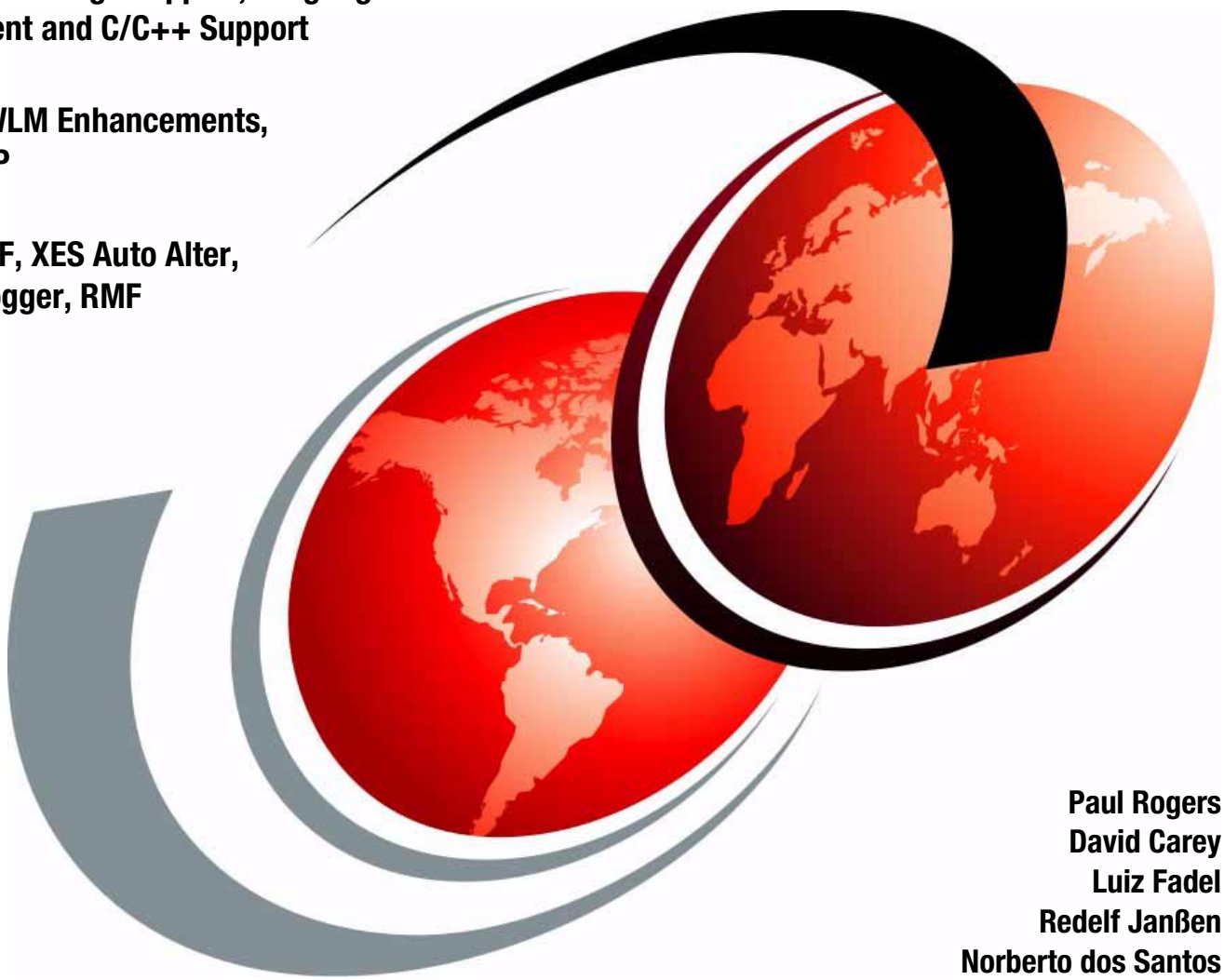
IBM

# OS/390 Version 2 Release 10 Implementation

- 64-bit Real Storage Support, Language Environment and C/C++ Support
- DFSMS, WLM Enhancements, Telnet, FTP
- JES2, SDSF, XES Auto Alter, System Logger, RMF



Paul Rogers
David Carey
Luiz Fadel
Redelf Janßen
Norberto dos Santos

Redbooks

ibm.com/redbooks

International Technical Support Organization

SG24-5976-00

# OS/390 Version 2 Release 10 Implementation

March 2001

**Take Note!**

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special notices” on page 199.

**First Edition (March 2001)**

This edition applies to OS/390 Version 2 Release 10, Program Number 5647-A01.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ  Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

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

# Contents

**Preface** . . . . . ix
The team that wrote this redbook . . . . . ix
Comments welcome . . . . . x

**Chapter 1. OS/390 Version 2 Release 10 overview** . . . . . 1
1.1 OS/390 64-bit architecture . . . . . 1
1.2 Workload Manager (WLM) . . . . . 1
1.2.1 WLM migration tool . . . . . 1
1.3 SDSF enhancements . . . . . 2
1.4 System Logger enhancements . . . . . 2
1.5 XES enhancements . . . . . 2
1.6 DFSMS enhancements . . . . . 3
1.6.1 Web Server environment flexibility . . . . . 3
1.6.2 Large tape block size . . . . . 3
1.6.3 Writing data to different media. . . . . 3
1.6.4 Multiple address spaces for DFSMShsm . . . . . 3
1.6.5 VSAM striping . . . . . 3
1.7 Language environment . . . . . 4
1.8 RACF program control enhancements . . . . . 4
1.9 BCP enhancements . . . . . 4
1.10 RMF enhancements . . . . . 5
1.11 OS/390 UNIX System Services . . . . . 5
1.12 SMB file/print enhancements . . . . . 6
1.13 OS/390 installation overview . . . . . 7

**Chapter 2. 64-bit real storage support** . . . . . 9
2.1 A brief history of storage management . . . . . 9
2.2 Storage overview . . . . . 10
2.3 Storage management components . . . . . 11
2.3.1 Real or main storage . . . . . 11
2.3.2 Expanded storage . . . . . 12
2.3.3 Virtual storage . . . . . 12
2.3.4 Auxiliary Storage Manager (ASM) . . . . . 13
2.3.5 Virtual Storage Manager (VSM). . . . . 13
2.3.6 Real Storage Manager (RSM) . . . . . 14
2.4 System Resource Manager (SRM) . . . . . 14
2.5 z/Architecture system programmer migration considerations . . . . . 16
2.5.1 Determining the architecture mode . . . . . 16
2.5.2 Displaying the architecture mode . . . . . 17
2.5.3 Dealing with real addresses . . . . . 17
2.6 Reconfiguring storage above 2 GB . . . . . 18
2.6.1 Changes to the CONFIGxx parmlib member . . . . . 18
2.6.2 Changes to the CONFIG command . . . . . 19
2.7 Changes introduced with 64-bit real storage support . . . . . 19
2.8 Changes to RSM in z/Architecture mode . . . . . 20
2.9 Changes to SRM in z/Architecture mode . . . . . 20
2.10 Channel program changes to support 64-bit real storage . . . . . 21
2.10.1 CCW indirect data addressing . . . . . 21
2.10.2 IOS support for 64-bit real storage . . . . . 21
2.10.3 Using EXCP and EXCPVR . . . . . 21
2.10.4 Instructions to perform I/O . . . . . 22

2.11 Changes to Access Method Services, EXCP and EXCPVR . . . . . 23
2.12 Changes to VSM in z/Architecture mode . . . . . 24
2.12.1 New parameters for GETMAIN and STORAGE . . . . . 24
2.12.2 CPOOL macro support. . . . . 25
2.13 Changes for both ESA/390 and z/Architecture mode . . . . . 25
2.14 Unformatted dump changes in OS/390 Release 10 . . . . . 26
2.15 Service aids support for 64-bit real storage . . . . . 26
2.15.1 Changes to the Interactive Problem Control System (IPCS). . . . . 26
2.16 Slip trap processing changes. . . . . 27
2.17 Invocation of system services in 64-bit mode. . . . . 28
2.17.1 Changes based on the addressing mode . . . . . 28
2.18 z/Architecture addressing mode considerations . . . . . 29
2.18.1 Non-modal instructions . . . . . 29
2.18.2 Modal instructions . . . . . 29
2.18.3 Changing addressing mode . . . . . 30
2.18.4 Addressing mode instructions . . . . . 30
2.19 Dual architecture support. . . . . 30
2.20 Migration scenarios . . . . . 32

**Chapter 3. System Logger** . . . . . 35
3.1 System Logger enhancements in Release 3 . . . . . 35
3.2 System Logger enhancements in Release 4 . . . . . 36
3.3 System Logger enhancements in Release 10 . . . . . 37
3.3.1 XES Auto Alter. . . . . 37
3.3.2 Logger latching performance enhancements . . . . . 39
3.3.3 Logger rebuild ENQ contention reduction. . . . . 39
3.3.4 Multi-block support for browse . . . . . 39

**Chapter 4. System Display and Search Facility (SDSF)** . . . . . 41
4.1 SDSF Primary Option Menu reorganization . . . . . 42
4.2 Improved handling of WTORs . . . . . 43
4.2.1 Filtering WTORs on log panels . . . . . 43
4.2.2 New panel to display system requests . . . . . 45
4.3 Mixed case column titles . . . . . 48
4.4 New action characters and overtypes . . . . . 48
4.5 New SDSF commands. . . . . 49
4.6 WHO command enhancement . . . . . 51
4.7 Improved management of SDSF parameters . . . . . 52
4.7.1 Conditional processing capability in dynamic parms. . . . . 53
4.7.2 System symbol support . . . . . 54
4.8 The SDSF server. . . . . 55
4.8.1 New ISFPARMS statements for sysplex data. . . . . 55
4.8.2 Server registration with ARM . . . . . 55
4.9 Sysplex system management and MQSeries . . . . . 55
4.10 MQSeries for OS/390 considerations. . . . . 56
4.10.1 MQSeries facilities. . . . . 56
4.10.2 MQSeries queues . . . . . 56
4.10.3 OS/390 MQ libraries . . . . . 57
4.10.4 MQSeries configuration . . . . . 57
4.10.5 Storage estimates . . . . . 58
4.10.6 Queues . . . . . 58
4.11 Communication between queue managers . . . . . 59
4.12 Server group . . . . . 59

4.12.1 Server group examples . . . . . .60
4.12.2 Configuration examples . . . . . .61
4.13 Sysplex-wide panels . . . . . .65
4.13.1 Sysplex device display support . . . . . .65
4.13.2 Improved SDSF browse and log display . . . . . .66
4.14 Configuration Assistant for enabling sysplex panels . . . . . .67
4.14.1 SAF security . . . . . .68
4.14.2 SDSF Configuration Assistant functions . . . . . .68
4.14.3 Accessing the SDSF Configuration Assistant . . . . . .69

**Chapter 5. DFSMS for OS/390 Release 10** . . . . . .71
5.1 DFSMSdfp enhancements . . . . . .71
5.1.1 VSAM striping . . . . . .72
5.1.2 Support for large tape block sizes . . . . . .76
5.1.3 DADSM rename duplicate data set . . . . . .77
5.1.4 UNIT=AFF support for tape libraries . . . . . .77
5.1.5 Coupling facility structure rebuild for catalogs . . . . . .78
5.2 DFSMShsm enhancements . . . . . .78
5.2.1 Concurrent copy enhancement . . . . . .79
5.2.2 Data set backup direct to tape . . . . . .79
5.2.3 Data set backup multitasking . . . . . .80
5.2.4 Fast subsequent migration . . . . . .81
5.2.5 Multi-address-space DFSMShsm . . . . . .83
5.3 DFSMSrmm enhancements . . . . . .85
5.3.1 Multivolume set retention and movement . . . . . .85
5.3.2 Support for Tivoli OPC . . . . . .86
5.3.3 Pre-ACS interface support . . . . . .87
5.3.4 SMS ACS support . . . . . .87
5.3.5 Virtual tape server support . . . . . .88
5.3.6 Fast tape positioning with DFSMSrmm . . . . . .89
5.3.7 Audit support for CDS against TCDB and library manager . . . . . .89

**Chapter 6. DCE DFS and SMB support** . . . . . .91
6.1 DFS enhancements in Release 10 . . . . . .93
6.1.1 DFS LFS enlarged file size . . . . . .93
6.2 SMB Enhancements in Release 10 . . . . . .94
6.2.1 SMB dialect NT LM 0.12 . . . . . .94
6.2.2 Record File System . . . . . .94
6.2.3 New environment variables . . . . . .96
6.2.4 Login mechanisms . . . . . .98
6.3 Setting up an SMB RFS connection . . . . . .98
6.3.1 Accessing files and printers . . . . . .99
6.3.2 User requests for files and printing . . . . . .101

**Chapter 7. Resource Measurement Facility enhancements** . . . . . .103
7.1 Online Monitoring with RMF Monitor II and Monitor III . . . . . .104
7.1.1 VSAM RLS support . . . . . .104
7.1.2 Multi-system enclave support . . . . . .106
7.1.3 OMVS process data report . . . . . .107
7.1.4 Parallel Access Volume (PAV) support . . . . . .110
7.1.5 Enterprise Storage Server . . . . . .110
7.1.6 ESS performance features . . . . . .112
7.1.7 S/390 64-Bit architecture . . . . . .116
7.1.8 Dynamic central processor upgrade . . . . . .119

7.2 Online monitoring with PM of OS/390 . . . . . 119
7.3 Long-Term reporting with the RMF postprocessor . . . . . 120
7.3.1 WebServer performance reporting . . . . . 121
7.3.2 Lotus Domino support . . . . . 123
7.3.3 FICON director support . . . . . 124
7.4 Performance analysis with the Spreadsheet Reporter . . . . . 125
7.4.1 Spreadsheet enhancements . . . . . 125

**Chapter 8. Language Environment and C/C++ enhancements** . . . . . 127
8.1 Downward compatibility . . . . . 127
8.2 XPLINK performance enhancement . . . . . 128
8.2.1 XPLINK applications in an LE environment . . . . . 128
8.2.2 Compiling and Linking XPLINK applications in LE . . . . . 129
8.2.3 Debugging an XPLINK application . . . . . 130
8.2.4 Comparison of non-XPLINK and XPLINK Register conventions . . . 130
8.3 Large file support . . . . . 131
8.4 Additional Language Environment OS/390 Release 10 enhancements . 131
8.5 C/C++ Enhancements . . . . . 132

**Chapter 9. Communications Server for OS/390 V2R10** . . . . . 133
9.1 Telnet enhancements . . . . . 133
9.2 Telnet security enhancements . . . . . 133
9.3 File Transfer Protocol (FTP) enhancements . . . . . 134
9.3.1 FTP security . . . . . 134
9.3.2 FTP functionality enhancements . . . . . 136
9.3.3 FTP user exit support . . . . . 137
9.3.4 FTP JES support . . . . . 137
9.4 Performance enhancements . . . . . 137
9.4.1 Service policy enhancements. . . . . 137
9.4.2 Dynamic Virtual IP Addressing (VIPA) takeover enhancement . . . . 138
9.4.3 Sysplex workload distribution . . . . . 139
9.4.4 Network traffic access controls . . . . . 140
9.4.5 Traffic Regulation Management (TRM) . . . . . 141
9.5 Security enhancements . . . . . 141
9.5.1 User control of network access . . . . . 142
9.5.2 User control of port access . . . . . 143
9.5.3 User control of stack access . . . . . 144

**Chapter 10. XES Auto Alter support** . . . . . 145
10.1 Requesting structure size . . . . . 146
10.2 Structure full monitoring . . . . . 146
10.3 Automatically altering structures . . . . . 147
10.3.1 Understanding when a structure is automatically altered . . . . . 148
10.3.2 Considerations for duplexed structures . . . . . 149
10.3.3 Relieving coupling facility storage constraints . . . . . 149
10.4 CFRM policy changes . . . . . 149
10.5 IXLCONN macro changes . . . . . 150
10.6 Interactions and dependencies . . . . . 150
10.7 Exploiters . . . . . 151
10.8 Externals . . . . . 152
10.8.1 Messages . . . . . 152
10.8.2 Commands . . . . . 152
10.9 Migration . . . . . 153
10.10 Coexistence . . . . . 153

**Chapter 11. Workload Manager (WLM)** . . . . . 155
11.1 WLM/SRM 64-bit support . . . . . 155
11.1.1 SRM changes due to removal of expanded storage . . . . . 156
11.2 WLM server task/thread management . . . . . 159
11.2.1 Assumptions . . . . . 160
11.2.2 General changes . . . . . 160
11.2.3 Potential exploiters . . . . . 160
11.2.4 UNIX System Services extensions . . . . . 161
11.3 Defining special protection options for critical work . . . . . 161
11.3.1 CPU protection . . . . . 162
11.3.2 Storage protection . . . . . 163
11.3.3 CICS and IMS region management . . . . . 164
11.3.4 CICS and IMS use of CPU and storage protection . . . . . 166
11.4 New classification qualifier types . . . . . 167
11.4.1 System name and system name group . . . . . 168
11.4.2 Sysplex name . . . . . 169
11.4.3 Subsystem collection name . . . . . 169
11.4.4 Scheduling environment . . . . . 170
11.5 Scenarios for managing CICS and IMS workloads . . . . . 170
11.5.1 Production regions running normal transactions . . . . . 171
11.5.2 Production regions running conversional transactions . . . . . 173
11.6 Workload management migration . . . . . 175
11.6.1 Migration considerations for CICS or IMS enhancements . . . . . 176
11.6.2 Other migrations issues . . . . . 176

**Chapter 12. OS/390 Version 2 Release 10 JES2 enhancements** . . . . . 177
12.1 Current JES2 processing overview . . . . . 177
12.1.1 Writing data to spool . . . . . 177
12.1.2 Reading data from spool . . . . . 177
12.2 JES2 spool I/O changes in Release 10 . . . . . 178
12.2.1 JES2 Release 10 spool writes . . . . . 178
12.2.2 JES2 Release 10 spool reads . . . . . 178
12.2.3 JES2 EXCPVR enhancement . . . . . 179
12.3 JES2 spool allocation changes . . . . . 179
12.3.1 JES2 volume fencing . . . . . 179
12.3.2 JES2 Release 10 fencing enhancements . . . . . 180
12.3.3 System affinity for spool volumes . . . . . 181
12.3.4 Spool management example . . . . . 181
12.4 Multi-system dumps . . . . . 182
12.5 Purging jobs affecting JES2 restarts . . . . . 183
12.5.1 New options to remove a job . . . . . 183
12.6 JES2 Release 10 Migration considerations . . . . . 184
12.6.1 JES2 levels supported by OS/390 . . . . . 184
12.6.2 JES2 coexistence . . . . . 185

**Chapter 13. OS/390 installation overview** . . . . . 187
13.1 Changed base elements . . . . . 187
13.2 New and changed optional features . . . . . 187
13.3 Removed elements and features . . . . . 188
13.4 System requirements . . . . . 189
13.4.1 Driving system hardware requirements . . . . . 189
13.4.2 Driving system software requirements . . . . . 189
13.4.3 Target system hardware requirements . . . . . 190

13.4.4 Target system software requirements . . . . . 191
13.4.5 OS/390 Release 10 coexistence requirements. . . . . 192
13.5 Installation improvements . . . . . 192
13.5.1 Web-based wizards . . . . . 192

**Appendix A. MQ definitions for MQSeries Queue Managers** . . . . . .195

**Appendix B. Special notices** . . . . . .199

**Appendix C. Related publications** . . . . . .201
C.1 IBM Redbooks . . . . . .201
C.2 IBM Redbooks collections . . . . . .201

**How to get IBM Redbooks** . . . . . 203
IBM Redbooks fax order form . . . . . .204

**Index** . . . . . 205

**IBM Redbooks review**. . . . . 209

# Preface

This IBM Redbook contains information related to many of the changes made in OS/390 Version 2 Release 10. You can use it to help you install, tailor and configure Release 10.

This redbook gives a broad understanding of a new architecture for 64-bit real storage addressing. Other topics discussed are:

- Changes to the System Logger
- New SDSF enhancements
- Enhancements to DFSMS
- Native Windows SMB support for accessing files and printers
- RMF enhancements
- Language Environment and C++ enhancements
- Changes to VSAM
- Communication Server enhancements
- Support for automatic tuning of coupling facility structures
- Workload Manager enhancements to support migrations to goal mode
- JES2 Release 10 enhancements

## The team that wrote this redbook

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

**Paul Rogers** is a Consulting IT Specialist at the International Technical Support Organization Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on various aspects of OS/390. Before joining the ITSO 12 years ago, he worked in the IBM Installation Support Center (ISC) in Greenford, England as OS/390 and JES support for IBM EMEA.

**David Carey** is a Senior IT Availability Specialist with the IBM Support Center in Sydney, Australia, where he provides defect and nondefect support for CICS, CICSPlex/SM, MQSeries, and OS/390. David has 20 years of experience in the information technology industry, and was an MVS systems programmer for 12 years prior to joining IBM.

**Luiz Fadel** is a certified Consulting IT Specialist from Sao Paulo, Brazil. He has more than 30 years of experience with MVS and OS/390. He has written extensively on OS/390.

**Redelf Janßen** is an Advisory IT Specialist in IBM Global Services ITS Bremen, Germany. He has a degree in Computer Science from Bremen University and joined IBM Germany in 1988. His areas of expertise include OS/390, Parallel Sysplex, UNIX System Services and DFSMS/MVS. He has written redbooks on OS/390 Release 3 and 4.

**Norberto dos Santos** is a System Programmer in Unibanco, Sao Paulo, Brazil. He has 19 years of experience in OS/390 system programming. His areas of expertise include OS/390, Parallel Sysplex, DFSMS/MVS, and related products.

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

**Rich Conway**
International Technical Support Organization, Poughkeepsie Center

**Bob Haimowitz**
International Technical Support Organization, Poughkeepsie Center

**Bob Rogers**
IBM Poughkeepsie

**Ken Jonas**
IBM Poughkeepsie

## Comments welcome

Your comments are important to us!

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

- Fax the evaluation form found in “IBM Redbooks review” on page 209 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

# Chapter 1. OS/390 Version 2 Release 10 overview

This chapter provides an overview of the new and enhanced functions introduced in OS/390 Version 2 Release 10.

## 1.1 OS/390 64-bit architecture

OS/390 Release 10 can operate on both the current ESA/390 architecture and the new z/Architecture. With this architecture, OS/390 Release 10 is able to support more than two gigabytes of central storage, thereby eliminating the overhead of paging between central storage and expanded storage.

OS/390 Release 10 has implemented the new architecture support in such a way that programs that are unaware of the new architecture function as expected.

For a detailed description of OS/390 large real storage support, see Chapter 2, “64-bit real storage support” on page 9.

## 1.2 Workload Manager (WLM)

Workload Manager (WLM) in goal mode continues to grow in its role and importance on the S/390 platform. Each new release of the operating system and supporting subsystems brings further exploitation of WLM goal mode for improvements and efficiencies in system performance and workload balancing. Goal mode is critical to the implementation of many strategic solutions to be delivered over the next several years.

A number of functions that have been considered by some customers to be inhibitors to the migration from WLM compatibility to goal mode will be addressed in the Release 10 time frame via APAR service. These items include:

- Enhanced CPU management to provide protection for critical work across major workload shifts
- Selective management of CICS or IMS regions with either velocity goals or transaction response time goals
- Storage isolation for critical regions over long, idle periods
- Classification by system group, system name, or “subsystem collection” name (e.g., JES2 MAS)
- Support for heterogeneous Parallel Sysplex clusters

**Note**: The operating system release scheduled for availability in the second half of 2001 will be the last release to support WLM compatibility mode. Goal mode will be the only supported mode starting with the operating system release scheduled for the first half of 2002.

### 1.2.1 WLM migration tool

To assist in this migration, a goal mode migration tool at no charge will be made available at the end of the first half of 2000. This tool can be accessed from the WLM Web page at:

```
http://www.s390.ibm.com/wlm/
```

## 1.3 SDSF enhancements

SDSF in Release 10 provides several enhancements to simplify system management in a Parallel Sysplex environment:

- The SDSF Initiator and Printer panels can now display a device defined to any JES in the MAS, regardless of the system the user is logged on to.
- Similarly, when browsing the SYSLOG or a jobs output, users now see the most recent data, regardless of the system they are logged on to.
- SDSF now supports the new functions in OS/390 JES2, such as:
  - Additional columns on the Printer panel
  - Support for mixed-case system commands, new codepages, and the OS/390 SDSF Configuration Assistant.
- The new sysplex function on the Initiator, Printer, Output Data Set, and SYSLOG panels requires the installation of MQSeries for OS/390 Version 2 Release 1. If MQSeries is not installed or available, the panels operate as they did in prior releases. The other SDSF enhancements do not require MQSeries.

SDSF enhancements are described in detail in Chapter 4, “System Display and Search Facility (SDSF)” on page 41.

## 1.4 System Logger enhancements

There are several enhancements to System Logger in OS/390, that are covered in detail in Chapter 3, “System Logger” on page 35. These enhancements provide a set of Logger performance improvements that include removing inhibitors between Logger services and providing access to more log stream log block data on a single browse request. They are related to the following topics:

- XES auto alter
  This function that allows XES to monitor and tune the CF structure size and storage ratios in real time in response to changing structure usage.
- Logger latching performance enhancements
  These remove the direct competition between browse and delete Logger requests with write and offload requests for a log stream.
- Logger recovery processing resource cleanup
  This identifies and fixes latent problems in Logger resource cleanup.
- Multi block support for browse
  This enhances the Logger IXGBRWSE service to return as many log blocks on a single request as fit into the invoker’s buffer.

## 1.5 XES enhancements

XES auto alter support, which is described in Chapter 10, “XES Auto Alter support” on page 145, gives you a new way to monitor and tune the coupling facility structure sizes and ratios in real-time to changing structure object usage. This happens via a system-initiated alter. It minimizes efforts in calculating initial structure sizes and ratios. It also helps in avoiding *structure full* conditions.

## 1.6 DFSMS enhancements

In OS/390 Release 10, DFSMS continues to add enhancements to performance, availability, system throughput, and usability for data access and storage management. In addition, DFSMS in Release 10 is the first release of DFSMS that will be available solely with OS/390. DFSMS is packaged and shipped with OS/390 Release 10 and offers customers the ease of installation, integration, and ease of maintenance inherent in the OS/390 product. You can read more about the DFSMS enhancements in Chapter 5, “DFSMS for OS/390 Release 10” on page 71.

### 1.6.1 Web Server environment flexibility

DFSMShsm provides the S/390 Web Server environment with additional flexibility and throughput in backup processing while maintaining data availability for the 24x7 Web environment. You now have full capability and the choice of doing backup via a timed event or by invoking it via batch or macros when a job runs and synchronization is required. Also, all backup processes can now proceed with integrity while leaving the data in full READ/WRITE mode and maintaining 24x7 availability by using proven functions like concurrent copy.

### 1.6.2 Large tape block size

Large business intelligence applications often place production data on both disk and tape. In Release 10, DFSMS introduces changes in tape block size support that allow certain applications to take better advantage of the speed and storage capabilities of newer tape devices such as the 3590, as well as older devices such as the 3480 and 3490.

### 1.6.3 Writing data to different media

Data Warehouse applications, some Enterprise Resource Planning (ERP) applications, and native OS/390 subsystems have requirements to write data to different media (disk and tape) using native OS/390 allocation techniques such as UNIT=AFF. Release 10 has the following new capabilities:

- Allows for seamless use of these allocation techniques.
- Minimizes allocation failures across device types.
- Improves the availability of these applications and subsystems.

### 1.6.4 Multiple address spaces for DFSMShsm

DFSMShsm allows multiple host address spaces per OS/390 image, up to a total of 39 host address spaces per Parallel Sysplex cluster. This can help reduce contention and improve system throughput in a highly multitasked system. Also, each host address space can be given a different dispatching priority or velocity goal. Functions can be assigned to specific DFSMShsm address spaces, or spread across multiple address spaces.

### 1.6.5 VSAM striping

IBM introduced sequential data set striping with DFSMS 1.1, providing significant throughput improvements for large sequential accesses. New with DFSMS in Release 10, VSAM can now also take advantage of data set striping. VSAM data

sets can be striped across multiple volumes. It also allows VSAM applications such as DB2 to substantially reduce run times and shorten batch windows.

## 1.7 Language environment

OS/390 Release 10 now provides downward compatibility support through Language Environment. Assuming that required programming guidelines and restrictions (documented in the *Language Environment Programming Guide*) are observed, this support enables programmers to develop applications on higher release levels of OS/390, for deployment on execution platforms that are running lower release levels of OS/390. For example, a company may use OS/390 Release 10 (and Language Environment) on a development system where applications are coded, link-edited, and tested, while using any supported lower release of OS/390 (and Language Environment) on their production systems where the finished application modules are deployed.

Enhancements to the Language Environment are as follows:

- Extra Performance Linkage (XPLINK) is an enhanced function call linkage between programs that can significantly improve the performance of C and C++ programs by reducing function call overhead. This new linkage also allows for a common linkage for C and C++ programs, which helps function pointers to work as on other platforms. With XPLINK, you can more easily develop applications on other platforms and deploy them on OS/390.
- Language Environment in OS/390 Release 10 provides downward compatibility support. This new function offers more flexibility to application developers. Language Environment downward compatibility allows the applications created on higher release levels of OS/390 to run on lower releases of OS/390.
- Language Environment provides large file support for 31-bit applications, which improves porting capabilities of C/C++ applications accessing HFS and NFS files larger than 2 GB. This is accomplished by changing some C runtime library I/O functions to support the long long data type for recording for file offsets.
- Language Environment provides support to allow C and C++ applications to exploit 64-bit arithmetic capabilities and access 64-bit register information.

## 1.8 RACF program control enhancements

Updates were made to the SecureWay Security Server (RACF) that enhance the configuration of program control for both traditional MVS libraries and OS/390 UNIX files. It will now be easier for customers to determine which programs they need to define as controlled programs to allow OS/390 UNIX server and daemon programs to run with good security and integrity. In addition, it will be easier to prevent the introduction of uncontrolled programs into the execution environment of the server or daemon. This can prevent trojan horses from compromising the security or integrity of the server daemon.

## 1.9 BCP enhancements

Enhancements to the OS/390 Release 10 BCP include:

- The enablement of automatic Coupling Facility structure alteration when an installation-defined percent full threshold has been reached. This process may increase the size of the structure, enable the reapportionment of objects within the structure, or both.
- Significantly improved elapsed time performance with BSAM, QSAM and EXCP support for 3590 tape device block sizes up to the maximum supported by the device. BSAM and QSAM will support a maximum block size of 64 KB on all other magnetic tape cartridge devices.
- Mutliblock support for the System Logger browse function has been introduced to enable multiple log blocks to be returned with a single IXGBRWSE request when using the MULTIBLOCK keyword.

## 1.10 RMF enhancements

RMF enhancements include:

- VSAM RLS monitoring support for transactional VSAM (TVS), which enables concurrent access to recoverable data sets by CICS online and batch.
- Multisystem enclave monitoring support for transactions that originate on any system and continue (in parallel) on other systems within the Parallel Sysplex.
- Performance measurement support for UNIX System Services.
- Parallel Access Volume (PAV) performance statistic generation.
- Web Server performance and usage information using SMF Type 103 records.
- Lotus Domino server load reporting using SMF Type 108 records.
- FICON Director activity report.

## 1.11 OS/390 UNIX System Services

OS/390 Version 2 Release 10 UNIX System Services includes new enhancements, as follows:

- Support for C/C++ applications.
- Support for 64-Bit Real and Arithmetic Hardware enhancements.
- Support for 64-bit real addressing, which improves the performance and response time of applications that have very large memory and DASD storage demands, use Data in Memory, or need to access very large databases.
- Large File support.
- Support for utilities that perform file operations for large (2 GB or larger) HFS files.
- Kernel support for LE XPLink. OS/390 UNIX provides support for LE XPLink (eXtra Performance Linkage), which improves the execution performance and compile times of OS/390 applications written in C/C++.
- Shell and Utilities support for New Long Long Data Types. Long long support eases the task of porting programs that use 64-bit integers (such as JAVA Virtual Machine).
- RAS enhancements that include improvements for diagnostics and serviceability of the OS/390 UNIX environment with tools that identify

problems in setup, enable users to gather better dumps, and improve the analysis of dumps. These improvements include:

  - Security enhancements to AF_UNIX PFS. These enhancements allow an AF_UNIX datagram server to receive the identity of the sender of each message it receives, providing for better troubleshooting of data passed from the syslog daemon to the joblog.
  - A sysconf() performance enhancement that is a valuable tool that improves performance and allows application programs to retrieve data from the system. New flags are added to meet UNIX98 standards.
  - Relative addressing exploitation (USS RAS) improves the performance of heavily used kernel modules through conversion to compiler/assembler relative addressing support, which reduces the size of the kernel modules in LPA.
  - dbx support of long long compiler symbolic and arithmetic to provide for debugging of C/C++ applications that include long long and unsigned long long data types.
  - dbx support of XPLink allows for debugging of new code associated with XPLink.

- Performance improvements in Release 10 include:
  - Enhanced reporter support, which allows more kernel-related data to be made available to report applications like RMF, improving the ability of the OS/390 UNIX platform to manage UNIX workloads.
  - make/c89 integration that integrates make, c89, and the Binder to cause the build process to run faster on larger applications.
  - Application notification of stack recycle enhances common INET to notify servers when a new transport provider stack is initialized, so that servers do not have to be manually recycled.

## 1.12 SMB file/print enhancements

The SMB server provides print serving support for Windows clients, allowing the SMB protocol to be used to send print requests to the OS/390 Infoprint Server, thus removing the need for additional print client code or unique printer setup steps on the user workstation.

The SMB server support is integrated within the Distributed File Service (DFS) element, which also provides DCE DFS client and server support. The SMB support does not require DCE, but the same server can optionally support DFS clients, SMB clients, or both.

In Release 10, in addition to its support for workstation access to OS/390 data stored in HFS using SMB protocols, the OS/390 DFS/SMB Server now supports workstation access to OS/390 data stored in SAM, PDS(E), and VSAM files to further expand the S/390 support for application development.

The Release 10 SMB server now supports the NT LM 0.12 level of the SMB protocol dialect used by the Windows NT networking support, thereby providing additional password encryption, file sizes greater than 4 GB, and other capabilities allowed by this level of the SMB protocol.

DFS continues to provide performance improvements for customers that use the DCE DFS support. DFS also enables the OS/390 SMB File/Print Server to support the latest SMB protocol used by the Windows NT networking, thereby providing the additional capabilities allowed by this protocol. This enhancement also includes SMB file serving support for OS/2 clients.

You can find more information about DCE/DFS and SMB support in Chapter 6, “DCE DFS and SMB support” on page 91.

## 1.13 OS/390 installation overview

In Chapter 13, “OS/390 installation overview” on page 187 we give an overview of new and updated items related to the installation process of OS/390 Release 10. These include changed base elements, new and changed optional features, removed elements and features, installation changes, system requirements, and installation improvements.

# Chapter 2. 64-bit real storage support

The 64-bit real storage support provides for up to 256 gigabytes of central storage to be configured to a single OS/390 image of the new z/Architecture of the 2064 processors. This removes the current ESA/390 architecture limit of 2 gigabytes configured to a single OS/390 image.

## 2.1 A brief history of storage management

The evolution of the OS/390 system from its humble beginnings in 1964 when the S/360 24-bit architecture was developed and real memory was counted in kilobytes, to the current OS/390 system, highlights the scalability of the large system processor as depicted in Figure 1.

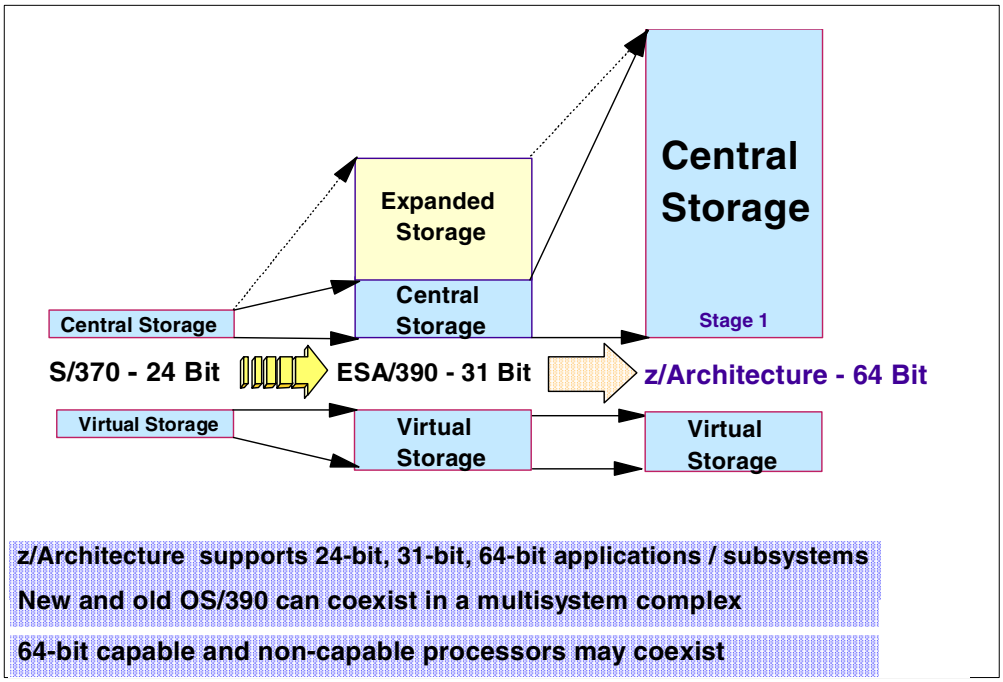



*Figure 1. Memory architecture evolution*

The year 1971 introduced us to the OS/VS1 system with its impressive virtual storage capability of 16 megabytes. In 1974 we saw the start of the multiple address space era with the Multiple Virtual System/370 architecture, commonly referred to as MVS. The MVS system could manage multiple address spaces up to 16 megabytes in size (the 24-bit limit), and utilized a sophisticated process to move programs and data from auxiliary I/O devices into the CPU in small manageable increments, while at the same time enabling multiple tasks to process seemingly simultaneously. The process consisted, in essence, of “paging” data that was occupying storage but had not been referenced for a period of time, to an external disk device, or “swapping” out a task, but retaining its allocated storage, until the System Resource Manager decided that this task could continue processing. The paged data would then be retrieved from disk.

The next major development came in 1981 when the MVS/XA system was announced. The XA represented eXtended Architecture and made available 31-bit addressing. This enabled 2 gigabyte addressability for real and virtual memory. While this provided us with the ability to process much larger applications, the weak link was still the paging process, which relied on slow external disk devices and I/O subsystems when compared with processor speeds.

The performance overhead associated with slower I/O devices and the paging process was positively influenced in 1985 with the development of the 3090 processor. The major enhancement was the introduction of expanded storage, which was, in essence, an extension of real storage that could be used for paging and data storage. This changed our reliance on the external I/O devices for paging.

The MVS/ESA system was available in 1988, the ESA identifying the futuristic "Enterprise System Architecture," which was then followed in 1990 with the S/390 processor, and then the MVS Open Edition operating system in 1993, which embraced the distributed, multiplatform structure that is now an integral part of the IT environment.

OS/390 is the latest generation in the development of what started in 1970 with S/370. The introduction of 64-bit large real storage support with OS/390 Version 2 Release 10 is the next step in the evolution.

## 2.2 Storage overview

The current ESA/390 architecture limits the amount of central storage that can be configured to a single OS/390 image to 2 GB. A major enhancement implemented in OS/390 Version 2 Release 10 is the removal of the 2 GB real storage restriction by utilizing the new 64-bit architecture, when running on the new "Freeway X3" (2064/116) processor. OS/390 now supports up to 128 GB of central storage, when running in z/Architecture mode. The 128 GB limit is a software restriction that was implemented to prevent the Page Frame Table from exceeding 1 MB.

As processors increase in speed, there is a need for larger amounts of central storage to ensure system optimization. Today's S/390 processors provide memory allocation of up to 2 GB of central storage augmented with expanded storage. However, as the ratio of central storage to expanded storage decreases, the amount of overhead to move data into and out of expanded storage increases. To eliminate this overhead, there is a growing need to support all processor memory as central storage, hence the move to 64-bit real storage support.

The initial problem of reducing the excessive I/O overhead related to task paging and swapping was addressed with the introduction of expanded storage in 1985. The term used to define this use of expanded storage was Data-in-Memory. This design included a 32-bit compatible method for addressability that enabled applications to take advantage of additional memory without, in most cases, changing code. Often, a simple recompile was all that was necessary. DB2 was one of the great beneficiaries of this function, which enabled large amounts of data to be stored in memory, eliminating the previous requirements for unnecessary I/O activity to retrieve data from external disk devices.

It should be noted that while the 64-bit architecture enables large real storage support above the 2 GB limit, a processor that only has 2 GB of memory does not benefit from running in z/Architecture mode. In fact, the additional overhead required to maintain the larger 64-bit registers in this type of system would be counter productive. The ESA/390 31-bit mode in this case would be a better option.

**Note:** Implementing OS/390 Release 10 on a 9672 G5 or G6 processors does not enable the 64-bit real storage capability.

## 2.3 Storage management components

In S/390 the following types of *storage* can be defined: processor storage, which can be either central, often referred to as main or real storage and/or expanded storage; virtual storage; and auxiliary storage, which usually resides on direct access storage devices (DASD) (see Figure 2).

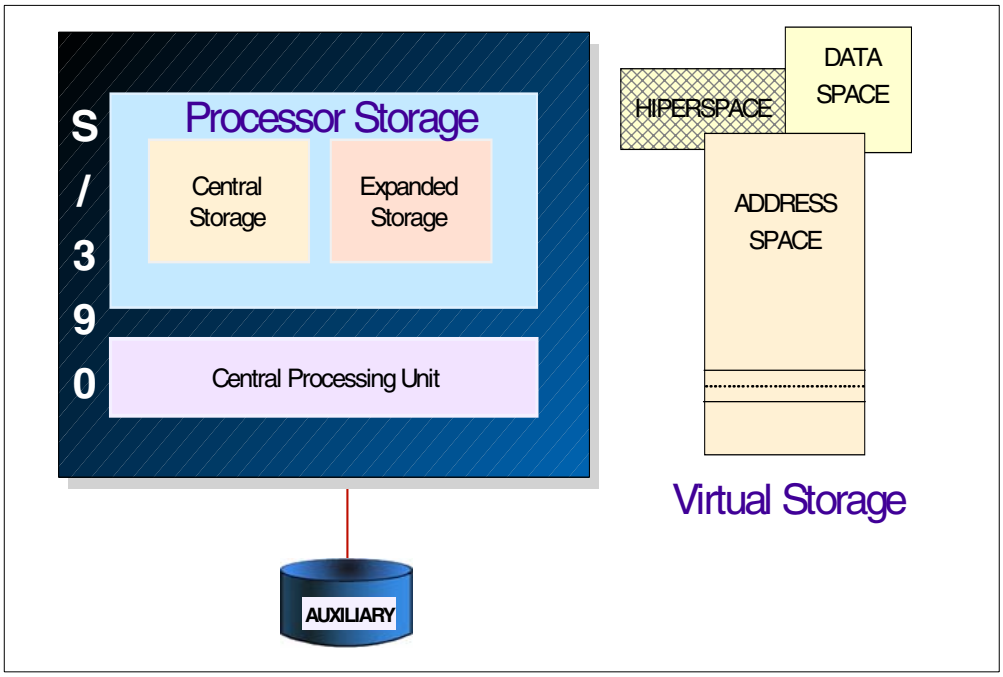



*Figure 2. S/390 storage*

### 2.3.1 Real or main storage

Real storage is directly addressable storage that enables high-speed processing of data by the CPUs and the channel subsystem. Both data and programs must be loaded into real storage, often called memory, from input devices before they can be processed. The storage is available in multiples of 4-Kbyte blocks. The amount of real storage that can be allocated to an OS/390 system is dependent on the physically available and configured memory.

**Note:** The amount of real storage supported on 64-bit capable processors is increased from 2 GB to 256 GB. A software-limited value of 128 GB has been implemented in OS/390 Release 10.

### 2.3.2 Expanded storage

Expanded storage is memory that has been configured as expanded. It can be accessed by all CPUs in the configuration by means of instructions that transfer 4-Kbyte blocks of data from expanded storage to main storage or from main storage to expanded storage. Programs do not execute in expanded storage, but data required by applications can be stored in expanded storage for access by applications executing in real storage. The operating system uses expanded storage to more efficiently manage executing workloads, and to minimize the movement of data to and from slower external disk devices.

**Note:** Expanded storage is not supported by OS/390 Version 2 Release 10 when executing in 64-bit mode. It should be noted that you have the option to run your system in ESA/390 mode, in which case expanded storage is supported, but the 2 GB central storage limitation still exists.

### 2.3.3 Virtual storage

Virtual storage is the foundation that enables the processing of workloads in an OS/390 system. Virtual storage enables programs larger than the available real storage to execute. It unburdens the application programmer from the task of managing physical memory resources and enables the focus to be directed to the business logic. Furthermore, it allows many independent applications to concurrently share a limited physical resource, in this case, real storage, commonly called memory.

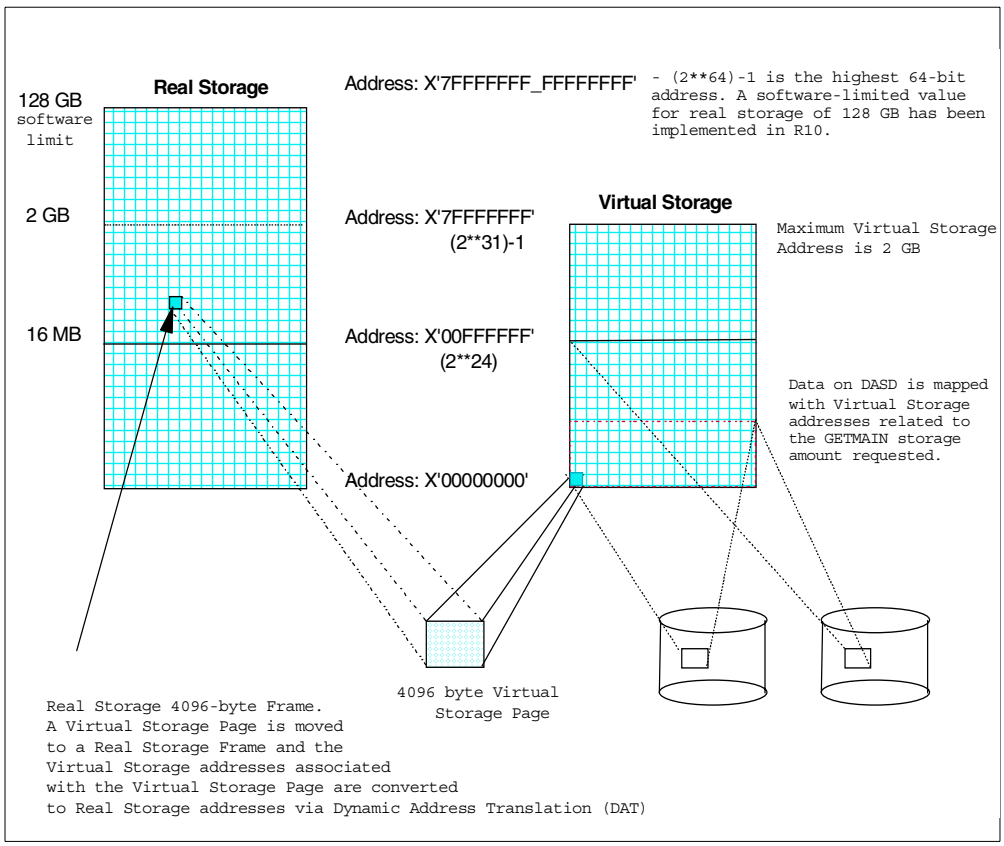



*Figure 3. Real and virtual storage overview*

Virtual storage does not occupy either real or auxiliary storage, but is conceptually a mapped representation of the resources, which include programs and data that are required to process an application. The operating system manages the physical memory, both real and expanded, as well as auxiliary storage on disk, while at the same time maintaining this conceptual simple linear address space of virtual memory that enables the application and operating system to interact.

Virtual memory is divided into 4 KB blocks, called pages. These pages are moved into and out of real storage by the operating system, and not all of them are accessible to the hardware at any given time. As far as the application is concerned, the storage it is using is contiguous, whereas the operating system is actually managing the processing in noncontiguous memory.

The mapping assigns virtual addresses to these resources, which in essence builds pointers to their location on auxiliary storage. The virtual address is what is used to control the movement of programs and data into real storage and these addresses group the data into 4 KB blocks, known as pages. When a virtual address is used for an access to main storage, it is translated by means of dynamic address translation (DAT) to a real address, which is then further converted by prefixing to an absolute address.

**Note:** Expanded storage is not supported by OS/390 Version 2 Release 10 when executing in 64-bit mode. It should be noted that you have the option to run your system in ESA/390 mode, in which case expanded storage is supported, but the 2 GB central storage limitation still exists.

Storage management in the OS/390 system is performed by the following components:

- Auxiliary Storage Manager
- Virtual Storage Manager
- Real Storage Manager

### 2.3.4 Auxiliary Storage Manager (ASM)

The Auxiliary Storage Manager is responsible for the transfer of virtual pages between real and auxiliary storage. This is performed as either a paging operation, one 4 KB page at a time, or a swapping operation, one address space at a time. The overhead associated with paging and swapping to auxiliary paging data sets is extensive, and for this reason the Real Storage Manager currently uses expanded storage to minimize DASD-related activity.

### 2.3.5 Virtual Storage Manager (VSM)

Virtual storage is addressable space that appears to the user as central (real) storage. Virtual storage is requested with the GETMAIN or STORAGE OBTAIN macro and is returned to the VSM with the FREEMAIN or STORAGE RELEASE macro. Instructions and data are mapped from virtual storage into central storage, where they are executed.

The process that maps virtual 4 KB pages with real memory 4 KB frames is called Dynamic Address Translation (DAT) and is implemented through a set of translation tables. The translation tables keep track of whether pages are backed in real storage, and, if they are not, they are marked as “invalid”. When the system needs to access a page of a program that is marked as “invalid”, an

exception condition, or page fault, is raised. This exception is handled by the operating system and is transparent to the application. This initiates the retrieval of data associated with the relevant virtual page into a real memory frame, and DAT then sets the translation table entry for the virtual page to point to the associated real memory frame. The application is then resumed and processing continues.

For changes in OS/390 Release 10, see 2.12, “Changes to VSM in z/Architecture mode” on page 24.

### 2.3.6 Real Storage Manager (RSM)

The Real Storage Manager’s task is to control the use of real storage frames. A frame is a 4 KB block of real storage. RSM also manages the movement of pages among central, expanded, and auxiliary storage. RSM acts together with the ASM to support the virtual storage concept, and with VSM to ensure that a GETMAINED page is backed with a real storage frame.

For changes in OS/390 Release 10, see 2.8, “Changes to RSM in z/Architecture mode” on page 20.

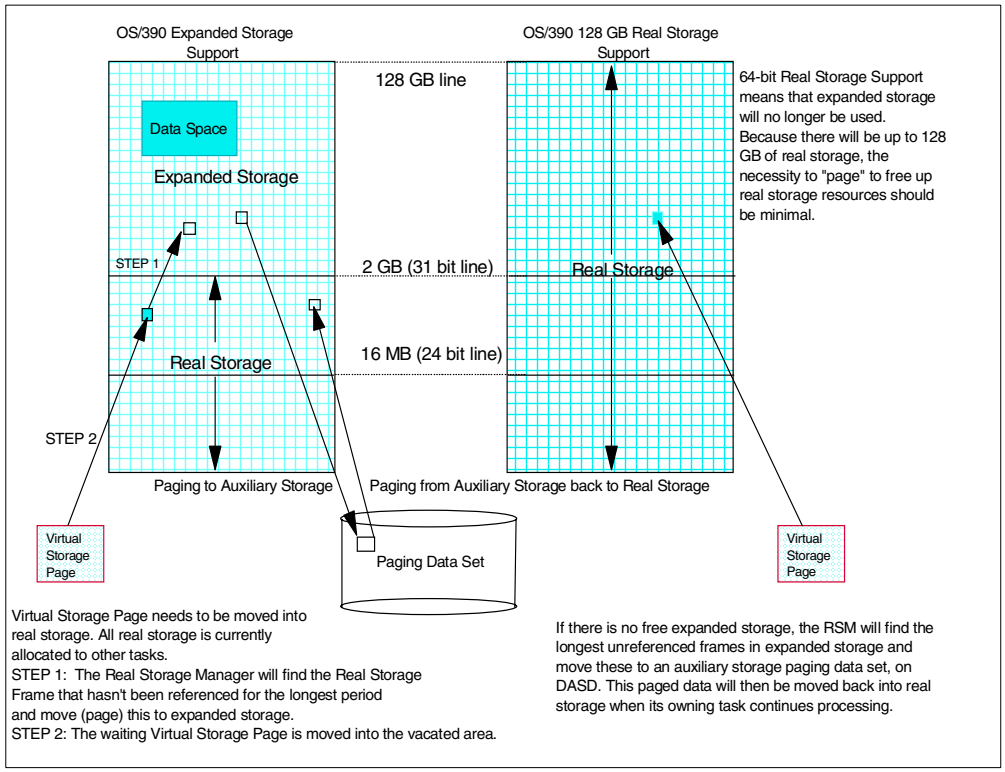



*Figure 4. Real and Virtual Storage Manager overview*

## 2.4 System Resource Manager (SRM)

SRM is a component of the system control program. It determines which address spaces, of all active address spaces, should be given access to system resources and the rate at which each address space is allowed to consume these resources (refer to Figure 5 on page 15).

One of the tasks SRM performs is to periodically monitor the availability of three types of storage and attempt to prevent shortages from becoming critical. The three types of storage are:

- Auxiliary storage
- SQA
- Pageable frames

Pageable frame stealing is the process of taking an assigned central storage frame away from an address space to make it available for other purposes, such as to satisfy a page fault or swap associated with another address space, or to satisfy a request for storage.

A page fault occurs when a task attempts to reference a page in real storage that is now being used by another task. The reason this occurs is that when there is a demand for pageable frames, SRM steals those frames that have gone unreferenced for the longest time and returns them to the system. The unreferenced interval count (UIC) of each frame indicates how long it has been since it was last referenced by an address space. This count is updated periodically by SRM and RSM. Each address space is examined, along with the common service area (CSA) and the pageable link pack area (PLPA). The frames to be “stolen” are then backed to expanded storage, or auxiliary storage, and the storage that was associated with that frame is reassigned to another task.

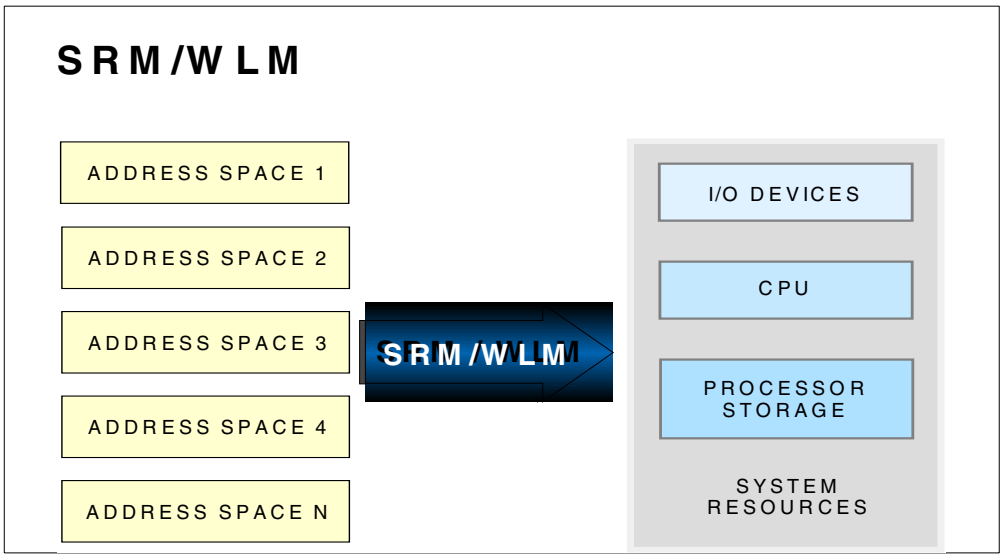



*Figure 5. SRM/WLM overview*

Stealing takes place strictly on a demand basis, that is, there is no periodic stealing of long-unreferenced frames. When the original task is “swapped in” by RSM to continue processing and attempts to reference the frame that has now been reassigned to another task, a page fault occurs, and SRM in conjunction with RSM retrieves the backed-up frame from either expanded or auxiliary storage and processing continues.

## 2.5 z/Architecture system programmer migration considerations

For the most part, the large real storage support is highly compatible with previous releases of OS/390. That makes migration to OS/390 Release 10 and the new processor very flexible.

The following system programmer steps and considerations to use an OS/390 image in z/Architecture mode are:

- Set the architecture mode.
- Make adjustments for the following removed functions:
  - Deletion of virtual fetch

    Any installation that still uses virtual fetch with IMS/DC must convert to using LLA for the IMS application modules in order to avoid a loss of performance due to the deletion of virtual fetch.
  - Deletion of duplexing common and PLPA data sets

    Any installation still duplexing the common data sets and PLPA page data sets must provide another way of preventing these data sets from being a single point of failure.
  - Deletion of swap data set support

    Any installation that still defines swap and page data sets must change the page data set definitions in IEASYSxx to no longer define swap data sets and to add additional local page data sets to restore the capacity previously provided by the swap data sets. The PAGTOTL parameter in the IEASYSxx is now specified in Release 10 as:

    PAGTOTL=(ppp)

    This specification causes the system to allow for the total of page data sets (ppp). The valid range is 0 to 256. This value is the maximum allowable number of page data sets that may be in use in the paging configuration at a time. Space for the PLPA and common data sets is reserved. Therefore, the maximum number of local page data sets is 253. The default value is 5.

    **Note:** The old specification allowed for swap data sets with the sss specification in PAGTOTL=(ppp,sss).

### 2.5.1 Determining the architecture mode

The IPL/NIP process prepares the system for initialization in either ESA/390 or z/Architecture mode according to what is specified in a new LOADxx statement, ARCHLVL.

The nucleus load module, IEANUC0x, is split into several components for OS/390 Version 2 Release 10. There is a common base, IEANUC0x, and two architectural extensions, IEANUC1x for ESA/390, and IEANUC2x for a/Architecture. When the ARCHLVL statement in LOADxx specifies ARCHLVL **1**, the system IPLs in ESA/390 mode. When ARCHLVL 2 is specified, as shown in Figure 6 on page 17, the system IPLs in z/Architecture mode. The system builds the nucleus from the base and incorporates the appropriate extension.

If the hardware does not support the requested architecture (e.g., requesting ARCHLVL 2 on an ESA/390 machine), wait state 088-10 results.

The new reason codes associated with wait state 088 are:

- 0E - Could not locate nucleus extension IEANUC1x or IEANUC2x.
- 10 - z/Architecture extension was requested but the hardware does not support the z/Architecture mode.

.

```
NUCLEUS   1
NUCLST    XX
ARCHLVL   2
IEASYM    XX
SYSPLEX   PLEX1     Y
IODF      ** SYS6       L06RMVS1  01  Y
SYSCAT    SBOX011    MCAT.PLEX1.VSBOX11
PARMLIB   SYS1.SYSPROG.PARMLIB
PARMLIB   SYS1.PARMLIB
PARMLIB   CPAC.PARMLIB
PARMLIB   SYS1.IBM.PARMLIB
*------------DEFINITION FOR SC63-------------*
HWNAME    SCZP601
LPARNAME  A8
SYSCAT    SBOX111    MCAT.PLEX1.R10.VSBOX11
*------------DEFINITION FOR SC64-------------*
HWNAME    SCZP601
LPARNAME  A9
SYSCAT    SBOX111    MCAT.PLEX1.R10.VSBOX11
*------------DEFINITION FOR SC65-------------*
HWNAME    SCZP601
LPARNAME  A10
SYSCAT    SBOX111    MCAT.PLEX1.R10.VSBOX11
```

*Figure 6. LOADxx member for z/Architecture mode*

### 2.5.2 Displaying the architecture mode

The architecture mode can be displayed with the Display IPLINFO command, as shown in Figure 7.

```
D  IPLINFO
IEE254I   16.02.46  IPLINFO  DISPLAY  753
  SYSTEM  IPLED  AT  08.15.15  ON  07/12/2000
  RELEASE  OS/390  02.10.00
  USED  LOADS8  IN  SYS0.IPLPARM  ON  0CD0
  ARCHLVL  =  2
  IEASYM  LIST  =  XX
  IEASYS  LIST  =  (R3)  (OP)
  IODF  DEVICE  0CD0
  IPL  DEVICE  3A00  VOLUME  O10RA1
```

*Figure 7. D IPLINFO command*

### 2.5.3 Dealing with real addresses

When there are situations where code must be changed to support operations under the new architecture, two approaches can be taken:

1. Provide the necessary function in a way that works under both architectures. The use of Test Protection (TPROT) as a replacement for Load Real Address (LRA) in certain cases is an example of this approach.
2. Use "dual-path" code to provide two distinct logic paths, one for each architecture environment. In support of this, a flag in the CVT is provided for execution-time testing of which architecture is being used. Also, the SYSSTATE macro has been enhanced to include an ARCHLVL option to address situations where macros may expand differently depending upon the architecture being used.

## 2.6 Reconfiguring storage above 2 GB

To support real storage above 2 GB, the CONFIG command and the CONFIGxx parmlib member have new options.

### 2.6.1 Changes to the CONFIGxx parmlib member

To support storage above 2 GB, the CONFIGxx parmlib member now allows the following specifications to configure sections of central storage. Multiple ranges can be specified. The parameter can be specified as STOR or STORAGE. The syntax is as follows:

ddddddddK-ddddddddK

**ddddddd** This is one to seven decimal digits, followed by a K, that are the starting and ending addresses of the section, and cannot exceed a value of 4194303. Each address represents a multiple of 1024 bytes. If necessary, the system rounds the low address down to the next lower 4K boundary. (This rounding is done to begin and end a section of storage on a 4K boundary.)

**Note:** The system does not reconfigure a section of central storage (in response to a CONFIG command) when the section is specified in this manner.

xxxxxxxxxxxxxxxx-xxxxxxxxxxxxxxxx

**xxxxxxxxxxxxxxxx** This is one to sixteen hexadecimal digits that address the first and last bytes of the section. If necessary, the system rounds the low address down to the next lower 4K boundary. This rounding is done to begin and end a section of storage on a 4K boundary.

**Note:** The system does not reconfigure a section of central storage (in response to a CONFIG command) when the section is specified in this manner.

dddddX-dddddX

**ddddd** This is one to five decimal digits, followed by a multiplier, that are the starting and ending addresses of the section and cannot exceed a value of 16383. The valid multipliers (X) are shown in 2.6.2, “Changes to the CONFIG command” on page 19.

**CONFIGxx example**

```
STOR 8192K-32768K,ONLINE
```

This example indicates that a section of central storage (location 8,388,608 through location 33,554,431) is to be verified as online.

### 2.6.2 Changes to the CONFIG command

The CONFIG command with the MEMBER option enables the operator to reconfigure the system according to the options in the specified CONFIGxx member. This reconfiguration is effective until the operator issues a different CONFIGxx MEMBER command or until the operator IPLs the system. With this command, the operator can reconfigure or verify the configuration of available channel paths, processors, central storage, central storage elements, expanded storage elements, and Vector Facilities. These are defined by the CHP, CPU, STOR(E=id), ESTOR(E=id), and VF parameters in CONFIGxx.

The CONFIG command can be used to reconfigure the amount of real storage above 2 GB for an OS/390 image. To set the amount of central storage to be reconfigured, specify up to five decimal digits followed by a multiplier (dddddX), where X is as follows:

**M** megabytes (2**20)

**G** gigabytes (2**30)

**T** terabytes (2**40)

**P** petabytes (2**50)

Check the configuration of your processor to see which size storage increments are supported. The value for ddddd must be a multiple of the storage increment size (usually 2, 4, or 8), and cannot exceed 16383P.

There are other formats for making the storage specifications:

**X'xxxxxxxxxxxxxxx'**This is a hexadecimal specification without a multiplier.

**X'xxx'X** Instead of specifying a decimal amount, you may specify a hexadecimal amount, with a multiplier, in the format shown in 2.6.2, "Changes to the CONFIG command" on page 19.

**dddddX-dddddX** The starting and ending addresses of the central storage section to be reconfigured. Specify up to five decimal digits followed by a multiplier (M-megabytes, G-gigabytes, T-terabytes, P-petabytes) for each address. The value for each ddddd must be a multiple of the storage increment size (usually 2, 4, or 8), and cannot exceed 16383P. The starting and ending addresses must not be the same.

**X'xxxxx'-X'xxxxx'** Instead of specifying the range using decimal numbers, you may specify it in hexadecimal without a multiplier.

**X'xxx'X-X'xxx'X** Instead of specifying the range using decimal numbers, you may specify it in hexadecimal, with a multiplier.

## 2.7 Changes introduced with 64-bit real storage support

The support for 64-bit large storage is totally transparent to problem state applications and to authorized programs outside the OS/390 BCP and DFP components. Programs using Load Real Address (LRA) instructions are affected.

The major difference is that with more central storage, paging to expanded storage is eliminated. There is also the possibility to configure applications to take advantage of more central storage.

## 2.8 Changes to RSM in z/Architecture mode

The following changes have been made to RSM when running in z/Architecture mode:

- Non-fixed pages, including disabled reference (DREF) storage, can be backed anywhere in real storage.
- The type of frame used, above or below 2 GB, when a page is fixed, is determined by an attribute specified at the time the virtual storage is obtained.
- Paging I/O is done directly to and from any real frame.
- VIO uses frames above 2 GB.
- Nucleus, SQA and LSQA, except DREF, is backed below 2 GB.
- Support for expanded storage is removed. The Hiperspace APIs and internal users of expanded storage are implemented to use real storage.
- The DSPSERV macro interface is changed to allow the invoker to specify whether pages of a data space can be backed anywhere in real storage when used for I/O.

## 2.9 Changes to SRM in z/Architecture mode

There are three categories of changes for SRM to enable support of 64-bit real storage. These include:

1. Expanded storage is not supported when the system is running in z/Architecture mode. Due to the removal of support for expanded storage, Hiperspace and VIO pages occupy real storage that retains support for applications that use these services. This change means that a page-out (DirectPO) sysevent must decide whether a VIO or standard Hiperspace page should be placed in real storage or sent to auxiliary storage when running in z/Architecture mode, rather than the current process of deciding between expanded and auxiliary storage.

   Currently, DirectPO sysevent decisions are based on the expanded storage migration age, which does not exist in a system running z/Architecture mode. Changes have been made to use the system high Unreferenced Interval Count (UIC), which is compared to the criteria table value of the page type, either VIO or Hiperspace. If the system high UIC is less than the criteria table value, the page is put in real storage. Otherwise it is sent to auxiliary storage.

2. Additionally, the UIC has been changed to ensure that a steal of CASTOUT(NO) Hiperspace frames is minimized, except when the system has a high level of contention for real storage. The current ESA/390 UIC interval is 1 second. This represents 1 second when a frame was not referenced, which means that the UIC is updated every second. To reduce the overhead of UIC update processing associated with the current 1 second frequency, this interval has been increased to 10 seconds when running in z/Architecture mode.

3. Finally, in this category, SRM has been changed to allow frames assigned below 16 MB to be stolen from logically swapped address spaces when no expanded storage is online when running in ESA/390 mode.

   Changes have been introduced to handle pageable frame shortages between 16 MB and 2 GB as a separate action, as distinct from overall pageable frame shortages. As OS/390 images grow larger than 2 GB, the area between 16 MB and 2 GB becomes a valuable resource with application dependencies for real storage in this range.

## 2.10 Channel program changes to support 64-bit real storage

Programs that build real channel programs may need to operate on multiple hardware platforms at different levels of the operating system referencing a variety of device types.

### 2.10.1 CCW indirect data addressing

Channel Control Word (CCW) indirect data addressing permits a single channel-command word to control the transfer of data that spans noncontiguous pages in real main storage. The use of CCW indirect data addressing also allows the program to designate data addresses above 16 MB for both format-0 and format-1 CCWs.

Channel access to real storage above 2 GB can be performed using format-0 or format-1 CCWs, in conjunction with a new 64-bit indirect-data-address word (IDAW) (refer to Figure 8 on page 22). CCW indirect data addressing is specified by a flag in the CCW which, when 1, indicates that the data address is not used to directly address data. Instead, the address points to a list of words, called indirect-data-address words (IDAWs).

The 64-bit IDAW is 8 bytes in length and must be aligned on a doubleword boundary, and the “span” of a 64-bit IDAW is 4 KB as opposed to the 2 KB used for 32-bit IDAWs. The ability to utilize 64-bit IDAWs is dependant on whether they are supported by the processor, the operating system, or the device. IOS modifies the UCB to indicate whether 64-bit IDAWs are supported for a specific unit. If addresses referenced within IDAWs are all below the 2 GB line, then 31-bit IDAWs can be used. If an I/O request specifies a 64-bit IDAW, but the UCB indicates that these are not supported, IOS posts an error.

### 2.10.2 IOS support for 64-bit real storage

The I/O support for 64-bit real storage is as follows:

- IOS is aware, through UIM definition, which device types support the 64-bit IDAWs and prevents their use with devices that do not support them.
- I/O involving real storage above 2 GB is supported only for DASD and TAPE.
- The STARTIO interface supports the use of channel programs using the 64-bit IDAWs.

### 2.10.3 Using EXCP and EXCPVR

Programs that use EXCP are always responsible for obtaining I/O buffers. Since the channel program provided by the application is always translated into a real channel program, EXCP translates it in a way that supports I/O buffers backed

above 2 GB in real storage. This is supported for DASD and TAPE. For other device types the UCB should be interrogated.

Programs using EXCPVR have the responsibility to page fix all I/O areas and to build real channel programs. To take advantage of real storage above 2 GB, they must get buffers that can be fixed above 2 GB and be able to handle 64-bit real addresses when constructing the channel programs.

Consider the following when using EXCP and EXCPVR:

- EXCP supports buffers backed above 2 GB without external changes.
- EXCPVR supports I/O into real storage above 2 GB using 64-bit IDAWs (the channel program must reside bellow 16 MB).
- EXCPVR provides a flag, IOBEFMT1, shown in Figure 8. If the flag is on, EXCPVR verifies that the device type allows the use of 64-bit IDAWs. The IOB Extension Block also contains a flag that indicates that the IDAWs in the channel program are using the new extended 64-bit format.

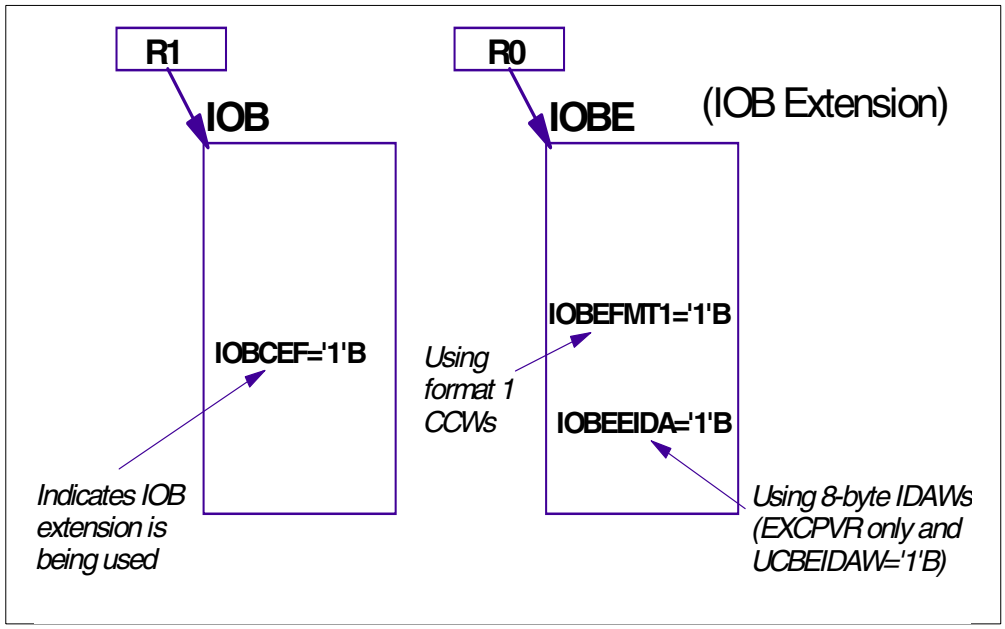



*Figure 8. Using EXCP and EXCPVR*

- Program Fetch does not load programs directly into real storage above 2 GB.
- Most users of Media Manager exploit real storage above 2 GB.

### 2.10.4 Instructions to perform I/O

Existing programs are written to use 32-bit general purpose registers. The architecture now allows for 64-bit general purpose registers, but using them requires additional supporting program changes. The following considerations should be taken when referencing large real addresses:

- The ESA/390 Load Real Address (LRA) instruction cannot be used to deal with real addresses greater than 2 GB because it places the result in a 32-bit general purpose register. Therefore, a new instruction is introduced to aid in

channel program construction which does not require the use of 64-bit registers. It is called Store Real Address (STRAG).

- The STRAG instruction has been added to accept a virtual address as one argument and then store the translated 64-bit real address at an 8-byte virtual storage location specified as the other argument.
- Many programs use the condition code returned by the LRA instruction to check whether a virtual page has been backed, but this never proved that a page was in fact fixed. It is recommended that the Test Protection (TPROT) instruction be used in z/Architecture mode to validate whether a page has been backed, but this still does not resolve the “is the page fixed?” issue. The LRA instruction can still be used if it is known that the target page is backed by a real frame below the 2 GB line. For example, if the storage was obtained with LOC=(24,24), or LOC=(...,31).

**Note:** All programs that issue an LRA instruction must be prepared to handle a 31-bit result if the virtual storage address specified could have been backed with central storage above 16 megabytes, or a 64-bit result if the virtual storage address specified could have been backed with central storage above 2 gigabytes. Issue LRA only against areas that are fixed. The TPROT instruction can be used to replace the LRA instruction when a program is using it to verify that the virtual address is translatable and the page backing it is in real storage.

## 2.11 Changes to Access Method Services, EXCP and EXCPVR

Some of the OS/390 access methods support large real storage. An attempt to enable large real storage exploitation by an application that uses an access method that does not support this enhancement leads to errors if the access method does not yet tolerate buffers backed by frames above the 2 GB location in real storage.

Some access methods acquire I/O buffers on behalf of the application, and in these cases no application changes are required to take advantage of large real storage. However, if the application acquires the buffers, it is responsible for obtaining storage with the correct attributes. The access methods specifically able to utilize large real support are:

- SAM, BSAM and QSAM, which provide DASD and TAPE support.
- VSAM, which provides support for Extended Format data sets.

Applications that use EXCP directly, as opposed to an OS/390 access method, are always responsible for obtaining the I/O buffers. Since the channel provided by the application is always translated into a real channel program, EXCP has the opportunity to translate in such a way that it supports I/O buffers backed above 2 GB in real storage. However, not all device types support buffers located above 2 GB. DASD and TAPE are supported for this release, but for other device types the UCB should be interrogated. For device types that do support large real storage, it is advisable to use the LOC parameter on the GETMAIN or STORAGE macro to allow buffers to be backed above 2 GB.

EXCPVR processing has been modified to enable the use of large real storage addressing for fixed buffers above the 2 GB line. If 64-bit IDAWs are being used, authorized programs must indicate to EXCPVR what type of IDAW is being used. If a program needs to use 64-bit IDAWs, but does not currently create an

Input/Output Block Extension (IOBE), it must be changed to create an IOBE, and set IOBCEF in the IOB to indicate that an IOB extension is being used. The 64-bit IDAW indicator is set in IOBEEIDA in the IOBE. The IOBE address must be passed to the EXCPVR in Register 0, as shown in Figure 8 on page 22.

## 2.12 Changes to VSM in z/Architecture mode

The macros GETMAIN, STORAGE OBTAIN, and CPOOL allow the caller to indicate that the virtual storage can be backed by storage above 2 GB when page fixed. In OS/390 Release 10, these macros have new parameters to support real storage above 2 GB.

Note: The request for 64-bit real storage backing on the GETMAIN and STORAGE OBTAIN is downward compatible. The macro expansion can execute on previous levels of OS/390 or MVS and is treated as a request for 31-bit real storage backing on systems that do not have real storage above 2 GB.

A request for virtual storage, which is obtained using the GETMAIN and STORAGE macros, can be tailored to identify the type of real storage that must back it when it is page fixed for I/O. The attribute is set by using the second parameter of the LOC keyword. The values and meanings of the LOC keyword on the GETMAIN macro are as follows:

**LOC=RES** This is the default, which acts like LOC=24 or LOC=31 depending upon the residency of the code that makes the request.

**LOC=24** Specifies virtual below 16 MB, backed by real storage below 16 MB when fixed

**LOC=(24,31)** Specifies virtual below 16 MB, backed by real storage below 2 GB when fixed.

**LOC=(24,64)** Specifies virtual below 16 MB, backed anywhere in real when fixed.

**LOC=31** Specifies virtual below 2 GB, backed by real below 2 GB when fixed.

**LOC=(31,64)** Specifies virtual below 2 GB, backed anywhere in real when fixed.

**Note:** The old values of BELOW and ANY continue to be accepted and relate to 24 and 31, respectively.

Since OS/390 backs non-fixed pages with any frame available, a page that must be backed below 16 MB or 2 GB when it is fixed may need to be moved if it is not already backed by the right kind of frame. If the application is acquiring the I/O buffers, and using an access method that supports large real storage, it might be changed to request storage with LOC=(24,64) or LOC=(31,64).

### 2.12.1 New parameters for GETMAIN and STORAGE

Another enhancement to the GETMAIN and STORAGE macros is the enabling of support relating to “containing boundaries” and “starting boundaries,” which can be specified using the parameters CONTBDY and STARTBDY.

Prior to OS/390 Version 2 Release 10, storage was obtained on a doubleword boundary, except when BNDRY=PAGE was specified, which could only be used for subpools that supported page boundaries.

**STARTBDY** The STARTBDY parameter indicates that the allocation must begin on a specified power of 2 alignment boundary. The STARTBDY is analogous to the BNDRY=PAGE parameter and is applied on all GETMAINs and STORAGE OBTAINs. It is not restricted to certain subpools as is the current restriction of the BNDRY=PAGE process.

**CONTDBY** This can be used to ensure that allocated storage must not span a particular power of 2 alignment boundary, and must reside within the area identified by the containing boundary. Naturally, the amount of storage being requested must not be larger than the containing boundary size.

#### 2.12.1.1 GETMAIN examples

In the following example, STARTBDY=12 represents a 4096-byte page boundary storage acquisition. The valid range is 3 to 31 and specifies the power of 2 that determines the starting boundary (for example, 12 means that the boundary is 2**12 or 4096).

```
GETMAIN STARTBDY=12,......
```

With the addition of the CONTBDY=12 parameter on GETMAIN, this ensures a containment boundary of 4096 bytes and that the storage obtained would not cross the specified boundary.

```
GETMAIN STARTBDY=12,CONTBDY=12......
```

**Note:** STARTBDY and CONTBDY are not valid with LOC=EXPLICIT or BNDRY=PAGE.

### 2.12.2 CPOOL macro support

Prior to OS/390 Release 10, a cell pool managed by CPOOL was aligned on a doubleword boundary. In OS/390 Release 10, CPOOL now supports alignment on a quadword boundary, as requested by the BNDRY=QWORD parameter. The user must ensure that the cell pool is a quadword multiple, which ensures quadword boundary alignment.

## 2.13 Changes for both ESA/390 and z/Architecture mode

The following enhancements are related to both ESA/390 and z/Architecture modes:

- Boundary alignment and containment options have been added to the GETMAIN and STORAGE macros.
- Quadword alignment for CPOOL.
- ESTAE request for SDWA above 16 MB.
- SVC UPDATE and SVC SCREENING facilities have been enhanced to support extended SVC routines.
- The IPCS IPCSDATA command has been enhanced.

## 2.14 Unformatted dump changes in OS/390 Release 10

All unformatted OS/390 R10 MVS dumps are written using a new format. Release 10 supports a limited number of utilities that accept either dumps from prior releases or dumps from Release 10. Utilities supplied with the most current release must be used. Those supplied with prior releases are not upgraded to support the latest.

While the unformatted dump logical record length remains at 4160 bytes, the 64-byte record prefix has been changed to accommodate the new architecture. The first 2 bytes in Release 10 dump records contain the character string “DR2.” Prior releases are identified with the “DR1” string.

Bytes 20-27 of the dump record prefix now contain an 8-byte (64-bit) address. Prior to OS/390 Release 10, these 8 bytes contained a 4-byte (31-bit) address plus a 4-byte dump sequence number. The dump sequence number in Release 10 now occupies bytes 28-31of the dump record prefix.

Only the OS/390 Release 10-supplied Interactive Problem Control System (IPCS) should be used to process Release 10 dumps.

## 2.15 Service aids support for 64-bit real storage

Service aids that have been upgraded to support 64-bit real storage include:

- IPCS, which is used for dump and trace analysis, has undergone some major enhancements that are discussed in more detail in 2.15.1, “Changes to the Interactive Problem Control System (IPCS)” on page 26.
- SPZAP, which enables you to dynamically update programs and data sets.
- System trace (SYSTRACE), which records system event data.
- Generalized Trace Facility (GTF) for more selective tracing.
- Component Trace (CTRACE) for event recording related to specific components and subcomponents.

### 2.15.1 Changes to the Interactive Problem Control System (IPCS)

IPCS has been enhanced to enable interrogation of 64-bit addresses and lengths. As with changes to other system components implemented in this release, the IPCS 64-bit updates represent some basic functional changes. These changes are the early evolution changes that are enhanced throughout the product’s life cycle. The IPCS inventory is compatible with earlier OS/390 releases to enable compatibility when sharing a sysplex dump directory.

A major change has been the introduction of the underscore character to delimit the low-order 8 bytes from the high-order 8 bytes. For example, a 64-bit address would be represented as “FFFFFFFF_00F6AC36”.

IPCS display and browse commands have been changed to handle 64-bit addressing, but essentially remain unchanged for displays relating to addresses below 2 GB. The difference is that when the address is above the 2 GB line, it displays the low-order 7 hexadecimal digit address prefixed with an underscore.

Displays showing offsets have also been changed to include both offsets and addresses as captions to the left of the data, but only when the range of offsets can be accurately represented with 5 hexadecimal digits. Offsets larger than 5 digits generate a format that only displays the address. The following IPCS general purpose line mode display example shows the changes reflected when reviewing data that resides above the 31-bit bar.

```
CVT - Communications Vector Table
LIST FFFFFFFF_00FD1DD0. ASID(X'0001') POSTITION(X'-0028') LENGTH(X'0528')
      STRUCTURE(CVT)
ASID(X'0001') ADDRESS(FFFFFFFF_00FD1DA8) KEY(00) COMMON
-00028 _0FD1DA8.                  E2D7F64B F04BF840  |         SP6.0.8
-00020 _0FD1DB0. C8C2C2F7 F7F0F340 40404040 40404040 |HBB7703
-00010 _0FD1DC0. 40404040 40404040 00009672 F0F3F840  |         ..o.038
-00000 _0FD1DD0. 00000218 00FDD560 00FD67EC 00FD23B8  |......N-........
+00010 _0FD1DE0. 00000000 00FF9FFC 00FF6CEE 00FE83B4  |..........%...c.
```

Using the IPCS Browse option would display the storage as follows:

```
ASID(X'0001') ADDRESS(FFFFFFFF_00FD1DA8.) STORAGE-------------------------------
Command ===>                                                  SCROLL===> CSR

_0FD1DA8                      E2D7F64B  F04BF840  |.        SP6.0.8
_0FD1DB0     C8C2C2F7  F7F0F340  40404040  40404040  |HBB7703
_0FD1DC0     40404040  40404040  00009672  F0F3F840  |         ..o.038
_0FD1DD0     00000218  00FDD560  00FD67EC  00FD23B8  |......N-........
_0FD1DE0     00000000  00FF9FFC  00FF6CEE  00FE83B4  |..........%...c.
```

Data displayed using dumps captured in z/Architecture mode for addresses below the 31-bit bar would appear as in the current ESA/390 IPCS displays, without the leading underscore.

## 2.16 Slip trap processing changes

Prior to OS/390 Release 10, a SLIP command used R to indicate a register. For example, SLIP DATA=(1R,EQ,3). OS/390 Release 10 introduces a new attribute to handle 64-bit registers. The G attribute is used to ensure that comparisons and calculations are performed using all 64 bits of the specified register. For example, SLIP DATA=(1G,EQ,5).

The R register continues to relate to the low-order 32 bits when running in z/Architecture mode on a 64-bit compliant processor, or the available 32 bits when running in ESA/390 mode. If the G register is used when running in ESA/390 mode, it is processed exactly as the R register.

Prior to OS/390 Release 10, SLIP traps could specify 24-bit (“%”) and 31-bit (“?”) indirection attributes. For example, 2R?+64?. OS/390 SLIP processing has been enhanced to include 64-bit indirection support via “!”. Since OS/390 does not support 64-bit virtual storage, this has limited use, but has laid the groundwork for future support.

Support for hexadecimal strings longer that 8 digits has also been implemented, along with the underscore to manage the longer strings more efficiently. For example, DATA=(15G,EQ,1122334455667788) can be managed by specifying DATA=(15G,EQ,11223344_55667788), or even DATA=(15G,EQ,1122_3344_5566_7788), both of which are easier to read than the continuous string. This delimiting can be used for all register types, for any string lengths.

## 2.17 Invocation of system services in 64-bit mode

In general, BCP services do not support 64-bit interfaces, unless explicitly stated otherwise. However, depending upon the mechanism of invocation, services may tolerate invocation in 64-bit mode, in which case their operation is as when the invocation is in 31-bit mode. Following is a description of the behavior based on mechanism of invocation:

**Branch entry** Invocation in 64-bit mode is not supported. Results are unpredictable. It is the responsibility of the invoker to switch out of 64-bit mode before invoking these services.

**SVC** Since this form of invocation involves a switch to a system-defined addressing mode, invocation in 64-bit mode is tolerated. However, all addresses passed on the interface are treated as in previous releases for 31-bit mode invocation.

**Non-stacking PC** Invocation in 64-bit mode is not supported. There is an architectural protection against a 64-bit mode program invoking a 24- or 31-bit routine with a non-stacking PC. A program exception occurs. It is the responsibility of the invoker to switch out of 64-bit mode before invoking these services.

**Stacking PC**: Since this form of invocation involves a switch to a system-defined addressing mode, invocation in 64-bit mode is tolerated. However, all addresses passed on the interface are treated as in previous releases for 31-bit mode invocation.

### 2.17.1 Changes based on the addressing mode

Some interfaces may be different on OS/390 Release 10 operating in z/Architecture mode depending on whether they are invoked in 24-bit, 31-bit, or 64-bit addressing mode.

As mentioned previously, several areas of the operating system are affected by the implementation of OS/390 Version 2 Release 10 real storage support as follows:

- The necessity to support two different architectures, the ESA/390 and the z/Architecture mode
- Changes to I/O processing for the EXCP, SAM, BSAM, QSAM, and VSAM access methods
- Enhanced register status information
- Hiperspace usage
- Page steal process changes

In a 64-bit real storage environment the terms “above the bar” and “below the bar” are used to identify the areas between 2**31 and 2**64-1, and 0 and 2**31-1 respectively. For example, any address in the range 0 to 7FFFFFFF would be below the bar, and addresses in the range FFFFFFFF to 7FFFFFFF_FFFFFFFF would be above the bar. This is basically an alteration to the 2 GB 31-bit terminology that related “below the line” to 24-bit storage, and “above the line” to 31-bit addresses.

## 2.18 z/Architecture addressing mode considerations

z/Architecture mode introduces a new addressing mode: 64-bit mode. Programs may execute in either 24-, 31-, or 64-bit mode. Although the architecture supports 64-bit addressing mode and limited areas of the operating system actually use it, most programs can completely ignore its existence. Nevertheless it is important to understand how 64-bit addressing mode works and how one can switch from one addressing mode to another.

As depicted in Figure 10 on page 32, the z/Architecture mode PSW has 128 bits, of which the low-order double-word represents the instruction address. The addressing mode bits are located in the high-order doubleword. The 128-bit PSW is remapped to 64-bit when stored in control blocks (see Figure 10 on page 32). Address generation in 64-bit mode uses 64-bit Base and Index values and produces 64-bit addresses.

### 2.18.1 Non-modal instructions

Existing 31-bit operand instructions behave as in ESA/390 regardless of the addressing mode; this includes RR and RX instructions like the following:

```
LR, AR, ALR, L, A, and AL
```

The high-order word of the 64-bit GPRs is unmodified.

New 64-bit operand instructions operate on 64-bit operands regardless of addressing mode; this includes RR, RX and RS instructions such as:

```
LGR, AGR, ALGR, AGR, AG, and ALG
```

All 64-bits of the GPRs are modified.

New 64-bit register and 32-bit storage operand instructions operate on 64-bit and 32-bit operands, regardless of the addressing mode; this includes RR and RX instructions such as:

```
LGFR, AGFR, ALGFR, LGF, AGF, and ALGF
```

Bits 0-31 of the 64-bit GPR are either cleared or the sign is propagated.

### 2.18.2 Modal instructions

For modal instructions that have operand addresses in a GPR, the address is interpreted according to the addressing mode. For instructions that return addresses in a GPR, the bits of the GPR are set according to the addressing mode. Examples of these instructions are:

```
LA, MVCL, TRT, BXLH, and BASSM
```

### 2.18.3 Changing addressing mode

There are three instructions that change the addressing mode without branching:

- Set Addressing Mode to 24-bit (SAM24)
- Set Addressing Mode to 31-bit (SAM31)
- Set Addressing Mode to 64-bit (SAM64)

There are 2 instructions that change addressing mode and branch:

- Branch and Save and Set Mode (BASSM)
- Branch and Set Mode (BSM)

In all modes, BASSM and BSM switch the addressing mode based on the contents of R2 in bits 63 and 32 (see Figure 9 on page 30).

**64-bit mode** BASSM sets a 64-bit return address with R1 bit 63 set to 1.

BSM sets bit 63 in R1 to 1 when R1 is not 0. The instructions BALR and BASR do not set bit 63 to 1 in R1.

**24/31-bit mode** These instructions set a return address as in ESA/390, and leave R1 bits 0 through 31 unchanged.

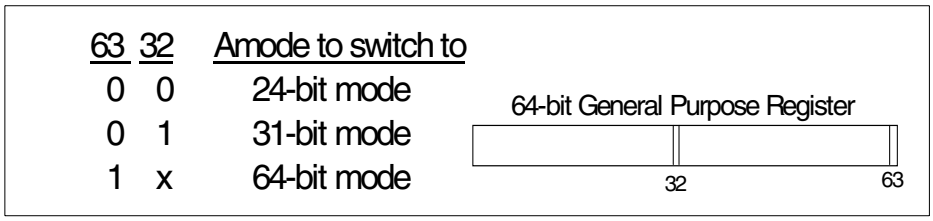



*Figure 9. Mode switching branches*

### 2.18.4 Addressing mode instructions

There are several other new instructions that are related to addressing mode:

**LLTG, LLTGR** Load Logical Thirty-one - Effectively loads the low-order 31 bits from a word in storage into the low-order 31 bits of a register and clears high-order 33 bits to zero.

**TAM** Test Addressing Mode - Sets the condition code based on current addressing mode.

**LMH, STMH** Load/Store Multiple High - Loads or Stores the top halves of a range of registers.

**BRCL, BRASL** Branch Relative on Condition Long (BRCL) and Branch Relative and Save Long (BRASL**)** - Like the BRC and BRAS instructions, but they have a 32-bit signed immediate value.

**LARL** Load Address Immediate Long

## 2.19 Dual architecture support

The support for the z/Architecture mode includes:

- 64-bit General Purpose Registers
- 64-bit Control Registers

- New format Program Status Word (PSW) (see Figure 10 on page 32)
- 8K Prefix Area
- New format Page and Segment Tables
- New format Linkage Stack, Entry Table and Address Space Second Table Entries
- Mode Tracing and new format Trace Entries
- New format IDAWs

There are, however, some restrictions in the first implementation of z/Architecture mode as follows:

- Central storage is limited to 128 GB.
- No support for 64-bit virtual addresses.
  - User address spaces and data spaces are limited to 2GB
  - PER ranges must be bellow 2GB.

When using z/Architecture mode 64-bit support, real storage is used, and the requirement to page to expanded storage is not supported, nor should it be necessary for the system to page at all, unless you attempt to use more real storage than is available. The overhead when all of real storage is exhausted is greater because paging is from real storage to auxiliary storage devices, and not from real storage to expanded as before.

Support for 64-bit General Purpose Registers (GPRs)

OS/390 Release 10 running in z/Architecture mode maintains the full 64-bit General Purpose Registers (GPRs) (see Figure 10 on page 32), but it is not expected that applications will make use of this feature in the immediate future due to the fact that virtual storage is limited to the 2 GB upper boundary. This virtual storage constraint also limits the scope for systems services to utilize 64-bit GPRs, but as this is the first generation supporting 64-bit addressing, in this new OS/390 evolutionary phase, the implementation of extending 64-bit functionality will itself be an evolutionary process.

There are, however, some considerations on the way the 64-bit GPR interface is used:

- For registers that are defined as input, the contents of the high-order word of 64-bit GPRs are ignored.
- For registers containing output values, the result is in the low-order word and the contents of the high-order word are unpredictable.
- For registers that are defined as preserved on an interface, the high-order word is also preserved.
- The register is 64-bits for:
  - Address generation in 64-bit mode
  - GPR operands of modal instructions in 64-bit mode
  - GPR operands of non-modal 64-bit instructions
- The register is 32-bits for:
  - Address generation in 24/31-bit modes
  - GPR operands of modal instructions in 24/31-bit modes

– GPR operands of non-modal 32-bit instructions

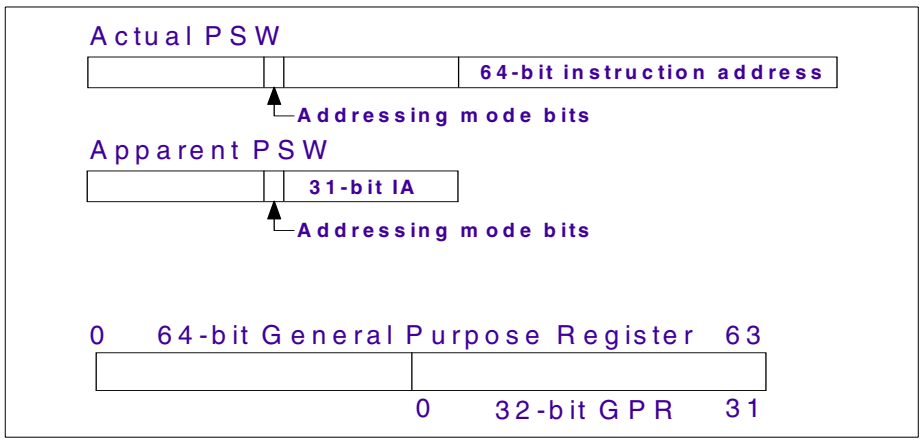



*Figure 10. z/Architecture mode General Purpose Registers and PSW*

## 2.20 Migration scenarios

As shown in Figure 11, migration should be implemented one step at a time. It can start by either migrating to OS/390 Release 10 using the current processor, or keeping the current OS/390 and replacing the current processor. The next step is to install the new processor (if OS/390 Release 10 has already been installed) or to migrate to OS/390 Release 10 (if the new processor has been installed).

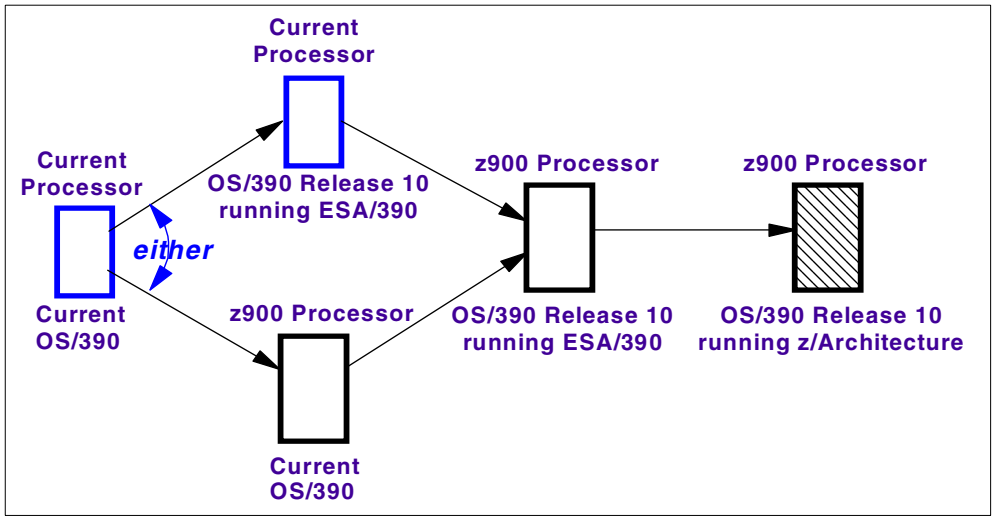



*Figure 11. Migration flexibility*

In this stage, testing of the 64-bit support can be performed. As shown in Figure 12 on page 33, this testing is nondisruptive; the installation can start running its system as shown in the initial configuration: three partitions, two running production and one set aside for testing. The testing partition can have its processor memory configured as central storage without disrupting the work in any other partition. The change that must be done to the operating system is merely to update an initialization parameter to bring the system into z/Architecture mode. To achieve maximum benefits, other customizing actions can optionally be taken.

Once the operating system is thoroughly tested and the 64-bit system is ready to be deployed, the other two partitions can be redefined and start fully utilizing the capabilities provided by OS/390 Release 10.

If there is a problem during the test, it is possible to back out, that is, to redefine the storage in partition 2 as central and expanded, and continue its utilization as before.

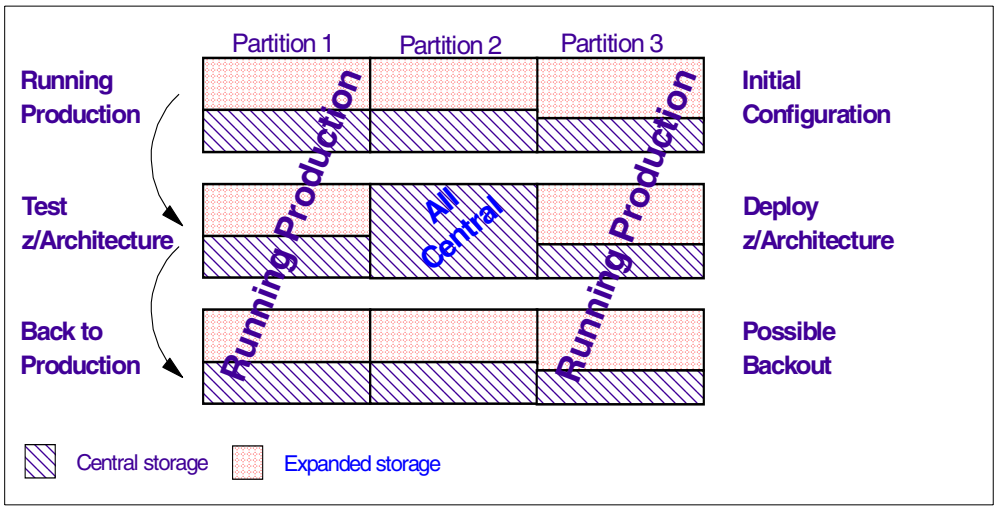



*Figure 12. Nondisruptive testing and migration*

Migration to 64-bit support is mostly transparent. No application programming interfaces are changed incompatibly. All the differences in the architectural environment are absorbed by the operating system. Even low-level authorized services remain compatible. The only exception is authorized programs dealing with real storage addresses. These programs must be examined to determine if any change is required.

If the installation is running in Parallel Sysplex, it can intermix systems running 32- and 64-bit architectures, as shown in Figure 13. The installation can have multiple sysplexes with mixed processors, CFs, and architectures, and still run a single level of the OS/390 operating system.

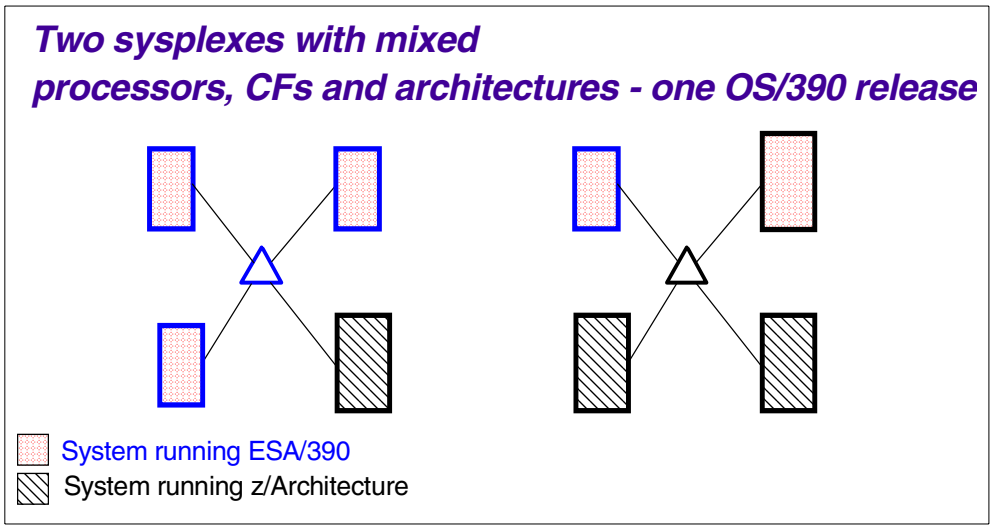



*Figure 13. Single OS/390 level across an enterprise*

OS/390 Release 10 is capable of running on both architectures, giving the installation the benefits of running a single level of the operating system.

# Chapter 3. System Logger

The System Logger was introduced with MVS/ESA Version 5.2. It is a set of services that allows an application to write, browse and delete log data. You can use System Logger services to merge data from multiple instances of an application, including merging data from different systems in a sysplex.

An OS/390 System Logger configuration includes the System Logger address space in each system of a sysplex, the LOGR couple data set, a log stream structure in a coupling facility, DASD log data sets for off-loaded data from the coupling facility log stream, and optionally either staging data sets or a dataspace for a backup copy of the log blocks residing in the log stream structure. See Figure 14.

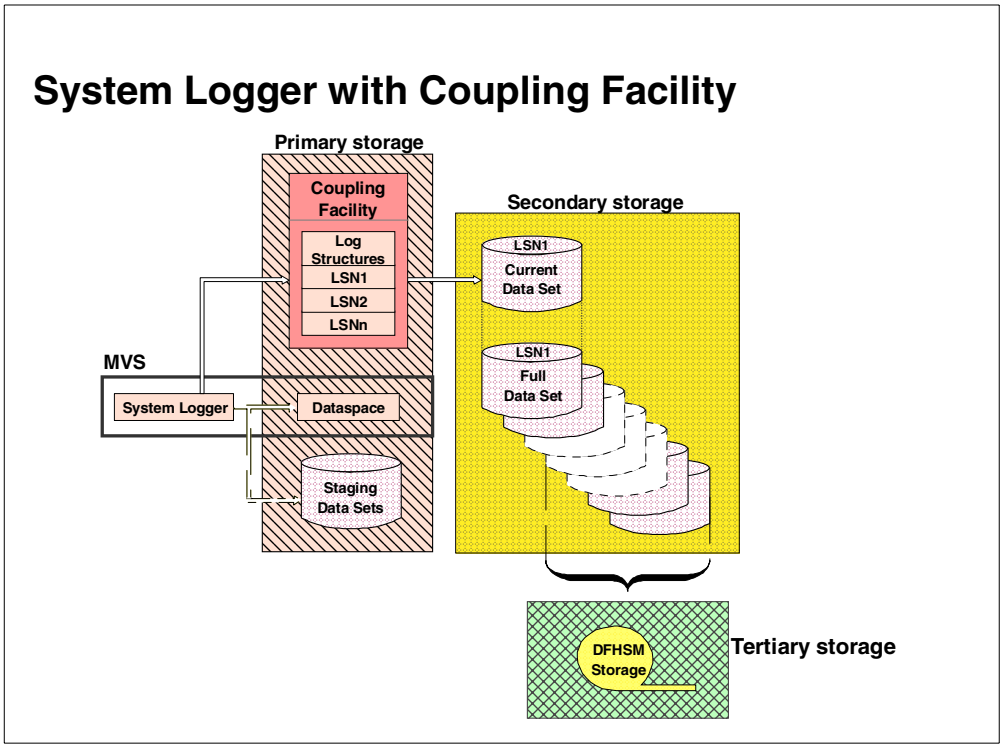



*Figure 14. System Logger working with a coupling facility*

## 3.1 System Logger enhancements in Release 3

At that time you could only use it with structures inside the coupling facility. In OS/390 Release 3, there were some major enhancements as follows:

- Automatic deletion of log data whose user-defined retention period has expired.
- Providing archival support for log data to remain in the log stream as long as required by the customer.
- Removal of the limit of 168 log data sets per log stream. It was replaced by a virtually unlimited number of data sets that can be defined.

- Ability to allocate staging data sets during rebuild failures, to dump the data space log data relating to the log stream. This ensures that a non-volatile copy of the log data is maintained and protected between the failure and its correction.
- Support for remote site recovery, which allows a log stream data at a local site to be duplicated at a remote or secondary site. In this case, the logger allows transmitted log data to be written into a log stream at the remote site with a block ID and GMT time stamp identical to the block ID and GMT time stamp at the local site.
- Restructuring of the System Logger connect processing to provide more parallelism. The results are better performance at connect time, because connects for log streams in different coupling facilities can be processed in parallel.

Before OS/390 Release 3, only SYS1.LOGREC and OPERLOG were able to write to log streams. Starting with Release 3, other products and services requiring System Logger were introduced, as follows:

- CICS Transaction Server for OS/390
- IMS/ESA Common Queue Server
- Remote Site Recovery
- Resource Recovery Services (RRS)

There were incompatibilities between the OS/390 version of the System Logger and its predecessors, as follows:

- The size of the couple data sets, due to a new record type called DSEXTENT, which is used for the number of directory extents for which the LOGR couple data set should be formatted.
- The ability of allocating staging data sets for log streams during rebuild recovery of the log stream structure.

These incompatibilities resulted in reformatting the LOGR couple data sets using the IXCL1DSU utility.

## 3.2 System Logger enhancements in Release 4

With the introduction of OS/390 Release 4, a second type of log stream was introduced, the DASD-only log stream. DASD-only log streams support non-sysplex environments or sysplex environments where only one system maintains a log stream. This environment supports products like CICS Transaction server, which require System Logger services. In a DASD-only log stream, the log data is contained in local storage buffers in a dataspace that is backed up by DASD staging data sets. Upon reaching a threshold in the dataspace, records are then off-loaded to DASD log data sets.

A DASD-only log stream has the following scope:

- A single system scope.
- Only one system can connect to it or write to it.
- Multiple applications from the same system can connect and write to it.

A system that uses DASD-only log streams has to be either in multisystem or monoplex mode. A DASD-only logger configuration is shown in Figure 15.

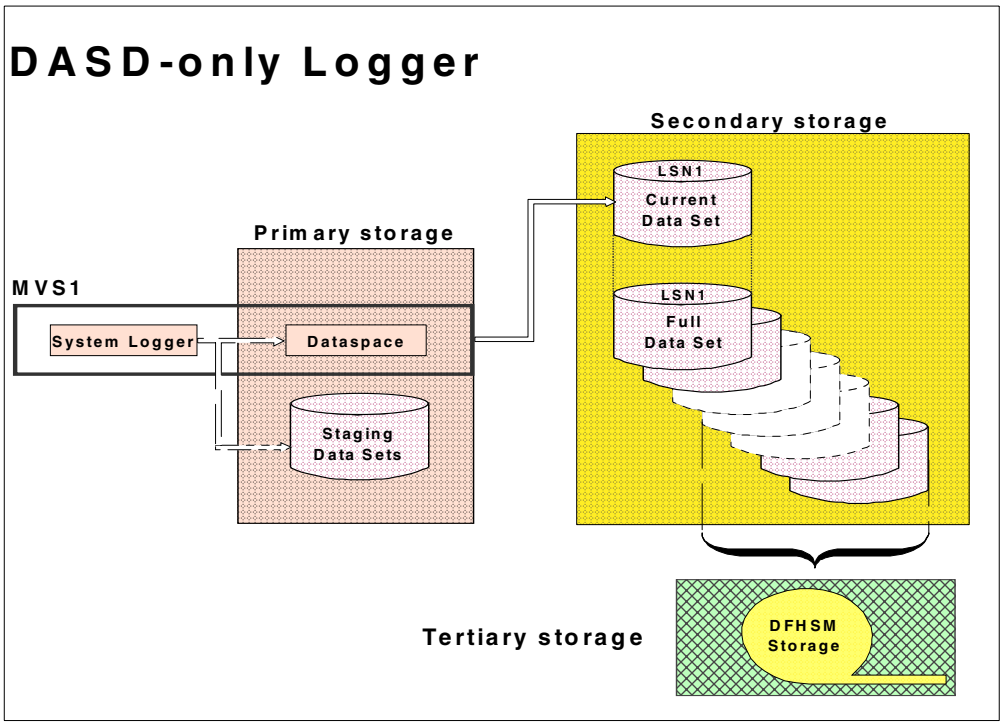



*Figure 15. DASD-only logger*

## 3.3 System Logger enhancements in Release 10

System Logger provides a set of performance enhancements in OS/390 Release 10, as follows:

- XES auto alter
  This enhancement provides a function so that XES monitors and tunes the coupling facility structure size and storage ratios in real time in response to changing structure usage.
- Logger latching performance enhancements
  Logger browse or delete requests via the logger services IXGBRWSE and IXGDELET can inhibit other work from occurring for the same log stream on the same system.
- Rebuild ENQ contention reduction
  This reduces unnecessary ENQ contention between logger systems participating in a rebuild.
- Multi-block support for browse
  With the introduction of the MULTIBLOCK keyword, the IXGBRWSE service has been enhanced to return multiple log blocks with a single request.

### 3.3.1 XES Auto Alter

When defining structures at the OS/390 Release 10 level, you can specify whether you want the system to automatically alter a structure when it reaches an

installation-defined or defaulted-to percent full threshold. This new support may do one of the following or both:

- Increase the size of the structure
- Rearrange the objects within the structure

#### 3.3.1.1 Using Auto Alter

For a structure to be eligible to automatically be altered, the following is required:

- Define a new parameter in the structure definition for the CFRM policy as follows:

  ALLOWAUTOALT(NO|YES)

  If this parameter is used by System Logger with the value YES, then it may affect those structures.

- The application specifies that the structure may be altered.

  When an application connects to a coupling facility list structure using the IXGCONN macro, the System Logger issues the following request to connect to the coupling facility list structure:

  IXLCONN ..... ALLOWALTER(YES)

  The System Logger detects that ALLOWAUTOALT(YES) was specified in the list structure definition in the CFRM policy. Cross System Extended Services (XES) can then automatically alter the size and ratio attributes of the structure as needed.

  System Logger manages the usable space within its own structure by allowing each log stream to receive 1/n of the portion of the number of active connections. System Logger does off loading of log data from the coupling facility to DASD staging data sets when their specified thresholds are met. So it is very possible that logger structures will appear to be only half full and logger structures are also good candidates for XES to take its space for other constrained structures.

Because of these facts you should consider allowing Auto Alter processing for other structures than logger structures first. If you specify ALLOWAUTOALT(YES) in the CFRM policy, then we recommend that you also specify the parameter MINSIZE(n) in the same policy. The value of *n* should be chosen in a way that enables the logger to handle the expected number of concurrent active log stream connections.

Figure 16 on page 39 shows an example for a CFRM policy with the above mentioned parameters.

```
STRUCTURE NAME(STRUCT1) SIZE(7000)
          INITSIZE(4000)
          MINSIZE(2000)
          FULLTHRESHOLD(75)
          ALLOWAUTOALT(YES)
          PREFLIST(CF01,CF02,CF03)
```

*Figure 16. CFRM policy with new parameters ALLOWAUTOALT and MINSIZE*

### 3.3.2 Logger latching performance enhancements

Another enhancement for logger in OS/390 Release 10 is the log stream latch contention. This enhancement improves the overall performance of application and subsystem use of System Logger. Restart times for IMS/ESA Common Queue Services (CQS) are improved. IMS/ESA CQS uses the browse function (IXGBRWSE) of System Logger.

### 3.3.3 Logger rebuild ENQ contention reduction

The reduction of unnecessary ENQ contention between logger systems leads to significant improvement when a number of systems are connected to a set of log streams and the structure that they are connected to is rebuilt.

This enhancement is shipped in OS/390 Release 10 first, but is also rolled back to Release 5 via PTF. Refer to APAR OW40811.

### 3.3.4 Multi-block support for browse

Before OS/390 Release 10, only one log block could be read per IXGBRWSE request at a time, although the logger did some buffering of data read from a log stream. This led to linkage overhead and front-end processing overhead.

With the introduction of the MULTIBLOCK keyword on the IXGBRWSE macro, the need for repetitive calls to the IXGBRWSE service to obtain consecutive log blocks is greatly reduced. This is done by increasing the DASD I/O buffer size used by the logger from approximately one quarter cylinder to half a cylinder. IMS/ESA Common Queue Service (CQS) was the initial reason for implementing this interface.

The IXGBRWSE macro has the following types of requests:

**START** Establishes a browse session and sets the cursor position.

**READCURSOR** Reads the first or next consecutive log block (this was done block by block in the past).

**READBLOCK** Reads the selected log block.

**RESET** Resets the cursor to either the beginning or the end of the selected log stream.

**END** Ends a browse session.

This function is shipped with OS/390 Release 10. It is also rolled back to OS/390 Release 6. Refer to APAR OW42631.

The related APARs for IMS/ESA are PQ38036 (Version 6.1) and PQ38039 (Version 7.1).

When you use the IXGBRWSE macro in your application, you should be aware of GPR 15 (general purpose register). If it contains a return code of 4 and GPR 0 contains a reason code of 0416, this means data end, but one of the blocks read may be in error.

You should also be aware of return code 4 and reason code 0417. This indicates an exception condition like end-of-file. Data has been read, but there are some warnings.

If you get one of these reason codes you can do the following, depending on your application and on how critical your data is:

- Accept this condition and continue with reading.
- Stop processing the log.
- If it is possible, attempt to get the problem rectified, and then attempt to re-read the log data.

# Chapter 4. System Display and Search Facility (SDSF)

OS/390 Release 10 SDSF requires the following operating system environment:

- BCP at OS/390 Release 10
- JES2 at OS/390 Release 4 through Release 10

SDSF is enhanced in OS/390 Release 10 to improve its support of the sysplex environment. SDSF provides improved system management capabilities in the sysplex environment, and addresses some critical customer requirements related to sysplex.

SDSF's job displays have always been MAS-wide. In previous releases, SDSF added support for sysplex-wide Display Active and Operlog displays. SDSF's device displays, however, have remained limited to a single system: the system the user is logged on to. As sysplex becomes increasingly prevalent, and as the number of systems in customer sysplexes increases, the limitation of the device displays to a single system becomes an impediment to implementing sysplex. With OS/390 Release 10 SDSF, the Initiator and Printer displays are enhanced to show sysplex-wide data.

**Note:** The new sysplex function on the Initiator, Printer, Output Data Set, and SYSLOG panels requires the installation of MQSeries for OS/390 Version 2 Release 1. If MQSeries is not installed or available, the panels operate as they did in prior releases.

The following SHARE Requirements have been satisfied by this release:

*Table 1. User requirements satisfied by SDSF*

| User requirement | Function requested |
|---|---|
| **REQ00072211 (SSJES298251)** | Provide a mechanism to support a full WTOR list. |
| **REQ00068439 (SSJES296001)** | Data set display should work the same across a sysplex. |
| **REQ00031376 (SOJES290001)** | Allow INIT and PR commands on a MAS. |
| **REQ0062273 (SSJES294010)** | PR display enhancement to support PSF modify commands. |

The enhancements in OS/390 Release 10 SDSF include:

- SDSF Primary Option Menu reorganization
- Improved handling of WTORs
  - Filter WTORs on log panels
  - New panel for system requests
- Mixed case column titles
- New action characters and overtypes
- New SDSF commands
- WHO command enhancement

- Improved management of SDSF parameters, simplifying preparation of SDSF server initialization statements through the use of conditionals and system symbols.
- New ISFPARMS statements
- Server registration with ARM
- MQSeries support for SDSF sysplex-wide displays
- Sysplex-wide panels
  - Sysplex device display support
  - Improved SDSF browse and log display
- The SDSF Configuration Assistant simplifies the migration to SDSF's new sysplex support. This wizard collects user input and generates results, such as JCL or RACF commands. As SDSF enhances and adds displays, it increases the complexity of the already complex task of defining security for SDSF function. SDSF adds a Web-based tool to assist in defining SAF security. This tool requires a Web browser. The minimum level for the most common Web browsers, Netscape Navigator and Microsoft Internet Explorer, is 4.0.

Other improvements to SDSF include support for new function in OS/390 JES2, additional columns on the Printer panel, support for mixed-case system commands, new codepages, and extending displays of printers and initiators to show sysplex-wide data.

SDSF's function to browse a job's output has been limited by the inability to show the most recent data when the job is running on a system other than the one the user is logged on to. The function to browse the syslog has the same limitation. This limitation is removed in OS/390 Release 10 SDSF.

These new functions, as well as their implementation and the problems they solved, are described in the following sections.

## 4.1 SDSF Primary Option Menu reorganization

For readability, and to accommodate new panels, the SDSF main menu is reorganized to group panels by type (job, device, system resource). It uses two columns to take better advantage of the screen width, and abbreviates the next describing each display. The menu continues to include only those panels the user is authorized to access.

The main menu reorganizations will affect the general users because they will see the menu options in an abbreviated, two-column format, as shown in Figure 17 on page 43.

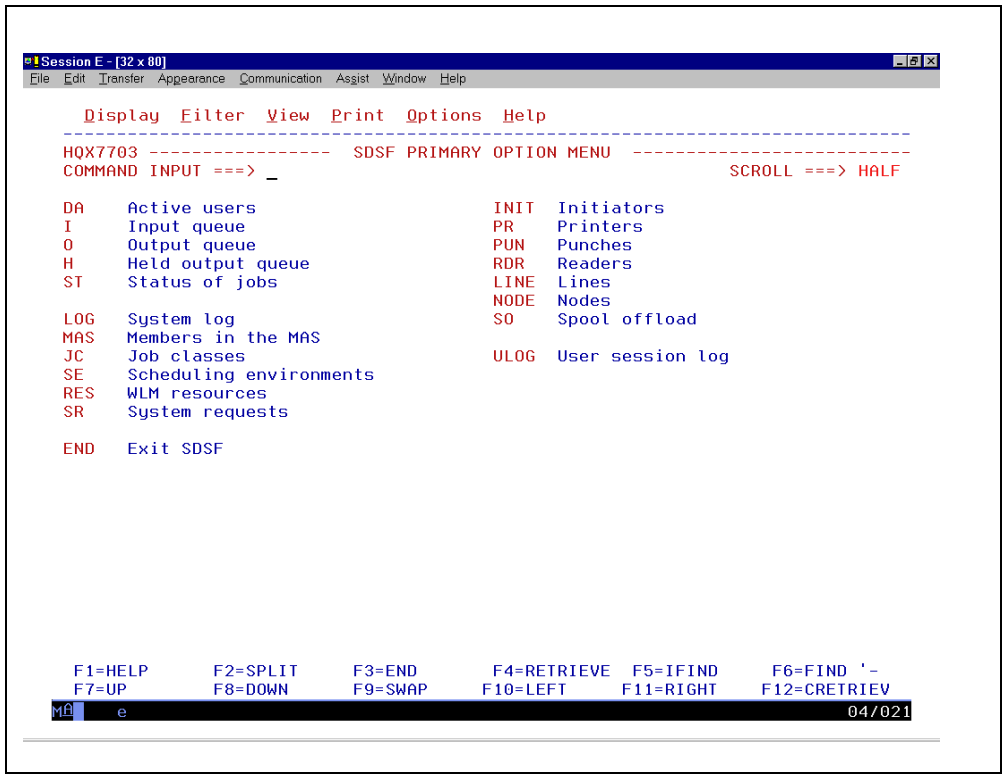



*Figure 17. SDSF Primary Option Menu*

## 4.2 Improved handling of WTORs

The ability to easily see and respond to outstanding WTORs is a key function of SDSF's log display. SDSF displays the outstanding WTORs for all systems after the last line of syslog data on the log (Syslog and Operlog) display. The high volume of messages in the sysplex environment can impact the usability of this design, as the WTORs are often forced off the screen. OS/390 R10 SDSF adds enhancements to make it easier to display and respond to system requests in a sysplex environment. There are two enhancements:

- Filtering WTORs on the log panel
- A new panel to display system requests

### 4.2.1 Filtering WTORs on log panels

SDSF adds the RSYS command to allow users to limit the WTORs to one or more systems. Authorization to the RSYS command is controlled through the AUTH parameter of ISFPARMS or an SAF resource. Users with authorization to the RSYS command can filter WTORs by system. The total number of outstanding WTORs (as opposed to the number being displayed after filtering) is shown on the title line of the log panels.

SDSF uses the RSYS parameter in ISFPARMS to set an initial default. RSYS can be used in the following three ways:

1. Specify RSYS with a ? to display the current setting for RSYS, as shown in Figure 18 on page 44. Also shown are systems SC63, SC64, and SC65 at the

bottom of the log. This indicates that the current value of RSYS is for all systems or RSYS *.

2. Specify RSYS system-name to display only WTORs from that system.
3. Specify RSYS with no parameters to display WTORs from the system you are logged on to.

There are other SDSF commands that affect the use of RSYS, as follows:

**ACTION** Limits WTORs by action code or turns them off in the log.

**FILTER** Limits the OPERLOG panel, so be careful not to turn off the system-name.

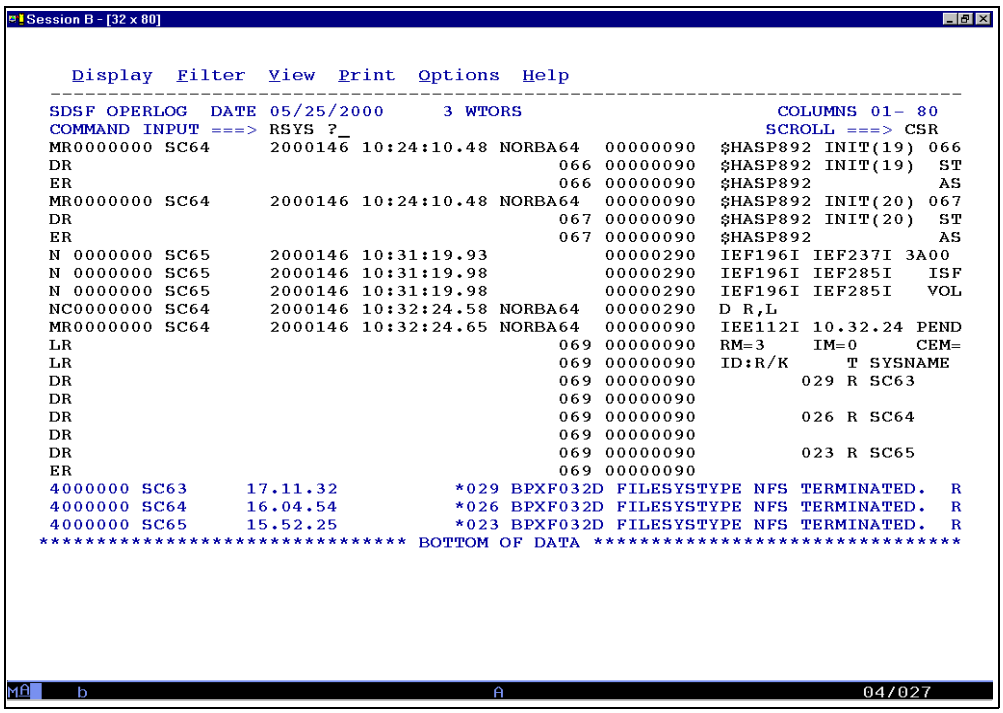



*Figure 18. RSYS command to display the current settings for RSYS*

Entering RSYS ? then displays Figure 19 on page 45, which shows the current value of RSYS. You can overtype the value * with SC64 to change the systems to be displayed.

The result of overtyping SC64 is shown in Figure 20 on page 45, where only WTORs for SC64 are shown.

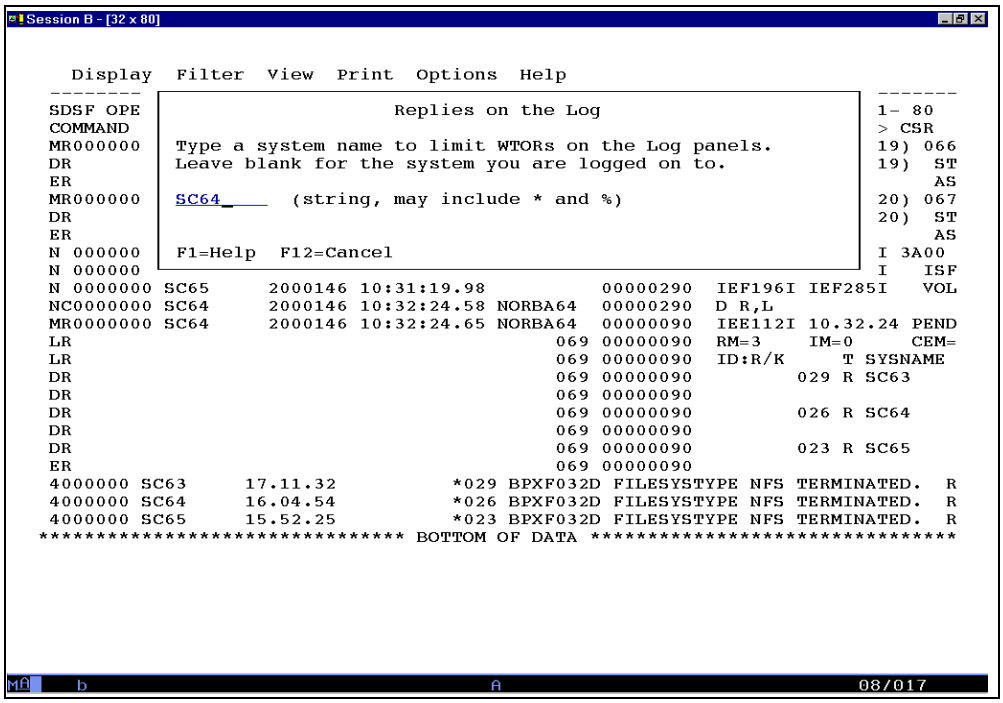



*Figure 19. RSYS ? created pop-up to display current, or change, system name*

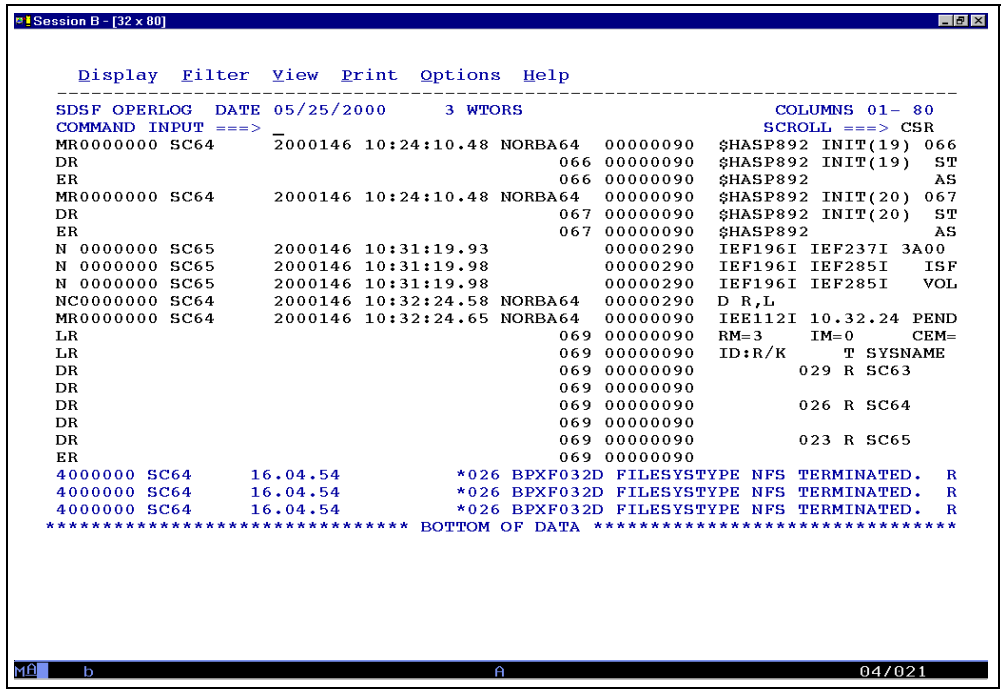



*Figure 20. RSYS display showing WTORs for SC64*

### 4.2.2 New panel to display system requests

OS/390 R10 SDSF adds a new tabular System Requests (SR) panel for outstanding system requests, which includes WTORs and action messages. With the new panel, users can easily find and respond to these messages. The panel

has all the benefits of SDSF tabular panels, such as the ability to sort rows, arrange columns, filter rows, and so on.

The system request display can be entered from the SDSF Primary Option Menu, shown in Figure 17 on page 43, by entering SR. Authorization to the SR panel is controlled with the AUTH parameter of ISFPARMS or an SAF resource.

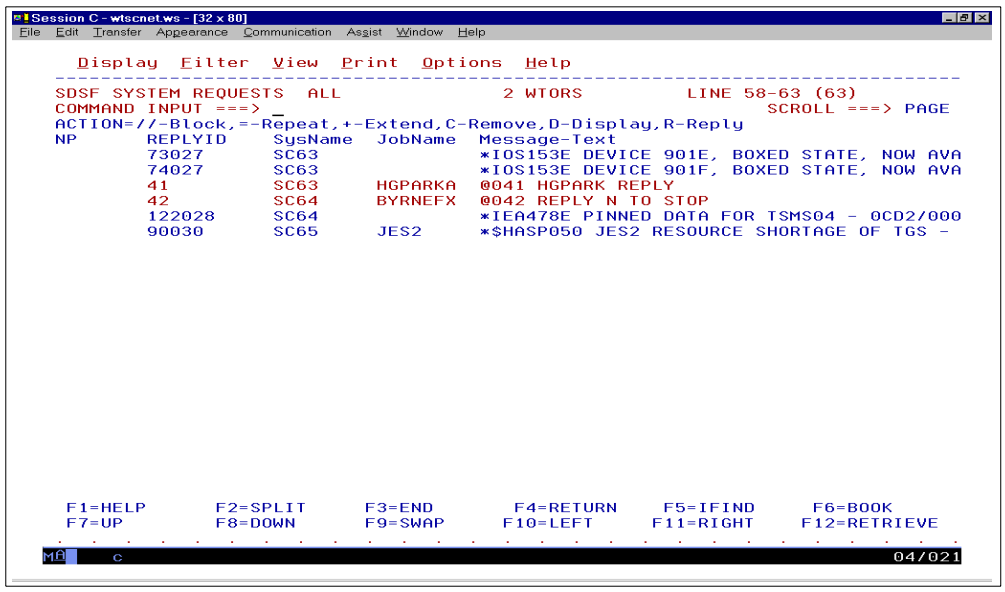



*Figure 21. SR command display panel*

This displays all system request messages. Parameters on the command that displays the panel (SR) can be used to limit the display, for example, to only show replies, tape mounts, or action messages, as follows:

- SR ALL
- SR ACTIONS
- SR MOUNTS
- SR REPLIES

This new panel compliments the RSYS command, which allows users to filter WTORs on the log panels. Filtering the new panel is accomplished with the filter function, which is common to all tabular panels and provides greater flexibility than RSYS.

In previous releases, SDSF showed the outstanding WTORs for all systems below the last line of syslog messages on the log display, as shown in Figure 18 on page 44. This design has limitations in a busy sysplex environment, due to the high volume of syslog messages and WTORs. For example, new syslog messages can quickly fill the screen, pushing the WTORs out of view. There is no ability to sort or filter the data to highlight the most important records. In addition, action messages, as shown in Figure 21, are not isolated from other messages.

Providing a tabular display of the WTORs removes these limitations and improves user productivity. This display also solves the problem of restricting which jobs a user can reply to, as the action character for replying to a message is protected. The new display is available even when JES2 is not available.

Ensure that AMRF is active. If AMRF is not active, the panel shows only reply messages, as shown in Figure 22.

**Note:** The SR R command also displays the same information.

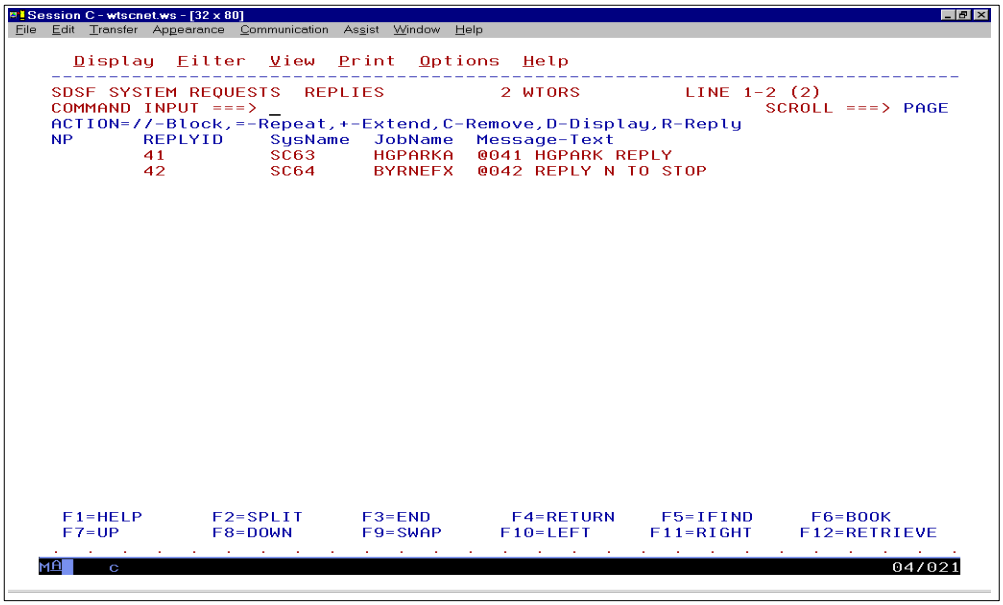



*Figure 22. Display of outstanding reply messages*

Entering the “r” action character in the action field for replyid 41 and hitting Enter displays Figure 23. You then enter your reply text and press Enter.

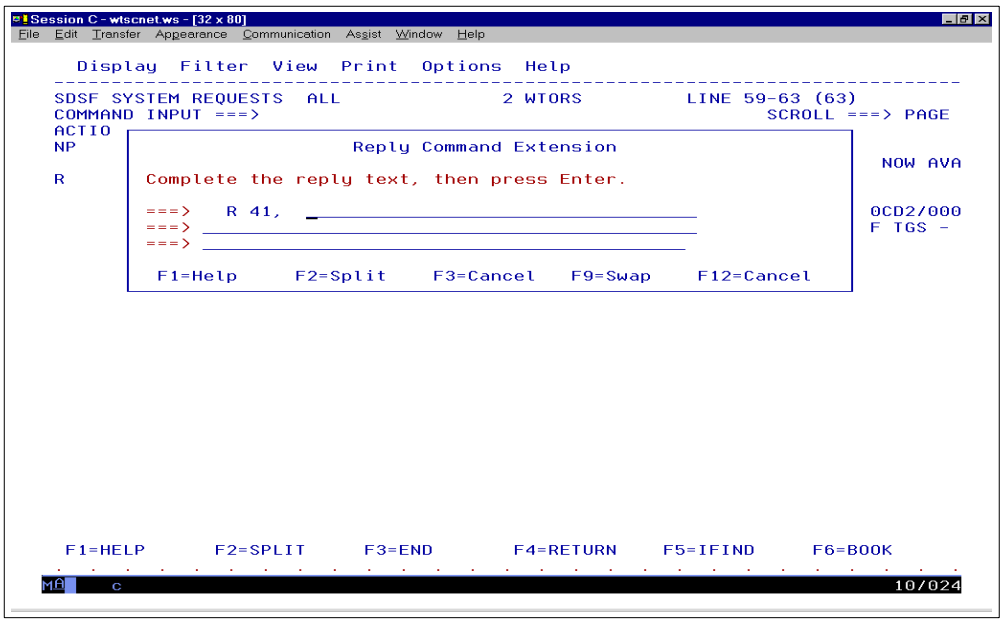



*Figure 23. Pop-up to enter a reply for the action message*

## 4.3 Mixed case column titles

For readability, the headings for all the columns in the variable field list of all panels supplied with SDSF are changed to use mixed case when they are displayed. To set it apart, the fixed field on each panel remains in uppercase. System commands are no longer folded to uppercase when entered via the command extension pop-up.

**Note:** Mixed-case column titles are not available if the language is set to Japanese.

The CTITLE parameter of ISFPARMS can be used to force column titles to be folded to uppercase.

## 4.4 New action characters and overtypes

SDSF makes the Scheduling-Env column on the I and ST panels overtypable and adds an overtypable Scheduling-Env column to the JC panel.

SDSF also adds new support for FSS on the PR panel, as shown in Figure 24, consisting of:

- New columns: FSSname and FSSproc. The FSSname column can be overtyped.
- A new action character K, which forces termination of the FSS. K is valid only for FSS printers.

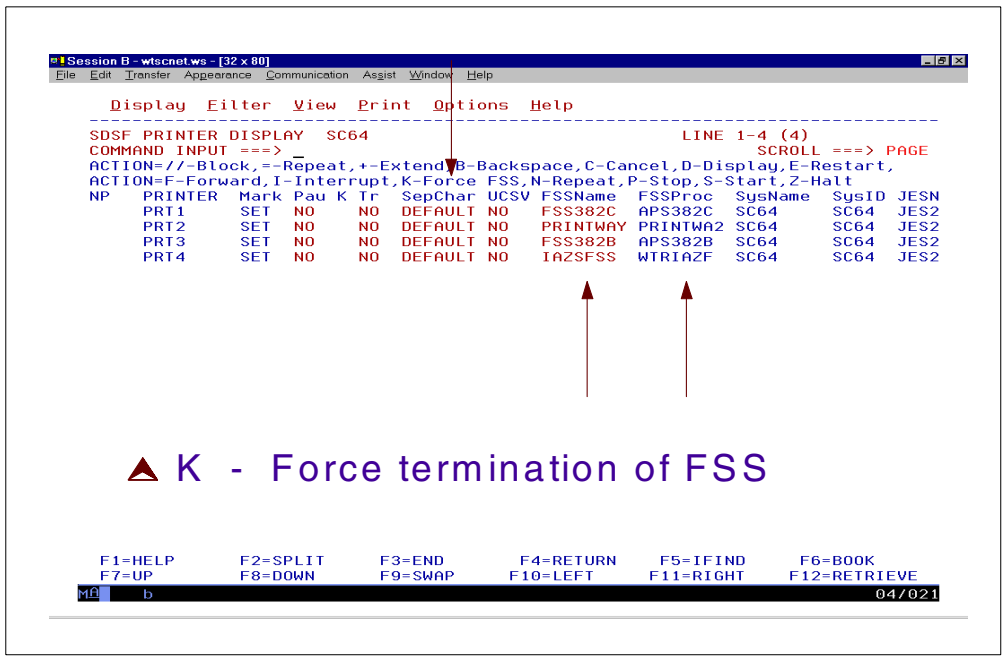



*Figure 24. New columns and overtypes for PR display*

Authorization to overtype the columns or issue the action character is controlled by CMDLEV in ISFPARMS, or SAF resources.

The default appearance of the new columns can be overridden with the fields list in ISFPARMS.

The new Scheduling-Env and FSSName overtypes are protected with CMDlev in ISFPARMS. They require command level 3. The new action character requires command level 3.

## 4.5 New SDSF commands

New SDSF commands are available:

- SYSNAME

  Use the SYSNAME command to select the systems in the sysplex displayed on the DA, INIT and PR panels.

  Figure 25 shows the DA panel with all three systems in the sysplex, SC63, SC64, and SC65.

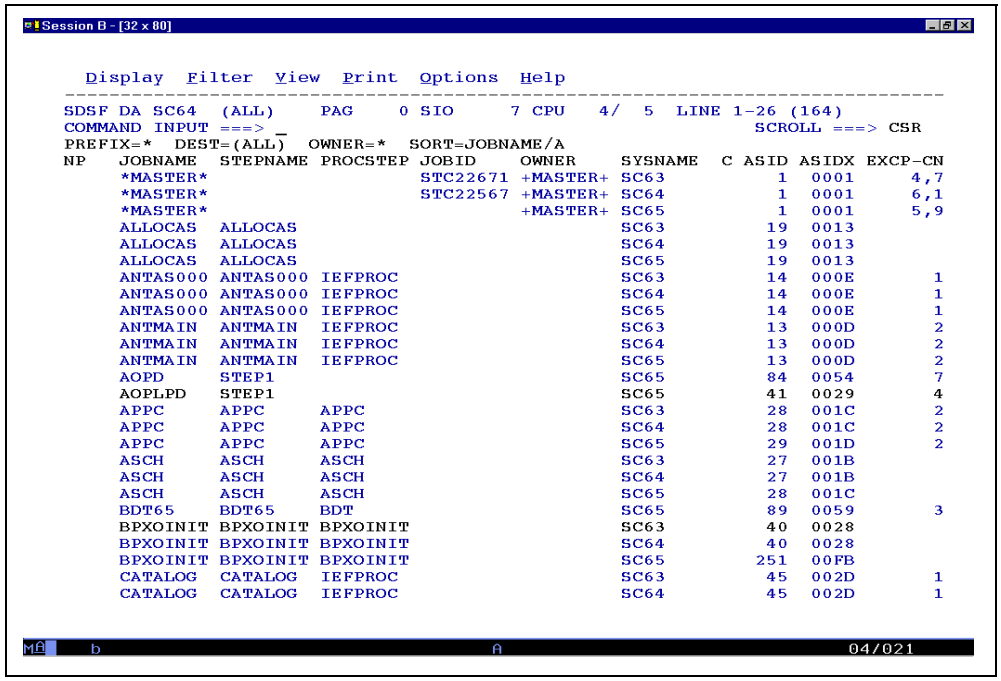



*Figure 25. DA command with sysplex view*

If you wish to change the DA display to only show system SC64, enter:

SYSNAME SC64

as shown in Figure 26 on page 50.

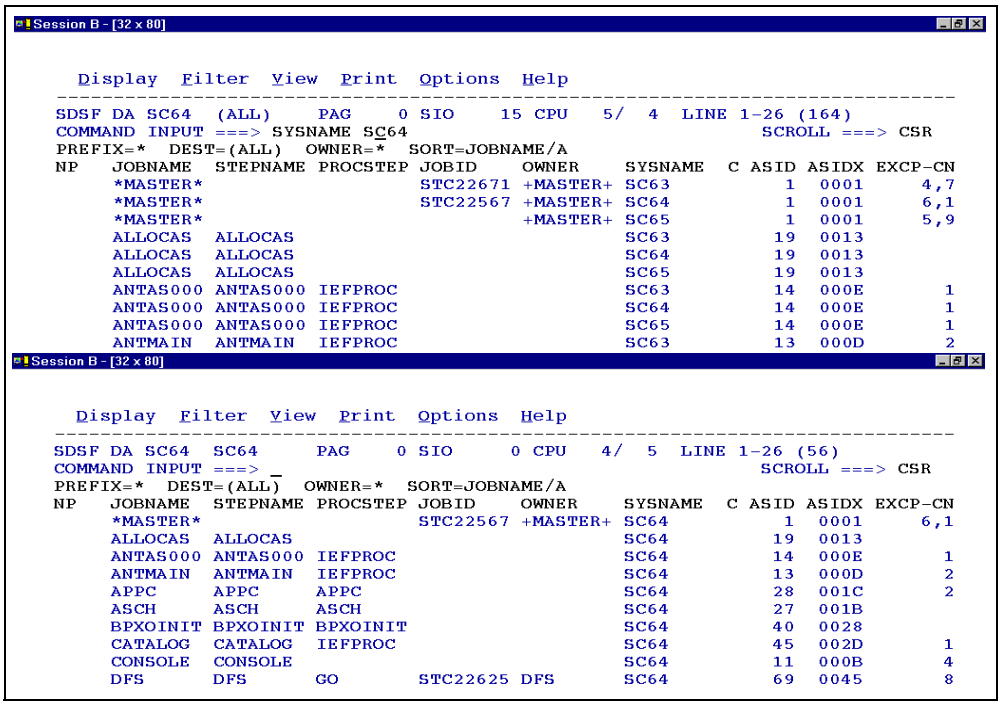



*Figure 26. SYSNAME command to select the systems on the DA panel*

- SET TIMEOUT

  Use the SET TIMEOUT command to set the default timeout value for MQSeries message responses for sysplex data. It has effects in the PR, INIT, ODS, and LOG commands. For example:

  SET TIMEOUT 30

  sets the timeout value to 30 seconds. The valid range is from 0 to 9999 seconds. Figure 27 on page 51 shows the ability to display the current timeout value.

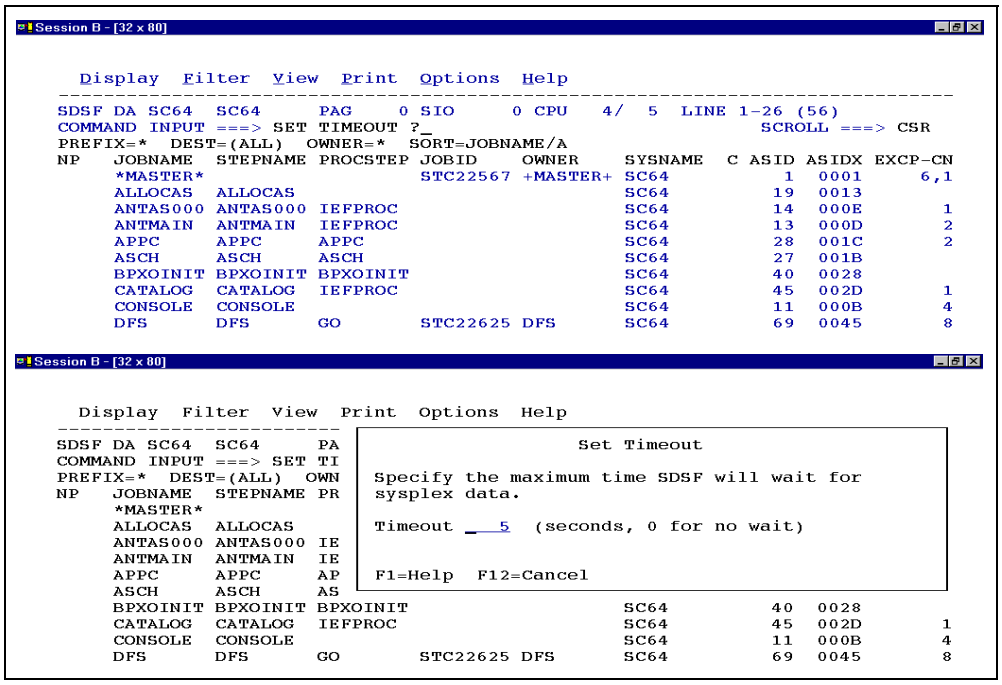



*Figure 27. SET TIMEOUT command*

## 4.6 WHO command enhancement

The WHO command was enhanced to display data related to ISFPARMS communication status; for example, sysplex capable, as shown in Figure 28.

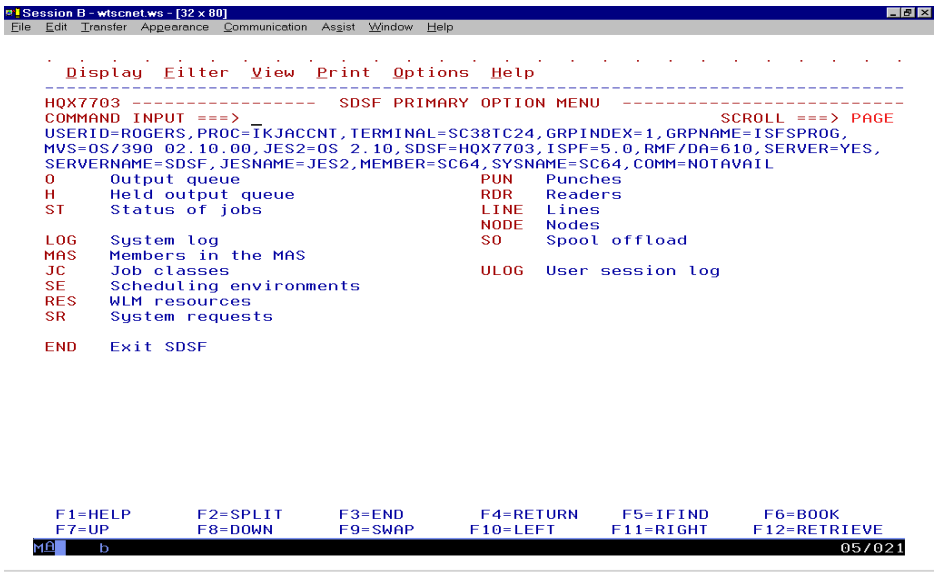



*Figure 28. Example of the WHO command*

The changes to WHO are as follows:

**SDSF=** This field now shows the SDSF FMID, HQX7703 for OS/390 Version 2 Release 10 SDSF.

**COMM=** This field shows information about communications between SDSF servers as follows:

**ENABLED** If communications are enabled

**DISABLED** If communications have been disabled with an error, such as an I/O error.

**NOTAVAIL** If communications are not available because the server group is not active or the SDSF server is not started.

**SUSPENDED** If communications have been temporarily disabled; for example, with the Set Communications timeout pull-down choice or the SET TIMEOUT command.

**Member=** The JES2 member name

**SYSNAME=** The system name of the system where the display occurs.

The same information can be displayed by placing the cursor under View and hitting Enter. Then choose Option 5 for the WHO display shown in Figure 29.

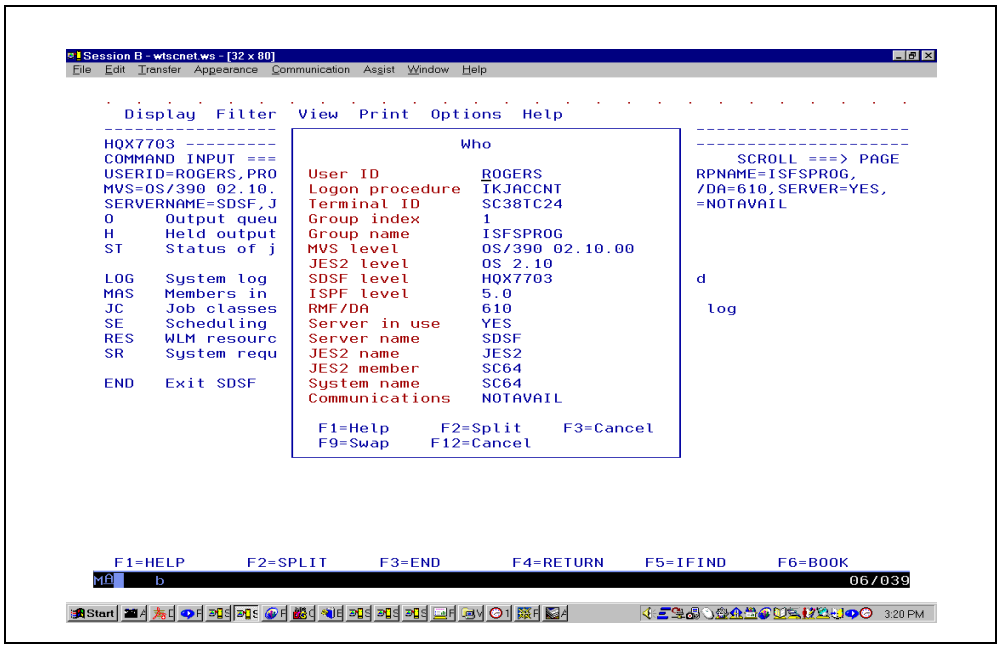



*Figure 29. WHO command and pop-up*

## 4.7 Improved management of SDSF parameters

SDSF makes it easier to manage its initialization parameters; in particular, to use a common set of initialization parameters for multiple systems. SDSF adds support for:

- Conditional processing in dynamic parms

The addition of a WHEN statement allows installations to identify parameters that apply to a particular system in order to facilitate using a common ISFPARMS for multiple systems.

**Note:** This eliminates the need to have separate ISFPARMS statements for individual systems.

- System symbols in the ISFPARMS statements

  The symbol service allows system symbols to be substituted in strings, such as the dynamic parms processed by the SDSF server. The server is enhanced to support system symbols anywhere within the dynamic parms.

  **Note:** Conditional processing is not available with the assembler format of ISFPARMS.

### 4.7.1 Conditional processing capability in dynamic parms

The WHEN statement can be used to conditionally process an entire ISFPARMS statement, which includes OPTIONS, GROUP, and the following:

- The WHEN statement specifies one or more conditions that are compared to the current environment. All of the conditions must be true for the statements that follow to be processed.
- The WHEN statement cannot be used to process a single parameter within a statement.

The WHEN statement supports the following conditions:

- Name of the LPAR
- Name of the system
- Name of the sysplex
- Name of the CPC
- User ID of the VM system under which MVS is running
- Name of the SDSF server

The example shown in Figure 30 on page 54 shows the use of the system name to determine which statements are to be processed. As each statement is processed, the WHEN statement system name specification is checked against the current system. If it matches the current system, the statements that follow the WHEN statement are processed until the next WHEN statement is found, or the end of the file is reached.

For WHEN statements that do not match the current system, the statements that follow the WHEN statement are checked for syntax but not processed, until the next WHEN is found.

```
WHEN SYSNAME(SC64)
 OPTIONS ATHOPEN(YES),     /* Use authorized open for datasets  */
  FINDLIM(5000),           /* Maximum lines to search for FIND  */
  IDBLKS(4096),            /* HASPINDX blocksize                */
  INDEX(ISF.SY1.HASPINDX), /* HASPINDX dataset name             */
  .
  .
  .
WHEN SYSNAME(SC65)
 OPTIONS ATHOPEN(YES),     /* Use authorized open for datasets */
  FINDLIM(5000),           /* Maximum lines to search for FIND  */
  IDBLKS(4096),            /* HASPINDX blocksize                */
  INDEX(ISF.SY2.HASPINDX), /* HASPINDX dataset name             */
  .
  .
  .
```

*Figure 30. WHEN statement example*

### 4.7.2 System symbol support

System symbol support can be used in the dynamic ISFPARMS statements:

- When your HASPINDX data set is “system.HASPINDX” on each system and is specified in one of the following manners:
  - &SYSNAME is a symbol for the system name (defined in IEASYMxx).
  - Symbolic name referenced on INDEX statement in dynamic ISFPARMS:

    INDEX(ISF.&SYSNAME..HASPINDX).

With the use of the system symbol, &SYSNAME, the previous example using the WHEN statement can be replaced by the following example shown in Figure 31. This results in the HASPINDX data set name being correct for each system in the sysplex.

```
OPTIONS ATHOPEN(YES),       /* Use authorized open for datasets  */
  FINDLIM(5000),            /* Maximum lines to search for FIND  */
  IDBLKS(4096),             /* HASPINDX blocksize                */
  INDEX(ISF.&SYSNAME..HASPINDX),  /* HASPINDX dataset name       */
  .
  .
  .
```

*Figure 31. ISFPARMS statement using system symbols*

## 4.8 The SDSF server

The SDSF server is an address space that is required if you:

- Define your ISFPARMS using statements rather than assembler macros. To process ISFPARMS, the server must be active on each system that contains SDSF users.
- Provide sysplex data on the PR, INIT, and browse panels. To provide sysplex data, the server must be active on each system that is to be included on SDSF panels.

  **Note:** This sysplex data support is new with SDSF Version 2 Release 10.

You can use the WHO command or pop-up to verify that the server is in use.

Multiple SDSF servers may be run on the same system. However, you must assign them unique names. Only one server with a particular name can be active on the system. The level of server must match the level of the SDSF application.

You can control the server through the MVS operator START, STOP, and MODIFY commands.

### 4.8.1 New ISFPARMS statements for sysplex data

Three new statements and corresponding assembler macros are added to the ISFPARMS definitions:

**SERVERGROUP** Defines a group of servers.

**SERVER** Defines the server and systems in the sysplex.

**COMM** Provides sysplex data and the communication between the servers.

### 4.8.2 Server registration with ARM

The SDSF server is registered with ARM if ARM is enabled on the system by making the following definitions in the ARM policy:

**Element name:** ISFserver-name@&sysclone

**Element type:** SYSSDSF

**Termtype:** ELEMTERM

The registration is controlled with a new START command option.

## 4.9 Sysplex system management and MQSeries

The support for sysplex-wide displays requires MQSeries for OS/390 Version 2.1 on each system that is to participate. Note that MQ is required only for the sysplex support. If MQ is not present, the displays continue to be single-system.

The support for sysplex displays is implemented using the SDSF server. It includes:

- SDSF server on all systems in the sysplex.
- Use of (dynamic) ISFPARMS statements rather than the assembler version.
- Use of SAF for security is highly recommended but not required.

**Note:** Sysplex support requires no additional SAF profiles.

- New ISFPARMS statements to define communications configuration

  **Note:** The ISFPARMS conversion utility, ISFACP, is recommended. The SDSF Server requires dynamic ISFPARMS statements.

- Assuming that MQSeries is installed and operational, the sysplex implementation requires no additional MQ customization.
- The SDSF server defines and maintains a single dynamic request queue at servergroup creation time. The SDSF MQ request queue name construct is:

  qprefix.SERVER.servername.sysname.REQUESTQ

  **Note:** ISF is the default qprefix. It can be installation defined through the dynamic ISFPARMS statements.

- ISFPARMS SERVERGROUP keyword parameter defines an SDSF group that a server must communicate with to provide sysplex data and is required.
- A new SERVER command is implemented to display information about the Server Group Communications that is used to display information about servers and communications between SDSF servers in a server group as follows:

  F server-name,DISPLAY,COMM...

## 4.10 MQSeries for OS/390 considerations

To provide sysplex support for the PR and INIT panels, for browse, and for the SYSLOG panel, MQSeries for OS/390 (MQ) along with the SDSF server must be installed.

### 4.10.1 MQSeries facilities

MQSeries provides facilities to enable application programs to communicate with each other using messages and queues. By using a common programming interface, in this case, the Message Queue Interface (MQI), applications written on one platform can be transferred to another platform.

Using MQSeries means that the application creating the message can continue processing while MQ processes the send as well as the reply from the receiver. If the receiver is unavailable, or cannot be contacted, MQ queues the message and processes it when the receiver is available.

### 4.10.2 MQSeries queues

In MQSeries, queues are managed by a component called the *queue manager*. The queue manager provides messaging services for the applications and processes the MQI calls they issue. When an application connects to a queue, the queue manager returns a *connection handle*, which must be identified by the application each time it issues an MQI call. An application connects to only one queue manager at a time and all MQI calls are processed by that queue manager, known as the local queue manager until a disconnect call is issued.

When the message is to be retrieved, the application must open the queue to GET the message. When the open is successful, an *object handle* is returned, and the application will specify this, plus the connection handle to PUT or GET a

call. The *message descriptor* that is added during an MQI call process is used to ensure that the message is processed correctly.

MQSeries is supported on over 35 platforms, including Digital, HP, TANDEM, Windows, SUN, and, of course, IBM's OS/2, OS/390, VSE/ESA, VM/ESA, AS/400, and AIX.

### 4.10.3 OS/390 MQ libraries

The MQ load libraries must be accessible to both the SDSF server and the SDSF client, which is the SDSF user running under TSO/E. This can be done by including the libraries in the LNKLST, or with a STEPLIB.

If you use a STEPLIB, be sure all libraries in the STEPLIB concatenation are APF-authorized.

The MQ libraries must be APF-authorized, including SCSQLOAD. Both the SDSF server and SDSF client need access to the modules in SCSQLOAD.

### 4.10.4 MQSeries configuration

Minimal MQSeries customization and configuration is required for SDSF:

- Review the values in the MQSeries system parameter module, CSQZPARM. In particular, you may need to increase the number of background and foreground connections, which are defined with the IDBACK and IDFORE parameters. The SDSF server establishes several connections with MQ, usually a minimum of 11 connections to a maximum of approximately 31. Your IDBACK value should reflect this usage. Similarly, the SDSF client establishes a connection with MQ for each SDSF logical session. Multiple SDSF sessions can be started using ISPF's split screen. Your IDFORE value may need to be adjusted to accommodate this. See *MQSeries for OS/390 Systems Management* for more information.

*Table 2. Summary of possible MQSeries system parameters changes*

| Parameter | MQSeries Default | Possible changes for SDSF |
|---|---|---|
| IDBACK | 20 | Change to reflect SDSF server connections with MQ. Up to 31 can be specified. |
| IDFORE | 100 | Change to reflect the maximum number of SDSF client logical sessions that are connected to MQ. |

- To separate the SDSF message usage from your existing applications, you may want to define a separate queue manager to be used by SDSF. You control which queue manager SDSF uses by coding its name on the COMM statement associated with the server definition in ISFPARMS.
- Review your MQSeries page sets. Storage estimates are described in 4.10.5, "Storage estimates" on page 58.
- If your installation has a large number of devices, you may need MQSeries APAR PQ33000 installed on your system. This APAR provides support for very large MQ messages.

### 4.10.5 Storage estimates

The number of messages is proportional to the number of users, the number of requests for data (caused, for example, by a user pressing Enter), and the number of servers in the server group.

The size of the messages varies with the data being requested. In general, a request is approximately 300 bytes. A response consists of a 300-byte header followed by the response data. The response varies with the panel and the number of rows returned. SDSF compresses the response data, so the actual data sent through MQ may be less than the maximum. The effectiveness of compression relates directly to the contents of the data being returned.

### 4.10.6 Queues

You do not need to define any queues for SDSF's use of MQ. The queues SDSF uses are:

- A model queue, used in creating other queues. The SDSF server defines this if it is not already defined.
- Temporary, dynamic queues used to communicate between the SDSF server and the user. MQ creates these with the use of the model queue.

  The SDSF client queues work to the SDSF server by using an alias for the server request queue. This is called a queue alias. Although it is not required, you can use an MQ DEFINE command to define the queue alias. The DEFINE command used to define the queue alias to the server request queue if it does not exist is:

```
DEFINE
QALIAS('prefix.CLIENT.servername.sysname.REQUESTQ') +
NOREPLACE                          +
DEFPRTY(5)                        +
DEFPSIST(NO)                       +
DESCR('SDSF Server Request Queue Alias')      +
PUT(ENABLED)                        +
GET(DISABLED)                       +
TARGQ('prefix.SERVER.servername.sysname.REQUESTQ')
```

*Figure 32. MQ DEFINE command - queue alias - member name SDSF64*

Figure 32 shows the DEFINE command in the MQ startup parameters and in the example, prefix is the queue prefix, which is defined on the COMM statement of ISFPARMS.

```
//CSQINP2   DD DSN=MQ210.SCSQPROC(CSQ4INSG),DISP=SHR
//         DD DSN=MQ210.SCSQPROC(CSQ4INSX),DISP=SHR
//         DD DSN=MQ210.SCSQPROC(CSQ4INYG),DISP=SHR
//         DD DSN=MQ210.SCSQPROC(CSQ4INYC),DISP=SHR
//         DD DSN=MQ210.SCSQPROC(CSQ4INYD),DISP=SHR
//         DD DSN=MQ210.SCSQPROC(CSQ4DISP),DISP=SHR
//         DD DSN=MQSB.SCSQPROC(SDSF64),DISP=SHR
```

*Figure 33. Member SDSF64 has the statements to define QALIAS at MQSeries start-up*

## 4.11 Communication between queue managers

The addition of queues used by SDSF may require you to perform some MQ customization so that the queue managers for those queues can communicate.

When a queue manager needs to put messages on a queue managed by a different queue manager, it locates the target queue by the queue name and the queue manager name. For example, SDSF's server request queue is accessed by all SDSF servers in the server group. To locate that queue, an MQ queue manager would need the following:

- The queue name, queue-prefix.SERVER. server.system.REQUESTQ.
- The queue manager name, which is specified on the COMM statement in ISFPARMS.

There are several ways to define the remote queues and queue managers to the local queue managers. SDSF is designed to simplify this task by allowing you to use the queue manager alias technique. A queue manager alias relates a queue manager name to a transmission queue. (The transmission queue is a special kind of queue on which messages are stored until they can be transmitted to the remote queue manager. MQ uses a channel and a transmission queue on the remote system to ensure that message gets routed properly.) The queue manager alias is convenient because only a single definition is needed to route all requests to all queues managed by a remote queue manager. If you don't use a queue manager alias, you need a remote queue definition for each remote queue. This results in many more definitions.

See Appendix A, "MQ definitions for MQSeries Queue Managers" on page 195 for examples of how to establish communication between two MQSeries queue managers.

## 4.12 Server group

To enable support for sysplex data, the system programmer must define a server group for each SDSF server. A server group is the group of SDSF servers that the server must communicate with to provide sysplex data. SDSF uses MQSeries for OS/390 to communicate between servers. In defining a server group, the system programmer defines the systems and their related JES2s that participate

in a sysplex-wide request. This gives the installation the flexibility to define any combination of systems, and primary and secondary JES2s, to process. Each combination requires a separate instance of the SDSF server. All SDSF servers must be in the same sysplex, and all associated JES2 systems must be in the same MAS, as shown in Figure 34.

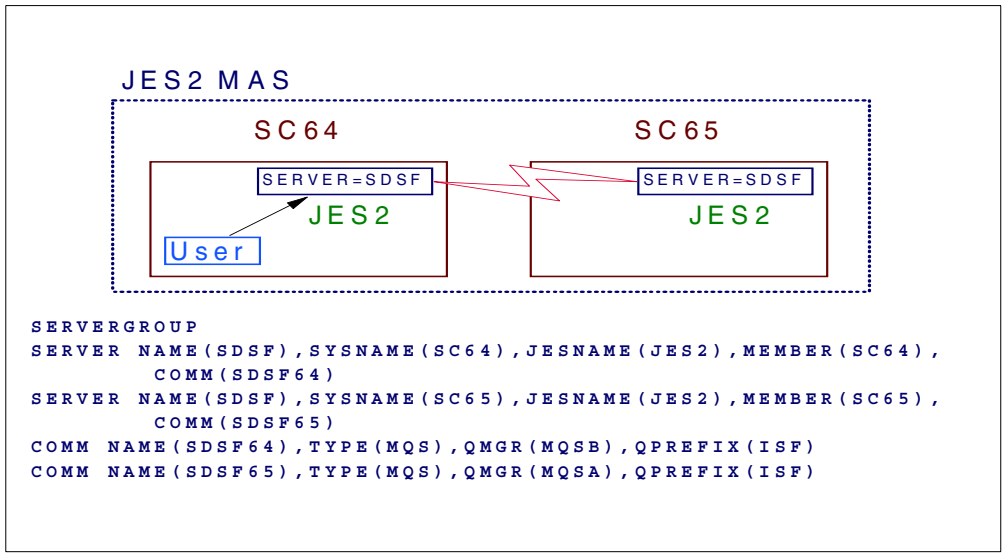



*Figure 34. Server group example*

The user connects to the server that is specified in ISFPARMS, or, alternatively, the server specified on the command used to invoke SDSF. The server that the user is connected to is known as the local server. The ability to override the server defined in ISFPARMS is controlled with an SAF resource. With the addition of server groups, SDSF extends control of access to the server to include access to the server group.

**Note:** If a server group is not defined, SDSF does not use the new sysplex support, that is, it does not use the server and MQSeries to gather the data. Instead, SDSF functions as it did in previous releases.

## 4.12.1 Server group examples

Figure 35 on page 62, Figure 36 on page 63, Figure 37 on page 64, and Figure 38 on page 65 illustrate some server group definitions. Before you can activate the server group, the SDSF server must be active.

### 4.12.1.1 Starting the SDSF server

There are several ways to start the SDSF server, as follows:

```
S SDSF,M=Y4
```

```
IEF403I SDSF - STARTED - TIME=11.18.36 - ASID=0048.
ISF724I SDSF level HQX7703 initialization complete for server SDSF.
ISF726I SDSF parameter processing started.
ISF170I Server SDSF ARM registration complete for element type SYSSDSF,
element name ISFSDSF@64.
ISF739I SDSF parameters being read from member ISFPRMY4 of data set
SYS1.PARMLIB.
ISF401I Server SDSF communications initialization in progress.
```

```
ISF728I SDSF parameters have been activated.
ISF402I Server SDSF communications ready.
```

To start the SDSF server from the current parmlib member after updating the member with the SERVERGROUP definitions, issue the following command:

```
F SDSF,REFRESH
```

The operator sees the following messages:

```
ISF304I Modify REFRESH command accepted.
ISF726I SDSF parameter processing started.
ISF739I SDSF parameters being read from member ISFPRMY4 of data set
SYS1.PARMLIB.
ISF728I SDSF parameters have been activated.
```

#### 4.12.1.2 Display current server status

The SDSF server status can be displayed by issuing the following command:

```
F SDSF,D
```

The following messages are issued:

```
ISF304I Modify DISPLAY command accepted.
ISF312I SDSF Display
    Server status:  Active
    Communications: Inactive
    Parms: ISFPRMY4 / SYS1.PARMLIB
    Trace: Not active  Mask: 00000000
```

#### 4.12.1.3 Start communications in a server group

The following command can be issued to start communications between servers in a server group when a connection is lost or is stopped by the STOP command:

```
F SDSF,START,COMM
```

The following messages are issued:

```
ISF304I Modify START command accepted.
ISF415I Server SDSF system SC64 started, current status is Active.
ISF415I Server SDSF system SC65 started, current status is Defined.
ISF310I SDSF Communications
   Id Server   Status       System   JESN Member
   01 SDSF     Active/L     SC64     JES2 SC64
   02 SDSF     Defined      SC65     JES2 SC65
```

### 4.12.2 Configuration examples

This section offers several configuration examples that you can make to set up an environment for your installation as follows:

- Two primary JES2 systems
- Two primary and two secondary JES2 systems
- Two secondary JES2 systems
- Using independent server groups

#### 4.12.2.1 Two primary JES2 systems

Figure 35 on page 62 shows a sysplex with MVS systems SC64 and SC65. When a user logs on to system SC64 and invokes SDSF, that user is connected to the server named SDSF.

This is the user's local server. The server processes ISFPARMS, which contains a server group definition consisting of the two servers named SDSF, one on system SC64 and the other on system SC65. The servers each gather data for the JES2 subsystem on the system where they are running. The user's panels for the PR, INIT, and browse displays show data from both of the JES2 systems.

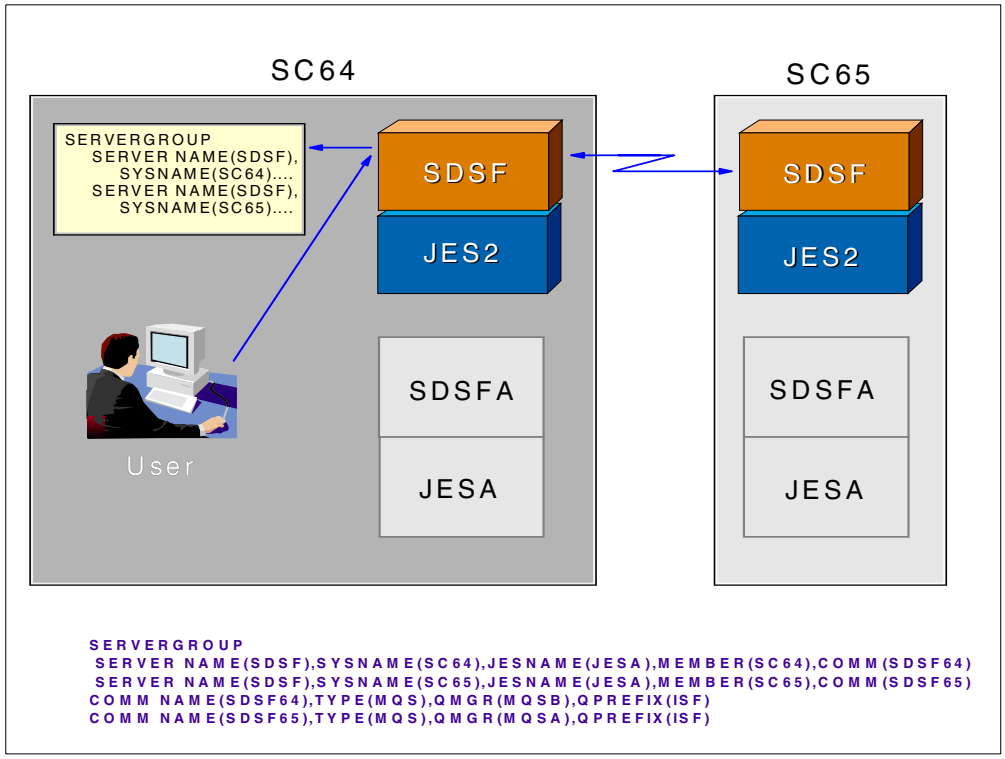



*Figure 35. Server group example*

### 4.12.2.2 Two secondary JES2 systems

Other SDSF servers are running on these systems, each named SDSFA. These servers, which gather data for secondary JES2s, are not part of the server group. Data from the secondary JES2 systems is not included on the users' panels.

Figure 36 shows the same sysplex. This time, the SDSF user is connected to the SDSF server SDSFA, which collects data for the alternate JES. (The user might, for example, have invoked SDSF with the command s.server(sdsfa).) The server group defined in the ISFPARMS processed by that server consists of the two servers named SDSFA. The user's panels will show data from the two JESAs.

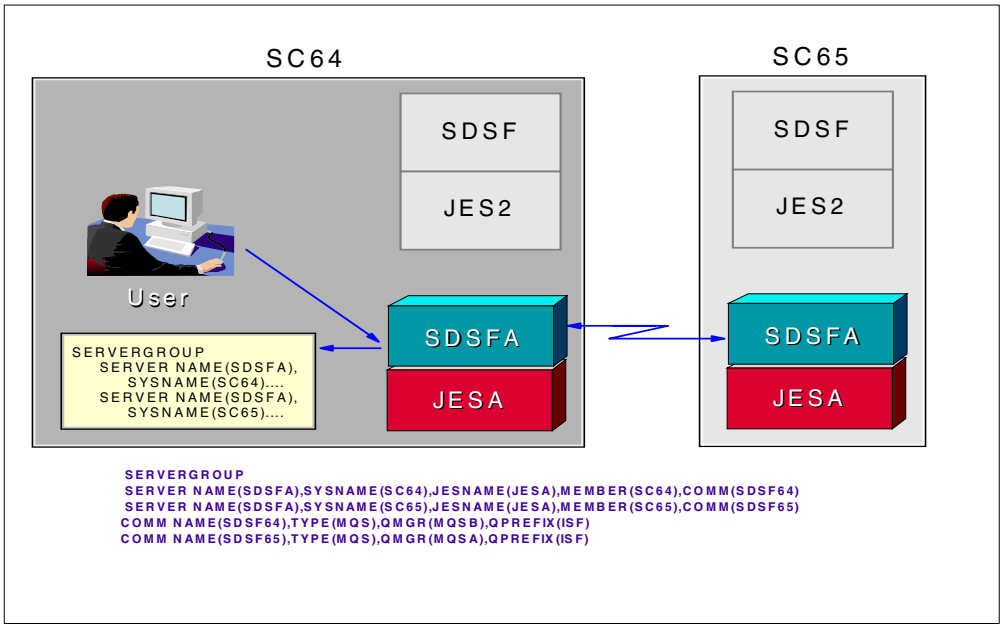



*Figure 36. Server group example - secondary JES2s*

### 4.12.2.3 Two primary and two secondary JES2 systems

Figure 37 shows the same sysplex. The SDSF user is connected to the SDSF server SDSF3. The server group defined in the ISFPARMS processed by that server is made up of all four servers. The user's panels will show data from the two primary and alternate JESs

.

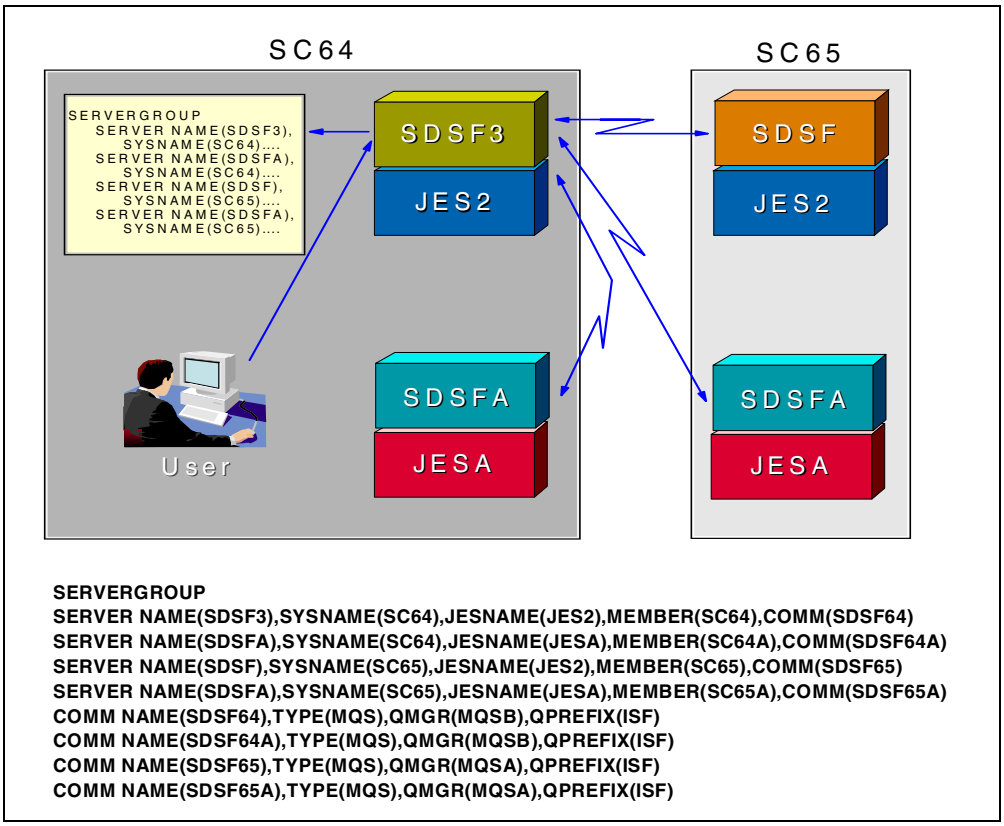



*Figure 37.* Server group example – all servers

### 4.12.2.4 Independent server groups

Note that the server groups defined for each server are independent of each other. In the previous example, server SDSF3 on SC64 and server SDSF on SC65 are in the same server group. However, server SDSF on SC64 may have a different server group defined in the ISFPARMS it processes than server SDSF3 has in the ISFPARMS it processes.

Figure 38 on page 65 shows a user logged on to SC65 and connected to server SDSF there, which processes an ISFPARMS with a server group consisting of three servers.

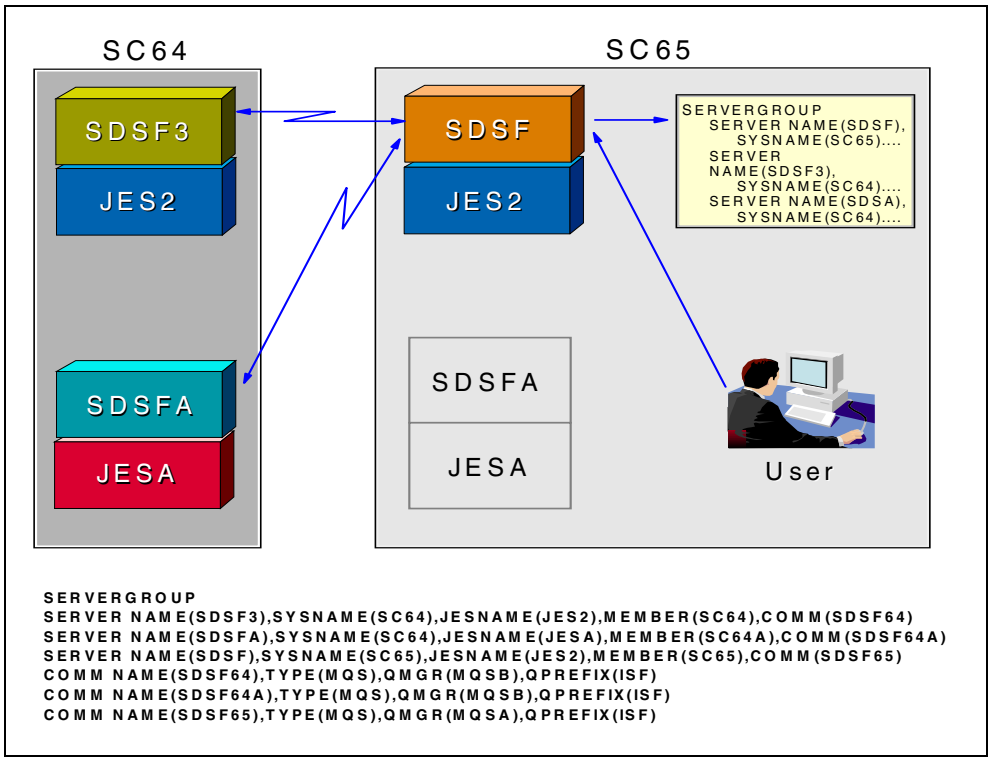



*Figure 38.* Server group example -- server group on SY2

## 4.13 Sysplex-wide panels

SDSF provides the installation with functions that allow sysplex systems management without requiring logon to individual systems to manipulate resources.

### 4.13.1 Sysplex device display support

In previous releases, SDSF's device displays showed devices for the system the user is logged on to. OS/390 Release 10 SDSF makes the most used of the device panels, Initiator and Printer, sysplex-wide. The panels are changed to show all initiators and printers for all systems in the MAS, regardless of which system the user is logged on to. Users are given the ability to control which systems are to be included in the display, and a column is added to indicate which system the device is on.

SDSF adds sysplex support to the PR and INIT device panels and to its browse function. It is assumed that all systems in the sysplex are in the same MAS.

With this support, the PR and INIT panels are enhanced to show all printers and initiators for all systems in the MAS, regardless of which system the user is logged on to. New columns were added to PR and INIT panels and JESname, SYSid and SYSname are now shown.

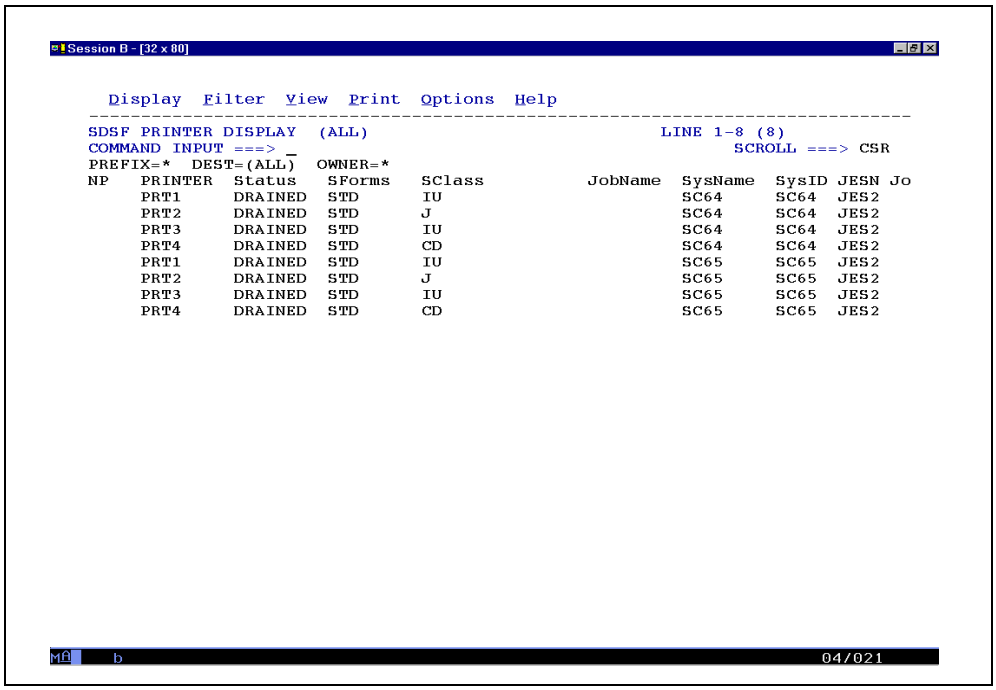



*Figure 39. Sysplex printers display*

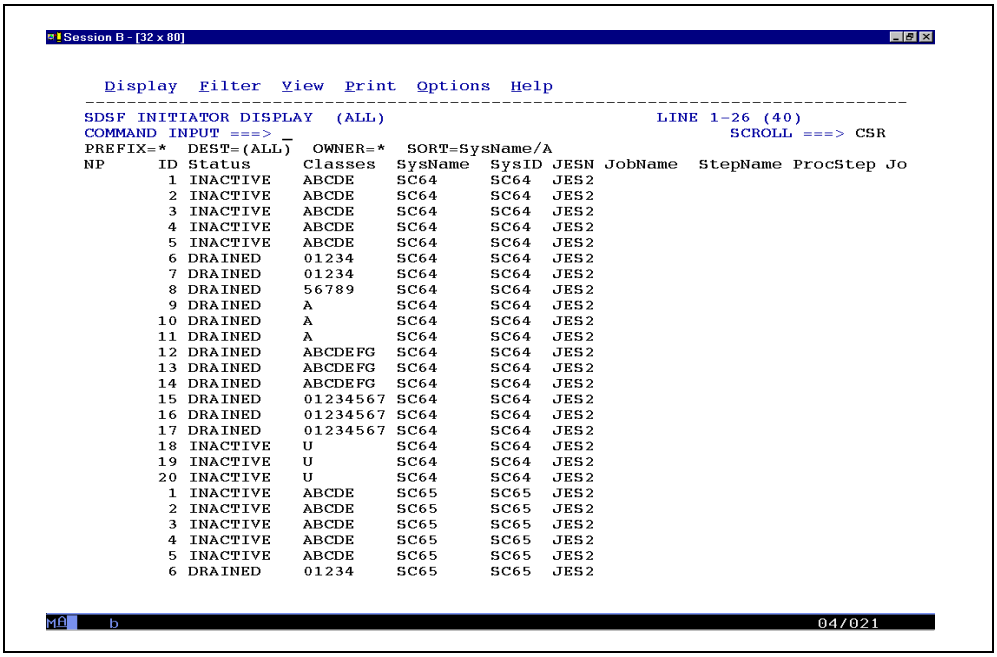



*Figure 40. Sysplex initiators display*

### 4.13.2 Improved SDSF browse and log display

SDSF's browse function, which is used to browse a job's output and the syslog, is enhanced to include data not yet written to spool from any system in the sysplex. SDSF's browse now always shows the most complete and current information for active jobs and syslog, regardless of which system the user is logged on to.

SAF profiles in the MQSeries for OS/390 classes are used to protect the queues used by SDSF.

The SDSF server processes ISFPARMS statements on each system and is also used in collecting sysplex data. The SDSF servers that collect data for display are defined in a new SERVERGROUP statement.

The amount of time SDSF waits for sysplex data is controlled with the TIMEOUT parameter in ISFPARMS or with the SET TIMEOUT command.

Fields are added to the user exit, to indicate the system name, JES2 subsystem name, MAS member name, and OWNER ID.

## 4.14 Configuration Assistant for enabling sysplex panels

SDSF provides a Web User Interface (UI) to simplify the tasks required to exploit the new sysplex-wide displays. The Web interface, called the SDSF Configuration Assistant, collects user input and generates results, such as JCL or RACF commands, that can be cut and pasted, or uploaded, into MVS data sets. It also uses graphics and hypertext links to simplify finding specific information, such as required SAF resources and RACF commands.

Although the SDSF Configuration Assistant is organized to support enabling the sysplex displays, some of the tasks it describes, specifically setting up a server and defining SAF security, are of general interest.

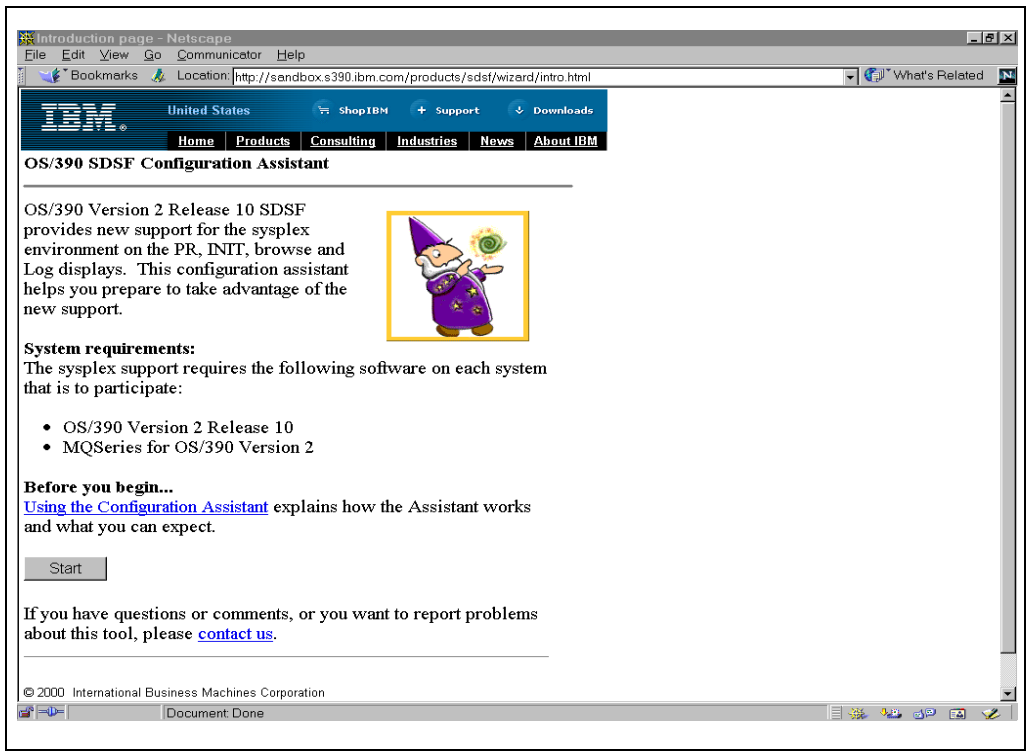



*Figure 41. WEB-based assistant*

To see the sysplex-wide data on the PR, INIT and browse displays, customers must perform some tasks that are either new or newly required:

- Convert SDSF's internal parameters from assembler format to a dynamic format.
- Define JCL for an SDSF server.
- Establish SAF security definitions.
- MQSeries administration.

### 4.14.1 SAF security

In addition, the use of SAF for security is recommended, though not required. These tasks involve a variety of skills and multiple products (SDSF, RACF and MQ). The SDSF Configuration Assistant brings the tasks together in one place. For some tasks, such as creating JCL, it provides simple forms that the user can fill out. The output that it generates reduces keystrokes and improves the system programmer's productivity.

The task of setting up SAF security is the most difficult of these tasks. Its complexity has prevented some customers from moving to SAF, even though the capability has been available for years, and SAF offers clear advantages over SDSF's internal security scheme (ISFPARMS). In the past year, a vendor has begun marketing a product that simplifies setting up SAF security for SDSF. The SDSF Configuration Assistant simplifies the task by providing an easy-to-use graphic interface to SAF resources and RACF commands. This information is intended to be used standalone or in conjunction with the RACF conversion tool that was included with the product in OS/390 Release 5.

### 4.14.2 SDSF Configuration Assistant functions

In this first release, the SDSF Configuration Assistant does not interact directly with the OS/390 system. For example, it does not read a customer's ISFPARMS.

The SDSF Configuration Assistant is made up of HTML pages connected by hypertext links and viewable by a Web browser. The Assistant uses two frames (refer to Figure 41 on page 67):

- A standard banner at the top of the page
- A main frame for content and the standard footer

The Assistant helps in performing the following tasks:

- Creating dynamic ISFPARMS

  Instructions describe how to use the ISFACP utility to convert assembler ISFPARMS to dynamic ISFPARMS.

- Defining the SDSF server

  An online form collects information for defining the JCL for an SDSF server. The user can request that the SDSF Configuration Assistant display the resulting JCL, and can then cut and paste or upload the JCL.

- Defining SAF security

  The dynamic ISFPARMS shipped with SDSF (ISFPRM00) is represented in the contents frame. Each parameter that is related to security has a hypertext link to the appropriate SAF resource, and, optionally, RACF commands. Where there is no direct SAF equivalent, or when there are multiple options, the User Interface displays conceptual and reference information.

- Defining MQ security

  An online form collects user input for the required SAF profiles and RACF commands to protect SDSF's use of MQ. The user can request that the SDSF Configuration Assistant display the resulting statements and RACF commands, and can then cut and paste them for use in defining security.

### 4.14.3 Accessing the SDSF Configuration Assistant

The SDSF Configuration Assistant is made available through the Internet at:

```
http://s390.ibm.com/products/sdsf/wizard/intro.html
```

Users can link to it from the SDSF Web page. It may also be accessible from the pages for one or more Web User Interfaces being developed, such as the Sysplex Configuration Web User Interface. The URL for the Web User Interface is added to the SDSF online help, in a new Help pull-down choice. In addition, the URL is published in the SDSF documentation.

Providing the SDSF Configuration Assistant over the Internet means that it can easily be updated at any time. In addition, there is no cost to the customer, and no installation required.

# Chapter 5. DFSMS for OS/390 Release 10

With OS/390 Release 10, the name for the storage management component has changed. As you may remember, up to OS/390 Release 9, its name was DFSMS/MVS Version 1 Releases 1 to 5. The new name in OS/390 Version 2 Release 10 is DFSMS. As of OS/390 Release 10, all DFSMS components are exclusive to OS/390. This means that DFSMS/MVS is no longer marketed.

OS/390 consists of two components that build the necessary base: The Base Control Program (BCP); and DFSMS, which is used to perform the essential data, storage, program, and device management functions of an OS/390 system.

DFSMS is the central component of both system-managed and non-system-managed storage environments. In a system-managed storage environment, the components of DFSMS automate and centralize storage management based on installation-defined policies for availability, performance, space, and security. The user and administrator interface called Interactive Storage Management Facility (ISMF) is the tool for defining and maintaining these policies. The underlying Storage Management Subsystem (SMS) governs these policies for the system or sysplex.

DFSMS consists of four functional components:

**DFSMSdfp** This is a base element of OS/390 and is used for storage management, tape mount management, data management, program management, device management and distributed data access.

In addition to the base element DFSMSdfp, you can order the following optional priced combinations as separate features:

**DFSMSdss** This is used for data movement and replication, space management, data backup and recovery, and data set and volume recovery.

**DFSMShsm** This is used for storage management, space management, tape mount management, and availability management.

**DFSMSrmm** This works as a tape management system. It is used for library management, shelf management, volume management, and data set management.

With OS/390 Release 10, there are enhancements to all four components. This chapter describes these enhancements.

## 5.1 DFSMSdfp enhancements

The enhancements in the DFSMSdfp component are improvements to performance and availability.

The performance enhancements are:

- VSAM striping
- Support for tape block size greater than 32K

The improved availability enhancements are:

- DADSM rename duplicate data set
- UNIT=AFF support for tape libraries
- Coupling facility structure rebuild for catalogs

### 5.1.1 VSAM striping

Placing stripes of data across multiple DASD volumes can improve your batch throughput. Now with DFSMS, VSAM striping is available for all VSAM extended format data set organizations. It includes:

- Key-sequenced data sets (KSDS)
- Entry-sequenced data sets (ESDS)
- Relative-record data sets (RRDS)
- Variable-length relative-record data sets (VRRDS)
- Linear data sets (LDS)

VSAM striping spreads control intervals (CI) in a control area (CA) across multiple devices. With this technique the effective throughput on sequential access increases proportional to the number of stripes. This happens unless there is contention at the level of the device, subsystem, path or channel. If you use striping, you should be aware that the effective throughput is limited by the slowest stripe when the data is spread across multiple devices that have dissimilar performance characteristics. On the other hand, striping does not impact performance of random accesses to those volumes.

If you plan to stripe VSAM data sets across multiple devices, you must define them with the following attributes:

- Extended addressability
- Compression
- Partial release

If you have SMS active, you have to define or modify a data class as shown in Figure 42 on page 73.

```
DATA CLASS ALTER              Page 3 of 3
Command ===>

SCDS Name . . . : SYS1.SMS.SCDS
Data Class Name : STRIPE

To ALTER Data Class, Specify:
  Data Set Name Type  . . . . . . EXT  (EXT, HFS, LIB, PDS or blank)
    If Ext  . . . . . . . . . . . R    (P=Preferred, R=Required or blank)
    Extended Addressability . . . Y    (Y or N)
    Record Access Bias  . . . . . U    (S=System, U=User or blank)
  Reuse . . . . . . . . . . . . . N    (Y or N)
  Initial Load  . . . . . . . . . R    (S=Speed, R=Recovery or blank)
  Spanned / Nonspanned  . . . . .      (S=Spanned, N=Nonspanned or blank)
  BWO . . . . . . . . . . . . . .      (TC=TYPECICS, TI=TYPEIMS, NO or blank)
  Log . . . . . . . . . . . . . .      (N=NONE, U=UNDO, A=ALL or blank)
  Logstream Id  . . . . . . . . .
  Space Constraint Relief . . . . N    (Y or N)
    Reduce Space Up To (%)  . . .      (0 to 99 or blank)
```

*Figure 42. Altering an SMS data class to data set type extended for striping*

You can have a maximum of 16 stripes of a VSAM data set. But only the data component can be striped. Striping is not possible for the index component or an alternate index (AIX). There is also no support for VSAM record-level sharing (RLS) and no RESET and REUSE processing. The number of stripes depends on the following conditions:

- The sustained data rate value of a storage class for non-guaranteed space data sets
- The number of volumes that are available for guaranteed data sets

In Figure 43 on page 74, you can see how striping works with four volumes. The assumptions are that you have a data CI size of 4K, a physical block size of 4K, 12 4K blocks per 3390 track, and a stripe count of four.

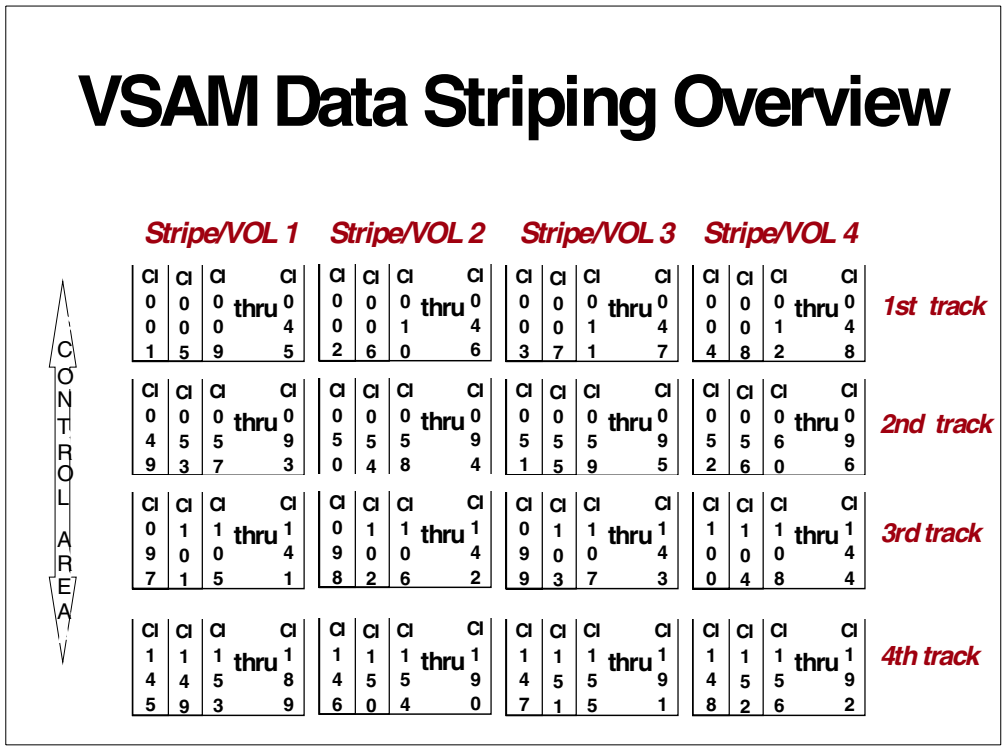



*Figure 43. VSAM data striping overview*

Multi-layering is the ability for a stripe to extend to a new volume. A layer is the relationship of those volumes that make up the total number of stripes that belong to a specific VSAM data set. Those volumes participate as part of a so-called “I/O packet”. You can see an example of multi-layering in Figure 44.

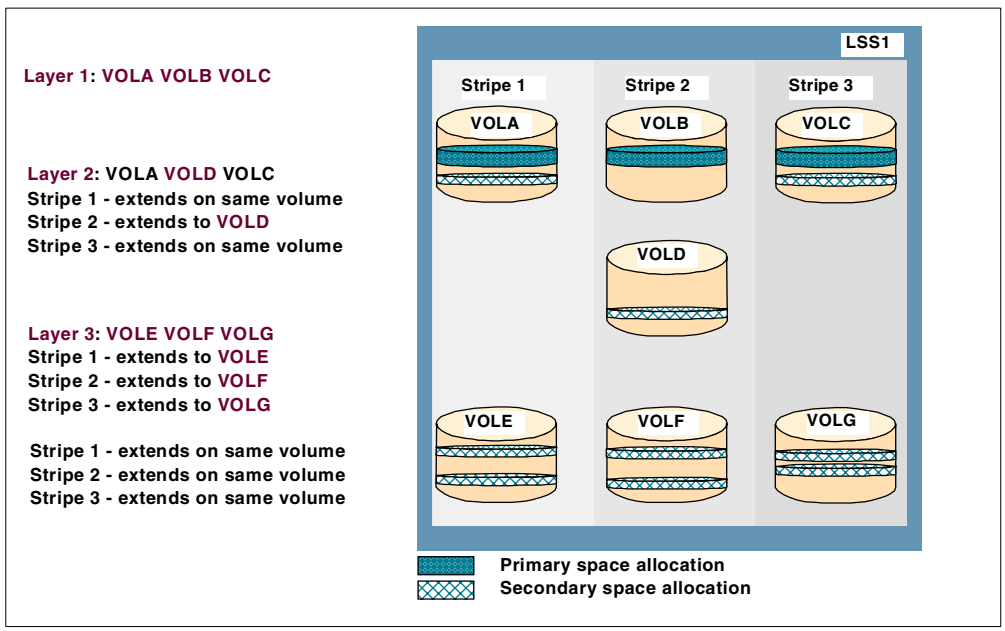



*Figure 44. Example of multi-layering*

VSAM stripe extent processing works this way: If no more space is available on one volume, a new volume must be acquired for that stripe. If the stripe is multi-layered, the total number of extents equals 255 times the number of stripes. A maximum of 123 extents per volume is possible. Space reduction percentages used for space constraint relief cannot be used for a striped VSAM data set.

You can check whether a data set is striped by setting up a LISTCAT job on a data set. The STRIPE-COUNT attribute must be greater than 1. Figure 45 shows an example.

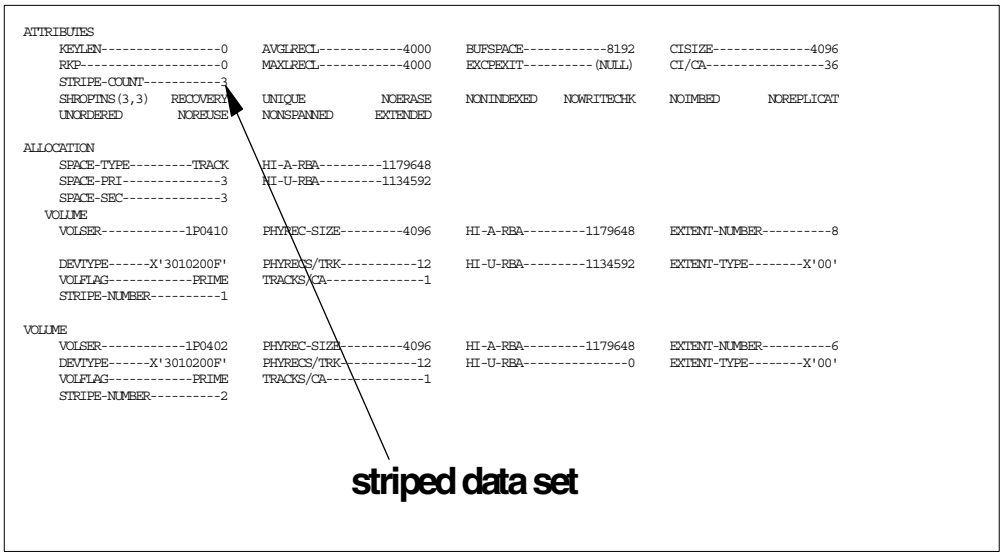



*Figure 45. Sample LISTCAT job with striped data set*

DFSMSdss supports the following functions for striped data sets:

- Data set copy

  Concurrent copy and SnapShot are supported under the following circumstances:

  – Volumes have the same track formats.
  – Data sets have the same number of stripes.
  – Data sets have the same characteristics.
  – Each target stripe must be contained on a single volume (no multi-layered stripes).
  – Source data set must not be multi-layered.
- Full volume copy
- Logical data set dump and restore
- Full volume dump and restore
- Partial release

DFSMSdss does not support physical data set dump and restore.

DFSMShsm supports all commands that can be used for non-striped VSAM data sets.

If you plan to convert VSAM data sets from non-striped to striped, you should update one or more SMS storage classes to set the sustained data rate (SDR) to a value greater than the minimum for that specific device type. Then you should take the IDCAMS REPRO command to load the new data set from the old one.

DFSMSdss does not provide conversion facilities. This means that if you plan to convert a striped data set back to a non-striped, you have to use data set copy or logical data set restore.

Toleration is provided for DFSMSdss to prevent lower releases from processing striped data sets. This is described in APARs OW24645 and OW31473. APAR OW41297 contains toleration maintenance for error messages that older releases can produce when an attempt is made to open or extend a VSAM extended format data set that is striped.

### 5.1.2 Support for large tape block sizes

Tape performance on modern tape drives, such as IBM Magstar 3590, is limited by the 32 KB block size limit for BSAM or QSAM. When we take a look at other platforms, we see much larger blocks used. Larger blocks make more sense in tape processing, because they result in fewer blocks read or written. This reduces overhead and improves performance. With DFSMS, the following maximum block sizes are now valid:

- 256 KB for 3590 tape drives
- 64 KB for all other tape drives
- <=32 KB as in the past for DASD devices and SYSOUT
- 2 GB as a new limit, but currently not used by any device

Tapes must have IBM standard labels, nonstandard labels or no labels. For IBM standard labels, a new 10-character field in bytes 71-80 contains the block length, if the 5-character block length field of data set label 2 (HDR2, EOV2, EOF2) contains all zeros.

You can specify the block size via JCL parameters, the TSO ALLOCATE command or SVC 99. If you do an allocation via JCL, you can specify it as follows:

```
BLKSIZE=262144
BLKSIZE=256K
BLKSIZE=2048M
BLKSIZE=2G
```

The following software components provide support for this large block size:

- Assembler programs using BSAM or QSAM
- IEBGENER
  IEBGENER accepts large block size as input (SYSUT1) and produces large block size output (SYSUT2) if it is requested for the specific device type. Input maximum block size must be specified, output block size may be specified.
- Volume mount analyzer (VMA)
- DFSMShsm ABARS for user tape data sets (ABARS tapes remain at 32 KB)
- DFSMSrmm
- DFSORT (supports large block interface (LBI))
- COBOL with OS/390 Release 10 as target platform

SMS modifications are also done for this item. A new read-only variable for SMS-ACS routines is introduced. It is called &BLKSIZE and is available for tape and DASD data sets. In an ACS routine you can check for the value of

&BLKSIZE, and if it is greater than 32760, then you can direct the data set to tape and not to DASD. If you try the allocation on DASD, it will fail. The ACS routines get the value of &BLKSIZE from a DD statement, dynamic allocation, or TSO ALLOCATE. For VSAM data sets, the value of the variable is 0.

A new field called BLKSIZE is added to the first page of the ACS Test Case Define panels in ISMF.

For coexistence with DFSMS/MVS V1R2, V1R3, V1R4 and V1R5 you should apply the PTFs referred to in APARs OW40414, OW40629, OW41030 and OW41865. The PTFs ensure that an ABEND is issued if a program detects larger block sizes and block formats.

### 5.1.3 DADSM rename duplicate data set

This is a very interesting new item, because prior to this support the renaming of a data set in use was not possible if its name was used anywhere in a GRS complex. This behavior made it sometimes difficult to clone systems, because system data sets like SYS1.PARMLIB, SYS1.PROCLIB, or even SYS1.VTAMLIB get serialized at IPL time and stay serialized as long as the system is up and running.

Starting with DFSMS, the new function DADSM RENAME allows the rename of duplicate non-VSAM, non-SMS-managed data sets, that have the same name as active data sets. This gives continuously available systems the ability to handle inactive data sets (from the perspective of the cloning system) with the same name as active data sets.

A System Authorization Facility (SAF) call is added to determine if the requester is authorized to rename a serialized data set. A new RACF facility class, STGADMIN.IGG.DPRN.dsn, is introduced to permit access to the RENAME function.

SMF record type 18 (rename non-VSAM data set status) now has a new indicator that gives information that RENAME was completed using STGADMIN.IGG.DPRN.dsn:

- The old field SMF18RV1 (X'84') was defined as reserved for three bytes.
- Now SMF18RV1 is at X'85' and is 2 bytes long.
- A new 8-bit field, SMF18FLG, has SMF18FAC with bit 0, indicating that serialization is not performed.

### 5.1.4 UNIT=AFF support for tape libraries

Unit affinity means that two or more volumes are assigned to the same device within the same job step allocation. It is used to minimize the number of tape drives for a job and also to stack multiple data sets on one specific tape.

Starting with Release 10 the system can determine whether a referencing data set on a DD statement resides on SMS-managed DASD, SMS-managed tape, or on non-SMS-managed volumes. This relates also to data sets that are not stacked. In the past this was not possible for non-stacking jobs that specified UNIT=AFF without VOL=REF=. Now you can use your SMS ACS routines to direct allocations to either DASD or tape, depending on data characteristics. You do not need to change your JCL.

The read-only variable &UNIT is changed so that it can contain several new values, which are:

```
&UNIT='AFF=SMSD'
&UNIT='AFF=SMST'
&UNIT=NSMS'
```

Figure 46 shows how you can code your ACS routines with these new values. The old value &UNIT='AFF=' did not say whether the data set was directed to SMS-managed DASD or tape, or to non-SMS volumes. This could lead to job failures.

This new support enables you to increase the utilization of your automated tape library or virtual tape server.

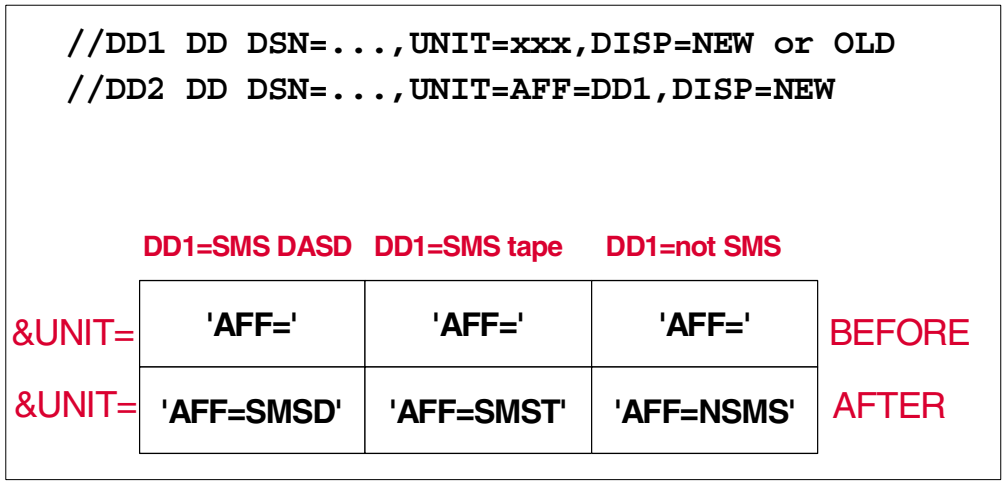



*Figure 46. ACS read-only variable &UNIT with new values*

### 5.1.5 Coupling facility structure rebuild for catalogs

When this function is enabled, a loss of coupling facility connectivity or structure failure is detected by the enhanced catalog sharing (ECS) function. In this case, you have to set up the following command at a system console:

```
SETXCF START,REBUILD,STRNAME=SYSIGGCAS_CAS
```

No user interaction is required after a rebuild.

## 5.2 DFSMShsm enhancements

The DFSMShsm enhancements to Release 10 consist of improved availability and improved system throughput.

The improved availability enhancement is:

Concurrent copy on data set backups

The enhancements in the area of improved system throughput are:

- Data set backup direct to tape
- Data set backup with multitasking
- Fast subsequent migration

- Multi-address-space DFSMShsm

### 5.2.1 Concurrent copy enhancement

The concurrent copy function of the 3990 DASD controller is now supported with the DFSMShsm data set backup commands. Concurrent copy is specified in management classes for SMS-managed data sets. If you use it within DFSMShsm, it overrides SMS management class specifications. Here it can also be used for non-SMS-managed data sets. The BACKDS keyword for concurrent copy is CC. Possible values for CC are:

**STANDARD** That is, without using concurrent copy.

**PREFERRED** Specifies that concurrent is the preferred backup method, if available and you are authorized. Otherwise, STANDARD is taken.

**REQUIRED** Concurrent copy must be used, otherwise the command fails.

**PHYSICALEND** Control returns to the application after physical end of the operation.

**LOGICALEND** Control returns to the operation when concurrent copy initialization completes.

If a concurrent copy session is initiated successfully, resource serialization is released to allow other jobs to access the original data set.

From a security point of view, you have to activate the RACF facility class STGADMIN.ADR.DUMP.CNCURRNT.

An example of a BACKDS command with concurrent copy required and logical end is as follows:

```
BACKDS USERA.PRIVATE.CNTL TARGET(TAPE) CC(REQUIRED LE)
```

### 5.2.2 Data set backup direct to tape

The target device category for backup can now be DASD or tape. Your data sets go to tape if you, as an end user, specify TARGET(TAPE) on the following data set backup commands:

- BACKDS
- HBACKDS
- ARCHBACK
- ARCINBACK

If you do not specify this parameter, DFSMShsm selects the output device (refer to Figure 47 on page 80).

An example of a BACKDS command for a cataloged data set is as follows:

```
BACKDS USERA.PRIVATE.CNTL TARGET(TAPE)
```

If this data set is not cataloged, the command is as follows:

```
BACKDS USERA.PRIVATE.CNTL VOLUME(OS3TS1) UNIT(3390) TARGET(TAPE)
```

### 5.2.3 Data set backup multitasking

In previous releases of DFSMS/MVS, data set backup was single threaded in the DFSMShsm component. Starting with this release, there is a new keyword, SETSYS DSBACKUP. This keyword has subparameters that specify the number of tasks directed to ML1 DASD or the number of tasks directed to tape. The sum of all tasks cannot exceed 64. This means that you can now backup up to 64 data sets at one time from one DFSMShsm host. You can also demount continuously mounted backup tapes. The DEMOUNTDELAY subparameter lets you tailor the times when DFSMShsm mounts or demounts tapes

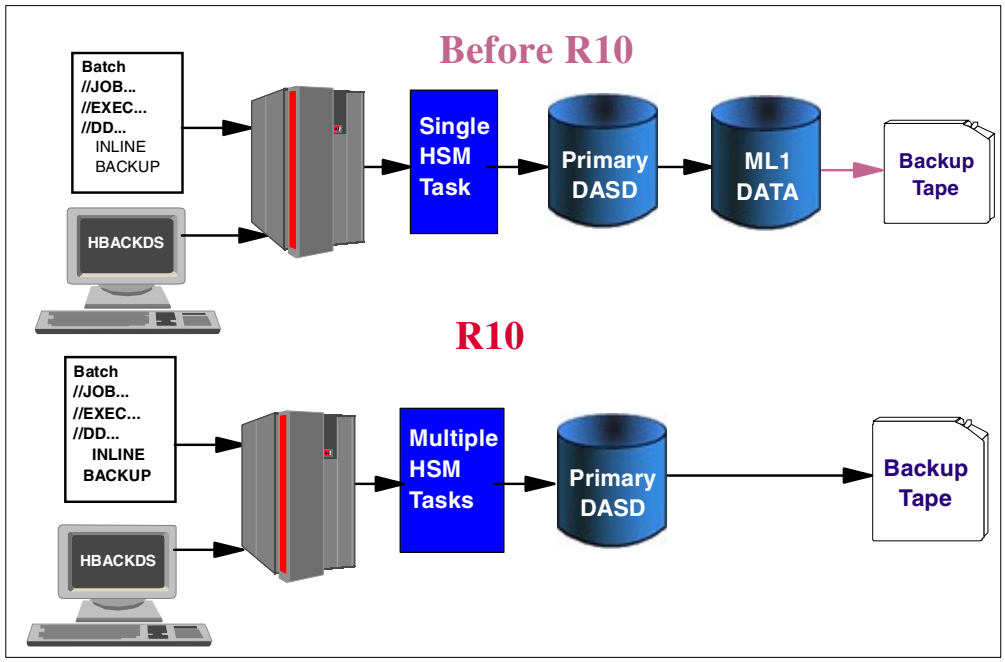



*Figure 47. DFSMShsm data set backup enhancements*

The effect of giving the users the ability to direct backup to tape might increase the number of tape drives and mount actions.

Here is the syntax of this new SETSYS command:

```
SETSYS DSBACKUP
   DASDSELECTIONSIZE(maximum standard)
   DASD(TASKS(aa))
   TAPE(TASKS(bb))
   DEMOUNTDELAY(MINUTES(nn) MAXIDLETASKS(drives))
```

The following defaults exist:

- *aa*, *bb* are set to 2.
- *maximum* is set to 3000 (KB); allowed range is 0..99999.
- *standard* is set to 250 (KB); allowed range is 0..99999.
- *nn* is set to 60; allowed range is 0..1440.
- *drives* is set to 0; allowed range is 0..64.

The subparameter DASDSELECTIONSIZE balances the workload between DASD and tape tasks for all WAIT type requests that do not go to tape. It is only usable if tape and ML1 DASD are allowed for command data set backup.

*maximum* is the size of the largest data sets that are directed to ML1 DASD, if a WAIT type request is processed. *standard* is the largest size of a small data set. Under the same circumstances as described, DFSMShsm directs this data set to DASD if a tape is not available for immediate processing.

The value of MINUTES is the number of minutes that you want DFSMShsm to wait before it deallocates the tape that is associated with continuously inactive command data set backup tasks. If you set its value to 1440, then DFSMShsm does not unmount the tape until the command data set backup tasks are stopped or DFSMShsm stops.

MAXIDLETASKS describes the maximum number of tape drives that DEMOUNTDELAY can accommodate.

### 5.2.4 Fast subsequent migration

Fast subsequent migration is a new DFSMShsm function. It is used for data sets that have been recalled from a tape volume but which are not changed after the recall function. They can now be remigrated to tape by not moving them physically, but by creating and updating the HSM control records. For this purpose, a new keyword for SETSYS TAPEMIGRATION is introduced: RECONNECT. It can be used with the following values:

**NONE** Specifies that no reconnection is done.

**ALL** Specifies that reconnection to migrating level 2 is done for each eligible data set.

**ML2DIRECTEDONLY** Specifies that only data sets that are normally migrated directly to ML2 are reconnected this way.

**Note:** The ML2DIRECTEDONLY keyword specifies that you want reconnection only on data sets that are eligible for direct migration to ML2 tapes. Data sets that are not eligible for direct migration to ML2 tapes migrate normally with no attempt to reconnect. If you specify either the RECONNECT(ALL) or the RECONNECT(ML2DIRECTEDONLY) parameters, ensure that no other product in the installation has reset the change flag in the VTOC data set entry. If it has, DFSMShsm will be unaware that the data set has changed since it was last recalled. Once the data set changes, the ML2 migration copy can no longer be used for reconnection.

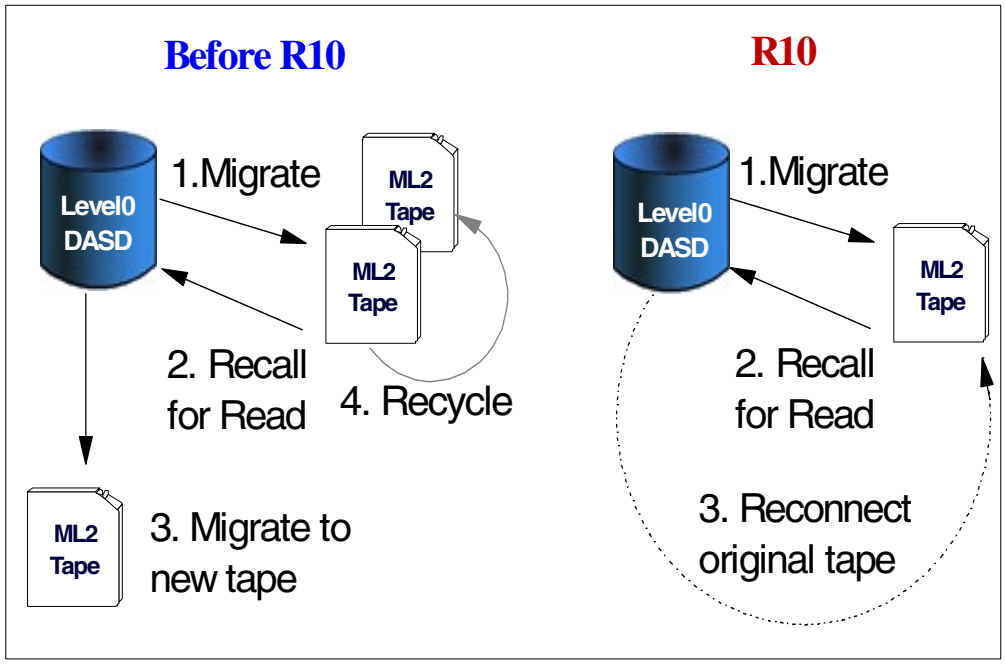



*Figure 48. DFSMShsm fast subsequent migration*

When a data set that has been migrated to ML2 tape is recalled, the migration copy of the data set remains on the tape. DFSMShsm CDS records recognize that copy as an invalid copy. When the data set that is recalled is remigrated in the normal manner, it is copied to a new tape, as shown in Figure 48 on page 82 in the before Release 10 example.

The fast subsequent migration function of Release 10 allows an unchanged data set that is recalled from a single ML2 tape (for input processing only) to be reconnected to that ML2 tape. This eliminates unnecessary data movement and tape mounts during remigration. It also reduces the need to recycle the tape. Reconnection can take place during individual data set migration or during volume migration.

Once a data set that is recalled from tape meets normal eligibility criteria, the fast subsequent migration (a fast remigrate) occurs during volume migration. Fast subsequent migrations occur ahead of migrations that cause data movement. Because the data is still on tape from the previous migrate, the fast subsequent migration involves both creating and updating DFSMShsm control records. It does not involve moving the data to tape.

#### 5.2.4.1 SETSYS MIGRATIONCLEANUPDAYS considerations

During the migration cleanup function, you can increase the likelihood of a successful reconnection with the SETSYS MIGRATIONCLEANUPDAYS command. If you use this command, you can delay the deletion of the MCD records according to a value that you have specified. DFSMShsm calculates the date that the migration cleanup function deletes MCD records. The basis for the date is the value supplied by the reconnectdays parameter. Remember that this ability to keep the MCD records may also cause your MCDS to grow.

Data sets that are eligible for ML2 reconnection with the fast subsequent migration function may have their MCD records retained for a longer period of time than for nonreconnectable data sets. This increases the likelihood of a reconnection when the data set next migrates. However, keeping the MCD records longer (while increasing the likelihood of reconnection) also causes growth in the size of the MCDS.

For migration and coexistence purposes you should consider that the migration and cleanup phase of secondary space management on down level systems (DFSMS/MVS V1R5 and lower) deletes the migration control data set records for such data sets that could be reconnected. This is based solely on the value of parameter:

SETSYS MIGRATIONCLEANUPDAYS (recalldays statdays reconnectdays)

recalldays Set a decimal number between 1 and 999 that specifies the number of days that DFSMShsm keeps MCDS data set records for recalled data sets that are not compacted, or are compacted and meet or exceed the current value of SETSYS COMPACTPERCENT. For data sets that are compacted but do not meet the current value of SETSYS COMPACTPERCENT, the MCDS data set records are retained for 90 days.

statdays Set a decimal number from 1 to 999 that specifies the number of days DFSMShsm keeps the daily and volume statistics records. The REPORT command uses these records, which the IDCAMS DCOLLECT function also collects.

reconnectdays Set the number of days that you want to add to the predicted remigration date to control retention of the MCD record for reconnectable data sets. The predicted remigration date is based on the number of days that the data set was unreferenced prior to the date of its last migration. MCD records for reconnectable data sets are kept until

**Note:** There is no coexistence PTF available. For your planning of secondary space management, this means that you should run it on a system that has OS/390 Release 10 running.

### 5.2.5 Multi-address-space DFSMShsm

Starting with OS/390 DFSMS Version 2 Release 10, you can use up to 39 HSM address spaces in a single HSMplex, as shown in Figure 49. All address spaces can then share the same set of control data sets (BCDS, MCDS, OCDS). Working together with workload manager (WLM), you can prioritize these address spaces by WLM velocity goals or by dispatching priority. You are also able to control the address spaces by console commands.

There are benefits for multi-address-space HSM. It offers less contention for the task I/O resource SYSZTIOT. This means each SYSZTIOT resource serializes within its own address space. Each HSM instance can be assigned to specific work and so get an appropriate dispatching priority for that type of work, as shown in Figure 49 on page 84 in the Release 10 part of the figure.

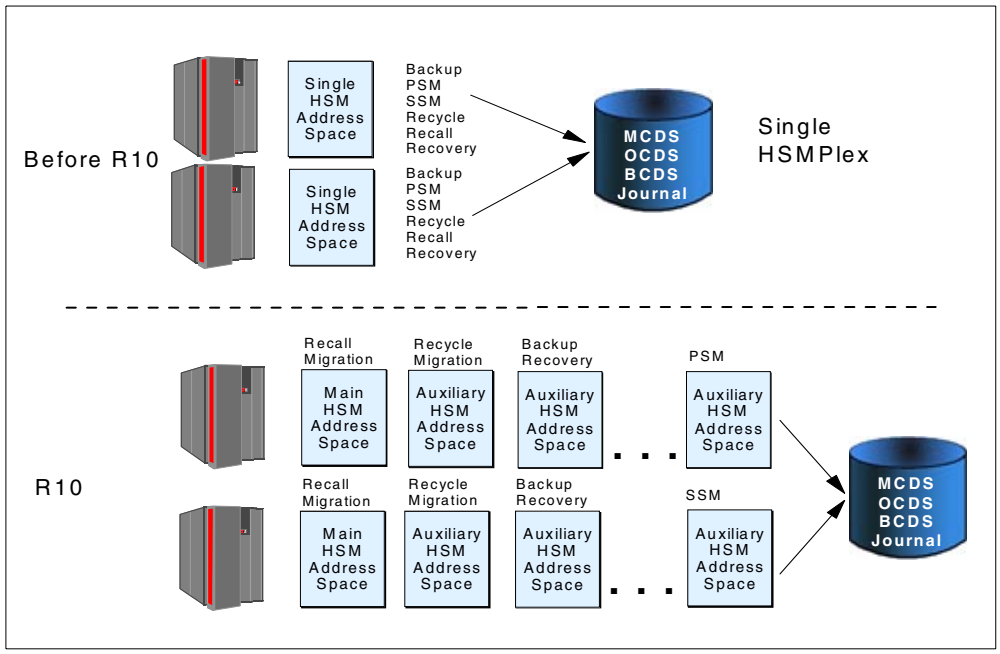



*Figure 49. Multiple address spaces with DFSMShsm*

There is a new keyword, HOSTMODE, on the DFSMShsm startup procedure. It can be used with two values:

**HOSTMODE=MAIN** This is the default. It processes implicit requests, such as recalls, and deletes migrated data sets from user address spaces. It processes all explicit commands from TSO, like HBACKDS. This mode also manages the ABARS secondary address space, allows MODIFY commands from a console and runs automatic backup, dump and space management.

**HOSTMODE=AUX** This value gives you the ability to use MODIFY commands from a console, and can also run automatic backup, dump, and space management.

Within one OS/390 image, only one DFSMShsm address space can run with HOSTMODE=MAIN. All other instances have to specify HOSTMODE=AUX.

Another new startup keyword, PRIMARY, specifies whether this is the primary HSM host within the HSMplex or not. It can be used with two values:

**PRIMARY=YES** This is the default and offers automatic backup and dump functions.

**PRIMARY=NO** This means that automatic primary and secondary space management can be performed on any host within the HSMplex.

If PRIMARY has no valid value and HOST=xy is set, then the following cases are possible:

- HOST=xY means that this is the HSM primary host.
- HOST=xN means that this is not the HSM primary host.

- If HOST= has no valid secondary character (Y(es) or N(o)), then this is the primary HSM host.

From a coexistence point of view, this means that hosts in this environment can share data with lower-level hosts on other OS/390 images when they are in the same HSMplex. But you cannot start an additional HSM host within one participating image if it is not specified as a host that is able to work in such an environment.

You must share HSM control data sets and journal data sets in such an environment. This requires either CDSQ=YES or CDSSHR=RLS for control data set serialization to be set in the DFSMShsm startup procedure.

The second alternative requires VSAM record level sharing to be implemented. If CDSR=YES is set, HSM serializes the control data sets with a shared ENQ/RESERVE. This means that all HSM hosts have to have HOSTMODE=MAIN active and use the same serialization technique. If you specify neither CDSQ nor CDSR, then you may have two further situations: Either HOSTMODE=MAIN (then DFSMShsm assumes itself CDSR=YES), or HOSTMODE=AUX, in which case you get error message ARC0093I. HOSTMODE=MAIN is the only allowed solution if CDSQ=NO is specified and CDSSHR=RLS is not specified. If CDSSHR=NO, DFSMShsm does no multiprocessor serialization and no other processors share control data sets. In this case, only HOSTMODE=MAIN is allowed.

## 5.3 DFSMSrmm enhancements

The DFSMSrmm component is enhanced in the following items:

- Multivolume set retention and movement
- Support for Tivoli OPC
- Pre-ACS interface support
- SMS ACS support
- Virtual Tape Server (VTS) support
- Fast tape positioning with DFSMSrmm
- Audit support for CDS against TCDB and library manager

### 5.3.1 Multivolume set retention and movement

This support provides an option to process multivolume, multi-data set tapes as aggregates for retention and movement. It makes DFSMSrmm more compatible with other tape management products during the period of migration to DFSMSrmm, and also gives you more flexibility in managing your data.

Member EDGRMMxx of SYS1.PARMLIB has two new parameters, OPTION RETAINBY and OPTION MOVEBY, that specify whether multivolume data sets are retained as sets or as individual volumes.

If you consider activating this function, ensure that all systems that share the DFSMSrmm control data set are on the same level.

### 5.3.2 Support for Tivoli OPC

DFSMSrmm now provides support for Tivoli OPC. This includes a set of programs and procedures to manage DFSMSrmm inventory management jobs. You can use the sample job EDGJLOPC located in SYS1.SAMPLIB to run the Tivoli OPC batch loader utility to set up DFSMSrmm as an application whose scheduling is managed by OPC. It uses the following workstations as defaults:

**STC1** Computer workstation for running started tasks

**CPU1** Computer workstation for running batch jobs

**PRT1** Workstation for printing movement reports

**TLIB** Manual workstation for use by tape library or operator to mark movement of volumes completed

If you use this tool, you must ensure that your workstations are defined to your OPC subsystem.

This interface performs regular tasks like the following:

- Scheduled tasks that run in a predefined order:
  - Verify
  - Backup
  - Main
  - INERS
  - Expire
  - Scrlist
  - Ejects
  - Report
  - Confirm
- Event-triggered tasks, as required:
  - Journal threshold is reached, backup is required to clear the journal.
  - Low on scratch, expiration processing is required to return pending volumes to scratch status and produce new scratch lists.

A special resource is provided to prevent the DFSMSrmm housekeeping task EDGHSKP from running more than one time in a given time frame.

If you want to set up the DFSMSrmm/OPC interface, you have to perform the following steps:

- Copy members EDGJLOPC, and EDGJHKPA of SYS1.SAMPLIB to your own proclib.
- Customize the batch loader statements in member EDGJLOPC.
- Ensure that the OPC workstations required by DFSMSrmm applications are defined to OPC.
- Make the customized jobs and procedures available to OPC and your running systems.
- Customize and run EDGJHKPA to define GDG bases and create first generations if necessary.
- Change parmlib member EDGRMMxx options for BACKUPPROC and SCRATCHPROC to use new OPC sample procedures.

- Run EDGJLOPC.
- Add the event trigger tracking entries to OPC using OPC dialog.

### 5.3.3 Pre-ACS interface support

In DFSMSrmm, pre-ACS exit EDGUX100 is used to set the values for read-only variables. These variables are then used as input to the ACS routines for tape data set processing. They are:

**&MSDEST** Specifies the destination.

**&MSPARM** Specifies any additional information and has variable length as an attribute.

**&MSPOLICY** Identifies a management policy.

**&MSPOOL** Specifies a tape pool name or an SMS storage group name.

These variables are commonly used to exclude data sets from tape mount management (TMM) or to direct data sets to a specific system-managed library.

Pre-ACS processing calls the DFSMSrmm exit EDGUX100 and sets the values of the read-only variables as an input to SMS ACS processing. This means that:

- &MSPOOL is set to scratch pool, with the ACL option not used.
- &MSPOLICY is set to vital record specification (VRS) management value, which is selected by the exit.

The DFSMSrmm exit EDGUX100 is updated in this release to support pre-ACS processing. Look at the EDGCVRSX member of SYS1.PARMLIB for the updated source of this exit.

You should use the mapping macro values of IGDACERO (in member EDGCVRSX in SYS1.SAMPLIB) to make decisions with EDGUX100. Consider the following macros:

- ACEROJOB for jobname
- ACERODSN for data set name
- ACEROEXP for data set expiration date
- ACERORTP for retention period

### 5.3.4 SMS ACS support

With this release, DFSMSrmm calls the SMS ACS routines to enable management class and storage groups to be used for non-system managed tape data sets. If such a situation occurs, the assigned storage group name is used as a scratch pool name. The management class is used in place of the EDGUX100-assigned VRS management value.

It is now possible to replace the EDGUX100 scratch pool and management value assignment with SMS ACS.

The selection order for a scratch pool is:

1. The setting of a storage group name by ACS processing overrides any other choice.
2. The decision of exit EDGUX100 is used.
3. The DFSMSrmm default system-based policy is used.

From a storage group point of view you have to define at least one system-managed tape library, which can be a non-existing library. You also have to define tape storage groups that can be associated with that library. The name for a storage group can be a VLPOOL NAME. It is used as a default storage group for all volumes that are added into this pool range. But it is also possible to choose another storage group name when you add a volume by using the STORGRP operand on the DFSMSrmm ADDVOLUME command.

The management class assignment requirements are:

- The SMS subsystem is active with a valid SMS configuration.
- DFSMSrmm requests that the ACS management class routine is executed.

### 5.3.5 Virtual tape server support

Starting with this release of DFSMSrmm, new support for a volume type of *stacked* is introduced. It allows the identification of the stacked volumes in a virtual tape server (VTS) and direct management of those when they are exported. DFSMSrmm retrieves information about stacked volumes from the library manager within an automated tape library when commands are issued and when stacked volumes are moved in and out of a library. If you want to use this stacked volume support, you have to enable the EDGUTIL utility of DFSMSrmm.

The DFSMSrmm subcommands ADDVOLUME, CHANGEVOLUME, DELETEVOLUME, and SEARCHVOLUME have been changed to support stacked volumes. An example for a changed ADDVOLUME command looks like this:

```
RMM ADDVOLUME VOL000 STATUS(MASTER) LOCATION(VTS1) TYPE(STACKED)
```

If you plan to change volume information, for example with the new subparameter TYPE(STACKED), use the CHANGEVOLUME command this way:

```
RMM CHANGEVOLUME VOL001 TYPE(STACKED) NORACK LOCATION(VTS1)
```

TYPE(STACKED) works only for virtual tape servers.

**Notes**

- For avoiding coexistence problems with down-level systems sharing the same DFSMSrmm control data set, you *must* look at APARs OW36349, OW36350, OW 37516, OW38469, and II12295.
- Once you enable stacked volume support, you cannot disable it again.

If you have stacked volume support enabled, you can no longer track the movement of logical volumes by using the LOCATION and DESTINATION fields of the volume information. A volume that is exported to a stacked volume has a value in the field “In Container” if you look at panel “DFSMSrmm Volume Details” within DFSMSrmm, and also has no location name assigned.

If you return stacked volumes to the VTS, you can build import lists either using the logical volumes or the stacked volumes.

### 5.3.6 Fast tape positioning with DFSMSrmm

With this new technique, DFSMSrmm enables the use of tape block IDs for applications that do not support those IDs. DFSMSrmm records the starting and ending tape block IDs and stores them into the control data set. The advantage of this function is that it enables high speed positioning to multiple stacked tapes. This makes sense when you use high-speed, high-capacity tape drives like IBM Magstar.

### 5.3.7 Audit support for CDS against TCDB and library manager

DFSMSrmm audit support checks the data in the control data set against the OAM TCDB and the library manager within the IBM 3494 tape library. It also does synchronizing of these data sets. The EDGUTIL VERIFY processing for SMSTAPE does the following when you specify:

| | |
|---|---|
| VERIFY(ALL) | DFSMSrmm resets the error indicator that is set when the control data set recovery processing was not successful. |
| VERIFY(VOLCAT) | DFSMSrmm compares TCDB information with information known to DFSMSrmm. |
| VERIFY(SMSTAPE) | DFSMSrmm also retrieves the Library Manager information for each system-managed volume and compares the information to the TCDB and DFSMSrmm information. |

Use EDGUTIL MEND(SMSTAPE) to synchronize the TCDB and Library Manager with DFSMSrmm. When you specify MEND, without SMSTAPE, EDGUTIL checks that DFSMSrmm is not active or that the DFSMSrmm control data set is not in use. MEND processing is performed the same way as VERIFY(ALL) and VERIFY(VOLCAT) processing. MEND processing cannot fix all discrepancies but those that can be fixed automatically are corrected by updating the control data set.

Before you use this function on your systems with a shared control data set, look at APAR OW37516 for toleration maintenance.

# Chapter 6. DCE DFS and SMB support

OS/390 Distributed File Service (DFS) supports two distributed file protocols: DCE DFS and SMB. DCE DFS is the abbreviation for Distributed File Service of the Distributed Computing Environment. SMB stands for Server Message Block and is the file and print protocol of the Windows environment.

DFS is a service that allows you to get access to remote files. As a server, DFS runs in a Distributed Computing Environment on most UNIX systems, Windows NT and also on OS/390. Clients are also available on all common operating systems, including the OS/390 platform. OS/390 DFS clients can connect to any DFS server implementation. This includes also OS/390 servers. OS/390 DFS servers can handle requests from any DFS client.

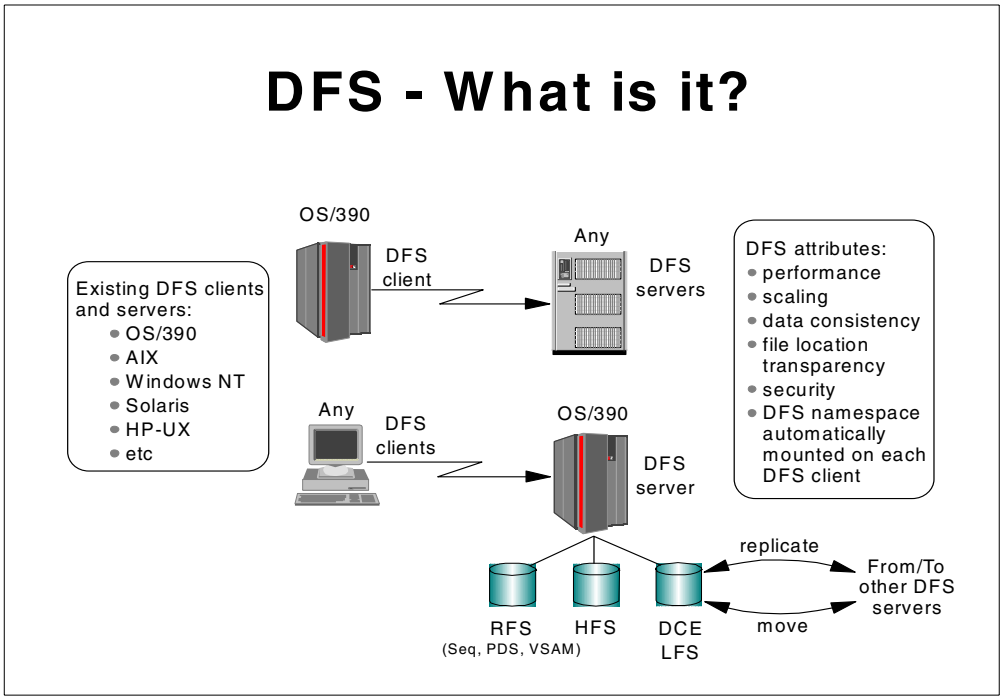



*Figure 50. DCE DFS overview*

The support of OS/390 DFS servers consists of three kinds of data:

- Record File System (RFS)

  RFS means normal OS/390 data sets. These include all types of VSAM, except linear data sets, partitioned data sets (PDS) and sequential data sets. From a client's perspective, they are presented as hierarchical files in the DFS global namespace.

- Hierarchical File System (HFS)

  HFS files are normally accessible from the OS/390 UNIX System Services shell, but they can also be made available to DFS clients.

- Distributed Computing Environment Local File System (DCE LFS)

  DCE LFSi s another file system, which comes from the UNIX platform and behaves very much like HFS does. It is part of DFS and provides functions in

addition to those known in normal UNIX file systems. With its replication mechanism, it provides administrators the ability to spread the data onto different DFS servers. The client access is transparent to users and applications. This is helpful in case of system outage. Moving files is also possible. If one DFS server is busy, an administrator can move files via command to another server. This is also transparent for users and applications.

You can find a graphical overview on DCE DFS in Figure 50 on page 91.

The OS/390 SMB server, which was first introduced in Release 9 (also as beta code in Release 8), supports PC clients for file and print serving. If you plan to use print serving, OS/390 Info Print Server must be installed on your system. On the client side, no special software is required. On the server side, OS/390 DCE must be installed. This is part of OS/390 base.

You do not need to configure or activate DCE. Communication between client and server is done via TCP/IP (NetBIOS over TCP/IP). In OS/390 Release 10, HFS and RFS are supported. The following PC client platforms are supported:

- Windows 95
- Windows 98
- Windows NT 4.0 (Workstation)
- Windows 3.11 (Windows for Workgroups)
- OS/2 WARP Version 4 (file serving only)

Figure 51 shows you how SMB and DCE DFS protocols work inside the DFS server.

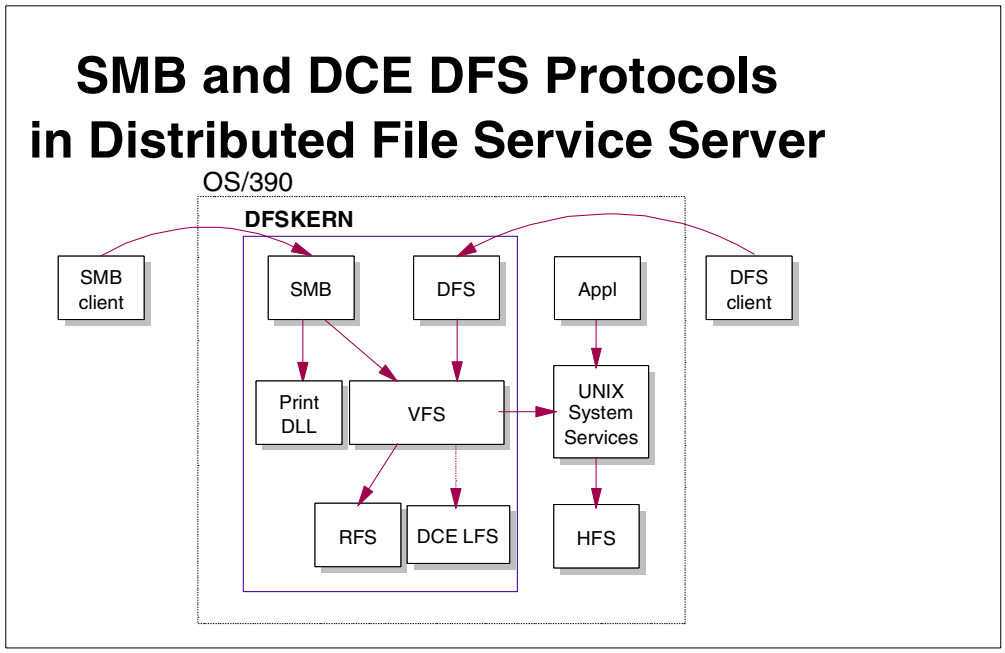



*Figure 51. SMB and DCE DFS protocols*

Here you have a detailed view of *dfskern*. It is the process in the DFS address space that receives file protocol requests. You can see that dfskern handles DCE DFS clients and also SMB clients. DFS clients can access Hierarchical File

Systems, Record File Systems and the DCE Local File System. SMB clients can get access to HFS, RFS and also to printers.

The OS/390 Distributed File Service server can be started with the SMB capability, the DFS capability, or both at the same time.

There are two abbreviations in Figure 51 on page 92, which are described now:

**VFS:** VFS means Virtual File system. This can be seen as a switch that determines whether clients want to access HFS, RFS or DCE LFS.

**DLL:** DLL means Dynamic Link Library. It is a set of routines that can be called without the need to link edit the routines into the calling load module. In this environment they are needed to handle print requests that come from the SMB protocol.

## 6.1 DFS enhancements in Release 10

In general, better serviceability and performance has been provided to the DFS component of OS/390 Release 10.

The major enhancement for DFS local file system (LFS) in this release is the ability is provide file sizes greater than 32 bit. This means that you can access files which are larger than 4 GB.

The DCE Local File System is a high-performance, log-based file system. It supports the use of *aggregates*. An aggregate is physically equivalent to a standard UNIX disk partition. It also contains specialized metadata about the structure and location of information on the aggregate. LFS maintains a log of al its modifications made to the metadata by operations like file creation and modification. The log is transparent to the LFS users and requires no administration. If the system terminates abnormally, LFS gets the logged information about the metadata back and uses it to rebuild the aggregate into a consistent state.

Aggregates also support the use of *filesets*. An LFS fileset is a hierarchical grouping of files managed as a single unit. It may vary in its size but is always smaller than a disk partition. Multiple filesets can be stored on a single aggregate, providing flexible disk usage.

### 6.1.1 DFS LFS enlarged file size

The maximum file size which can be processed by an OS/390 DFS server is based on the structure of the physical file system where the data resides. For DCE Local File System, the maximum file size is based on the following formulas:

```
maximum_file_size = minimum(4GB * fragment_size, 2GB * blocksize)
maximum_fileset_size = 4GB * fragment_size
maximum_aggregate_size = 4GB * 4K
```

For the fragment size, the following values are valid:

```
1024 < fragment_size < blocksize
```

The fragment size must be a power of 2.

The blocksize may have the following values:

```
4096 < blocksize < 65536
```

The blocksize must be a power of 2.

The default is that the fragment size is 1024 and the blocksize is 8192. This gives a maximum file size of 4 TB.

**Note:** We recommend you use Windows Workpad, instead of Notepad, when you plan to browse or edit especially large OS/390 files.

## 6.2 SMB Enhancements in Release 10

SMB in Release 10 also provides better serviceability and performance, as DFS does.

The other topics, which are also new or enhanced, are:

- Support for SMB dialect called NT LM 0.12
- PC access to Record File System data (RFS)
- New environment variables
- Changed login mechanisms

These enhancements are detailed in this chapter.

### 6.2.1 SMB dialect NT LM 0.12

SMB protocol consists of so-called dialects. At the time of OS/390 Release 10, NT LM 0.12 is the highest dialect of Server Message Block. In this dialect, a new SMB called SMB_CREATE_ANDX is introduced. This SMB is used to create files.

Additionally, with this dialect a new password encryption mechanism is introduced. In this case, passwords are *not* converted to upper case.

### 6.2.2 Record File System

Starting with Release 10, Record File System (RFS) support is added to SMB. This means that SMB clients can access OS/390 data sets. This includes sequential data sets, partitioned data sets and VSAM data sets. Data sets can be exported and shared to SMB clients. From a client's view they appear as hierarchical files. Partitioned data sets look like directories, their members again like hierarchical file within that directory. The data can be translated from EBCDIC to ASCII and vice versa. RFS file systems are not mounted locally (they are not real "file systems"). Each one is a separate, stand-alone tree.

OS/390 data sets are allocated when a PC client gets access to them.

In Figure 52 on page 95 you can see what a structure with an HFS file system and an RFS file tree looks like. Remember that the RFS is a stand-alone tree. This means that an RFS is neither mounted nor connected to any existing HFS hierarchy.

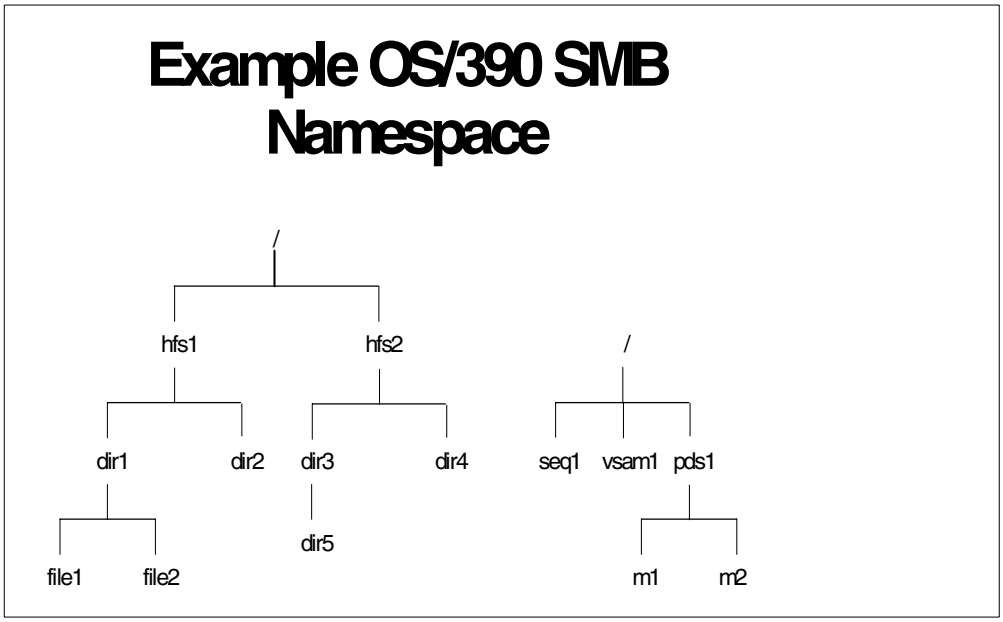



*Figure 52. OS/390 SMB namespace with HFS and RFS*

Figure 53 shows the same structure, but this time you can also see the file systems.

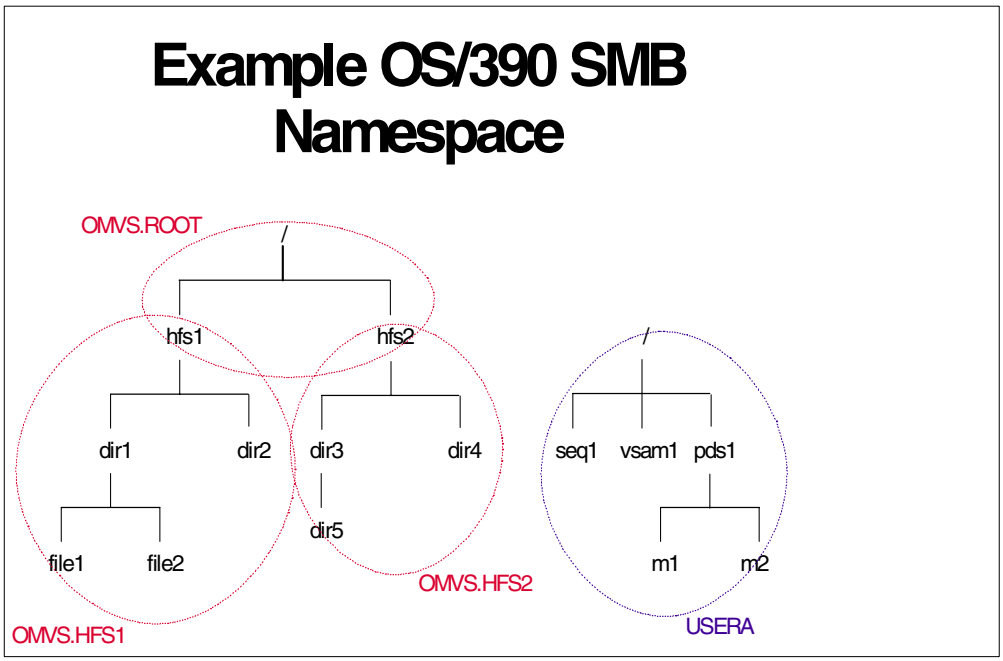



*Figure 53. OS/390 SMB namespace with HFS and RFS with file systems*

The RFS hierarchy on the right side of the figure is a “file system” with a virtual root. It consists of OS/390 data sets beginning with TSO prefix USERA.

This means MVS data sets with the following names exist:

- USERA.SEQ1, which is a sequential data set.

- USERA.VSAM, which is any VSAM data set, but without VSAM linear data sets.
- USERA.PDS1, which is a partitioned data set with members M1and M2.

If another sequential data set exists that has the name USERA.SEQ1.DATA, this one would appear directly under the RFS root directory (/) with a file name of seq1.data.

There are also new environment variables for RFS. You can find their descriptions and possible values in Table 3 on page 97. The variables are related to the dfskern.

Note that the variable _IOE_RFS_ATTRIBUTES_FILE can point, not only to an HFS file, but also to an MVS data set. If this is the case, the syntax is the following:

```
_IOE_RFS_ATTRIBUTES_FILE='//HLQ.DATASET(MEMBER)'
```

If it is an MVS data set, it can either be a fixed-block partitioned data set or a fixed-block sequential data set with a record size of 80.

The rfstab is a file or data set which describes the attributes used to manipulate RFS files in the SMB server. It also contains descriptions for:

- Data set creation attributes
- Processing attributes
- Site attributes

Here you can see an example of an rfstab file:

```
lrecl(80)
recfm(fb)
blksize(6160)
space(100,10),blks
dir(250)
lf
maplower
```

You may recognize that many of them look like JCL DD statements. `lf` means that the line feed character will be put at the end of each record for text files. `maplower` means that the OS/390 data set without the data set prefix will be mapped to lower case letters before returning the file name to the PC client. It is not possible to work with mixed case file names.

The SMB server can also use the same attributes file as the DFSMS/MVS Network File System server (NFS server). NFS server attributes that are not supported by the SMB server are ignored.

We describe the implementation of an RFS connection in 6.3, “Setting up an SMB RFS connection” on page 98.

### 6.2.3 New environment variables

Server Message Block are daemons that run as separate processes within the DFS address space. The DFS control task dfscntl is the parent process of all daemons. Under that server task run the child processes dfskern and export.

The export daemon communicates with dfskern to make a specified file system available on the network and stops after doing so. The dfskern daemon runs in a loop and handles incoming file and print requests.

The settings for these processes are specified by giving values to specific environment variables. The envar files contain these sets of environment variables. The envar file is located in the corresponding home directory of each process. It is an EBCDIC file, residing in a HFS file, which can be edited with OEDIT editor within UNIX system services.

In OS/390 Release 10 are some new environment variables, which are described in Table 3. They are related to performance and tuning.

*Table 3. SMB - new environment variables*

| Envar Name | Values (Default with underline) | Description |
|---|---|---|
| _IOE_LFS_IO_SEEK | ON, OFF | Changes the seek algorithm for LFS |
| _IOE_SMB_PATHCACHE_NODES | 0 (to disable it), 5<=x<=200, 30 | Number of SMB patch cache nodes per session |
| _IOE_SMB_OPLOCK__TIMEOUT | 0<x<2G, 35 | How long (in seconds), the server will wait for Oplock callback response before closing a file |
| _IOE_SMB_ISSUE_TIMEOUT_MSG | ON, OFF | Issue a message, if Oplock callback response never received |
| _IOE_RFS_ALLOC_TIMEOUT | >=30, 300 | Interval (in seconds) of inactivity that will cause the server to deallocate a data set |
| _IOE_RFS_ATTRIBUTES_FILE | pathname, /opt/dfslocal/var/dfs/rfstab | Name of HFS file that contains the table for describing attributes for RFS files (also called the rfstab) |
| _IOE_RFS_STATUS_REFRESH_TIME | >0, 600 | Interval (in seconds) for how often the server refreshes its cache of RFS file names and attributes |
| _IOE_RFS_TRANSLATION | ON, OFF | Parameter which decides whether server treats RFS data as text and converts from ASCII to EBCDIC and back |
| _IOE_RFS_WORKER_THREADS | >0, 1 | Number of threads to service open and close requests for RFS files |

### 6.2.4 Login mechanisms

In the initial release of SMB, password encryption was not provided. This changed with Release 9. Now you use the environment variable _IOE_SMB_CLEAR_PW, which has an initial value of REQUIRED. This means that password are transmitted in a clear form. The syntax in the envar file is:

```
_IOE_SMB_CLEAR_PW=REQUIRED
```

The variable and its value is used by the dfskern process. Other possible values for the variable are:

**ALLOWED** This specifies that passwords may be transmitted clear, Authentication is attempted, using encrypted passwords, if the client supports this.

**NOTALLOWED** This specifies that clear passwords are not allowed. Authentication is attempted using encrypted passwords only.

If passwords are sent in an encrypted way, you have to install and configure Open Cryptographic Services Facility (OCSF), which provides at least Security level 2 (with 56-bit DES encryption). OCSF is an optional element of OS/390.

There are two forms of encrypted passwords in OS/390 Release 10. These are LANMAN, which is used by the LANMAN dialect and supported since Release 9 and NT, which is used by the NT LM 0.12 dialect and supported now starting at Release 10.

## 6.3 Setting up an SMB RFS connection

In this section, we describe how to set up an RFS connection from a Windows NT workstation to an OS/390 SMB server on Release 10.

As previously mentioned, the OS/390 SMB server supports PC client access to OS/390 data sets (RFS). OS/390 data sets are presented to PC clients as hierarchical byte stream data (with restrictions). The devtab file is enhanced to allow the administrator to specify RFS data to be exported. After it is exported, an RFS “file system” directory can be shared with PC clients. A new file, rfstab, is defined to allow specification of OS/390 data set characteristics when RFS files are created. Several new environment variables are defined to control RFS characteristics.

A Record File System (RFS) is a collection of OS/390 data sets with a common data set name prefix. These data sets are then exported by the SMB server as a single “file system”. An RFS file system is not locally mounted, so it is not part of the HFS hierarchy. However, it can still be exported by the SMB server, even though it is not part of the HFS hierarchy.

An RFS file system must be exported by the SMB server before a directory contained in it can be shared. Exporting is done on an RFS file system basis. Sharing is done on a directory tree basis and allows PC clients to access data in an RFS file system. In order to allow a directory to be shared, the file system that the directory is contained in must be exported as follows:

- Ensure that the OS/390 DFS Server is started with SMB processing enabled (SMB Server).

- Determine which HFS “trees” you want to make available.
- Update export files (smbtab, dfstab and devtab) to indicate what data should be available.
- Issue the dfsshare command to make data available (or restart the SMB Server).

### 6.3.1 Accessing files and printers

An administrator needs to make files and printers available to PC users and inform them as PC users how to access the files and printers. The steps necessary to provide SMB processing are as follows:

- First, we need to make sure that the OS/390 Distributed File Service server is started with SMB processing enabled. This is controlled by an environment variable.
- Next, we need to decide what data we want to make available to PC clients.
- Then we need to update the export files and make the data available.

The file systems that the user accesses must be exported. The directory that is at the top of the “tree” that the PC user accesses must be shared. There are three export files involved with file sharing as follows:

***smbtab***
This tells the SMB server which directory is to be shared, the file system that the directory is in, and gives a name to the directory share for PC users to use. The parameters used are as follows:

- device name - A unique device name that refers to the file system device name in the dfstab
- share name - The directory share name that the PC user will use to “connect” to the HFS data
- device type - ufs refers to UNIX File System (that is, HFS); prt refers to a printer device type
- description - A text description that shows up on a net view for the OS/390 SMB server
- permission - r/o means the share can only be accessed in a read-only fashion; r/w means that PC users can read and write to the data (assuming they are authorized)
- max users - The maximum number of users that can “connect” to the directory share; 0 means unlimited
- directory - The directory name (within the file system referred to in the device name) to be shared with the PC users

An example of a smbtab file in the HFS is shown in Figure 54 on page 100. This file can be accessed in the HFS using the following command:

```
/etc/dfs/var/dfs/smbtab
```

```
# device  share    type share label    permissions max users path
#
/dev/ufs2 umikem   ufs  "Mike's home"  r/w           100    /
/dev/ufs3 nichola  ufs  "Als share  "  r/w           100    /
#/dev/ufs4 root     ufs  "wtsc43oe root" r/w          100    /
/dev/prt2 postscr  prt  "A PS Printer" pokeps "3130IBM"
/dev/prt1 text     prt  "A text Printer" poke "TextOnly"
/dev/prt3 text1    prt  "A text Printer" poke "TextOnly"
```

*Figure 54. smbtab file in the HFS*

### *dfstab*

dfstab provides file system information. The parameters to be specified are as follows:

- device name - unique device name of the file system to be exported
- file system name - unique file system name
- file system type - for HFS file systems, this must be ufs
- file system id - unique file system numeric identifier
- fileset id - unique fileset id to be associated with the HFS file system. It is a 64-bit number, so it is represented as two 32-bit numbers separated by two commas.

An example of a dfstab file is shown in Figure 55 on page 100. This file can be accessed in the HFS using the following command:

```
/etc/dfs/var/dfs/dfstab
```

```
# device  fs-name  type  fs-ID  fileset-ID
/dev/ufs2 homemike ufs   101    0,,1715
/dev/ufs3 homeal   ufs   102    0,,102
```

*Figure 55. dfstab file in the HFS*

### *devtab*

devtab provides the file system data set name. The parameters specified are as follows:

- device name - unique device name of the file system to be exported in the format define_ufs n, where n is the number in the device name in the dfstab and smbtab
- HFS data set name - the data set name of the HFS file system to be exported. This may be optionally followed by a character data translation parameter.

An example of a devtab file is shown in Figure 56 on page 101. This file can be accessed in the HFS using the following command:

```
/etc/dfs/var/dfs/devtab
```

```
* command  minor device
define_ufs 2
* HFS dataset
OMVS.SC43.MIKEM
define_ufs 3
OMVS.SC43.NICHOLS
*define_ufs 4
*OMVS.OS390R8.SC43.O38RC1.ROOT
~
```

*Figure 56. devtab file in the HFS*

### 6.3.2 User requests for files and printing

When PC users send requests (file or print) to the OS/390 SMB server, the requestor is identified by their SMB user ID. Since HFS data is authorized by a requestor's OS/390 user ID, the incoming SMB user ID must be mapped to its corresponding OS/390 user ID. This is accomplished by the OS/390 SMB server based on a configuration file called the smbidmap file. The SMB server administrator creates this file and puts an entry for each SMB user ID and its corresponding OS/390 user ID.

The administrator can also specify that if the SMB user ID is not found in the smbidmap file (or the password was invalid), the SMB user ID can be considered to be the OS/390 user ID. (The SMB user ID must be eight characters or less, in this case.) Finally, if the logon still fails, the administrator can specify that a default (or guest) user ID can be used. The default OS/390 user ID is specified in an environment variable and must, of course, be a valid OS/390 user ID. If there is no default OS/390 user ID, the SMB request is denied.

Authorization to files and directories is handled in the normal manner with the chmod command to change permissions or the chown command to change the owner.

The smbidmap file is located via the _IOE_SMB_IDMAP environment variable.

The smbidmap file is a text file that the administrator creates and maintains. It must be an HFS file. Any editor available on OS/390 USS may be used (for example, oedit, vi, etc.) The smbidmap file contains one or more mapping declarations and has the following general format:

```
SMB-user-ID1
OS/390-user-ID1

SMB-user-ID2
OS/390-user-ID2

*
=
```

Figure 57 on page 102shows the smbidmap file. This file can be accessed in the HFS using the following command:

```
/etc/dfs/home/dfskern/smbidmap
```

```
R O G E R S 0 1
R O G E R S

R C O N W A Y 1
R C O N W A Y

U S E R 1 0 0
U S E R 1

U S E R 2 0 0
U S E R 2

*
=
```

*Figure 57. smbidmap file*

Each entry has two elements: the SMB user ID and OS/390 user ID. A blank line is required between entries. The first line of each entry is the SMB user ID to be mapped. It can be either a simple SMB user ID or it can be qualified by a Domain name. (There may be two SMB users with the same ID in different Domains.) If it is not qualified by Domain, that means you don't care which Domain it came from (that is, it could have come from any Domain or no Domain).

The second line is the corresponding OS/390 user ID.

Several SMB user IDs can be mapped to the same OS/390 user ID.

A special entry allows you to specify that if the SMB user ID is not found in the smbidmap file, or the login was unsuccessful, then the SMB user ID should be considered to be an OS/390 user ID. This entry has * for the first line and = for the second line. This will only be done if the SMB user ID is eight characters or less.

Again, if all of the above fails to successfully login, then the _IOE_MVS_DFSDFLT environment variable is checked. If it exists and it specifies a valid OS/390 user ID, then the SMB requestor is mapped to that user ID. Otherwise, the request is denied.

# Chapter 7. Resource Measurement Facility enhancements

The Resource Measurement Facility (RMF) is the strategic IBM product for performance management in an OS/390 host environment.

RMF consists of several components:

- Monitor I - Monitor II - Monitor III
- Postprocessor
- Performance Monitoring of OS/390
- Client/Server Enabling
- Spreadsheet Converter / Reporter
- Sysplex Data Server

These components work together in providing the capabilities you need for performance management:

- Gathering data
- Reporting data
- Accessing data across the sysplex

Monitor I provides continuous long-term data collection for system workload and resource utilization. Single-system reports can be obtained directly as real-time reports for each completed interval. You can run the Postprocessor to create single-system reports or sysplex reports.

RMF Monitor II provides online measurement on demand for use of resources. It allows the user to evaluate the current level of resource utilization and helps in solving immediate problems.

RMF Monitor III provides continuous measurement of system status by collecting short-term data. It provides online reports that allow the user to identify delays for jobs that are important to overall system performance. These reports show the amount of delay that is being caused and identify the main impactor, helping the user in solving performance problems.

Running Monitor II and Monitor III simultaniously as ISPF applications is a practical way to get different views of the current performance of the system.

This chapter discusses RMF enhancements that are being provided by OS/390 Version 2 Release 10. These enhancements bring new and improved functions that allow better monitoring and reporting with this new release of RMF in several areas:

- Online Monitoring with Monitor II and Monitor III
  - VSAM RLS Support
  - Multi-system Enclave Support
  - UNIX System Services Support
  - Parallel Access Volume (PAV) Support
  - S/390 64-Bit Architecture
  - Dynamic Central Processor Upgrade
- Online Monitoring with PM of OS/390
  - PM of OS/390 Java Edition
- Long-Term Reporting with the Postprocessor

- WebServer Performance Reporting
- Lotus Domino Support
- FICON Director

- Performance Analysis with the Spreadsheet Reporter
  - Spreadsheet Enhancements

## 7.1 Online Monitoring with RMF Monitor II and Monitor III

The Monitor III gatherer session has a typical gathering cycle of one second, and consolidated records are written for a range which is typically set to 100 seconds.

You can collect short-term data and continuously monitor the system status to solve performance problems. You get actual performance data (response times, execution velocity) on a very detailed level for later comparison with performance policy goals.

### 7.1.1 VSAM RLS support

RMF provides online support for VSAM RLS performance analysis and problem determination. Online monitoring is necessary if several applications (for example, online CICS or batch programs) access the same set of files and can impact CICS transactions and batch throughput.

Monitor III offers some new reports to support monitoring of VSAM RLS measurements like buffering and locking activities, cache structure effectiveness, and resource contention:

- VSAM RLS activity by storage class or by data set
- VSAM LRU overview

This support is focusing especially on customers planning to exploit transactional VSAM (TVS) in their IT shop. TVS is based on VSAM RLS but adds logging to allow concurrent access on recoverable data sets by CICS online and batch.

#### 7.1.1.1 Gathering data for VSAM RLS support

In order to enable RMF in gathering data for VSAM RLS Support, you must specify the optional keyword for the Monitor III data gatherer; this option is depicted in Table 58.

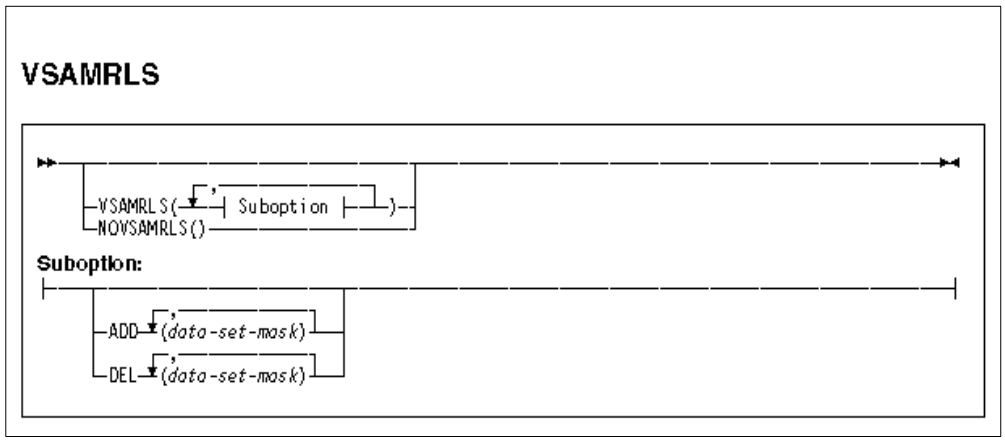



*Figure 58. VSAM RLS: data collection control options*

This option controls the collection of VSAM RLS activity data, so when you specify | VSAMRLS or allow the default value to take effect, activity data is gathered for | VSAM RLS by storage class. In addition, data set masks can be specified to collect | data by VSAM spheres, too. To suppress the gathering of VSAM RLS data, specify NOVSAMRLS.

The collection of VSAM RLS activity data by VSAM spheres can be controlled by following suboptions:

- **ADD** - Start collection for all VSAM data sets which are covered by the mask.
- **DEL** - Stop collection for all VSAM data sets which are covered by the mask.

### 7.1.1.2 Reporting on VSAM RLS support

You can select VSAM RLS related reports from RMF Monitor III sessions, as shown in Figure 59:

```
RMF Sysplex Report Selection Menu

Enter selection number or command for desired report.


 Sysplex Reports
         1 SYSSUM   Sysplex performance summary                (SUM)
         2 SYSRTD   Response time distribution                 (RTD)
         3 SYSWKM   Work Manager delays                       (WKM)
         4 SYSENQ   Sysplex-wide Enqueue delays                (ES)
         5 CFOVER   Coupling Facility overview                 (CO)
         6 CFSYS    Coupling Facility systems                  (CS)
         7 CFACT    Coupling Facility activity                 (CA)
         8 CACHSUM  Cache summary                             (CAS)
         9 CACHDET  Cache detail                              (CAD)
        10 RLSSC    VSAM RLS activity by storage class        (RLS)
        11 RLSDS    VSAM RLS activity by data set             (RLD)
        12 RLSLRU   VSAM LRU overview                         (RLL)

 Data Index
         D DSINDEX  Data index                                 (DI)
```

*Figure 59. Sysplex Report Selection Menu screen*

The sample output from VSAM RLS Activity by Storage Class report is shown in Figure 60 on page 106.

### 7.1.1.3 Report header

LRU Status reports the LRU status of local buffers under control of BMF (Buffer Management Facility). The status can be:

**Good** BMF is at or below its goal on all systems.

**Accelerated** BMF is over the goal on at least one system, and the buffer aging algorithms were accelerated.

**Reclaimed** BMF is over the goal on at least one system, and the buffer aging algorithms were bypassed to reclaim buffers.

```
Samples: 117     Systems: 1    Date: 12/16/99  Time: 12.21.30  Range: 120   Sec

LRU Status   : Good
Contention % :  0.0
False Cont % :  0.0

Stor Class  Access  Resp   -------- Read ----------  ------ BMF -------  Write
                    Time    Rate  BMF%   CF%  DASD%  Valid%  False Inv%  Rate

RLS
             DIR    0.001  15.46  86.6   0.0   13.4   99.1      0.92      0.03
             SEQ    0.000   0.00   0.0   0.0    0.0    0.0      0.00      0.00
RLS1
             DIR    0.001  15.80  90.5   0.0    9.5   98.5      1.55      0.04
             SEQ    0.000   0.00   0.0   0.0    0.0    0.0      0.00      0.00
RLS2
             DIR    0.002  11.63  89.7   0.0   10.3   98.0      2.03      0.04
             SEQ    0.000   0.00   0.0   0.0    0.0    0.0      0.00      0.00
RLS3
             DIR    0.002  11.78  89.5   0.0   10.5   97.7      2.32      0.05
             SEQ    0.000   0.00   0.0   0.0    0.0    0.0      0.00      0.00
RLS4
             DIR    0.002  12.64  78.9   0.0   21.1   85.6     14.44      0.05
             SEQ    0.000   0.00   0.0   0.0    0.0    0.0      0.00      0.00
```

*Figure 60. Sample output from VSAM RLS Activity by Storage Class report*

The contention % shows the true lock contentions: all external requests issued by connectors delayed due to contention on a lock. False contention percentage shows the percentage of false lock contentions: all external requests issued by connectors that experience “hash contention”.

## 7.1.2 Multi-system enclave support

An *enclave* is a transaction that can span multiple dispatchable units (SRBs and tasks) in one or more address spaces, and is reported on and managed as a unit. It is managed separately from the address space it runs in. CPU and I/O resources associated with processing the transaction are managed by the transaction’s performance goal and reported to the transaction.

Multi-system enclaves are transactions that originate on any system and continue (in parallel) on other systems in the Parallel Sysplex.

The Monitor III Enclave report has been enhanced to identify multi-system enclaves and their owners in a sysplex. It improves performance management for all applications (for example, DDF or ICSS Web server) using this new technology.

The Postprocessor Workload Activity reports now presents enclave transactions separate from address space transactions, and shows the amount of parallelism for multi-system enclaves.

### 7.1.2.1 Requesting the enclave report

To request the Enclave report from RMF Monitor III, select `O` on the Primary menu, and then select `5` on the Overview Report selection menu (shown in Figure 61 on page 107). Or you can enter the following command:

```
ENCLAVE [subsystem-type]
```

```
 RMF Overview Report Selection Menu
Selection ===>

Enter selection number or command for desired report.


 Basic Reports
        1 WFEX     Workflow/Exceptions                          (WE)
        2 SYSINFO  System information                           (SI)


 Detail Reports
        3 DELAY    Delays                                       (DLY)
        4 GROUP    Group response time breakdown                 (RT)
        5 ENCLAVE  Enclave resource consumption and delays     (ENCL)
        6 OPD      OMVS process data
```

*Figure 61. Overview Report Selection Menu*

### 7.1.3 OMVS process data report

The Monitor III OMVS Process Data report provides information about UNIX system services address spaces and server processes. It includes performance measurements for OS/390 UNIX processes (OS/390 UNIX System Services address spaces) and thus improve the ability of the OS/390 platform to manage the growing UNIX workloads.

UNIX System Services (USS) address spaces can consist of several processes which in turn might run one or more threads. Each process is typically associated with a UNIX command or service, consumes a certain amount of CPU time, and also provides state information. USS is the brand name for UNIX on MVS, and in the MVS context it is referred to as *open MVS* or, shortly, *OMVS*.

The new support provides data to answer questions like:

- What are the delayed processes?
- What command is associated with the process?
- What is the status of each of the processes?
- Which processes are high CPU consumers?

The data provided by the OMVS command is accessible via a callable interface. It includes CPU information about the different processes owned by a TSO user, process threads and their state, as well as global settings of the kernel address space itself. RMF provides this information in the Monitor III OMVS Process Data report and thus improves problem determination as well as performance management for UNIX workloads.

#### 7.1.3.1 Gathering data for UNIX System Services support

In order to enable RMF in gathering data related to UNIX System Services, you must specify the data gathering options of Monitor III, as listed:

**OPD** Specifies measurement for OMVS process data

**NOOPD** No measurement is done for OMVS process data

OMVS data is collected on a data space attached to RMFGAT, created when the OPD has been specified.

### 7.1.3.2 Reporting on UNIX System Services

In order to get the OMVS process data reports, you have to choose option `6` from the RMF Overview Report Selection Menu (see Figure 62) or type the command `OPD`:

```
 RMF Overview Report Selection Menu
Selection ===>

Enter selection number or command for desired report.


 Basic Reports
        1 WFEX     Workflow/Exceptions                          (WE)
        2 SYSINFO  System information                           (SI)


 Detail Reports
        3 DELAY    Delays                                      (DLY)
        4 GROUP    Group response time breakdown                (RT)
        5 ENCLAVE  Enclave resource consumption and delays     (ENCL)
        6 OPD      OMVS process data
```

*Figure 62. RMF Overview Report Selection Menu*

The report options defaults are shown on this panel (see Figure 63):

- Selection = 1 (Process ID view)
- ALL process IDs

As alternatives to presenting information for all or a specific process ID, the user can choose to request the report for:

- A specific ASID (where the ASID can be specified in decimal or hexadecimal format, but is always shown in decimal format on the report).
- A specific jobname
- A specific user name

Note that when invoking the report and no options have been set, regardless whether via command or selection, the default options are honored.

```
 RMF OMVS Process Data Report Options
Command ===>                                          Scroll ===>  HALF

Change or verify parameters. To exit press END.
Select one of the following options:

 1  1. Process ID  ===> ALL      ALL or a process ID
    2. ASID        ===>          ID of an address space in decimal or
                                 hexadecimal (with preceeding X) format
    3. Jobname     ===>          Jobname associated with a process
    4. User        ===>          User name associated with a process
```

*Figure 63. RMF OMVS Process Data Report Options*

The OMVS Process Data panel shows the information, according the criteria specified by the user. Figure 64 on page 109 shows the resulting report for the

selected options ALL Process IDs. The selection is shown on the right side of the header area.

The report is sorted by ascending process IDs.

```
RMF 2.10  OMVS Process Data                    Line 1 of 10
Command ===>                                          Scroll ===> HALF

Samples: 100     System: RMF6  Date: 02/10/00  Time: 13.03.20  Range: 100   Sec

Kernel Procedure: OMVS      Kernel ASID: 0014        Option: PID      ALL
BPXPRM: OMVS=(69)

------------------------------------------------------------------------------
Jobname   User      ASID        PID       PPID  LW  State  Appl%  Total  Server

BPXOINIT  OMVSKERN  0028          1          0       MF      0.0  0.152   FILE
INETD7    OMVSKERN  0032          6          1       1FI     0.0  0.113   N/A
BHOL7     BHOL      0064   16777238   50331672       HA      1.6  13.36   N/A
BHOL6     BHOL      0531   16777239   50331672       HA      2.1  6.358   N/A
BHOL3     BHOL      0534   33554457   50331672       HA      3.1  35.10   N/A
RMFGAT    STCUSER   0063   33554458          1       1FI     1.3  13.23   N/A
BHOL5     BHOL      0530   33554460   50331672       HA      1.9  11.71   N/A
BHOL      BHOL      0025   50331669          1       1FI     0.0  23.23   N/A
BHOL      BHOL      0025   50331672   50331669       1FI     0.0  23.23   N/A
BHOL4     BHOL     0535   50331675   50331672      HA     3.5  9.858   N/A
```

*Figure 64. OMVS Process Data panel*

On this panel, cursor sensitivity applies to the following fields:

- Jobname: the JOB report is invoked
- Parent PID (PPID): Report is invoked again with the parent process shown on top of the screen
- Elsewhere on the process line: OMVS Process Data Details pop-up panel, shown in Figure 65:

```
 RMF OMVS Process Data - Details

Press Enter to return to the Report panel.

Start Time/Date : 08.45.22 02/07/2000
Command         : BPXPINPR
Process-ID:        1    Parent Process-ID:        0
Jobname    : BPXOINIT    User Name         : OMVSKERN
ASID       :     0028    Hexadecimal ASID  :     001C

Appl%  :  0.0   Total CT     : 0.152   LW-PID     :        0

Server Information:
  Name : Init Process
  Type : FILE   Active Files:     0   Max. Files:     200K

Process State: MF
M: Multiple threads, no pthread create used
F: File system kernel wait
```

*Figure 65. RMF OMVS Process Data - Details panel*

You can also track OMVS address space performance data through the reports listed below (the class column is enhanced by an O, to indicate an OMVS address space):

1. Delay Report
2. Job Reports
3. Processor Delays Report

### 7.1.4 Parallel Access Volume (PAV) support

The new Enterprise Storage Server (ESS), known under its code name *Shark*, provides a feature that allows concurrent I/O to the same logical volume from an S/390 system. Such a logical volume is called a parallel access volume (PAV) and is defined as a set of devices consisting of one base device plus a variable number of alias devices. The Workload Manager can instruct the system to dynamically add alias devices in order to reduce I/O queue times caused by increasing I/O requests to one volume, and based on goals and importance for the workloads affected by such delays.

The maximum number of alias devices for a PAV is shown on the Device Activity reports (for example, the Monitor II Device Activity report). Device performance statistics are provided for the entire PAV.

Now, RMF Monitor I and II reports provide device performance statistics for the entire PAV, including the maximum number of alias devices for a PAV. No changes are required to get these reports.

### 7.1.5 Enterprise Storage Server

The Enterprise Storage Server (ESS) is the latest IBM storage product to be developed using IBM's Seascape architecture. It provides all the open system functions of the Versatile Storage Server, and it provides all the functions of the 3990 Storage Control too - and a lot more. However, this section describes only features and functions that are relevant in an OS/390 environment.

In ESS, there are exciting new features that significantly enhance performance. These features, which together with OS/390 software deliver a new world to the S/390 storage environment, are probably the biggest change since disk caching was introduced.

The Enterprise Storage Server is the natural successor to the IBM 3990. It provides all functions that were available on the 3990, including peer-to-peer remote copy (PPRC), extended remote copy (XRC), and concurrent copy. For the many customers who have installed the RAMAC Virtual Array (RVA), with its revolutionary log structure files (LSF) architecture, ESS is protecting their investment in this technology, too.

#### 7.1.5.1 Architecture

The Enterprise Storage Server is a high performance, high availability, high capacity storage subsystem. It contains two 4-way RISC processors with 6 GB of cache and 384 MB of non-volatile storage to protect from data loss. It has a maximum capacity of over 11 TB and is connected to S/390 through up to 32 ESCON channels.

The first two models of the Enterprise Storage Server that became available are:

- The IBM 2105-E20 Enterprise Storage Server: This model supports the full complement of 128 disks in two 2105 cages.
- The IBM 2105-E10 Enterprise Storage Server: This model has a limited capacity in terms of disk arrays; only 64 disks can be installed in the base rack. These occupy a single 2105 cage.

Ongoing and frequent ESS performance upgrades can be expected as a direct result of its underlying Seascape architecture. The first example of such upgrades, the ESS Models F10 and F20, incorporates 64-bit RISC processing, higher effective host adapter utilization, more PCI buses, larger cache, and other improvements at the component level. The maximum sequential throughput capability of the ESS Model F20 is approximately double that of the ESS Model E20.

Disks can be installed in the cages in groups of 8. These are called *disk 8-pack*s. Each group of 8 disks is configured as a RAID Rank (of either 6 Data + Parity + Spare, or 7 Data + Parity) or JBOD (Just a Bunch Of Disks) with no Parity.

For each 8-pack, you have the choice of three different disks for use:

- 9.1 GB - 10K RPM: Use this disk size for the highest performance RAID ranks.
- 18.2 GB - 10000 RPM: Use this disk size for the high performance and capacity - this is the optimum choice for most applications.
- 36.4 GB - 7200 RPM: Use this disk size for the high capacity and standard performance.

These guidelines are applicable for most workloads, however, disk performance is most applicable to cache-unfriendly operations that rely on disk performance characteristics for application performance.

A RAID rank is formatted as a set of Logical Volumes (LV). The number of LVs in a rank depends on the capacity of the disks in the array. The LVs are striped across all the data and parity disks in the array.

```
------------------------------------------------------------------------------------------------------------------
                                                RAID RANK ACTIVITY
------------------------------------------------------------------------------------------------------------------
ID   RANK    DA  HDD  SECT   -------- READ REQ -------    ------- WRITE REQ -------  ---------- HIGHEST UT
     TYPE            SIZE   RATE  AVG MB  MB/s  RTIME    RATE  AVG MB  MB/s  RTIME

*ALL                        5000   0.003    78      3     800   0.113    76      4
0500 RAID-5  13    7   524   2000   0.003    60      2      50   1.420    10      4  SK2440 SK2403 SK2510
0501 RAID-5  13    7   524   3000   0.003    90      3     750   0.107    80      4  SK3390
0502 RAID-5  13    7   524      0   N/A     0.0      0       0   N/A     0.0      0
```

*Figure 66. Raid Rank Activity Report*

A non-RAID rank, also called a JBOD rank, is very different because each disk in the group of 8 is a rank in itself. So there are 8 ranks in a JBOD group. Each JBOD disk can be defined as one or more LVs. It is not RAID-protected and, should a disk fail, all data on it is lost.

### *JBOD or RAID 5?*

A group of JBODs are normally used in a situation where you need high random write performance and you do not need RAID protection. With the improvement of ESS, the known disadvantages of the RAID 5 write penalty have been eliminated.

Therefore, IBM recommends that customers use RAID 5, which delivers very high performance, the best data protection, and fault tolerance. There is no known JBOD performance benefit on the ESS.

#### 7.1.5.2 OS/390 Parallel Sysplex I/O management

In the OS/390 Parallel Sysplex, the Workload Manager (WLM) controls where work is run and optimizes the throughput and performance of the total system. Until now, WLM management of the I/O has been limited. With ESS, there are some exciting new functions that allow WLM to control I/O across the sysplex. These functions include parallel access to both single system and shared volumes and the ability to prioritize the I/O based upon the WLM goals. The combination of these features can significantly improve performance in a wide variety of workload environments.

### 7.1.6 ESS performance features

This section covers the performance features of the ESS including PAV, multiple allegiance, and I/O priority queuing.

#### 7.1.6.1 Concurrent Access features

OS/390 retains device queuing features that were developed to ensure effective serial access to physical devices. These features were developed at a time when physical devices rather than emulated devices were the norm and long before RAID solutions were developed. They ensured that only one channel program could be active to a disk at any time. This ensured that there was no possibility of interference between channel programs.

OS/390 interacting with the ESS relaxes some of these restrictions. Two features are introduced to allow this:

- Multiple allegiance
- Parallel Access Volumes (PAV)

Although there may be a large number of concurrent operations active to a particular logical volume, the ESS ensures that no I/O operations that have the potential to conflict over access to the same data is scheduled together. Effectively no data can be accessed by one channel program that has the potential to be altered by another active program. Channel programs that are deemed to be incompatible with an active program are queued within the ESS.

#### 7.1.6.2 Multiple allegiance

S/390 device architecture has defined that a state of implicit allegiance exists between a device and the channel path group that is accessing it. This allegiance is created in the control unit between the device and a channel path group when an I/O operation is accepted by the device. The allegiance causes the control unit to guarantee access, no busy status presented, to the device for the remainder of the channel program over the set of paths associated with the allegiance. This concept has been expanded to support the ESS with the concept of multiple allegiance.

ESS's concurrent operations capability supports concurrent accesses to or from the same volume from multiple channel path groups, system images. The ESS's multiple allegiance support allows different hosts to have concurrent implicit allegiances, provided that there is no possibility that any of the channel programs can alter any data that another channel program might read or write.

Multiple allegiance requires no additional software or host support, other than to support the ESS. It is not externalized to the operating system or operator. Multiple allegiance reduces contention reported as PEND time.

Resources that benefit the most from multiple allegiance are:

- Volumes that:
  - Have many concurrent read operations
  - Have a high read-to-write ratio
- Data sets that:
  - Have a high read to write ratio
  - Have multiple extents on one volume
  - Are concurrently shared by many users

#### 7.1.6.3 Parallel Access Volumes

OS/390 systems queue I/O activity on a unit control block (UCB) that represents the physical device. High parallel I/O activity can adversely affect performance, because traditionally, high accesses correlate to high levels of mechanical motion. In subsystems with large caches and RAID arrays, performance was adversely affected because the volumes were treated as a single resource that was serially reused. This contention is worse for large volumes with numerous small data sets. The symptom displayed is extended IOSQ time. The operating system cannot attempt to start more than one I/O operation at a time to the device.

The ESS's concurrent operations capabilities also support concurrent data transfer operations to or from the same volume from the same system. A volume accessed in this way is called a Parallel Access Volume (PAV).

PAV exploitation requires both software enablement and an optional feature on your ESS. PAV support must be installed on *each* ESS. It enables the issuing of multiple channel programs to a volume from a single system, and allows simultaneous access to the logical volume by multiple users or jobs. Reads, as well as writes to different extents, can be satisfied simultaneously. The domain of an I/O consists of the specified extents to which the I/O operation applies. Writes to the same domain still have to be serialized to maintain data integrity.

Support is implemented by defining multiple UCBs for volumes. The UCBs are of two types:

- Base address

  This is the actual unit address of the volume. There is only one base address for any volume.

- Alias address

  Alias addresses are mapped back to a base device address. I/O scheduled for an alias is physically performed against the base by the ESS. No physical disk space is associated with an alias address; however, they do occupy storage within OS/390.

  Alias UCBs are stored above the 16 MB line.

The workloads most likely to benefit from the PAV function include:

- Volumes that have many concurrently open data sets, for example, volumes in a work pool
- Volumes that have a high read-to-write ratio per extent
- Volumes reporting high IOSQ times

Candidate data types are:

- High read-to-write ratio
- Many extents on one volume
- Concurrently shared by many readers
- Accessed using media manager or allocated as VSAM extended format

#### 7.1.6.4 I/O priority queueing

If I/Os cannot run in parallel due to, for example, extent conflicts, the ESS internally queues I/Os. This reduces operating system overheads incurred by having to post device busy, and redriving channel programs. The ESS queues I/Os in the order in which they are received. This helps to reduce problems that occur when one processor can respond to interrupts faster than a sharing one, and therefore monopolize a device.

You also have the option to enable priority queuing of I/Os to the ESS. WLM sets a priority bit in the CCW when running in goal mode. Priority queuing is within a sysplex and queuing is at a volume level. The ESS queues I/O requests in the order specified by WLM. I/O may be queued in the following situations:

- An extent conflict exists for a write operation.
- To allow servicing of a cache miss, the device reconnects when data has been staged to cache.
- A reserve request is issued and other accesses are current with a different path group ID.

#### 7.1.6.5 Cache Management with ESS

The ESS provides performance improvements over those provided by raw disk by using caching. Caching algorithms are executed by the storage directors and determine what data occupies cache. Attached hosts can offer information about their data access intentions that the storage directors use to help select the best algorithm to use for cache management. The following caching features are provided.

***Read Caching***

Read hits occur when all of the data requested for a data access is located in cache. The ESS improves the performance of read caching by using algorithms to store in cache tracks that have the greatest probability of being accessed by a read operation. An I/O operation that results in a read hit does not disconnect from the channel and data is immediately available to the application.

If the requested data is not located in cache, a write miss occurs and data is read from disk, returned to the program, and loaded to cache. This is called *staging*. While records are being read from disk the channel program is disconnected, allowing other applications to access the channel.

There are three types of staging:

**Record staging** Only those records accessed by the channel program are staged into cache

**Partial track staging** The required records and the rest of the track are staged to cache. This is the default mode of cache operation

**Full track staging** Based on the prediction of sequential I/O processing. This can be predicted either the previous behavior of the application or by the application signalling in the I/O request that sequential access is used.

Data transferred using inhibit cache load or bypass cache attributes is loaded into the cache, but is eligible for accelerated destaging.

The ESS offers new capabilities to support the optimization of sequential performance; improved and more sensitive pre-fetching algorithms, and new channel commands that improve overheads and provide increased bandwidth.

***Write Caching***

Two forms of write caching are supported: DASD Fast Write (DFW) and Cache Fast Write (CFW). Cache Fast Write is an OS/390 function that is intended for data that does not need to be written to disk. CFW is used only when explicitly requested by the application and should only be selected for transient data, that is, data that is not required beyond the end of the current job step and that can more easily be recreated from scratch than recovered from a checkpoint.

DFW is the more usual form of write caching. With DFW, the application is told that an I/O operation is complete once data has successfully been written to cache and Non-Volatile Storage (NVS). Data integrity and availability is maintained by retaining two copies of the data until it is hardened to disk, one copy in cache on one cluster and the second in NVS of the other cluster. NVS is protected by battery backup. Normal access to the data is from the copy retained in cache.

Destaging of data that is backed up in NVS from cache to disk is based on a Least Recently Used (LRU) algorithm. Data may be retained in cache after being written to disk based on the cache activity. Destaging from NVS and cache is anticipatory and threshold-based. The intention is to always have NVS and cache resources available to accept new data.

Tracks at the top of the cache LRU list are checked for updates in NVS that have not been destaged to disk. The ESS schedules tracks at the top of the NVS LRU for destaging, so that they can be allocated without a delay during destaging.

***Write Performance***

Caching benefits write performance, as almost all writes are at cache speeds. DFW minimizes any potential penalty of RAID 5 generation of parity. This performance benefit is clearly demonstrated by the published results of ESS performance tests.

In addition, write performance is enhanced by striping. The ESS automatically stripes logical volumes across all the drives in the RAID array. This provides automatic load balancing across the disk in the array, and an elimination of hot spots. This design should reduce the amount of effort that storage administrators spend in hand-placing data, while at the same time offering performance improvements.

The ESS RAID 5 implementation gives a minimal RAID 5 write penalty for sequential writes. When writing a sequential “stripe” across the disks in an array, the ESS generates the parity only once for all the disks. This is sometimes called a *RAID 3-like implementation*, and it provides high performance in sequential operations.

### 7.1.7 S/390 64-Bit architecture

In z/Architecture mode machines, the address range for real storage is extended from 2 GB to 128 GB (currently). When running in z/Architecture mode, there is no difference between central storage and expanded storage. The entire address range is treated as central storage. RMF was changed to run in z/Architecture mode machines. The RMF data collector is changed due to modified system interfaces, and the storage-related RMF reports are enhanced to be sensitive to the mode the hardware is running in.

When running in z/Architecture mode, all metrics related to expanded storage become obsolete. No data related to expanded storage is shown in any RMF report. New metrics regarding the 2 GB line are added.

The following changes have been implemented to support the z/Architecture mode:

- Monitor I/Postprocessor
  - SMF records 70-78: New bit indicating z/Architecture mode in the RMF Product section
  - New metrics related to z/Architecture mode in SMF Record Type 71
  - New metrics in Paging Activity report
  - New Overview/Exception conditions based on SMF 71 (Paging Activity) record
- Monitor II
  - SMF record 79: New bit indicating z/Architecture mode in the RMF Product Section
  - New metric related to z/Architecture mode in SMF Record Type 79.2
  - New field in the ARD report (fixed pages between 16 MB and 2 GB)
- Monitor III
  - All storage-related reports have been changed for z/Architecture mode
  - Unreferenced Interval Count (UIC)
  - All RMF reports display the UIC with 4 digits (instead of 3)

Bit 3 of byte 49 of the RMF Product section is added to indicate whether the system is operating in z/Architecture mode. With this information, all RMF reporter modules processing SMF type 70 to 79 records can recognize z/Architecture mode and flag all expanded storage-related report information as N/A.

RMF determines whether the system is operating in z/Architecture mode by checking a bit in the Prefixed Save Area, IHAPSA.

The Postprocessor Paging Activity Report is changed for systems operating in z/Architecture mode (refer to Figure 67 on page 117); the changes are:

- In z/Architecture mode, all metrics related to expanded storage become obsolete.

- Besides the OPT member, the mode is shown (ESA or z/Architecture).
- Central Storage Section.
  - No changes when in z/Architecture mode.
- Expanded Storage Section.
  - Not shown in z/Architecture mode.
- Real Storage Section.
  - New with z/Architecture mode - shows VIO and HIPERSPACE pages in real storage.
- Frame and Slot Counts section.
  - In z/Architecture mode, expanded storage-related values are set to N/A.
  - Fixed Frames: MIN/MAX/AVG shown for frames fixed between 16 M and 2 G
- Swap Placement Activity section.
  - In z/Architecture mode, expanded storage-related values are set to N/A.

```
                                P A G I N G   A C T I V I T Y

          OS/390                  SYSTEM ID SYS1            DATE 09/27/1999    INTERVAL 60.00.378
          Rel. 02.10.00           RPT VERSION 02.10.00      TIME 09.00.00      CYCLE 0.333 SECONDS

OPT = IEAOPT00   MODE = ESAME              REAL STORAGE MOVEMENT RATES - IN PAGES PER SECOND
-------------------------------------------------------------------------------------------------------

 HIGH UIC (AVG) =  610.6  (MAX) =  2540   (MIN) =   120


                    WRITTEN TO     READ FROM       *----- REAL STORAGE FRAME COUNTS -----*
                     REAL STOR     REAL STOR            MIN           MAX           AVG

 HIPERSPACE   RT          0.02          0.00             67           140           112
  PAGES

 VIO          RT          2.93          1.96             18         1,832           169
  PAGES
```

*Figure 67. Paging Report: Real Storage section*

The following changes were made to Monitor II reports in z/Architecture mode:

- Monitor II Report - Status Line:
  - In z/Architecture mode, MIG (migration age) is not displayed.
  - UIC is displayed with 4 digits.
- ARD/ARDJ Report (see Figure 68 on page 118):
  - LSQA pages in expanded storage are not displayed (LSQA CSF).
  - Fixed pages between 16 MB and 2 GB are displayed (FF 2G).
  - Fixed pages below 16 MB are renamed from FF BEL to FF 16M.

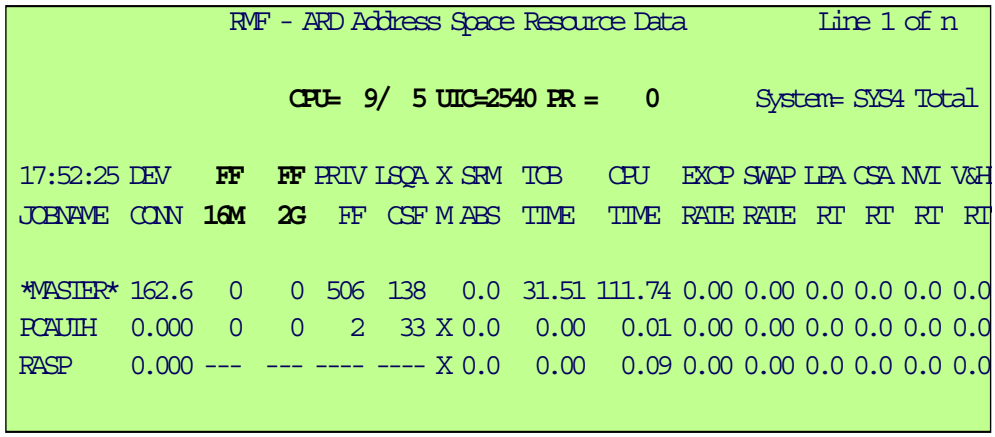



*Figure 68. ARD Report - z/Architecture mode*

- ASD/ASDJ Report:
  - Number of expanded storage frames (ESF) contains blank.
  - Rate of page movement from expanded storage (ES RT) blank.
- SPAG Report:
  - Rate of pages sent to expanded storage (ES RTE) contains blank.
  - Migration age and rate (MIG AGE and MIG RTE) blank.
  - Number of ES frames (ESF AVL) contains blank.
- SRCS Report:
  - LSQA frames in expanded storage (LSQA ESF) contains blank.

Monitor III reports, in z/Architecture mode, include the following changes in their various storage-related reports:

- JOB Report (Storage Delay)
  - ES Move Rate contains N/A in z/Architecture mode.
- Storage Delays Report (STOR)
  - Expanded Working Set: Column contains blank in z/Architecture mode.
- Storage Frames Report (STORF)
  - ES Rate: Column contains blank in z/Architecture mode.
- Storage Resource Delays Report (STORR)
  - Migration Age/Online Frames contain N/A in z/Architecture mode.
- Storage Delay Summary Report (STORS)
  - Same as STORR

With z/Architecture mode, the UIC is no longer restricted to be a number between 1 and 254. Its update takes place less frequently (every 10 seconds). Thus, the UIC may grow to 2540 to keep the view of the UIC as the “number of seconds” a page has not been referred to. The UIC is displayed with 4 digits and is shown in the following reports:

- Status area of all Monitor II reports
- Mon II SPAG report

- Mon II SRCS report
- Mon III Storage resource delays report
- Mon III Storage delay summary report
- Postprocessor Paging report

### 7.1.8 Dynamic central processor upgrade

Processor reporting supports the concurrent model upgrade function becoming available with new processor models (G5 and G6). Concurrent Central Processor upgrade is the capability to increase the capacity of the processor non disruptively.

The additional information returned by the system is reflected in the Monitor III System Information report, and in the Postprocessor CPU Activity report.

## 7.2 Online monitoring with PM of OS/390

The PM of OS/390 Java Edition provides the workstation client access to sysplex-wide OS/390 performance data from Windows 95 or NT as well as OS/2 platforms. It is a port of the PM Common Functions framework and the exploiting PM of OS/390 applications. With the availability of the Java Edition for PM of OS/390, the OS/2 version is withdrawn from service.

PM of OS/390 Java Edition is part of OS/390 RMF Release 10. It is platform-independent. It allows enterprise-wide performance monitoring of OS/390 hosts through the use of a graphical user interface. PM of OS/390 provides a selected subset of the information provided by the Monitor III gatherer.

To install the application, the user just needs a Web browser that is Java-enabled, Adobe Acrobat reader, and the InstallShield.

The InstallShield installation routine asks the user for which sysplex (TCP/IP Name or TCP/IP Address) it should configure an example PerfDesk. So, it's really very easy to start working with the application.

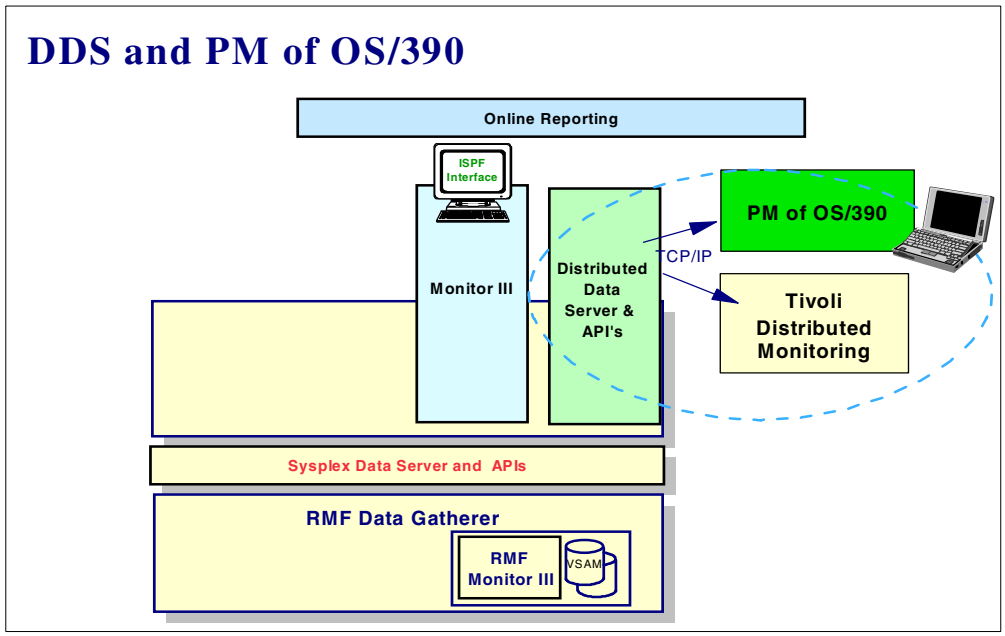



*Figure 69. DDS and PM of OS/390*

PM of OS/390 takes its input data from a single data server on one system in the sysplex, which gathers the data from RMF Monitor III on each MVS image. Therefore, this is called the Distributed Data Server (DDS). If you want to monitor several sysplexes, each needs to have an active DDS (see Figure 69).

PM of OS/390 provides a selected subset of the information provided by the RMF Monitor III gatherer: general performance data, performance data for jobs, and for systems running workload-related performance data like the following in goal mode:

- WLM workloads
- WLM service classes
- WLM service class periods
- WLM report classes

RMF Monitor III is the source of the data provided in "PM of OS/390". It is suited for monitoring and analyzing the performance in real time and in the near past. How far back in time historical data can be provided depends on the size of the Monitor III gatherer VSAM data sets allocated by the customer.

Not all metrics available by RMF are available for PM of OS/390, especially all items new for OS/390 V2R10:

- FICON channel
- Shark Enterprise Storage Server (ESS) support
- UNIX System Services

## 7.3 Long-Term reporting with the RMF postprocessor

RMF has enhanced the Postprocessor to accept the SMF record type 103 subtype 1 and 2 written by the WebServer, and offers the new HTTP Server

report, which provides usage statistics as well as performance information about the WebServer.

The Postprocessor has also been enhanced to accept the SMF record type 108 subtype 1 written by the Lotus Domino, to offer the new Domino Server report, which provides feedback on server load as well as the number and type of messages that the server handled.

The Postprocessor also provides a new FICON Director Activity report with information about director and port activities.

## 7.3.1 WebServer performance reporting

The WebServer is a strategic application in IBM's e-business portfolio. Therefore, sufficient support is required on the S/390 platform for tuning and capacity planning. A new RMF Postprocessor report, based on SMF record type 103 written by the WebServer, provides performance information and usage statistics about the WebServer. The data provided by this report helps the user in tuning and capacity planning

### 7.3.1.1 Requesting HTTP server reports

RMF does not write any SMF record related to WebServer performance. These records are created by the HTTP Server directly. The HTTP Server writes measurement data to SMF. It uses two directives to define how this is done:

- SMF defines which type of data is recorded: configuration data (103-1), performance data (103-2), both, or none.
- SMF Recording Interval defines how often performance records (103-2) are written to SMF by the elapsed time between recordings.

The SMF record type 103-1 data are taken from the server configuration file and are written once after the server daemon is fully initialized. That means that reports of especially long running servers may lack these data if the report interval does not include server start time.

The SMF record type 103-2 data are accumulated continuously. The cumulative values are counted over the life of the server process and do not reflect the single intervals.

For details, see *OS/390 HTTP Server Planning, Installing, and Using*, in Appendix B. Configuration directives. You can obtain softcopy of this book from the OS/390 Online Library Collection (SK2T-6700), the OS/390 PDF Library Collection (SK2T-6718), or the OS/390 Internet Library at:

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

Updates of this book are available in PDF and HTML formats on the HTTP Server Web site at:

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

The RMF Postprocessor has been enhanced to produce the HTTP Server Summary Report (see Figure 70 on page 122) and the HTTP Server Detail Report (see Figure 71 on page 122). The HTTP reports are invoked by the HTTP option in the Reports control statement.

```
                                  H T T P   S E R V E R   S U M M A R Y
                                                                                                                    PAGE    1
          OS/390                  SYSTEM ID 3090            DATE 02/11/2000           INTERVAL 30.00.000
                                                            TIME 18.40.00

SERVER NAME                     DURATION    REQUEST  RESPONSE  THROUGHPUT RATE    THREADS    CACHE SIZE   CACHE FILES  TIMEOUTS
                                              RATE     RATE       IN      OUT     MAX  USED   MAX  USED    MAX  USED

MVS071                          00.24.35      0.03     0.03      5.46   30.15      39  0.00  5120  0.90      0  0.00      1
```

*Figure 70. HTTP Server Summary report*

The summary report contains one line per server start/stop period within the reported interval.

If there are several HTTP Server records to be reported within one RMF reporting interval, then the values of the detail reports within the RMF reporting interval are accumulated in the summary report. That is, the summary report then is a kind of "duration report" for the RMF reporting interval size. For example, if the RMF reporting interval is 60 minutes and the HTTP Server recording interval is 5 minutes, then there will be 1 line in the summary report representing 12 detail reports.

```
                                  H T T P   S E R V E R   D E T A I L S
                                                                                                                    PAGE    2
          OS/390                  SYSTEM ID 3090            DATE 02/11/2000           INTERVAL 03.04.000
                                                            TIME 18.40.00
-------------------------------------------------------------------------------------------------------------------------------
                                    SERVER CHARACTERISTICS
-------------------------------------------------------------------------------------------------------------------------------

NAME:       MVS071                          SERVER ROOT IN HFS: /usr/lpp/internet/server_root
IP-ADDRESS: 9.67.116.9                      STARTUP:            02/11/2000-08.03.01
PORT:       8181                            SECURITY TYPE:      1
TYPE:       PROXY                           SSL-PORT:           443
APPL-LVL:   V5R2M0

--------- FLAGS ---------     MAX BUFFER          102K                    ------- CACHE -------     -- TIMEOUT THRESHOLDS --
DNS LOOKUP            YES     MAX THREADS           39                    CACHE             YES     INPUT                330
ACL SETTINGS          YES                                                 MAX SIZE         5120     OUTPUT              3600
META FILE              NO     -- GARBAGE COLLECTION --                   MAX FILES           0     SCRIPT               600
DIRECTORY ACCESS       NO     ENABLED               NO                    LIMIT 1           100     IDLE THREADS           0
SERVER IMBEDS HTML     NO     INTERVAL           10800                    LIMIT 2          4000     CACHE LOCK           160
NORMAL MODE           YES     MEMORY USE           500                    TIME MARGIN       120
GMT                    NO                                                 KEEP EXPIRED       NO
PROXY                 YES                                                 CONNECT            NO
```

*Figure 71. HTTP Server Details report*

The first part of the HTTP Server Details report contains the server characteristics as they are provided by the SMF type 103-1 records. If these data are not available, an appropriate message is printed.

Possible server types are: HTTP, PROXY, CACHING, and CACHING PROXY.

The second part of the HTTP Server Details report contains the server activity data as it is provided by the SMF type 103-2 records. This data is calculated by

two consecutive 103-2 records, which must not overlap nor gap. Note that the MIN, MAX, AVG response time values are related to the complete server runtime, not only to the current interval.

## 7.3.2 Lotus Domino support

Lotus Domino writes measurement data to SMF. The RMF Postprocessor supports the following SMF record types created by Lotus Domino:

- SMF record type 108-1 (containing server load data)
- SMF record type 108-3 (containing monitoring and tuning data) of record level 4 (as provided by Domino 5.0.3), which contains data needed for capacity planning and performance analysis

Based on these SMF records written by Lotus Domino, RMF Postprocessor provides the Lotus Domino Server report. This report provides feedback on server load and the number and type of messages that the server handled.

### 7.3.2.1 Requesting Lotus Domino server reports

Domino uses an ENF signal-based mechanism, which guarantees the synchronization of the Domino records with the RMF records. Shorter intervals may occur in case of server startup or server shutdown; the interval length, however, is recorded. The reports produced by the RMF Postprocessor (Domino option in the REPORTS control statement) are the Lotus Domino Server Summary report (see Figure 72 on page 123) and the Lotus Domino Server Details report (see Figure 73 on page 124).

```
                          L O T U S   D O M I N O   S E R V E R   S U M M A R Y
                                                                                PAGE    1
          OS/390                  SYSTEM ID LN21            DATE 01/30/2000          INTERVAL 30.00.000
                                                            TIME 18.40.00

NAME                            DURATION  ---- USERS -----  TASKS  TRANSACTION  ASYNC  I/O RATE      MAIL RATE       SMTP RATE
                                          CONNECTED ACTIVE           RATE     READS    WRITES  DELIVERED  SENT  READS    WRITES

BLUED1/BIGBLUE                  00.30.00       5000      4   5113     68.91     158.6     81.68      3.56   0.00   0.00      0.00
```

*Figure 72. Lotus Domino Server Summary report*

Like the HTTP Server Summary Report, the Lotus Domino Summary report contains one line per server within the reported interval. If there are several Domino Server records to be reported within one RMF reporting interval then the values of the detail reports within the RMF reporting interval are accumulated in the summary report. That is, the summary report then is a kind of "duration report" for the RMF reporting interval size.

```
                                L O T U S   D O M I N O   S E R V E R   D E T A I L S
                                                                                                  PAGE   2
          OS/390                   SYSTEM ID LN21            DATE 01/30/2000          INTERVAL 05.00.000
                                                             TIME 18.40.00

NAME: BLUED1/BIGBLUE               INTERVAL: 00.05.00

--- USER ACTIVITY ---   ------ TASKS ------   ------------- MESSAGES -------------   --- ACCESS RATES ---   --- DATABASE CACHE ---
MAX                 0   MAX            5115   MAILBOXES                       3   AS I/O READ    151.9   STATUS               OK
CONNECTED        5001   CURRENT        5113                   COUNT RATE AVG SIZE   AS I/O WRITE   78.02   MAX ENTRIES         384
ACTIVE              4   MAX UPDATES       0   MAIL DELIVERED   1652  3.30        4   POP3 READ       0.00   CURRENT ENTRIES     574
 WITHIN  1 MIN   1669   MAX REPLICS       0   MAIL SENT           0  0.00        0   IMAP READ       0.00   HIGH WATER MARK     576
 WITHIN  3 MIN   3172   COUNT REPLICS     0   SMTP RECEIVED       0  0.00        0   HTTP READ       0.00   INITIAL DB OPENS 19778
 WITHIN  5 MIN   4101                         SMTP SENT           0  0.00        0   HTTP WRITE      0.00   REJECTIONS          8671
 WITHIN 15 MIN   5000                                                                                      HITS                1775
 WITHIN 30 MIN   5000   - VIRTUAL THREADS -   - PHYSICAL THREADS -
                        MAX            5002   MAX                18                   --- AVAILABILITY ---   -- NSF BUFFER POOL ---
                        CURRENT        5000   CURRENT             6                   THRESHOLD           0   MAX                32768
                                              TOTAL             100                   INDEX             100   CURRENT                0
- TRANSACTION ACTIVITY -
MAX CONCURRENT  NO LIMIT
```

*Figure 73. Lotus Domino Server Details report*

The first part of the Domino Server Details report contains performance data provided by the SMF type 108-1 and 108-3 records. The second part contains transaction activity and port activity data provided by the SMF type 108-1 records.

## 7.3.3 FICON director support

The FICON Director Activity report is implemented as a Monitor I report and supplements the I/O-related Monitor I reports (Channel, Device, I/O Queuing). It provides configuration topology information and measurement values related to the usage of the active ports.

The data can be used to do capacity planning, analyze performance problems and bottlenecks, and identify contributors to device pending and disconnect times.

### 7.3.3.1 Gathering data for FICON director support

In order to enable RMF to gather data related to FICON Director, you must specify the data gathering options of Monitor I, as follows:

**FCD** Measurements for FICON Director data are done.

**NOFCD** No measurement is done for FICON Director data.

### 7.3.3.2 Requesting FICON Director Activity reports

In order to get FICON Director Activity reports (see Figure 74 on page 125), you have to specify the suboptions to the postprocessor in the option REPORT:

- FCD / NOFCD, or selective reporting with:
  - FCD(NMBR(xxxx:yyyy,zzzz),EXNMBR(aaaa:bbbb))

```
                                F I C O N   D I R E C T O R   A C T I V I T Y

          OS/390                    SYSTEM ID RMF2             START 03/02/2000-16.40.16  INTERVAL 000.19.43
          REL. 02.10.00             RPT VERSION 02.10.00       END   03/02/2000-17.00.00  CYCLE 1.000 SECONDS

IODF = 01   CR-DATE: 02/02/2000   CR-TIME: 16.57.20     ACT: POR

SWITCH DEVICE: 0555*  SWITCH ID: 00     TYPE: 009032   MODEL: 999   MAN: IBM   PLANT: 00   SERIAL: 000000010012

PORT    TYPE      ID     AVG FRAME     AVG FRAME SIZE     PORT BANDWIDTH (MB/SEC)     FRAME       LINK
                            PACING       READ   WRITE     -- READ --  -- WRITE --    ERRORS     ERRORS
 03     CHPID     FA             0       4040    3820           8.4          9.4         5          0
 09     CHPID     FB             0       1540    1780          12.8         64.2         0          3
 1B     CU        2222           0       3824    1460          77.1         19.0         0          0
        CU        3412
 1D     CHPID                    0       2840    2600          25.3         13.4         0          2


SWITCH DEVICE: 0AAA*  SWITCH ID: 01     TYPE: 009032   MODEL: 999   MAN: IBM   PLANT: 00   SERIAL: 000000020012

PORT    TYPE      ID     AVG FRAME     AVG FRAME SIZE     PORT BANDWIDTH (MB/SEC)     FRAME       LINK
                            PACING       READ   WRITE     -- READ --  -- WRITE --    ERRORS     ERRORS
 10     CU      3333             0       3820    3560          32.4         46.0         0          0
 19*    CHPID     FC             0       3214    3960          27.8         10.2         0          0
 1A*    CU      4444             0       2957    2910          10.4          4.4         1          0
        CU      5555
        CU      6666
 1E*    CU      AAAA             0       3420    4060          12.8         14.2         0          0
        CU      BBBB
```

*Figure 74. FICON Director Activity report*

## 7.4 Performance analysis with the Spreadsheet Reporter

The Spreadsheet Reporter (RMFPP) is the function in RMF that assists you in converting Postprocessor listings and Overview records into spreadsheets. In addition, it provides sample spreadsheets to help you in presenting and analyzing performance data at a glance.

The Spreadsheet Converter (RMF2SC) is part of the Spreadsheet Reporter for handling data from the RMF report displays or from report data sets (Monitor II, Monitor III, and Postprocessor) using techniques familiar to every spreadsheet user.

### 7.4.1 Spreadsheet enhancements

OS/390 Release 10 introduces a new spreadsheet macro and the following enhancements to Spreadsheet Reporter and Converter:

- I/O Subsystem Report
- Enhanced DASD Activity Report
- Enhanced Workload Activity Trend Report
- Enhanced Coupling Facility Trend Report
- Workload Overview Report - New Days/Week/Month selection

# Chapter 8. Language Environment and C/C++ enhancements

Enhancements to Language Environment in OS/390 Release 10 include:

1. *Downward compatibility* with previous releases of Language Environment. This means that you will be able to develop an application on a higher level of the operating system and execute the application on a lower level of the operating system, providing that the application has not exploited any new functions not available on the lower level version.
2. Language Environment support for Extra Performance Linkage (XPLINK) which will provide improved linkage performance for C and C++ programs, including DLLs. This is necessary because applications developed on other platforms can incur excessive overheads when processing function calls on OS/390.
3. Large file offset support for HFS files greater than 2 Gigabytes.

## 8.1 Downward compatibility

Language Environment (LE) has provided upward compatibility since pre-OS/390 days, which enabled applications link-edited with Language Environment to execute on future levels of Language Environment without the need to recompile or relink the application. For example, a C/C++ application compiled and link-edited on OS/390 R6 with Language Environment is guaranteed to run on OS/390 R10.

OS/390 Version 2 Release 10 will now guarantee Language Environment downward compatibility. Figure 75 displays the typical operating system upgrade, while at the same time ensuring application development continuity.

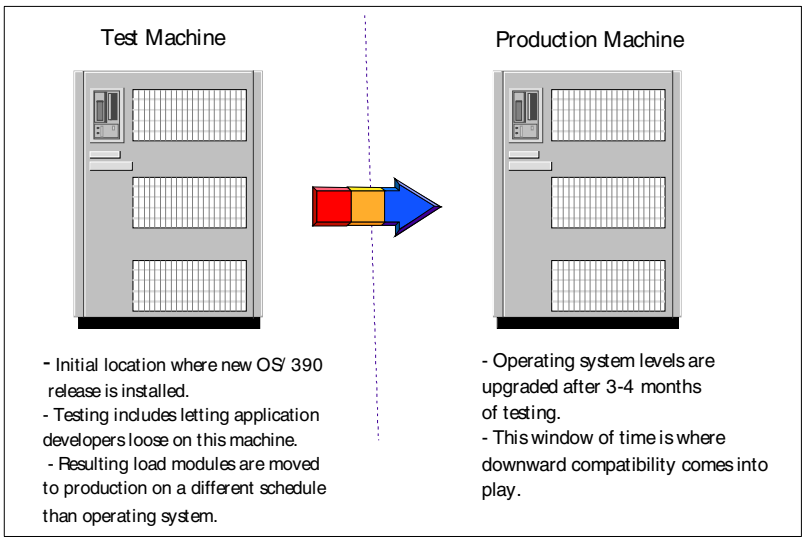



*Figure 75. Typical application dev. scenario highlighting LE downward compatibility benefit*

Toleration PTFs will be provided to ensure initialization and termination stub compatibility with OS/390 Release 10 LE and prior releases of LE. These PTFs must be applied to your production environment to ensure down-level compatibility when developing an application on a OS/390 R10 development

system and migrating these applications to a back level OS/390 production system.

## 8.2 XPLINK performance enhancement

Workloads written on other platforms and then ported to OS/390 often suffer because these applications can spend up to 25 percent of their time processing function calls. This is an excessive overhead. This is most noticeable in Object Orientated applications where there is a higher ratio of function calls to lines of application code.

The objective of XPLINK is to provide improved call linkage performance for applications using C/C++ subroutine linkages and executing on OS/390. An expected 50 percent reduction in linkage instructions is indicated for specific types of applications. This will also lead to a reduction of the memory requirements for functions processes, and will provide a common linkage for C/C++ applications and Dynamic Link Libraries (DLLs).

XPLINK application candidates would generally be highly modular with lots of small functions calls. In general you cannot bind XPLINK compiled and non-XPLINK compiled functions together in the same program, and it is advisable to minimize calls between XPLINK and non-XPLINK compiled functions. The DLL calling mechanism is the primary method for calling between XPLINK and non-XPLINK programs.

### 8.2.1 XPLINK applications in an LE environment

An XPLINK application is one in which at least one of the executables involved has been compiled XPLINK. XPLINK and non-XPLINK compiled source code cannot be link-edited together into the same executable, but XPLINK and non-XPLINK executables (for example, DLLs) *can* be mixed in the same application. Note that the performance advantage from XPLINK is increased as the percentage of XPLINK executables in an application increases.

The standard Language Environment stack is upward-growing. A main feature of XPLINK's more efficient program prolog code is a program stack which grows from higher to lower addresses. This provides implicit protection against exceeding available stack storage, rather than having to make an explicit test, and therefore reduces path length.

XPLINK utilizes a guard page which is a write-protected area of storage at the low address end of a downward growing stack segment. This allows a stack frame (smaller than the size of the guard page) to be allocated simply by storing into the low address of the stack frame. Stack segment overflow and extension is triggered by the exception resulting from a prolog storing into the guard page (implicit stack overflow detection).

With respect to XPLINK compatibility, “glue code” is inserted between XPLINK and non-XPLINK executables. This code converts the stack structure, registers, and parameter list into a format suitable for the called function, and then restores the environment upon return.

### 8.2.2 Compiling and Linking XPLINK applications in LE

The C/C++ XPLINK compiler option produces a program that uses the XPLINK calling conventions. Language Environment will initialize the enclave as an XPLINK environment if the initial program is compiled XPLINK or the XPLINK(ON) run-time option is specified. If the initial program is non-XPLINK but may call an XPLINK program later in its execution, then the XPLINK(ON) run-time option is required so that the XPLINK resources will be allocated and available when they are needed.

Applications that consist only of non-XPLINK functions (for example COBOL or PL/I) should not specify the XPLINK(ON) run-time option, because this option provides no benefit when not running an XPLINK application. In fact, for C/C++ non-XPLINK applications, enabling this run-time option could result in a performance degradation.

No AMODE 24 routines are allowed in an enclave that uses XPLINK. When an application is running in an XPLINK environment (that is, either the XPLINK(ON) run-time option was specified, or the initial program was compiled XPLINK), the ALL31 run-time option will be forced to ON.

When an application is running in an XPLINK environment (that is, either the XPLINK(ON) run-time option was specified, or the initial program was compiled XPLINK), the STACK run-time option will be forced to STACK(,,ANY). Only the third suboption of the STACK run-time option is changed by this action, to indicate that stack storage can be allocated anywhere in storage.

SCEEBIND is a new LE data set containing XPLINK compiled static routines (referred to as “stubs”), and SCEELIB, also a new data set containing LE DLL side decks. The Linkedit SYSLIB DD statement must include the SCEEBIND data set for XPLINK applications. The Linkedit SYSLIN DD statement should include the SCEELIB data set which contains the DLL side decks, CELHS003 (the “C” RTL DLL), CELHS001 (the LE AWIs), and CELHSCPP (the C++ DLL), concatenated with the program object data set.

Sample Linkedit JCL for XPLINK applications:

```
//LKEDX    EXEC PGM=IEWL,REGION=20M,
//          PARM='AMODE=31,RENT,DYNAM=DLL,CASE=MIXED,MAP,LIST=NOIMP'
//SYSPRINT DD   SYSOUT=*
//SYSLMOD  DD   DSNAME=USER.PDSELIB,UNIT=SYSALLDA,
//          DISP=(NEW,KEEP),SPACE=(TRK,(7,7,1)),DSNTYPE=LIBRARY
//SYSLIB   DD   DSNAME=CEE.SCEEBIND,DISP=SHR
//SYSLIN   DD   DSNAME=USER.OBJLIB(PROGRAM1),DISP=SHR
//          DD   DSNAME=CEE.SCEELIB(CELHS003),DISP=SHR
//          DD   DSNAME=CEE.SCEELIB(CELHS001),DISP=SHR
//SYSDEFSD DD   DUMMY
//SYSIN    DD   *
NAME  PROGRAM1(R)
/*
```

The LE run-time library data set SCEERUN, and a new XPLINK run-time data set, SCEERUN2, must both be available at execution time.

### 8.2.3 Debugging an XPLINK application

Stack storage is automatically created by Language Environment and is used for routine linkage and automatic storage. This section describes the way the XPLINK stack differs from the standard Language Environment stack. The prolog of a function usually allocates space (referred to as a “frame”, “Stack Frame”, or “DSA” - dynamic storage area) in the Language Environment-provided stack segment for its own purposes and to support calls to other routines.

The DSAs in the standard (upward-growing) Language Environment stack are allocated from lower to higher addresses. The XPLINK (downward-growing) stack is different, specifically in that the DSAs are allocated from higher to lower addresses, with the presence of the guard page to mark the bottom of the stack.

The XPLINK stack register (GPR 4) is “biased”, meaning it points to a location 2048 bytes *before* the stack frame for the currently active routine. It grows from numerically higher storage addresses to numerically lower ones; that is, the stack frame for a called function is normally at a lower address than the calling function. The stack frame is quadword-aligned.

XPLINK introduces a register scheme which is very different from standard OS linkage. The reasons for the new register conventions are to optimize the performance of saving and restoring registers in function prologs and epilogs.

### 8.2.4 Comparison of non-XPLINK and XPLINK Register conventions

Table 4 shows the differences in register usage for non-XPLINK applications, which would be the current standard for all applications running under LE, and the new convention for XPLINK applications.

*Table 4. Comparison of non-XPLINK and XPLINK Register conventions*

| Description | Non-XPLINK | XPLINK |
|---|---|---|
| Stack Pointer | Register 13 | Register 4 (Biased) |
| Return Address | Register 14 | Register 7 |
| Entry Point on entry | Register 15 | Register 6 (not guaranteed). A routine may be called by branch relative. |
| Environment | Register 0 (writable static) | Register 5 |
| CAA Address | Register 12 | Register 12 |
| Input Parameter List | Address in Register 1 | Located at offset 2112, X’840’, off Register 4 (fixed location in caller’s stack frame. First 3 words are passed in R1-R3. Floating point values in FPRO 2, 4, 6. |
| Return Code | Register 15 | Register 3, extended return value in R1,R2 |
| Start address of callee’s stack frame | Caller’s NAB value | Caller’s Register 4 minus DSA size |

| Description | Non-XPLINK | XPLINK |
|---|---|---|
| End address of callee's stack frame | Caller's NAB value plus DSA size | Caller's Register 4 |

## 8.3 Large file support

Language Environment provides large file support for 31-bit applications that will improve porting capabilities of C/C++ applications accessing HFS and NFS files larger than 2 GB. This is done by changing some C run-time library I/O functions to support the long data type for recording file offsets. Applications will be able to access HFS files up to the hardware limit of (2**43)-2 bytes.

Large file offset under LE is invoked in your C application when you #define_LARGE_FILES 1, and use the LANGLVL(EXTENDED) compiler option. The MAXFILESIZE parameter in SYS1.PARMLIB member BPXPRMxx should also be reviewed.

## 8.4 Additional Language Environment OS/390 Release 10 enhancements

Other LE enhancements introduced in OS/390 Release 10 include:

- *Transactional VSAM Support*, introduced in OS/390 Release 10, is added in C/C++ for transactional VSAM, which extends VSAM record level sharing (RLS) by adding commit and logging support for batch programs.
- The *CEEGPID* callable service now provides a fully programmable byte-oriented interface, mapping the Language Environment product ID, version, release, and modification levels in the fullword value returned as CEE_Version_ID.
- Language Environment adds improvements to the *c89*, C/C++, make, and Binder tools to work together seamlessly and to increase the speed of system builds, thereby reducing system downtime. *c89* compiles, assembles and binds OS/390 UNIX C applications. The *Binder* provided with OS/390 combines the object modules, load modules and program objects comprising an OS/390 application. It produces a single output program object or load module that can be loaded for execution.
- Language Environment provides a *timer parameter* to control the performance of signals based on *semop()* and *msgrcv()* functions.
- Language Environment provides code pages for:
  - IBM-1390 Japanese
  - IBM-1399 Japanese
  - IBM-1364 Korean
- Language Environment provides *sysconf* functions to allow for more data to be collected and to *improve sysconf performance*.
- Language Environment provides a C function to retrieve the CPUID of the current system.
- Changed ABTERMENC Run-Time Option Default - The default value for the ABTERMENC run-time option is changed from RETCODE to ABEND. in OS/390 V2R10. This support was also optionally provided in OS/390 V2R9.

**Note:** If the same default behavior is expected as in previous releases, the default for the ABTERMENC run-time option must be set to RETCODE.

## 8.5 C/C++ Enhancements

Along with the Extra Performance Linkage (XPLINK) enhancement discussed earlier in this chapter, the following C/C++ enhancements have been implemented in OS/390 Version 2 Release 10.

1. **GOFF** - The Generalized Object File Format (GOFF) is the strategic object module format for S/390. It extends the capabilities of object modules to contain more information than current object modules. It removes the limitations of the previous object module format and supports future enhancements. GOFF enables you to re-bind more easily and efficiently using incremental binds. It is required for XPLINK.
2. **IPA Level 2** - Under IPA Level 1, many optimizations such as constant propagation and pointer analysis are performed at the intraprocedural (subprogram) level. With IPA Level 2, these optimizations are performed across the entire program, which can result in significant improvement in the generated code.
3. **@STATIC Map** has been added into Compiler Listing. The @STATIC Map displays the contents of the @STATIC area, one per compilation unit. The @STATIC area is part of the object file and contains all the writable (i.e., not read-only) static variables, named or unnamed.

   For example, for the C statement "strcpy(string_var, "String Example")", the unnamed string "String Example" goes into the @STATIC area. Whenever several compilation units are linked together, all the individual @STATIC areas are grouped together and go into the Writable Static Area (WSA), which is one per executable program module.
4. The **COMPACT** Compiler option has been introduced. During Interprocedural Analysis (IPA) optimizations and the optimizations performed during code generation, choices must be made between those optimizations which tend to result in faster but larger code and those which tend to result in smaller but slower code.

   The **COMPACT** | **NOCOMPACT** option controls these choices. When the COMPACT option is used, the compiler favors those optimizations which limit the growth of the code. This feature gives you the flexibility to choose between faster but larger code, or slower and smaller code.

Support has been *removed* for the following:

- The IBM System Object Model (SOM) is no longer supported in the C++ compiler and the IBM Open Class Library. The SOM-enabled class library DLLs have been stabilized at the V2R9 level and continue to be shipped as a run-time environment only.
- The Model Tool is no longer available.

# Chapter 9. Communications Server for OS/390 V2R10

Communications Server for OS/390 Version 2 Release 10 offers enhancements to Telnet and FTP, improved sysplex exploitation, security and performance changes and improvements related to network management, usability and serviceability. An overview of the changes introduced in this release follows.

## 9.1 Telnet enhancements

The Telnet enhancements in OS/390 Release 10 include the following.

1. Autologon support, via the LOGAPPL parameter, allows a client to request a session with a Physical Logical Unit (PLU) and then wait for the PLU if its status is currently inactive. When the PLU becomes active, it will initiate a session with the waiting client.

   Previously, during Telnet simulation of SNA Secondary LUs, the session request was rejected and the SLU inactivated when an SLU requested a session with an unavailable LU. This enhancement provides the same basic support as the SNA 3270 AUTOTI and AUTORTRY VTAM star options.
2. Keepopen support allows Telnet to keep the SLU LU Access Control Block (ACB) open instead of closing it after a failed session attempt. With KEEPOPEN coded on the LUMAP statement, the ACB for a selected LU will be opened and will remain open during LOGON failure and session termination, either normal or abnormal.
3. Mapping support will allow TN3270E clients to specify an LU pool when choosing an SNA LU name. Prior to this enhancement, an SNA LU name was either assigned by the server or specified by the client.
4. Debug has been enhanced to provide a SUMMARY or DETAILed message when a major state change or error occurs.
5. Network Qualified Name (NQN) support, added in Release 8 and made available to Release 5 and above via an APAR, was limited to the USSCMD macro. NQN application names can now be entered on the Solicitor panel or Linemode entry.
6. Specific LU Takeover session support has been enhanced with the new statement TKOSPECLURECON, which, although mutually exclusive of the TKOSPECLU statement, does not drop the session.
7. Default redrive control if immediate session establishment is desired on the first attempt only can be set by coding the FIRSTONLY keyword on either the DEFAULTAPPL or LINEMODEAPPL statement.

## 9.2 Telnet security enhancements

Telnet security enhancements include the following.

1. Telnet Secure Sockets Layer (SSL) will now use OS/390's Cryptographic Services System Secure Sockets layer (System SSL). This will ensure TN3270 will pick up the latest fixes as soon as they are applied, which were previously supplied in a SSL toolkit packaged with TCP/IP.

The OS/390 SSL supports the use of RACF as a repository for the server's key ring, referred to as Common Key Ring Support. This provides another alternative to using the HFS or MVS data set to store the key ring.

2. Certificate Revocation List (CRL**)** processing for Vault Registry-issued client certificates is now supported. System SSL supports an optional query to an LDAP directory to determine if the client-supplied certificate is in the CRL that is maintained on the LDAP directory.
3. A TELNETGLOBALS block is added, and parameters added in this block apply to all TELNETPARMS blocks. This eliminates the need to specify a KEYRING statement for each SECUREPORT.
4. The BEGINVTAM information block in the Profile data set has been enhanced to include the PARMSGROUP and PARMSMAP statements.

   The PARMSGROUP statement defines CONNTYPE, CLIENTAUTH, ENCRYPT and DEBUG options for a subset of the ports connection. The PARMSMAP statement maps a connection (by IP address, host name or linkname) to a PARMSGROUP. Statements related to connections mapped to a PARMSGROUP override those defined in the TELNETPARMS block.
5. The TELNETPARMS SECUREPORT keyword has been enhanced with the CONNTYPE parameter. This enables the following options be specified:
   - CONNTYPE SECURE indicates that the traditional SSL handshake will be used to start the SSL connection.
   - CONNTYPE NEGTSECURE indicates that a TN3270 negotiation with the client first determines if the client is willing to enter into a secure connection. If the client agrees, SSL protocols will be used for all subsequent communication; if not, then the connection will be closed.
   - CONNTYPE ANY indicates that the client can connect as either secure or basic. TELNET will first try a standard SSL handshake. If the handshake times out, a negotiated SSL is attempted. If the client is willing to enter into a secure connection, SSL protocols will be used; otherwise, a basic connection is used.
   - CONNTYPE NONE indicates that a client is not allowed to connect in, unless the client's IP address, host name or linkname is mapped to a PARMSGROUP which must specify CONNTYPE.
   - CONNTYPE BASIC indicates that a basic, non-SSL, connection will be used.

## 9.3 File Transfer Protocol (FTP) enhancements

File Transfer Protocol (FTP) enhancements implemented in OS/390 Version 2 Release 10 to improve administration, security, functionality and performance are as follows.

### 9.3.1 FTP security

OS/390 FTP was enhanced to be consistent with other UNIX platforms when controlling anonymous client access. The FTP.DATA server configuration file has been enhanced to allow the administrator to limit the resources anonymous users can access.

The administrator can:

- Allow or prevent HFS or MVS file system access
- Restrict anonymous users to the HFS subdirectory
- Allow or preclude filetypes JES, SEQ or SQL
- Control permission bits on HFS resources created by anonymous users
- Require anonymous users to provide e-mail address as password at login

The new statements in the server FTP.DATA file to control anonymous user access are:

- ANONYMOUSLEVEL, where:
  - **1** (the default) preserves anonymous user controls at pre-V2R10 levels
  - **2** restricts anonymous users to the home directory of the anonymous user when the STARTDIRECTORY statement is set to HFS
  - **3** enables the full capabilities of OS/390 V2R10 where:
    - The STARTDIRECTORY HFS statement will cause a chroot() to the anonymous user's home directory.
    - The FTP client will not be allowed to switch back and forth between the anonymous user and a specific user, with the USER subcommand.
    - The chmod() command will not be permitted.
    - The anonymous user must enter the user's e-mail address as the password when logging in (unless the ANONYMOUS statement is coded as ANONYMOUS USERID).
  - Only if ANONYMOUSLEVEL=3 is specified are the following statements in effect:
    - ANONYMOUSFILEACCESS determines whether anonymous users have access to HFS files, MVS data sets, or both. Valid options are HFS, MVS or BOTH. The default is HFS.
    - ANONYMOUSHFSFILEMODE determines what permission bits will be assigned to files created in the HFS by anonymous users. Valid option: permission bit string (specified as 3 octal bytes).The default is 000.
    - ANONYMOUSHFSDIRMODE determines what permission bits will be assigned to directories created in the HFS by anonymous users. Valid option: permission bit string (specified as 3 octal bytes). The default is 333.
    - ANONYMOUSFILETYPESEQ determines whether anonymous users have access to FILETYPE=SEQ (HFS files and MVS data sets). Options: TRUE or FALSE. The default is TRUE.
    - ANONYMOUSFILETYPEJES determines whether anonymous users have access to FILETYPE=JES (JES jobs on input and output queue). Options: TRUE or FALSE. The default is FALSE.

- ANONYMOUSFILETYPESQL determines whether anonymous users have access to FILETYPE=SQL (ability to submit SQL queries). Options: TRUE or FALSE. The default is FALSE.
- EMAILADDRCHECK determines the level of checking applied to e-mail addresses entered as passwords when the anonymous user logs in.

  **NO** indicates that any value will be accepted. **WARNING** performs minimal checking, but allows login even if syntax is incorrect. **FAIL** will check syntax and fail login if incorrect.

  Figure 76 highlights some of the FTP enhancements.

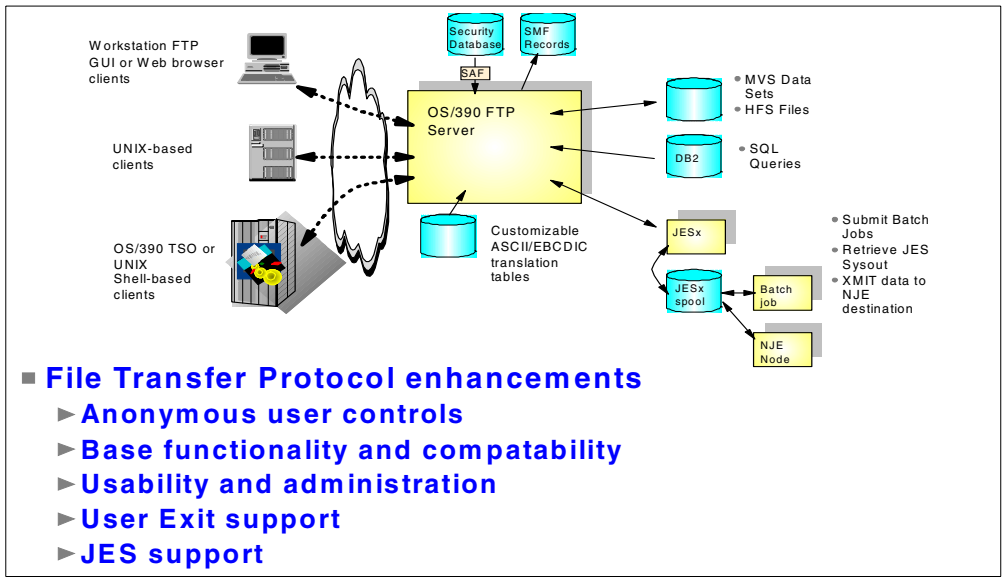



*Figure 76. FTP enhancements*

### 9.3.2 FTP functionality enhancements

Base functionality has been enhanced in the FTP server and client to include the following.

1. FTP Transfer of MVS load modules between OS/390 hosts without losing the load module characteristics that enable them to be executed, provided that both the FTP server and FTP client are running on a TCP/IP stack on OS/390 V2R10 or later. The transfer must be from a partitioned data set (PDS) to PDS, or partitioned data set extended (PDSE) to PDSE.
2. FTP Transfer by DDNAME on GET and PUT subcommands.
3. FTP Transfer of MVS data sets via URL without requiring the use of two slashes to identify the MVS data set name.
4. File SIZE and last modified (MDTM) extensions have been implemented.
5. New file allocation keywords have been implemented to enable the specification of multiple volumes for new file allocations. The UCOUNT, unit count, and VCOUNT, volume count, keywords have been added and the existing VOLUME keyword now supports a list of volume serial numbers.

### 9.3.3 FTP user exit support

The FTP.DATA server user exits have been enhanced with the introduction of a new exit. FTPOSTPR, the post transfer processing exit, gets control whenever a file transfer attempt has completed, successfully or otherwise. Use of this exit is optional.

Additionally, the FTCHKCMD user exit has been changed to enable the passing of a command string as input, and accept a modified command string as output.

### 9.3.4 FTP JES support

FTP JES interface support has been enhanced to enable the use of SAF definitions to determine which output can be accessed, and allows access to started tasks, APPN, and TSO, as well as batch jobs. It will also return more detailed information regarding job status, and will allow spool files from a single job to be retrieved individually. These functions are available when server FTP.DATA configuration statement JESINTERFACELEVEL=2 is specified.

## 9.4 Performance enhancements

The TCP/IP performance improvements implemented in OS/390 Version 2 Release 10 include new sysplex workload distribution functions, improved recoverability for dynamic Virtual IP Addresses (VIPAs), and enhanced service policy requirements.

### 9.4.1 Service policy enhancements

FTP server policy enhancements have been implemented to enable Quality of Service (QoS) enforcement for Differentiated Services type policies, improvement in policy agent support for the Sysplex Distributor function, and the introduction of a Traffic Regulation policy that can assist customers in defining what traffic levels are expected and help prevent denial of service attacks due to TCP flooding.

The policy agent performs two distinct functions to assist the Sysplex Distributor. It firstly provides a list of target stacks (outbound interfaces to target XCF addresses) to consider for inbound traffic, and secondly, provides QoS weight fraction from target stacks to be considered along with the WLM weight for those stacks. Figure 77 on page 138 displays the policy agent and Sysplex Distributor interaction.

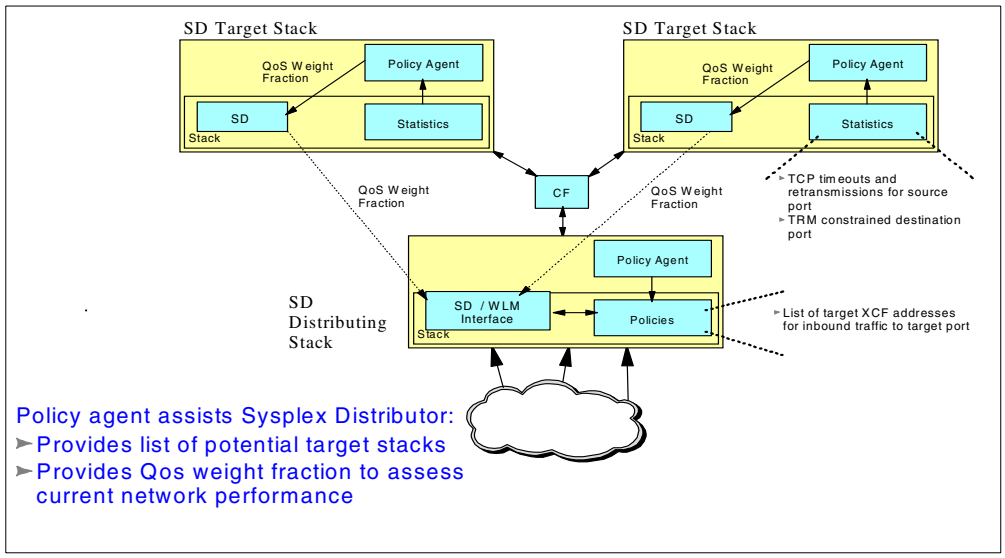



*Figure 77. Enhanced policy support - Sysplex Distributor assistance*

Policy rules prior to Release 10 were limited to source address and port, or destination address and port, or days of the week, time of day, etc. OS/390 Version 2 Release 10 adds the ability to filter by application name and by application data.

For example, the IBM HTTPD Server is enhanced to pass the URL to the Fast Response Cache Accelerator exits, enabling policy definitions to be filtered by URL, which could enable a higher priority to customers referencing specific web pages.

### 9.4.2 Dynamic Virtual IP Addressing (VIPA) takeover enhancement

The purpose of VIPA is to free other hosts from dependence on particular physical network attachments to an OS/390 TCP/IP stack. Prior to VIPA, other hosts got bound to one of the home IP addresses and, therefore, to a particular physical network attachment (for example, a device or adapter).

VIPA provides an IP address that is associated with an OS/390 TCP/IP stack without associating with a specific physical network attachment. Other hosts that connect to the OS/390 TCP/IP stack can send data via whatever paths are selected by the routing protocols.

VIPA Takeover requires that Dynamic VIPAs (as opposed to traditional “static” VIPAs) be defined as having a normal “home” stack, and optionally one or more backup stacks. The stacks all share information on Dynamic VIPAs using MVS XCF Messaging (the same mechanism as XCF Dynamics and Sysplex Sockets), so that all stacks know--for each Dynamic VIPA--which stack has it active, and which stacks--and in what order--will participate in backup if the active stack fails.

When a failure of a stack hosting an active Dynamic VIPA is detected, the first stack in the backup list automatically defines DEVICE. LINK, and HOME statements for the same Dynamic VIPA, and notifies its attached routing daemon of the activation, to be passed on to the routing network via dynamic routing protocols.

When the original “normal home” stack is reactivated, the Dynamic VIPA remains on the current backup stack as long as there are active connections to that Dynamic VIPA on that stack. Thus, takeback is nondisruptive to existing connections at the time of reactivation of the original stack--but takeback may be delayed, potentially indefinitely.

Changes have been implemented in OS/390 Version 2 Release 10 to ensure takeback is immediate and nondisruptive; new connections are routed to the restored stack; and existing connections with the prior owning stack will be preserved. Immediate takeback will be the default, but stack can be configured for deferred takeback.

### 9.4.3 Sysplex workload distribution

A major enhancement to VIPA takeover and sysplex interaction is the Sysplex Distributor, which is implemented in OS/390 Version 2 Release 10. The Sysplex Distributor will perform Interactive Network Dispatcher (IND)-like functions in conjunction with WLM to enable optimization of IP stack workloads.

IND requires direct connectivity to the application host, which in many instances is not possible as some installations do not connect all S/390 nodes directly to the routing network, and additional hardware is also required to process the IND requests. Current IND implementation requires a single hop to the target IP address, which means that VIPAs do not work particularly well, given that they appear to be two hops away from the IND router. Sysplex Distributor exploits Dynamic VIPAs.

Use of Sysplex Distributor means the stacks themselves will be aware of Sysplex Distributor function; the application hosts will identify to the routing stack when an application is actually listening to the Dynamic VIPA for work, so that work will not be distributed to a stack that does not at the time host a server instance that is ready to accept work.

CPU resources are consumed on the routing stack to keep information up-to-date in the CF, but not on the backup stack processing real-time updates, until the primary stack fails. Figure 78 on page 140 shows IP address representations within the sysplex.

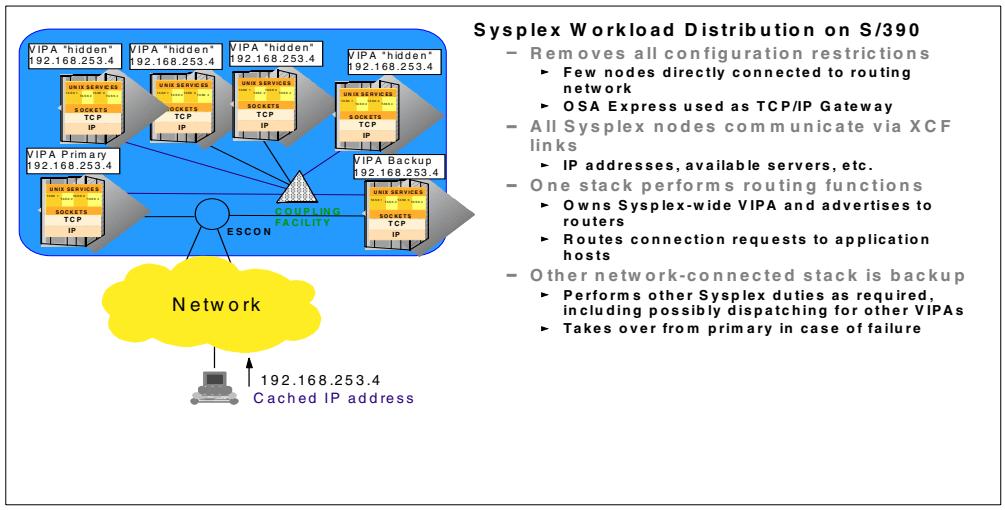



*Figure 78. Sysplex IP workload distribution*

WLM is periodically queried to obtain comparative “weights” for each target stack that are indicative of each stack’s workload. When a performance monitor policy is configured to the Service Policy Agent, the “weights” may be modified to account for network performance issues (for example, TCP retransmissions). The Service Agent Policy can also be used to restrict the target list for specific connections.

In addition, the functionality of Sysplex Distributor can be reduced to the simple case of a single application host which is also the stack routing the Dynamic VIPA. While there is no new function for the normal case and the initial failure case, when a failing Dynamic VIPA “normal home” stack is restarted, the Dynamic VIPA can be returned to its normal location, and all future requests serviced by applications on that stack, without disrupting existing connections to applications on the backup stack. In other words, the Dynamic VIPA returns to its “natural” home, but all connections to applications on the backup stack are maintained transparently.

Where Dynamic VIPAs are assigned to a specific application instance via BIND or IOCTL, such applications can now be moved to a new node by starting the identical application JCL package on a new node. Existing connections to the previous node are not disrupted, but that stack no longer receives new connections, because they go to the application instance at its new home. When the last connection to the “old” application instance is closed normally, the “old” application instance can be stopped, without disruption of any client operations.

### 9.4.4 Network traffic access controls

Network traffic regulation is managed by the Traffic Regulation Management (TRM) daemon via an API in conjunction with a service policy agent. It provides control over the number of inbound connections. This is accomplished by providing user control of network accesses via the management of security zones using SAF to identify users and groups who are allowed access, and by controlling who has TCP and User Datagram Protocol (UDP) port access.

Network Access control enables system administrators to assign permission for users to access certain networks or security zones; therefore, they can disallow

the ability of certain users to send data to certain networks. Port Access control enables administrators to control an application's ability to bind to specific TCP and UDP ports or port ranges, and Stack Access Control prevents users form getting access to a TCP/IP stack through a TCP or UDP socket.

What was developed was a method of regulating the number of connections allowed to a TCP port and a percentage of connections to a client. This was achieved by defining the total number of connections of a service (port), defining a controlling percentage, tracking the number of available connections of the port and tracking the number of connections from a host.

This results in the scenario where a request for a new connection will be established if the controlling connection/percentage ratio has not been reached. The exception is if there is a QoS policy applied to the host, and the QoS policy allows more than the connection/percentage ratio; then the request is permitted.

### 9.4.5 Traffic Regulation Management (TRM)

Traffic Regulation Management (TRM) configuration is performed via a Service Policy Agent (Pagent); this specifies the total number of connections (TotalConnections) allowed for a port and a controlling percentage. The policy rule has a scope of “TR” (Traffic Regulation), and these policies can only be defined in the Pagent configuration flat files, not in LDAP in this release. Figure 79 displays the TRM system overview.

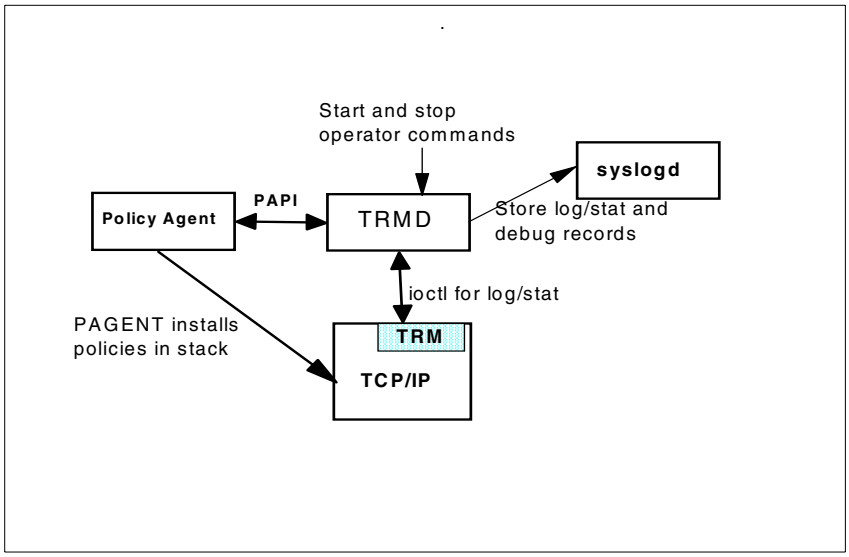



*Figure 79. Traffic Regulation Management system*

The Traffic Regulation Management Daemon (TRMD) must be started when TR policy processing is required. TCP/IP determines whether a TR policy applies to a particular TCP port and will determine whether a connection request from a host will be permitted.

## 9.5 Security enhancements

Network access security has been enhanced to enable SAF to control network access at the IP address level, port level and stack level.

### 9.5.1 User control of network access

User control of network access provides the ability to associate a network destination with a SAF resource. During TCP connects and UDP/TCP/RAW sends, we determine if a particular user has authority to send to that destination. This is done by creating a tree that stores the mappings of SAF resources and the destination network resources. This tree is referenced during TCP connects and all protocol data sends.

To use this function, the new profile statement NETACCESS must be added, RACF (or another SAF server) must be installed on the system, and the appropriate SAF resource names must be defined and access designated.

Use the NETACCESS statement to configure network access. Specifically, it allows for the one-to-one mapping between network and SAF resource name. The network specifications will be used to build an internal data structure representing all defined SAF resource name mappings.

The structure is checked to determine the user's permission to access the network resource. The most specific network entry in the tree (with its SAF resource name) will be used in the SAF authorization check. The format of the NETACCESS statement in the TCP/IP configuration profile is:

```
NETACCESS
ipaddr/num_mask_bits ipaddr address_mask saf_resname
DEFAULT
ENDNETACCESS
```

ipaddr/num_mask_bits specifies the network for which security product access control is required for outbound user requests. The num_mask_bits field is used to create an address mask that is bit-contiguous from left to right. This address mask is logically ANDed with the ipaddr value to create the network address for which access control is required.

ipaddr address_mask specifies the network for which security product access control of outbound user requests is required. The address_mask value is a bit mask (expressed in dotted-decimal form) that is bit-contiguous from left to right. The address_mask value is logically ANDed with the ipaddr value to create the network address for which access control is required.

DEFAULT specifies that security product access control of outbound user requests is required for any networks not specifically defined by other NETACCESS statement entries.

saf_resname specifies the final qualifier of a security product profile name. The maximum length is 8 characters. This qualifier is appended to a profile name with the following format:

```
EZB.NETACCESS.sysname.tcpname.saf_resname
```

As an example, note the following TCP/IP configuration sample:

```
NETACCESS
; Network                    SAF
   9.0.0.0/8                IBM
   9.155.0.0/16             CS390
   9.160.0.0 255.255.0.0    CS390
   9.155.128.0/17           DESIGN
   DEFAULT 0                WORLD
   ENDNETACCESS
```

This would require the following SAF resources to be defined within the SERVAUTH class.

```
EZB.NETACCESS.sysname.stackname.IBM
EZB.NETACCESS.sysname.stackname.CS390
EZB.NETACCESS.sysname.stackname.DESIGN
EZB.NETACCESS.sysname.stackname.WORLD
```

### 9.5.2 User control of port access

OS/390 Version 2 Release 10 enables system administrators to control user access to TCP and UDP ports via user ID. This is implemented by defining new RACF resources for port access which will verify a user ID's permission to access port at bind() time for OE socket applications. The verification will be performed at TcpOpen, UdpOpen or RawOpen for Pascal applications.

The SAF resname keyword has been added to the TCP/IP configuration profile PORT and PORTRANGE statement. This indicates that the port is reserved for users that are permitted to the RACF resource named following the keyword. This is optional and valid for TCP or UDP protocols.

Every PORT or PORTRANGE statement causes a port reservation entry to be created. At bind time, TCP/IP attempts to find a matching port reservation entry by using the jobname, which now includes the wildcard "*****" jobname.

Using the wildcard along with the SAF resource name enables RACF to manage all access for a defined port. When a port reservation entry is located, the SAF resource name associated with that entry (if present) is used to construct the resource name that is passed to the security product for access verification.

The format of the SERVAUTH class SAF resource name associated with the PORT or PORTRANGE statement is:

```
EZB.PORTACCESS.sysname/tcpname.SAFName
```

where SAFName is specified on the PORT Reservation statement.

An example of defining a port access resource is as follows:

**Defining SAF PORT access**

```
PORT 20 TCP * SAF FTPDATA
RDEFINE SERVAUTH EZB.PORTACCESS.system.TCPproc.FTPDATA
SETROPTS RACLIST(SERVAUTH) REFRESH
PERMIT EZB.PORTACCESS.system.TCPIPproc.FTPDATA CL(servauth)
ID(user1) AC(READ)
SETROPTS RACLIST(SERVAUTH) REFRESH
```

### 9.5.3 User control of stack access

Stack access can also be controlled via SAF using the following SERVAUTH class resource:

```
EZB.STACKACCESS.sysname.tcpproc
```

No new TCP/IP definition is required.

# Chapter 10. XES Auto Alter support

XES Auto Alter adds support for automatic tuning of CF structure size and ratio of structure objects in response to changing structure object usage. Entry to element ratios, structure size and event monitor controls are dynamically tuned by XCF to achieve more optimal use of CF storage resources when permitted by both the installation and the CF structure exploiter.

As of OS/390 Version 2 Release 9, Structure Full Monitoring adds support for real-time monitoring of CF structure utilization for entries (including list structure entries, cache directory entries, and lock structure record data entries), data elements (for both list and cache structures), and event monitoring control (EMCs, for list entries) into the XCF component of OS/390. This support serves to provide a common framework for performing automatic tuning functions for the monitored CF structures.

Structure Full Monitoring externalizes information about structures that are at or above a structure full threshold percentage in terms of the utilization of any or all of these monitored structure objects, in a consistent manner for all structures, via highlighted system message IXC585E. In OS/390 Version 2 Release 10, when the threshold structure full condition is detected, XCF may start an Alter request for the structure (either reapportioning the structure objects within the structure or expanding the structure size, or both) to provide additional CF resources of the type that are in short supply.

The system allocates CF storage for a structure at the time of the first connect to that structure. The CF apportions storage in accordance with ratios specified on the IXLCONN macro.

Connectors historically have had a difficult time selecting appropriate values for structure ratios (entry-to-element ratio for list structures, directory-to-data ratio for cache structures). For the subset that makes use of these resources in a known, fixed ratio, determining what ratio to specify on the IXLCONN macro is no problem.

For the rest, it is often very difficult to predict at structure allocation time what the ratios should be. Often the exploiting subsystem has simply passed this responsibility back to the installation, by providing an external on which the installation can, in effect, specify the directory-to-data ratio which is to be used; but the installation also has very little idea how to set the ratio correctly before the fact, in anticipation of actual usage.

The event monitor controls storage percent (for list structures) is even more difficult, because here, making the leap from specifying a particular storage percentage to estimating how many EMCs are created in the structure can be quite complicated.

Some structure storage goes to fixed controls, and thus does not figure into the EMC storage ratio calculation at all. The remaining storage is apportioned into either EMCs or entries/elements based on the EMC storage ratio, but one must understand how big an EMC is and how big entries and elements are in order to predict how many EMCs are created. And even if one could figure all of this out, it is still uncertain whether one would have a clear understanding of whether the

number of EMCs that are created are in fact sufficient when the structure is actually put into use.

## 10.1 Requesting structure size

The size of a coupling facility structure includes the total amount of storage required by both the application and by the coupling facility itself for structure control information, and is specified in 1K blocks. Determining the correct size is based on both the application's use of the structure and characteristics of the workload environment in which the structure is to be used.

The application provides guidance so that you can estimate an appropriate size for a structure. The S/390 Coupling Facility Structure Sizer is also available to help you estimate the amount of storage required for a structure; see the following site:

```
http://www.ibm.com/s390/pso/
```

The size of a structure can be specified by the application in its code or by the installation in its CFRM policy.

## 10.2 Structure full monitoring

Structure full monitoring was introduced in OS/390 R9 to add support for the monitoring of objects within a coupling facility structure. Its objective is to determine the level of usage for objects that are monitored within a coupling facility.

Structure full monitoring, running on a given system, periodically retrieves structure statistics for each of the active structure instances from each of the coupling facilities which it is currently monitoring. The retrieved information indicates the in-use and total object counts for the various monitored objects.

These counts are used to calculate a percent full value. For each structure in the CFRM policy, the percent full threshold can be specified by the installation to be any percent value between 0 and 100. Specifying a threshold value of 0 means that no structure full monitoring takes place. If no threshold value is specified, the default value of 80% is used as the full threshold percent value.

Only “active” structure instances are monitored by structure full monitoring. Active structures include:

- Simplex allocated structures
- Duplexed allocated structures

Active structure instances are eligible to be monitored by structure full monitoring regardless of whether or not they have any active, or failed-persistent, connectors at the time. Persistent structures with no connectors, or structures which have only failed-persistent connectors, are therefore monitored.

Structure full monitoring does not monitor “inactive” structures. Inactive structures include structures which are:

- In transition during connect or disconnect

- Pending deallocation as a result of a FORCE structure request that cannot be processed immediately
- Pending deallocation where the structure is being kept because of an associated structure dump which has not been written to a dump data set yet.

In most cases, structure full monitoring accurately represents the utilization of objects within the structure being monitored. There are a few special considerations. In the case of a cache structure, you can get a full condition that structure full monitoring cannot to detect, as in the following situations:

- Structure full monitoring does not monitor unchanged objects in a structure, so a cache structure is monitored only in terms of the changed or locked-for-castout objects that the structure contains.

  However, a cache structure may be full because all objects are in-use for unchanged objects, in which case structure full monitoring does not detect the full condition. This type of “full” condition normally causes reclaiming of resources, not a structure full condition.
- A directory only cache structure is not monitored because it has no changed or locked-for-castout objects for structure full monitoring to count.
- Structure full monitoring cannot monitor cache structures allocated on a CFLEVEL=0 coupling facility. Some of the required object counts are not available from this level coupling facility, so it appears to structure full monitoring that these counts are zero.
- A cache structure with multiple storage classes may experience a “full” condition because its available (unchanged and unlocked) objects are in a storage class where they are not available for satisfying a particular write request. Structure full monitoring cannot detect this kind of full.

Monitoring of coupling facilities by structure full monitoring is shared dynamically among systems in the sysplex. The ownership of monitoring by systems can change as systems come and go, or as losses of connectivity to the monitored coupling facilities occur. As a result, you cannot use automation that relies on any affinity between a system and the coupling facility being monitored. Structure full monitoring frequency is dynamically tuned.

The messages IXC585E, IXC586I and IXC587I are issued during the structure full monitoring processing.

## 10.3 Automatically altering structures

Starting with OS/390 Release 10, you can specify whether you want the system to automatically alter a structure when it reaches an installation-defined or defaulted-to percent full threshold as determined by structure full monitoring. The alter process may increase the size of the structure, reapportion the objects within the structure, or both. The ability to have a structure automatically be altered assumes the following:

- The application defining the structure has specified that the structure is capable of being altered.
- The installation has specified that the structure is allowed to be altered (ALLOWAUTOALT(YES) in the active CFRM policy).

- There is available coupling facility storage to accommodate the changed size or apportionment of the structure.

After issuing message IXC585E to externalize the structure full threshold condition, the system then stops any alters in progress against the structure and issues a new alter request against the affected structure. The objective of the alter request is either or both of the following:

- To expand the size of the structure to relieve the resource shortage when:
  - All monitored object types of the structure are over the structure full threshold value
  - At least one (but less than all) monitored object types are over the structure full threshold value, but all objects cannot be brought below the threshold without also increasing the size of the structure
- To reapportion the monitored structure objects to relieve the resource shortage for the object that is in short supply when at least one (but less than all) monitored objects are over the structure full threshold value, and all objects can be brought below the threshold without also increasing the size of the structure.

### 10.3.1 Understanding when a structure is automatically altered

As stated previously, a structure is “eligible” to be automatically altered when all of the following are true:

- Connectors to the structure allow alter.
- CFRM policy specifies ALLOWAUTOALT(YES).
- CFRM policy specifies a non-zero FULLTHRESHOLD value.
- The structure is not currently in the process of being rebuilt.

A system-initiated alter for an eligible structure begins when structure full monitoring determines that a structure contains monitored objects that are at or above the structure full threshold specified by FULLTHRESHOLD. After issuing message IXC585E to externalize the structure full threshold condition, the system then stops any alters in progress against the structure and issues a new alter request against the affected structure. The objective of the alter request is either or both of the following.

- To expand the size of the structure to relieve the resource shortage when:
  - All monitored object types of the structure are over the structure full threshold value.
  - At least one (but less than all) monitored object types are over the structure full threshold value, but all objects cannot be brought below the threshold without also increasing the size of the structure.
- To reapportion the monitored structure objects to relieve the resource shortage for the object that is in short supply when at least one (but less than all) monitored objects are over the structure full threshold value and all objects can be brought below the threshold without also increasing the size of the structure.

**Note:** A structure that is in the process of being rebuilt, either as a result of user-managed or system-managed rebuild, is not eligible to be altered automatically. The exception to this rule is for structures in a user-managed

duplexing rebuild environment. In that case, when the structures are in the duplex established phase, the structures are eligible to be altered automatically.

### 10.3.2 Considerations for duplexed structures

When the system determines that either one of a pair of duplexed structures needs to be altered, the system signals the system currently responsible for monitoring the coupling facility which contains the primary structure. The system then issues an alter request against the structure pair. The XES Auto Alter processing for user-managed duplexing alters the primary structure first, followed by the secondary structure.

### 10.3.3 Relieving coupling facility storage constraints

There are instances when the system automatically contracts a structure when the coupling facility is 80% full and coupling facility resources are not being used productively by the structure. When structure full monitoring determines that all monitored objects for a particular structure are below the structure full threshold value, the system starts an alter request against the structure to contract the structure.

The objective of the alter request is to contract a small percentage of resource at any one time, making sure that the reduction in structure size does not cause a percent full threshold in terms of any monitored objects to be exceeded. The alter processing does not allow the structure to be contracted below the structure the MINSIZE value specified in the active CFRM policy. The structure must have been defined to be eligible for XES Auto Alter processing.

## 10.4 CFRM policy changes

For each structure that is to be eligible for system-initiated alter processing, the following parameters in the CFRM policy can be specified:

1. ALLOWAUTOALT

Specify ALLOWAUTOALT(YES) to request that the structure be eligible for system-initiated alter processing. If ALLOWAUTOALT(NO) is specified and a value other than zero is specified or defaulted to for the FULLTHRESHOLD parameter, the system continues to monitor the structure but does not initiate an alter. The default is NO.

2. FULLTRESHOLD

The value you assign to the FULLTRESHOLD parameter is used by both Structure Full Monitoring and system-initiated alter processing to determine when a structure is “full”. The value is specified as a percentage and represents a percentage full threshold in terms of coupling facility structure objects.

If a value of zero is assigned to FULLTHRESHOLD, neither structure full monitoring nor automatic altering of the structure will occur. If a value other than zero is specified or defaulted to, the value is used as the percent full threshold for structure full monitoring, as well as being used by system-initiated alter processing to determine how to alter the structure. The default is 80%.

3. MINSIZE

The value you assign to MINSIZE specifies the smallest size to which the structure can ever be altered, either as a result of system-initiated alter processing or any other alter process. This also serves as a minimum bound for the structure size on all structure allocation requests that occur on OS/390 Release 10 and above (including connect and rebuild connect) with the following exception:

- During system-managed rebuild, new structure allocation may result in the structure being allocated with a size smaller than the value specified or defaulted to for MINSIZE. The MINSIZE must be less than, or equal to, the SIZE value if the INITISIZE is not specified. The default is 50% of INITSIZE, or 50% of SIZE, if INITSIZE is not specified

:

```
STRUCTURE NAME(STRUCT1) SIZE(7000)
          INITSIZE(4000)
          MINSIZE(2000)
          FULLTHRESHOLD(75)
          ALLOWAUTOALT(YES)
          PREFLIST(CF01,CF02,CF03)
```

*Figure 80. Example of CFRM structure definition using IXMIAPU utility*

## 10.5 IXLCONN macro changes

- ALLOWALTER(YES) should be specified
- Change the default value of the following parameters from 0 to 25 to avoid structures being contracted to a size smaller than intended when the coupling facility is storage constrained:
  - MINELEMENT
  - MINENTRY
  - MINEMC

## 10.6 Interactions and dependencies

The ability to have a structure automatically be altered requires:

- Hardware requirement
  - Coupling facility level 8 or greater
- Software requirements
  - OS/390 release 10
  - The exploiters/applications defining the structure have specified that the structure is capable of being altered.
  - The installation has specified that the structure is allowed to be altered. ALLOWAUTOALT(YES) has specified in the active CFRM policy.

## 10.7 Exploiters

The exploiters enabled to use XES Auto Alter for their structures are:

- XCF Signalling
- IMS IRLM Lock
- DB2 SCA
- DB2 IRLM Lock
- DB2 GBP
- DFSMS Enhanced Catalog Sharing
- DFSMS VSAM RLS Lock
- WLM Multi-System Enclaves

For some structures, being over the structure full monitoring percentage may be a normal condition that does not require any diagnostic or corrective actions. Examples of situations where structures may go over the structure full percentage without posing a problem are as follows:

- An exploiter may “format” all of the structure objects within a structure before using them. This formatting makes it appear to structure is in use, causing it to issue IXC585E message, even though there may be plenty of usable space in the formatted structure objects. An example of this kind of protocol is the coupling facility list structure JES2 uses for its checkpoint data.
- Some exploiters define their own “thresholds” for the structures they use. When the data in these structures exceeds the application-defined thresholds, data gets offloaded, or cast out, to DASD. During this process of accumulating data and offloading or casting out, these structures may exceed the structure full monitoring percentage.

  If the installation has defined a higher application threshold for that structure, message IXC585E is issued even though the structures have not exceeded the application offload or castout thresholds. An example of this kind of protocol is the logger structures.

The exploiters not recommended to enable XES Auto Alter processing for their structures are:

- MVS Logger
- DFSMS VSAM RLS cache
- CS/390 VTAM GR
- CS/390 VTAM MNPS
- CICS Temporary Storage
- CICS CF Data Tables
- CICS Named Counter Server
- IMS CQS Shared Message Queue
- IMS Shared FMH

## 10.8 Externals

The messages and commands that have been introduced with the XES auto alter support are described in the following sections.

### 10.8.1 Messages

- IXC588I AUTOMATIC ALTER PROCESSING INITIATED FOR STRUCTURE strname

  ```
  CURRENT SIZE: cursize K; TARGET SIZE: size K
  TARGET ENTRY TO ELEMENT RATIO: entries: elements
  [TARGET EMC STORAGE PERCENTAGE: emcs%]
  ```

  The system has initiated an alter request to correct a resource shortage problem for one or more of the structure objects that the structure contains. When XCF has detected that a structure is at or above its structure full threshold value in terms of one or more of the structure objects that the structure contains, XCF starts and alters the structure to correct the situation, when allowed.

- IXC589I AUTOMATIC ALTER PROCESSING FOR STRUCTURE strname

  ```
  text; rebuild_text
  ```

  The system-initiated structure alter processing has finished and failed. The contents of `text` describes the reason for the failure.

- IXC590I AUTOMATIC ALTER PROCESSING FOR STRUCTURE strname

  ```
  text
  CURRENT SIZE: cursize K; TARGET: targsize K
  CURRENT ENTRY COUNT: count; TARGET: count
  CURRENTE ELEMENT COUN: count; TARGET: count
  CURRENT EMC COUNT: count; TARGET: count
  ```

  The system-initiated structure alter processing has finished. The contents of `text` may be: `COMPLETE. TARGET ATTAINED` or `COMPLETE. TARGET NOT ATTAINED`.

  In the first case, the structure alter processing completed and the requested target was attained; in the second case, the processing was completed but the target was not attained.

### 10.8.2 Commands

The output of D XCF command has been altered, as shown in Figure 81 on page 153.

```
IXC360I  15.53.45  DISPLAY XCF 261
STRNAME: JES2CKPT_2
 STATUS: NOT ALLOCATED
  POLICY SIZE    : 4096 K
  POLICY INITSIZE: 2048 K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: N/A
  DUPLEX         : DISABLED
  PREFERENCE LIST: CF03     CF04
  ENFORCEORDER   : NO
  EXCLUSION LIST : JES2CKPT_1
```

*Figure 81. Output of D XCF,STR,STRNAME=JES2CKPT_2*

## 10.9 Migration

An OS/390 Version 2 Release 10, the system takes over ownership of Structure Full Monitoring and Auto Alter for a particular CF if a downlevel system already owns Structure Full Monitoring for the CF.

CF structure exploiters who specify ALLOWALTER=NO may want to change to allow alters in order to benefit from this function. CF structure exploiters who currently do their own dynamic structure tuning via alter may want to disable their tuning support and have the installation specify ALLOWALTER(YES) on the structure definition in the CFRM policy to allow Auto Alter to tune their structure.

A CFRM policy must be defined that specifies FULLTHRESHOLD(0) and then activated on a OS/390 Version 2 Release 10 or above system to disable Structure Full Monitoring and Auto Alter. This is also true in order to use a FULLTHRESHOLD value other than 80%.

To disable the Auto Alter function without disabling Structure Full Monitoring, a CFRM policy must be defined that specifies or defaults the ALLOWAUTOALT value to ALLOWAUTOALT(NO) and specifies or defaults the FULLTHRESHOLD value to a value other than zero.

## 10.10 Coexistence

There is no need for a compatibility PTF; no Structure Full Monitoring is performed by an OS/390 Version 2 Release 8 or lower system. The OS/390 Version 2 Release 10 system automatically assumes monitoring.

A structure which has connectors running on an OS/390 Version 2 Release 9 or lower system and an OS/390 Version 2 Release 10 could be allocated or altered to a size smaller than the MINSIZE specified or defaulted to in the CFRM active policy. This can occur in the following conditions:

- The first connector to the structure is running on an OS/390 Version 2 Release 9 or lower system.
- The alter process is being driven from an OS/390 Version 2 Release 9 or lower system.

# Chapter 11. Workload Manager (WLM)

In OS/390 Release 10, the MVS Workload Management provides support for the following:

- WLM 64-bit support allows WLM and the SRM to run in the OS/390 Release 10 z/Architecture mode.
- WLM Server Task and Thread Management allows WLM to manage the number of server instances for server environment address spaces per transaction environment.
- New protection options for critical work and new classification qualifier types to help with migrations to goal mode. These two new functional enhancements are designed to:

  a. Allow definition of special protection options for critical work; this removes several installation-identified shortcomings in WLM that are believed to inhibit installations from exploiting goal mode in production environments.

     These inhibitors consist primarily of CICS or IMS migration and segregation, pro-active rather than reactive protection of critical address spaces, and classification of work based on the system on which it executes. The new special protection options are:

     - Using the CPU protection to guarantee relative CPU priority
     - Storage protection for critical applications
     - Options for managing CICS and IMS workloads using transaction server management options

  b. Allow new classification qualifier types to assist migration to goal mode as follows:

     - Scheduling environment name
     - Subsystem collection name
     - Sysplex name
     - System name
     - System name group

**Note:** The special protection options and new classification qualifiers are available on OS/390 Release 10 by installing APARs OW43810, OW43812, OW43813, and OW43855. When any of these APARs are installed, because all of the items include changes to the WLM couple data set, exploitation of any of them requires a new functionality level 011 in the service definition.

**Note:** A coexistence PTF to support this implementation on a Release 9 or older system should be installed. See APAR OW43856 for the corresponding PTF number for the older release.

## 11.1 WLM/SRM 64-bit support

There are three categories of changes to SRM in support of 64-bit real storage:

1. Changes required because expanded storage is not supported by MVS in z/Architecture mode (64-bit mode). Because of this lack of support for expanded storage, hiperspaces and VIO pages are kept in real storage in z/Architecture mode to retain support for applications that use these services.

For SRM, this change means the DirectPO SYSEVENT must make the decision whether a VIO or standard hiperspace page should be put in real storage or sent to AUX in z/Architecture mode, rather than the decision between expanded and AUX that DirectPO currently makes.

In addition, UIC steal has new logic to avoid stealing CASTOUT(NO) hiperspace frames except when the system has a high level of contention for real storage.

Finally in this category, SRM must allow below 16 MB frames to be stolen from logically swapped address spaces when there is no expanded storage online.

2. The next category of SRM changes deals with pageable frame shortages between 16 MB and 2 GB as a separate action from overall pageable frame shortages.

   As MVS images grow much larger than 2 GB, the area between 16 MB and 2 GB become a scarce resource and there will probably always be applications with a dependency on real storage in this range.

3. The final category of changes deals with performance concerns in this new environment. One set of concerns is related to frame queues growing much larger than they are today. These large frame queues require changes to UIC Steal and UIC Update, so that the CPU cost of these functions does not grow too large, and the time that the processing for these functions holds locks and stays disabled does not become so long as to impact other functions in the system.

## 11.1.1 SRM changes due to removal of expanded storage

The changes to SRM due to the lack of expanded storage are limited, since today SRM must function on systems without expanded storage. Current logic sets a bit MCTESNA (expanded storage not available) in z/Architecture mode. The storage management policy adjustment algorithms and working set management are not changed and will function as they do today in an all real-storage system.

### 11.1.1.1 SYSEVENT DirectPO

One necessary change to SRM is that in z/Architecture mode, the DirectPO SYSEVENT must make a placement decision for hiperspace and VIO pages between real and AUX, as opposed to between expanded and AUX.

**Note:** In z/Architecture mode, DirectPO is only supported for VIO and hiperspace pages. The other page types (PageOut, Virtual Fetch, and Self Steal) are no longer supported.

The current compatibility mode DirectPO decision is based on the expanded storage migration age, which obviously is not a useful measure of system performance in z/Architecture mode. Instead, system high UIC is compared to the criteria table value of the page type (VIO or hiperspace). If the system high UIC is less than the criteria table value, the page is put in real storage. Otherwise, it is sent to AUX.

Increasing the UIC increment value from 1 second to 10 seconds--which increases the time interval a one-byte UIC can represent to about 42 minutes--makes system high UIC a better indicator to make this decision. See the discussion of “UIC update” on page 158.

In goal mode, the current DirectPO decision is based on the expanded storage policy of the address space making the request. Again, expanded storage policy is not meaningful in z/Architecture mode. Instead, the placement decision is based on the UIC bucket distribution.

If at least 10% of the system real storage is discretionary (UIC buckets 3 and 4) or there is a lot of real storage available, the following can occur:

- DirectPO directs hiperspace and VIO pages to real.
- If the address space is below its protective storage target, DirectPO directs pages to real.
- If the address space is above its restrictive real storage target, DirectPO directs pages to AUX.

#### 11.1.1.2 SYSEVENT STGTEST

OS/390 Release 10 has a SYSEVENT keyword called STGTEST. TYPE=BLOCK returns data for expanded storage, TYPE=BYTE returns data for central and expanded storage. In either case, three values are returned which are documented as:

1. Use of this number of frames affects system performance very little, if at all.
2. Use of this number of frames affects system performance to some degree.
3. Use of this number might substantially affect system performance.

The actual calculation of these numbers depends on whether the system is in compatibility or goal mode. In compatibility mode, the actual calculation of these numbers is:

1. Available frames. If the home address space has a minimum storage isolation target, also include the amount the space is below this minimum value. If the home address space has a maximum storage isolation target, this value will not be more than the maximum.
2. Value 1 + frames owned by spaces swapped out longer than the hiperspace criteria value. If TYPE=BLOCK, also 2% of expanded owned by swapped-in address spaces. Also adjusted by storage isolation.
3. Value 2 + 75% of the free AUX slots below the AUX shortage threshold

In goal mode, the values are as follows:

1. Basically the same as compatibility mode, with the storage isolation adjustments based on the storage targets that WLM may have set for the space.
2. Value 1 + frames from the old end of the UIC distribution, adjusted by storage targets.
3. Value 3 is equal to Value 2.

In z/Architecture, mode it is no longer meaningful to differentiate between TYPE=BYTE and TYPE=BLOCK. Therefore, in z/Architecture mode, the current TYPE=BYTE calculations is done whether TYPE=BYTE or BLOCK is specified. The TYPE=BYTE calculation works in an all-real system. Note the STGTEST documentation specifically discusses expanded storage, so it is updated to describe the z/Architecture mode support.

#### 11.1.1.3 CastOut(No) ESO hiperspaces

Another issue related to keeping hiperspace pages in real storage is how to protect frames owned by CastOut(NO) ESO hiperspace. ESO hiperspace pages

cannot go to AUX, so if the system needs to take these frames, the data is tossed.

In the case of a CastOut(NO) hiperspace, the system should only take frames in times of severe storage contention. RSM keeps CastOut(NO) hiperspace frames on the data space fixed frame queue and only attempts to take these frames during UIC steal when directed to by SRM with a new bit in the IARXS parameter list (STEALCAST).

SRM only steals CastOut(NO) frames when the system high UIC gets below a threshold value (maybe UIC of 20 seconds). RSM keeps a RAX count of CastOut(NO) frames owned by each address space so SRM knows how many of these frames can be stolen from each address space.

#### 11.1.1.4 Below 16M Fix Frame Steal

Currently SRM enables RSM to steal < 16M frames from logically swapped spaces when the space has between 1 and 10 fixed below frames and there is expanded storage available. RSM is then able to steal these below fixed frames to expanded storage while the address space remains logically swapped. SRM calls RSM to restore these stolen frames before either swapping the space in or converting the swap to a physical swap. The function of stealing below frames from logically swapped address spaces is also necessary in z/Architecture mode.

In fact it is probably more important because, as the amount of real owned by an MVS images grows, below frames become a scarcer and scarcer resource. To support this function in z/Architecture mode, RSM exchanges the below frames for above frames, rather than stealing below frames to expanded.

To enable this RSM function, SRM in z/Architecture mode does not require that expanded be available in order to make an address space eligible for stealing below 16 Meg fixed frames. The current SRM eligibility test also requires that the total amount of non-migratable expanded not be too large a percentage of online expanded. Because exchanging a below fixed frame for an above frame does not increase the total amount of fixed storage, there is no need to have an equivalent check in z/Architecture mode.

#### 11.1.1.5 Pageable frame shortages

z/Architecture mode introduces a new storage “line”: the 2G line. SRM manages the region between 16M and 2G as a separate resource when detecting pageable frame shortages. The logic is much like the processing for total pageable frame shortages, except it is driven by new RSM counts of frames between 16M and 2G.

There is a new return code on the IRA400E and IRA401E messages to identify a shortage between 16M and 2G. In z/Architecture mode, there is no difference between a DREF shortage and a total pageable frame shortage because without expanded, DREF pages must be in real.

#### 11.1.1.6 UIC update

The UIC update process is the other area of performance concern. Currently the UIC interval is 1 second. In other words, when a frame's UIC value is increased by 1, it represents 1 second when the frame was not referenced. To reduce the overhead of UIC update in z/Architecture mode, this interval is increased to 10 seconds. The current logic to reduce the frequency of UIC update beyond this

base interval is maintained, which allows UIC update frequency for swappable spaces to grow to 100 seconds and non-swappables to 200 seconds. There are several implications of this change:

- Swap working sets tend to grow, since a UIC 1 is used to identify the working set of an address space in most cases.
- Logical swap-in runs UIC for the space being swapped in. In z/Architecture mode, UIC update should only be run if it has been longer than 10 seconds since UIC update was last done for the space.

## 11.2 WLM server task/thread management

The objective of the WLM Server Task/Thread Management enhancement is to reduce complexity in exploiting OS/390 functionality by having the system determine the number of server instances per server address space, instead of forcing the installation to make that decision.

Since OS/390 1.3, the Workload Manager component gives applications the capability to distribute work requests to a set of server address spaces. While WLM manages the number of server address spaces per transaction environment, it requires the application programmer to determine the number of server instances tasks/threads which select work requests per server address space.

Some applications, such as DB2 Stored Procedures, may consume very different amounts of system resources per work request. For example, a DB2 Stored Procedure can be a simple database update or a complex update including a huge binary object which should be stored in the database.

These two requests require different amounts of system resources in terms of CPU and storage. Because tasks or threads of the same address space share system resources, for example virtual storage, it can be very difficult for the application programmer to determine the number of server tasks which run in parallel in a server address space.

If the installation is unable to determine a good number of server instances per address space, it is possible that only one instance per server address space runs to avoid resource contention or abnormal program terminations. This in turn can result in wasting system resources which influences the capacity of the OS/390 system.

With the introduction of WLM Server Task/Thread Management, the application can give WLM the ability to manage the server instances per server address space. WLM and SRM then measure the resource consumption of server instances running in the same address space and recommend the number of server instances to be started or stopped for the application. This support significantly reduces the complexity for installations to exploit the functionality of OS/390. In addition, it better exploits system resources and therefore helps to better utilize existing system resources.

For OS/390 Release 10, support is being introduced so that WLM/SRM can manage the number of server instances per server address space. WLM Server Task/Thread Management is an extension to WLM Queue MPL Management introduced with OS/390 Release 3.

The existing support allows WLM to start and stop server address spaces based on the number of work requests queued to WLM. But it requires the application or user to determine the number of server instances per server address space which can select work requests from WLM.

With the new enhancement, an application can give WLM the capability to manage the number of server instances for its server address spaces. WLM then monitors the number of server instances for each work queue (transaction environment) for the application environment and communicates the number of server instances to start and stop to the server address spaces of the application. The application must react accordingly to the recommendations from WLM.

### 11.2.1 Assumptions

WLM manages the number of server instances if the following conditions apply for a given application environment:

1. The system is in goal mode and the application environment is partitioned(1) and managed(2) by WLM.
2. There is a set of worker subtasks within the server address space that each obtains a queued request(3), processes the request, and then loops back to obtain the next request.

The existing external interfaces for server address spaces are enhanced to allow the exploiter to use Queueing and Server Manager services in exactly the same way as today, as well as to leave the adjustment of server instances to WLM.

In the latter case, the server managers of the application use a new service to obtain the number of server instances to start. If the number of subtasks must be reduced, WLM gives a new return code back on the IWMSSEL interface to inform the subtask that it should terminate, without terminating the whole server address space.

### 11.2.2 General changes

- Change application programming interface to server environment manager to allow an application to tell WLM that it wants to get its tasks managed. Change the external interfaces to allow WLM to tell server address spaces to start and/or stop tasks which select work.
- Change WLM server environment component to execute SRM recommendation for the number of server tasks running in server address spaces.
- Collect resource consumption data about virtual storage, lock contention, tcb processor utilization and delays to project the required number of server instances for a server address space.
- Enhance SRM processing to recommend the number of server instances per server address space in addition to the number of server address spaces per transaction environment that WLM already determines.

### 11.2.3 Potential exploiters

DB2 Stored SQL Procedure support can exploit, since DB2 V4.2, WLM Queueing Manager Services introduced with OS/390 Release 3. The existing exploitation of WLM Queueing Manager services requires that the user determines the number

server instances per server address space which can concurrently execute a stored procedure. If stored procedures can manipulate huge amounts of data (for example, by storing or retrieving binary large objects), it is difficult for the user to determine a good number for concurrently running worker tasks which select work per server address space.

By exploiting the new services, WLM/SRM monitor the server address spaces and adjust the number of server tasks selecting work from WLM based on the resource consumption of the server address spaces. If the user defines different service classes for stored procedures executing different types of work requests, WLM can adjust the number of server instances on a per service class basis.

OS/390 UNIX System Services provides extensions to the existing WLM Queueing Manager services by providing the capability to transfer application data from the queue manager to the server manager address spaces. UNIX System Services enhances their existing interfaces to allow server managers using UNIX System Service APIs to exploit the new WLM functionality.

WebSphere exploits UNIX System Services APIs and WLM Queueing Services to let WLM manage the server address spaces executing servlets. WebSphere exploits the extended queueing server interfaces provided by UNIX System Services.

### 11.2.4 UNIX System Services extensions

OS/390 UNIX System Services extends its WLM interfaces to make the new functionality available to exploiters like WebSphere. The general interface extensions for WLM Server Task/Thread Management assume that an exploiter uses a new service which returns the number of server instances to start and which is suspended by WLM if no information should be passed to the caller.

For UNIX System Services, the interface is slightly changed. UNIX System Services passes an ECB per server address space using the new interfaces to WLM. WLM posts the ECB if it wants to inform the server address space to increase the number of tasks. UNIX System Services invokes the new service to obtain the actual number of server instances which should be started and passes this information to the server address space.

## 11.3 Defining special protection options for critical work

The objective of the new special protection options is to remove the major problems in WLM that prevent wider adoption by installations. This is done by providing new externals to allow performance administrators to:

1. Protect CPU for critical regions more stringently by assigning CPU protection to assure that important address spaces always have a higher CPU dispatching priority than work of lower importance. (In this case, “importance” refers to the importance level of the goal assigned to the address space.) This provides a guarantee that CPU access for important address spaces is never impacted by work of lower importance.
2. Protect storage in a storage-constrained system. When storage is constrained, an online region that is idle for a long period of time may be exposed to paging delays when it becomes active again. The cause of this behavior is that, while the region is idle, WLM believes it requires little storage

to meet its goal. When the region becomes active, WLM must relearn its storage requirements.

3. Restrict which regions are managed using transaction goals, and assign performance goals to work based on the run-time environment. Some CICS or IMS regions cannot be effectively managed using transaction response times.

   However, once any CICS or IMS regions are switched to transaction response time goal management, *all* regions must be managed the same way, regardless of whether they are used in a production environment or for testing purposes.

### 11.3.1 CPU protection

In WLM goal mode, there is no guarantee that the dispatching priority of critical address spaces will be higher than that of the other work in the system. If the critical work is meeting its goals and there is less important work that is missing its goal, WLM may assign a higher dispatching priority to the lower importance work, as long as it does not impact the higher importance work.

APAR OW43855 provides internals to support the CPU CRITICAL = YES option in the WLM service class definition panel, as shown in Figure 82. With this support, regions or transactions with very stringent service level agreements can be marked CPU CRITICAL so that lower importance work will not have a higher dispatching priority.

```
  Service-Class  Notes  Options  Help
------------------------------------------------------------------------------
                          Create a Service Class              Row 1 to 2 of 2
Command ===> ____________________________________________________________

Service Class Name . . . . . . CLASS1    (Required)
Description  . . . . . . . . . CICS Special CPU Class
Workload Name  . . . . . . . . CICS      (name or ?)
Base Resource Group  . . . . . ________  (name or ?)
Cpu Critical . . . . . . . . . YES       (YES or NO)  <------------------------

Specify BASE GOAL information.  Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

        ---Period---  --------------------Goal---------------------
Action  #  Duration   Imp.  Description
  __
```

*Figure 82. Defining a service class as CPU critical*

#### 11.3.1.1 New service class attributes

CPU protection is part of the service class definition. Earlier designs treated CPU protection like storage protection; while this clustered all of the new externals in one place, there was no way to hide the SRM period-level CPU management. This led to SRM having to propagate CPU protection from the address spaces to periods.

CPU protection was moved to the service class definition (with WLM restricting it to single period service classes to satisfy SRM requirements) to make its specification and SRM implementation scope consistent. Specify yes or no field on the service class definition panel, as shown in Figure 82 on page 162.

#### 11.3.1.2 CPU protection considerations

CPU protection is always defined for a service class, never in classification rules. The following considerations should be understood:

- WLM is responsible for restricting CPU protection to single-period service classes.The decision to restrict CPU protection to single-period service classes follows from the SRM implementation of CPU management, which is period-level, and from SRM implementation restrictions about the order of dispatching priorities within multi-period service classes. This restriction does not adversely affect the primary target of this support (CICS and IMS regions), which typically run in single period service classes.
- Because SRM may ignore protection in certain cases, reporting interfaces that work at the address space level should distinguish between classification data and how the address space is being managed at the instant the interface is called. On performance-sensitive interfaces, it is acceptable to report only the instantaneous data.
- Reporting interfaces that provide total mode period level data rather than address space level data should report only the specification information such as whether CPU protection was assigned.

### 11.3.2 Storage protection

APAR OW43810 implements a new option called Storage Critical that can be specified in the goal mode classification rules. Storage Critical=YES means that the address space keeps close to its high water mark of storage used even if storage is not needed to meet its goals.

Storage protection may be assigned to a CICS or IMS transaction service class via the classification rules, or to individual address spaces via their classification rules.

If a CICS or IMS transaction is marked as storage critical, all regions serving the transactions service class receive protection.

The classification rule column for storage protection should only be displayed for subsystem types CICS, IMS, ASCH, JES, STC, OMVS, TSO. While only CICS, IMS, JES, and STC are required to support the CICS or IMS requirements, there are known cases where OMVS daemons might benefit from storage protection.

#### 11.3.2.1 Assigning storage protection using classification rules

CICS or IMS regions may be started by a job to JES or via a started task (STC). Therefore, the only two subsystem types in the WLM classification rules that support this new option are JES and STC.

When defining classification rules to assign a CICS or IMS region a service class, by scrolling right using PF11 twice, the screen then shows the Storage Critical and Manage Region Using Goals Of columns, as shown in Figure 83 on page 164.

**Note:** You can assign storage protection to all types of address spaces using classification rules for subsystem types ASCH, JES, OMVS, STC, and TSO.

```
   Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                Modify Rules for the Subsystem Type       Row 1 to 1 of 1
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : JES         Fold qualifier names?   Y  (Y or N)
Description  . . . Use Modify to enter YOUR rules

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                     <=== More
          --------Qualifier--------           Storage    Manage Region
Action    Type       Name     Start           Critical   Using Goals Of

  ____  1  TN         CICS1                    YES        REGION
****************************** BOTTOM OF DATA ******************************
```

*Figure 83. Defining storage critical in the JES classification rules*

#### 11.3.2.2 Storage protection considerations

For CICS or IMS regions, this allows performance administrators to do the following at an address space level:

- Assign long-term storage protection, so that once storage is acquired, WLM restricts storage donations, even if the region appears to have stopped serving transactions. A storage-protected address space is allowed to donate storage only in the following cases:

  a. Higher importance work needs the storage to meet goals.

  b. Work of equal importance needs the storage, is missing goals, and the protected address space is overachieving its goal. The protected address space can donate storage until it just meets goals.

  c. Work of equal importance needs the storage, is missing goals, and the protected address space is missing its goal by less than the receiver. The protected address space can donate storage until the performance indices are equalized.

  d. Basically, the previous cases say that we continue to use the same donor/receiver order when dealing with importances at or above the protected address space, and prevent donations by protected address spaces to lower importance work.

  For installations converting from compatibility mode, this solves an aspect of the problems cited with the removal of storage isolation, as well as sparse arrivals.

  **Note:** Storage protection may be specified in conjunction with:

  - A short response time goal of twenty seconds or more
  - A single service class
  - All goals except a discretionary goal

### 11.3.3 CICS and IMS region management

In Release 10 with APAR OW43812, new WLM panels have been implemented to tell WLM that a region is ineligible to be managed according to the response times of the CICS or IMS transactions that it is processing. This new option is documented as an exemption from transaction server management.

This solves the CICS or IMS migration granularity problem by allowing installations to migrate one region at a time, and allows them to avoid both setting goals for transactions that run in test regions and managing the test regions to the transaction goals.

This APAR provides the support to have some CICS and IMS regions be managed by transaction response time goals while others are managed with velocity goals. This is specified through a new column (Manage Region Using Goals Of) in the Classification Rules for susbsystems STC and JES. The default is that the regions are to be managed to the goal of the TRANSACTIONS unless REGION is specified in the new column in the classification rules.

#### 11.3.3.1 Assigning CICS or IMS region management

As stated previously, CICS or IMS regions may be started by a job to JES or via a started task (STC). Therefore, the only two subsystem types in the WLM classification rules that support this new option are JES and STC.

When defining classification rules to assign a CICS or IMS region a service class, by scrolling right using PF11 twice, the screen then shows the Storage Critical and Manage Region Using Goals Of columns, as shown in Figure 84 on page 166.

Use the Manage Region Using Goals Of: field in the Modify Rules for the Subsystem Type panel, as also shown in Figure 84 on page 166, to declare that a specific CICS or IMS region is not managed to the response times of the CICS or IMS transactions. Instead, it is managed to the performance goal of the service class assigned to that region. In other words, it will no longer be managed as a server.

**Note:** This option can only be used on the STC or JES classification rules that assign the service class to the address space.

CICS or IMS regions can be managed either to transaction response time goals or to region goals based on subsystem type. Once classification rules have been specified for CICS or IMS, all CICS or IMS regions are managed to the goals of the transactions they are serving.

**Note:** There is a need to differentiate between regions of the same subsystem.

#### 11.3.3.2 CICS and IMS and CPU and storage protection

The primary targets for the CPU and storage protection support are environments which have extremely stringent response time requirements for CICS or IMS workloads, often in the form of service level guarantees. CPU protection and storage protection are direct responses to those requirements.

Two separate issues, addressed by the exemption attribute, concern CICS and IMS workloads. The first is the granularity of migration from velocity management of regions to management based on transaction response times; the second is test versus production regions. The exemption option allows installations to tell WLM by region whether or not regions should be managed by transaction response times, and also allows installations to manage test regions using a “measure of progress” external which more closely matches what they are used to and wish to keep.

#### 11.3.3.3 New classification rule attributes

A new classification rule panel in the WLM administrative application is displayed when the subsystem type supports storage protection or exemption from transaction server management, as shown in Figure 84.

Similar in appearance to the panel for comments, the new panel is reached by scrolling right from the comments panel. The new panel replaces the comments column with either one or two new columns, depending upon the subsystem type. The new panel shows only those attributes which apply to the subsystem type being processed by the user. If none of the new attributes apply, the user should not be shown the new panel in response to a scroll right request.

```
 Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                 Modify Rules for the Subsystem Type       Row 1 to 7 of 7
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : JES         Fold qualifier names?   Y  (Y or N)
Description  . . . Use Modify to enter YOUR rules

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                    <=== More
          --------Qualifier--------           Storage    Manage Region
Action    Type       Name     Start           Critical   Using Goals Of

 ____  1  TN         COM*     ___             NO         TRANSACTION
 ____  2    UI         COMBLD   ___           NO         TRANSACTION
 ____  2    UI         COMFTP   ___           YES        TRANSACTION
```

*Figure 84. Classification rule panel to define “Manage Region Using Goals Of”*

### 11.3.4 CICS and IMS use of CPU and storage protection

The other new options are also available to help performance administrators protect critical regions. Although applicable to several other subsystem types, CICS and IMS work will particularly benefit from:

- Storage protection
- CPU protection

#### 11.3.4.1 Storage protection for CICS and IMS

Use the Storage Critical option on the Modify Rules for the Subsystem Type panel, as shown in Figure 85 on page 167, to assign long-term storage protection to a CICS or IMS transaction service class. Even if the region appears to have stopped serving transactions, WLM restricts storage donations unless other work of greater importance needs it, or unless work of equal importance needs it “more”, that is to say, work of equal importance that is failing its goals while the protected work is either overachieving its goals or failing by a lesser margin.

The objective of storage protection is to give an installation a way to guarantee that certain applications are proactively storage-isolated. This allows protective storage isolation, avoiding impact on sudden changes in storage demand, as well as protections to page faults on regions with low overnight activity when interactive users return in the morning.

```
---------------- Modify Rules for the Subsystem Type        Row 1 to 2 of C
===> _____________________________________________ SCROLL ===> PAG

Subsystem Type . : STC          Fold qualifier names?   Y  (Y or N)
Description  . . . IBM-defined subsystem type
Action codes:  A=After    C=Copy         M=Move     I=Insert rule
               B=Before   D=Delete row   R=Repeat   IS=Insert Sub-rule
                                                 <=== More
           -------Qualifier---------      Storage    Manage Region
 Action    Type       Name     Start      Critical   Using Goals Of
  ____  1  SY         SYST1    ___        NO         TRANSACTION
  ____  2    TN         CICSTEST ___      NO         REGION
  ____  2    TN         CICS*    ___      YES        TRANSACTION
```

*Figure 85. Classification rules showing Storage Critical and Region Goals*

This storage protection can also be assigned to the specific address space in which the service class is running, using the STC or JES classification rules that assign the service class to the address space.

#### 11.3.4.2 CPU protection for CICS and IMS

CPU protection specification is part of the service class definition and is restricted to single period service classes with velocity or service time goals.

Use the CPU Critical option on the Modify a Service Class panel (as shown in Figure 82 on page 162) to assign CPU protection to a CICS or IMS service class. This assures that this work always has a higher dispatching priority than work of lower importance.

This CPU protection can also be assigned to the specific address space in which the service class is running, using CPU critical=yes on the service classes that run in that address space.You can also assign CPU protection to service classes handling address space-oriented work, enclave work, but the service class must:

- Have a single period
- Not have a discretionary goal

If a CICS or IMS region is managed as a server by WLM; for example, managed to the response time goals of the transactions it serves, and any of the transaction service classes it serves is assigned CPU protection, then the CICS or IMS region itself automatically has CPU protection.

## 11.4 New classification qualifier types

OS/390 Release 10 addresses user requirements to set performance goals based on the run-time location of the work. This release provides the ability to classify work based on system name (or named groups of them).

Classification rules are the rules you define to categorize work into service classes, and optionally report classes, based on work qualifiers. A work qualifier is what identifies a work request to the system. The first qualifier is the subsystem type that receives the work request.

There is one set of classification rules in the service definition for a sysplex. They are the same regardless of what service policy is in effect; a policy cannot override classification rules. You should define classification rules after you have defined service classes, and ensure that every service class has a corresponding rule.

The list of work qualifiers introduced in Release 10 and their abbreviations are:

**SY** System name
**SYG** System name group
**PX** Sysplex name
**SSC** Subsystem collection name
**SE** Scheduling environment name

**Note:** Different subsystem types support each qualifier type.

- The PX qualifier is supported by all IBM-defined subsystem types.
- The SE is supported by DB2 and JES.
- The SSC is supported by DB2, DDF and JES.
- The SY and SYG are supported by ASCH, OMVS, STC and TSO.

### 11.4.1 System name and system name group

These are supported for address spaces whose execution system is known at classification time (ASCH, TSO, OMVS, STC). JES is not supported because the system on which conversion occurs (also where classification occurs) may not be the system on which the job runs. Since JES uses WLM queueing services, changing the service class after the job is queued would pollute the queue delay data that WLM uses to decide when to start more initiators. Subsystem-defined transactions (CICS or IMS) and enclaves (the remainder) are not bound to an execution system at classification time either.

Since the system name is known to WLM when classification runs, no changes to the IWMCLSFY parameter list are required. SRM/WLM does not return any attributes on the classification query services that cannot explicitly be passed on IWMCLSFY.

**Note**: DB2 sysplex parallelism, prior to exploitation of multisystem enclaves, queries the classification attributes of the originator of the parallel query and passes the result to participating systems. Each participating system uses these attributes to allow the installation to assign the same performance goal to each piece of the parallelized query (each piece runs in an independent enclave).

If system name or system name group is used as a classification attribute and the same performance goal is desired for each piece of the parallelized query, it is the installation's responsibility to ensure that all systems to which the query might be parallelized are included.

In order to make sure that installations are not surprised by this, subsystem type DB2 (used for remote sysplex query parallelism tasks) does not support classification by system name or system name group even though DB2 clients (TSO) will support it. Thus an installation that attempts to play by the rules and copies classification rules containing SY/SYG qualifiers from subsystem type TSO to DB2 will get an error message on the copy.

This is not a problem once multisystem enclaves are used for DB2 sysplex query parallelism instead of multiple enclaves per DB2 query, as long as the entire parallelism group is in WLM goal mode.

The system name or system name group qualifier is used to assign goals based on the execution system.

The system name qualifier is supported for address spaces whose execution system is known at classification time. Note that JES is not eligible for this qualifier, as the system on which classification occurs may not be the system on which the job is run. Subsystem-defined transactions (CICS or MS) and enclave-based transactions are not bound to an execution system at classification time, and are therefore not eligible either.

The following subsystems support the system name classification type:

**ASCH** The name of the execution system, as identified at classification time.
**OMVS** The name of the execution system, as identified at classification time.
**STC** The name of the execution system, as identified at classification time.
**TSO** The name of the execution system, as identified at classification time.

### 11.4.2 Sysplex name

This has the same considerations as qualifier type system name if the work crosses sysplex boundaries. To provide the most flexibility for installations and exploiters, this attribute is supported for all subsystem types.

The sysplex name qualifier is used to assign goals based on the execution sysplex. This facility makes it easier to use a single service definition across multiple sysplexes.

For work that crosses sysplex boundaries, this qualifier can be used by all subsystems.

### 11.4.3 Subsystem collection name

This is supported for subsystem types (JES, DB2, DDF) which have names for groups of subsystem address spaces that define a boundary relevant to the subsystem as follows:

**JES** For JES2, the MAS name. For JES3, the JESplex name.
**DB2** The data sharing group name.
**DDF** The data sharing group name.

The value of the attribute must be communicated to SRM; for address spaces (JES), it is added to an existing sysevent parameter list (IRAICSP) built by the initiator. JES2 and JES3 need to pass the value to the initiator using the Subsystem Interface (SSI) in macro IEFSSJS.

The subsystem must pass a new value to IWMCLSFY. As a result, a new keyword (SUBCOLN) is added, and a new version number is required.

Since this is address space classification data, it should be reported in the SMF 30 record for the address space. This requires SRM to pass the value to SMF. The value must be reported by the classification query services for both address spaces and enclaves. If the value changes after classification occurs and before

the job starts, the subsystem must reclassify the work. This probably does not occur for any of the existing exploiters.

The subsystem collection name gives the installation the possibility to differentiate work been executed within different subsystems. A single job class can have multiple definitions within a sysplex:

### 11.4.4 Scheduling environment

The scheduling environment qualifier is supported for subsystem types (JES) which exploit WLM scheduling environments. Base OS/390 support already passes the value to SRM for SMF 30 reporting on Sysevent JobSelect, so the only change is to also specify the value during classification. This new qualifier is supported also for subsystem type DB2 because DB2 V5 sysplex query parallelism must support the qualifier types of any possible query originators (including batch).

Because this qualifier type takes values longer than 8 bytes, the starting position, shown as Start in Figure 86 on page 170, indicates that all jobs that have a scheduling environment name with the characters 'BANS' starting in the tenth position are assigned to JESMED.

```
Subsystem Type . . . . . : JES
Description  . . . . . . . . All JES2 service classes

   -------Qualifier-------------            -------Class--------
   Type       Name     Start                Service     Report
                                  DEFAULT: JESLOW      ________
1  SE         BANS*     10                  JESMED      ________
```

*Figure 86. Scheduling environment qualifier in JES classification rule*

The subsystem must pass a new value to IWMCLSFY. As a result a new keyword (SCHEDENV) is added, and a new version number is required. If the value changes after classification occurs and before the job starts, the subsystem must reclassify the work.

The use of scheduling environment names has the following meaning:

**JES** The name of the scheduling environment in which the JES work is assigned.

**DB2** Scheduling environment name associated with the originator of the query.

## 11.5 Scenarios for managing CICS and IMS workloads

While the CPU and storage protection attributes apply equally well to other types of work, they are really oriented toward improving management or migration for CICS and IMS environments. It is common for a single system to have several hundred regions to manage. The following section gives an overview of the various installation scenarios, and how the installations are expected to respond when using this support. All of these scenarios assume that the installation is unhappy with default WLM server management.

### 11.5.1 Production regions running normal transactions

**Scenario 1:** A CICS production region is running normal (non-conversational) transactions, for which response time goal management makes sense, and the regions have enough activity so that WLM manages them as servers essentially all the time.

- Region: **CICSPROD**
- Service class: **PROD1**

The scenario describes certain transactions that are critical and must never be delayed by less important work, even if overall system throughput must be sacrificed to provide this level of service.

- Transaction name: **ABC**
- Service class: **ABC1**

Transactions ABC are very important business transactions and deserve both CPU and storage protection, and they execute in CICS region CICSPROD. The following WLM specifications should be made.

#### 11.5.1.1 Define a service class for PROD1

The production region CICSPROD has a service class of PROD1 and CPU critical is not specified, as shown in Figure 87.

```
Service-Class  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                         Modify a Service Class              Row 1 to 2 of 2
Command ===> ____________________________________________________________

Service Class Name . . . . . : PROD1
Description  . . . . . . . . . CICS production region
Workload Name  . . . . . . . . STC       (name or ?)
Base Resource Group  . . . . . ________  (name or ?)
Cpu Critical . . . . . . . . . NO        (YES or NO)

Specify BASE GOAL information.  Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

        ---Period---  ---------------------Goal---------------------
Action  #  Duration   Imp.  Description
```

*Figure 87. CICS region CICSPROD service class definition*

**Note:** Since the main concern is the transactions running in the CICS region, simply protect the transactions and that protection will be inherited by the CICS region where the transactions are running.

#### 11.5.1.2 Classification rule for CICSPROD region

This classification rule assigns the service class PROD1, not shown because by shifting right with PF11 twice, you can specify TRANSACTION for the CICS production region CICSPROD as shown in Figure 88.

```
 Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
               Create Rules for the Subsystem Type        Row 1 to 1 of 1
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type STC   (Required)    Fold qualifier names?   Y  (Y or N)
Description  . . . CICS production region

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                     <=== More
          --------Qualifier--------           Storage    Manage Region
Action    Type       Name     Start           Critical   Using Goals Of

 ____  1  TN         CICSPROD ___             NO_        TRANSACTION_
****************************** BOTTOM OF DATA ******************************
```

*Figure 88. Classification rule for STC for CICS region CICSPROD*

Protection based on transaction service classes is sufficient. This gives the installation the advantage of not having to worry if the topology (relationship between transaction service classes and regions serving them) changes.

#### 11.5.1.3 Assign CPU protection to ABC1

Since CPU protection is needed for service class, assign CPU protection to the transaction service class ABC1 in the service class definition in the Cpu Critical field, as shown in Figure 89.

```
 Service-Class  Notes  Options  Help
------------------------------------------------------------------------------
                        Create a Service Class              Row 1 to 2 of 2
Command ===> ____________________________________________________________

Service Class Name . . . . . . ABC1      (Required)
Description  . . . . . . . . . Service class for TN ABC
Workload Name  . . . . . . . . CICS      (name or ?)
Base Resource Group  . . . . . ________  (name or ?)
Cpu Critical . . . . . . . . . YES       (YES or NO)

Specify BASE GOAL information.  Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

        ---Period---  ---------------------Goal---------------------
Action  #  Duration   Imp.  Description
```

*Figure 89. Specify CPU critical for service class ABC1*

#### 11.5.1.4 Assign storage protection to ABC1

Since storage protection is needed, assign storage protection to the transaction service class using the classification rules in subsystem types CICS, as shown in Figure 90.

```
 Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                Modify Rules for the Subsystem Type       Row 1 to 1 of 1
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : CICS        Fold qualifier names?   Y  (Y or N)
Description  . . . CICS classificaiton rules

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                       <=== More
          --------Qualifier--------           Storage    Manage Region
Action    Type       Name     Start           Critical   Using Goals Of

 ____  1  TN         ABC      ___             YES        N/A
****************************** BOTTOM OF DATA ******************************
```

*Figure 90. Define storage critical for transaction ABC*

This specification for the production regions running non-conversational transactions is necessary for regions having periods of low activity during which WLM stops managing them as servers. The problem is that during periods of low activity, the region's pages are stolen by competing workloads.

For reporting, the regions never show that CPU or storage protection was specified, but does show that they are protected (if it was assigned above) while serving the transactions. Any service class reporting will show protection as specified above.

Therefore, protection based on transaction service classes is useful. This gives an installation the advantage of not having to worry if the topology (relationship between transaction service classes and regions serving them) changes, and protects the regions during periods of normal activity.

### 11.5.2 Production regions running conversional transactions

**Scenario 2:** Production regions running conversational transactions and test regions.

Conversational transactions are not amenable to response time management, and many installations want to simply run test regions with velocity goals.

These regions must be managed using a velocity or discretionary goal. Protection of the regions themselves is also required, however, since WLM does not manage the regions as servers during the low activity periods.

A CICS production region is running normal (non-conversational) transactions.

- Region: **CICSCONV**
- Service class: **PRODC**

The scenario describes certain transactions that are critical and must never be delayed by less important work, even if overall system throughput must be sacrificed to provide this level of service.

- Transaction name: **CONV**
- Service class: **CONVC**

Transactions named CONV are very important business transactions and deserve both CPU and storage protection, and they execute in CICS region CICSCONV. The following WLM specifications should be made.

### 11.5.2.1 Define a service class for CONVC

The production region CICSCONV has a service class of PRODC. If CPU protection is needed, assign CPU protection to the regions' service class, as shown in Figure 91 on page 174.

```
Service-Class  Notes  Options  Help
------------------------------------------------------------------------------
                          Create a Service Class              Row 1 to 1 of 1
Command ===> ____________________________________________________________

Service Class Name . . . . . . PRODC__   (Required)
Description  . . . . . . . . . CICS production (Conversational)
Workload Name  . . . . . . . . STC_____  (name or ?)
Base Resource Group  . . . . . ________  (name or ?)
Cpu Critical . . . . . . . . . YES       (YES or NO)

Specify BASE GOAL information.  Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

        ---Period---  --------------------Goal---------------------
Action  #  Duration   Imp.  Description
```

*Figure 91. CICS region CICSCONV service class definition*

**Note:** Since the main concern is the transactions running in the CICS region, assigning CPU protection to the region protects the transactions that are running.

### 11.5.2.2 Classification rule for CICSCONV region

This classification rule assigns the service class PRODC, not shown because by shifting right with PF11 twice, you can specify REGION for the CICS production region CICSCONV, as shown in Figure 92.

```
Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                Modify Rules for the Subsystem Type       Row 1 to 2 of 2
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : STC         Fold qualifier names?   Y  (Y or N)
Description  . . . CICS transaction -Conversational

Action codes:   A=After     C=Copy       M=Move     I=Insert rule
                B=Before    D=Delete row R=Repeat   IS=Insert Sub-rule
                                                       <=== More
          --------Qualifier--------           Storage    Manage Region
Action    Type       Name     Start           Critical   Using Goals Of

  ____  1  TN         CICSPROD ___             NO         TRANSACTION
  ____  1  TN         CICSCONV ___             YES        REGION
****************************** BOTTOM OF DATA ******************************
```

*Figure 92. Classification rule for STC for CICS region CICSCONV*

If storage protection is needed, assign storage protection to the regions using classification rules in subsystem types STC and/or JES which assign service classes to the regions, as shown in Figure 92 on page 174.

For reporting, regions serving transactions in this service class will show that CPU and/or storage protection was specified based on the region's storage protection value and the CPU protection value of the region's service class.

Table 5 shows how to assign protection based on the new CPU and storage assignments.

*Table 5. Summary of new attributes*

| When you... | WLM... |
|---|---|
| Assign CPU protection to a service class used to manage address spaces and/or enclaves. | Protects any address space or enclave managed according to the goals of that service class. Address spaces being managed as servers are managed according the goals of the served transactions. |
| Assign storage protection to a ASCH, JES, OMVS, STC, or TSO address space. | Protects any address space which matches the classification rule, regardless of its server status. Address spaces in multiperiod service classes or in service classes with a short response time goal are excluded from protection. |
| Assign CPU or storage protection to a CICS or IMS service class. | Protects any regions recognized as serving that CICS or IMS service class, unless you prevent WLM from managing the regions as servers. |
| Prevent WLM from managing a CICS or IMS region according to the response times of the transactions it is running. | Does not recognize the region as a server. The region is managed using the goal of the service class assigned to the region. Transaction data is not reported in the service classes to which the transactions are classified, but is reported in their report classes if assigned. |

When a production region is running conversational transactions, response time goals are not appropriate; or perhaps it is a test region, and the installation wishes to use a velocity goal. By exempting the region from management to the transaction response time goals, it will instead be managed according to the goal of the service class assigned to that region. If either storage or CPU protection is needed, that goal must be a velocity goal, as discretionary goals are not eligible for storage or CPU protection.

## 11.6 Workload management migration

The OS/390 Release 10 version of the WLM administrative application only works with a WLM couple data set allocated by OS/390 Release 4 or above. Because of changes to the structure of the WLM couple data set, you must use a CDS format level that is appropriate to your version of the administrative application.

At present there are three CDS format levels. CDS format level 3 is required for use with the OS/390 Release 10 version of the WLM administrative application. If your WLM couple data set was allocated by OS/390 Release 3 or below (CDS format level 1 or 2), you must reallocate it. This is required regardless of whether you want to use the new functions in OS/390 Release 4 and above.

Previous versions of the WLM administrative application can be used with a WLM couple data set formatted by OS/390 Release 4 or above, provided that new function is not exploited.

### 11.6.1 Migration considerations for CICS or IMS enhancements

On OS/390 Release 10 with APAR OW43812 installed, you should be aware of the options available to help system administrators protect critical CICS and IMS regions, and how these options may affect other work.

If storage protection is selected for a CICS or IMS address space or transaction service class, storage donation to other work is restricted. Storage will only be donated when higher importance work needs the storage to meet its goals, or when work of equal importance needs the storage more than the protected work. Storage will never be donated to work of lower importance.

If CPU protection is selected for a CICS or IMS address space or transaction service class, that work will always have a higher CPU dispatching priority than work of lower importance.

If a CICS or IMS region is exempted from management according to the response times of the CICS or IMS transactions it is processing, it will instead be managed according to the goal of the service class assigned to that region.

### 11.6.2 Other migrations issues

Functionality level 011 imposes the following restrictions on the service definition:

- Service classes used for CICS and IMS transactions cannot be used by other subsystem types.
- Multi-period service classes must have periods whose importances do not decrease, meaning later periods must be no more important than earlier ones.
- Running pre-Release 10 JES2 or JES3 on OS/390 Release 10:
  - Classification by system or sysplex name now works.
  - Classification by scheduling environment name and subsystem collection name is *not* supported.
- Pre-OS/390 Release 10 systems ignore rules with new qualifier types.
- SRM may reclassify jobs using different classification parameters than JES uses.
- Only OS/390 Release 10 systems honor CPU protection, storage protection and CICS or IMS region management.

**Recommendation**: Do not use classification by scheduling environment name or subsystem collection name until all systems are both OS/390 Release 10 and JES2 or JES3 Release 10.

# Chapter 12. OS/390 Version 2 Release 10 JES2 enhancements

Enhancements to JES2 implemented in OS/390 Version 2 Release10 include:

- Spool I/O efficiency improvements
- JES2 spool allocation changes
- Changes relating to systems dumps
- Purging jobs on JES2 restarts

## 12.1 Current JES2 processing overview

JES2's spool data set can be thought of as a multi-volume, fixed block data set. Each record is the same size, which is specified by the installation. In general, the default of 3992 bytes (the largest allowed) is used. However, the size can range from 1944 to 3992 bytes.

JES2 writes SYSOUT to spool data sets from the address space that has the SYSOUT data set allocated. Each SYSOUT data set is processed independently (that is, each has its own buffers and I/O stream). When JES2 is writing to the spool, it builds a channel program to write a single spool record and uses EXCP to perform the I/O.

JES2 does some buffering of data. There can be up to 20 “protected” buffers queued for a single SYSOUT data set. There is logic in the channel end appendage to redrive an I/O with the next buffer (if it is for the same volume). However, the buffers are always written one at a time.

### 12.1.1 Writing data to spool

When JES2 is writing data to the spool, there are two methods of assigning records to data sets:

1. Normally, records are assigned one at a time, as needed, to each SYSOUT data sets in an address space. This can cause records for SYSOUT data sets to be jumbled within the spool data set. Adjacent buffers for a SYSOUT data set can be placed on separate tracks, or even on separate volumes. This maximizes spool utilization, but makes any attempt at reading more than a single record difficult.
2. A second method of assigning spool space called *track cells*. The use of this option is controlled by the TRKCELL option on the OUTCLASS statement. In track cell allocation, a data set is assigned multiple adjacent spool records when space is needed (a track cell amount). The process for writing these records is not changed (still one record at a time).

### 12.1.2 Reading data from spool

Reading SYSOUT data sets from the spool volume can be done either one record at a time, or it can take advantage of the track cell allocation of spool records. The standard method uses one pending buffer and one active buffer. A single record is read from the spool volume into the first buffer. The channel end appendage redrives the I/O if the second buffer is available (has been processed) and the next record is on the same volume. Otherwise, the I/O is stopped and restarted when the current buffer is completed.

Taking advantage of the track cell allocation can greatly improve the overall performance of spool reads. Instead of reading a single spool record, a full track cell amount of buffers are read. Two buffer sets are maintained (each set having enough buffers for a full track cell). This reduces the number of I/Os and takes advantage of DASD cache algorithms that read ahead.

Unfortunately, track cell reads are limited to FSS devices (and then only if requested via the PRTnnn statement). This means that other methods of reading data from spool (SAPI, PSO, spool browse) cannot take advantage of track cell spool allocation.

## 12.2 JES2 spool I/O changes in Release 10

The enhancements introduced for JES2 spool activity result in fewer I/Os, more efficient channel programs, and the utilization of parallel I/O processing.

### 12.2.1 JES2 Release 10 spool writes

For spool writes, JES2 Version 2 Release 10 chains many spool records into a single I/O. This is done in two ways:

1. If the record length of a PUT does not fit into a single spool buffer, JES2 ensures that any I/O that is started includes all spool records needed to write the record. In other words, if a PUT for a 8 K record is received, JES2 builds the three spool buffers needed for the I/O and writes them using a single I/O. (Currently, JES2 would start the first and queue the next two.)
2. Additionally, the JES2 spool write process maintains a new table for a track cell index block. This block is pointed to by a Spool Control Record (SCR) at the beginning of the data set. This is a list of all the track cell addresses used by this data set and is created for all track celled SYSOUT data sets, except spin data sets. The track cell index block remains an instorage control block until a threshold of entries have been recorded. At that point, space for the block is assigned on spool and the index block written.

### 12.2.2 JES2 Release 10 spool reads

There are three ways of reading data from spool: the older *single record reads* (for non-track celled data); *standard track cell reads* for track celled data that does not have a track cell index block (and needs one); and *advanced track cell reads*. The method used depends on the nature of the data and the historical patterns of the requester.

Advanced track cell reads use four track group buffer sets. Each set is big enough to hold an entire track group's worth of data (with a maximum of three tracks). Two parallel I/Os are started to read an entire track group's worth of data into a buffer. The track cell *index* is used (or assumed, for special case data sets) to know where to read next. Where possible, entire tracks are read using a Read-Track-Data CCW.

There are some special data sets that can use track cell reads and advanced track cell reads even if no track cell index exists. These are spin data sets, SYSIN data sets (these are written sequentially because of the nature of input processing), and SYSOUT from NJE (again, the nature of how these are written

makes them sequential). In all these cases, the corresponding allocation IOT can be used as an index of track groups for the data set.

### 12.2.3 JES2 EXCPVR enhancement

I/Os performed using the HASP Access Method (HAM) are currently done using EXCP. In OS390 Version 2 Release 10, these have been changed to use EXCPVR.

Additionally, Format 1 CCWs are now used (this is newly supported by EXCPVR in OS/390 Release 10). Format 1 CCWs can reside above the line, thus reducing the amount of below-the-line storage needed to perform spool I/O.

The checkpoint channel program has also been updated to use format 1 CCWs. This allows a number of large page fixed data areas used by CKPT to move above the line.

## 12.3 JES2 spool allocation changes

Changes have been implemented to provide new schemes for JES2 fencing and to improve the way the BLOB (the JES2 cache of available spool space) is managed. These changes ensure that I/O speeds become faster, and that the availability of spool space to running programs does not become an issue. Fencing and performance are often at odds, as the following explanation shows:

- Fencing reduces the impact of losing a spool volume by minimizing the number of volumes a job uses. However, a job that rapidly creates a large amount of SYSOUT is slowed by the limited number of track groups in the BLOB for that volume. These changes have been introduced to provide a compromise between performance and minimizing the impact of a failure.
- Modification has also been made to ensure that, as spool space is allocated, the volumes used alternate (within the mask of allowed volumes). For example, if the current track group is on volume X, then the code that allocates a new track group attempts to allocate it to a volume *other than* X. This ensures that when parallel I/Os are used, they will be going to different volumes.

### 12.3.1 JES2 volume fencing

Prior to OS/390 Version 2 Release 10, JES2 allocated spool space using one of two algorithms:

- Minimum volume fencing

  In minimum volume fencing, the mask starts out as zero. The first track group for the job is assigned randomly from the available volume (that is, from the volumes in the BLOB). The bit in the mask which corresponds to the spool volume of the selected track group is set to 1.

  At this point, all spool space for that job is assigned from just that one volume. If all the volumes in the mask become full and the job needs another track group, then a track group is obtained from another volume (at random), assigned to the job, and the corresponding bit in the mask is turned on.

- No fencing

In an unfenced system, the mask is initially set to all bits on. This allows the spool space to be assigned randomly to the job. It also ensures that the job is spread over the most spool volumes. This provides the best performance because a job that creates a large amount of SYSOUT  be spread over all volumes.

The problem is that a failure of a single spool volume affects nearly all jobs in the system, and probably all large jobs.

#### 12.3.1.1 Fencing with JES2 exits

In addition, a set of JES2 exits (exits 11 and 12) allowed the installation to implement their own type of fencing. All fencing in JES2 was implemented via a spools allowed mask. This mask controls where spool space for a job can be allocated. Each bit in the mask represents one spool volume. A bit being on means that space can be allocated from the corresponding volume.

Using JES2 exits, an installation can manipulate the spools allowed mask. This would allow them to assign allowed spool space based on any one of a number of criteria. Some installations assign masks based on JOBCLASS, while others use a table of JOBNAMEs. Whatever scheme that is used must still work with the new changes being proposed.

### 12.3.2 JES2 Release 10 fencing enhancements

JES2 Release 10 introduces two changes to the fencing algorithm, which allow a compromise between performance and limiting the impact of the failure:

- A new fencing volume limit has been established

  – FENCE=(ACTIVE=YES|NO,VOLUMES=nn) on the SPOOLDEF statement

  The fencing volume limit fences a job to an installation-defined number of volumes. With this form of fencing active, the job starts with a zero spools allowed mask. As a job allocates spool space, it is selected from random volumes and the corresponding bit set in the spools allowed mask.

  When the number of bits set in the mask reaches the installation-defined limit, the job is forced to only use volumes that have bits on in the mask. If all those volumes are full, then the mask is expanded (like the current fencing algorithm).

  Setting the new fencing volume limit to one is the same as today's minimum volume fencing. See 12.3.4, “Spool management example” on page 181 for an example of the use of this new FENCE option.

- A system binding of spool volumes

  – $T SPOOL or $S SPOOL(nnnnnn),SYSAFF=(sys,sys,...) commands

  A second type of fencing associates members with spool volumes. Each spool volume has a mask of systems that can allocate space on the volume. This mask limits the entries that can be placed in the BLOB for that system.

  Since all track group allocations come from the BLOB, jobs are limited to the spool volumes associated with a system. If all spool volumes that are associated with a system become full, spool space starts being allocated from any spool volume that has space.

  The assigning of systems to a spool volume is via the new $T SPOOL or $S SPOOL command. There are no initialization options (since there is no spool

initialization statement). See 12.3.3, “System affinity for spool volumes” on page 181 for examples of the use of this command.

The new options reduce the performance impact of limiting a job to a subset of the spool volumes.

### 12.3.3 System affinity for spool volumes

The $T SPOOL ($tspl) command can be used to associate or assign a system affinity for a spool volume to one or more systems; refer to Figure 93.

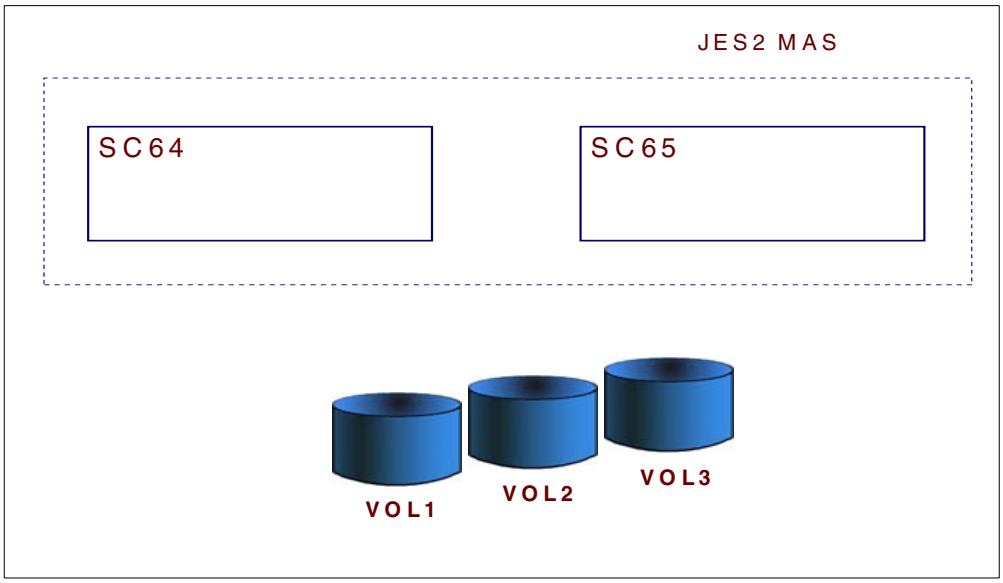



*Figure 93. JES2 MAS*

**$tspl(vol3),sysaff=(sc65)**

```
$HASP893 VOLUME(VOL3)
$HASP893 VOLUME(VOL3) STATUS=ACTIVE,SYSAFF=(SC65),
$HASP893                TGNUM=2000,TGINUSE=716,
$HASP893                TRKPERTGB=3,PERCENT=35
$HASP646 35.8000 PERCENT SPOOL UTILIZATION
```

The following command removes the system affinity for spool volume vol3 from system SC65.

**$tspl(vol3),sysaff=-sc65**

```
$HASP893 VOLUME(VOL)
$HASP893 VOLUME(VOL3) STATUS=ACTIVE,SYSAFF=(),
$HASP893                TGNUM=2000,TGINUSE=716,
$HASP893                TRKPERTGB=3,PERCENT=35
$HASP646 35.8000 PERCENT SPOOL UTILIZATION
```

### 12.3.4 Spool management example

Using the following operator commands to create a JES2 spool environment with fencing and system affinities creates the job track allocation, shown in Figure 94 on page 182.

$T SPL(SPOOL1)  SYSAFF=(ANY)

```
$T SPL(SPOOL2)  SYSAFF=(ANY)
$T SPL(SPOOL3)  SYSAFF=(SC64)
$T SPL(SPOOL4)  SYSAFF=(SC65)
SPOOLDEF  FENCE=(ACT=Y,VOL=2)
```

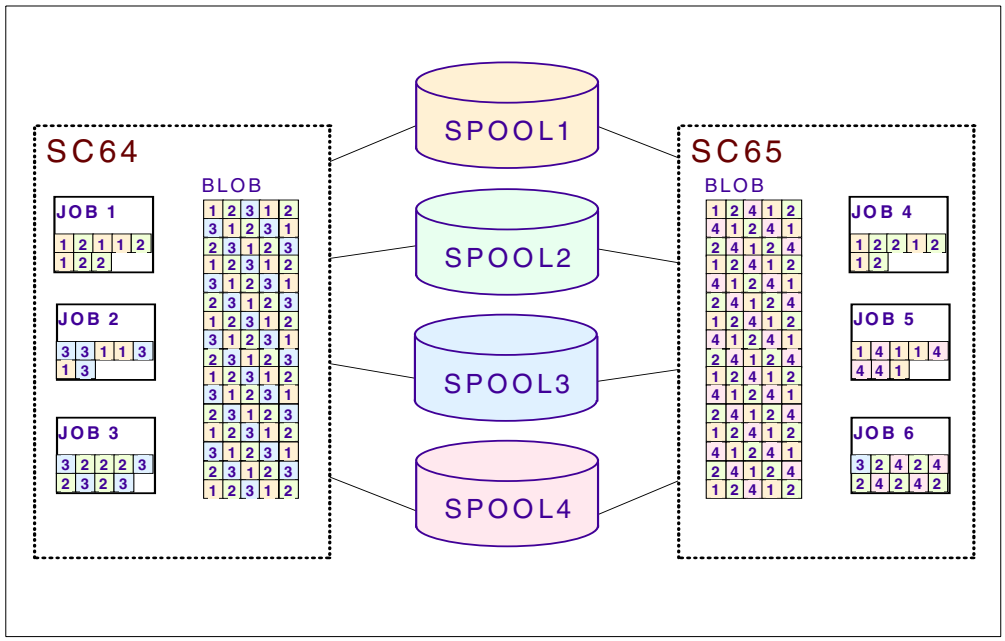



*Figure 94. Track allocation to jobs executing on systems SC64 and Sc65*

In Figure 94, SPOOL4 is not used by jobs executing on SC64 and SPOOL3 is not used by jobs executing on system SC65 due to the definitions in the operator commands specifying SYSAFF= those specific systems.

## 12.4 Multi-system dumps

There is a class of errors in JES2 that requires information from multiple systems to enable correct diagnosis. This is especially true of JES2 work queue errors. To address this, JES2 now supports dumping all members within a sysplex when certain errors are encountered. A new keyword is added as follows:

```
$ERROR macro (DUMP=MAS | SINGLE)
```

This indicates what systems are to be dumped. The $SDUMP macro supports a AFFIELD= parameter that is a bit map of systems to dump (AFFIELD=ACTIVE is also supported). If multiple systems need to be dumped, the invocation of SDUMPX in HASPRAS indicates the systems and address spaces on those systems to dump.

This introduces the problem of how to know *what* dataspaces or additional data the other MAS members may need dumped; a test is made to see if all JES2 members are at the same level.

To ensure all data is captured, the tailored SVC DUMP exit is exploited by JES2. This service allows JES2 to identify a dynamic exit which gets control whenever a SDUMP is requested (on the member taking the dump). The exit, supplied with

Release 10, detects that the JES2 address space is being dumped and adds the required dataspaces to the dump.

## 12.5 Purging jobs affecting JES2 restarts

On occasion, JES2 restart errors are caused by a job that is in a state that warm start and purge processing cannot handle. The only way out of these situations has been either a cold start or to ZAP the checkpoint data set to eliminate the offending control block. Typical errors that might occur are:

- A job causes an ABEND while processing
- A job cannot be purged from the system
- A warm start of the system fails

In Release 10, a new service routine has been introduced to remove an offending job from the system. This service locates the offending JQE by scanning the JQE CTENT for a matching job number and job name (optionally, a job key). When found, its JQEs are removed from any chains and placed on the free queue. The JQE is removed from all chains and added to the free queue. A new SYMREC is issued to report on the success of the operation.

A job queue validation is done at the start of the routine and repeated at the end. If the first validation is successful, then the second must also be successful; otherwise, JES2 is terminated.

### 12.5.1 New options to remove a job

The service routine gets control as the result of attempting to remove a job in one of two ways:

1. A new initialization statement (**ZAPJOB**)
2. A new operator command (**$ZAPJOB**).

At least one of the following parameters for the INIT statement and operator command must be specified:

- JOBID
- JQEOFF
- JQEINDEX
- JOBKEY
- JOBNAME

**Note:** Specify as many parameters as are known, to ensure the correct job is selected to be removed. JQEOFF, JQEINDEX, and JOBKEY can be obtained from a dump.

The ZAPJOB implementation is intended only as an option to prevent cold starts and is intended to operate on jobs that have incorrect data. ZAPJOB cannot verify whether the job is not busy, since the job's data may be incorrect and makes the specified job go away even if there are PCEs using the job. Executing ZAPJOB on the wrong job may result in other problems (and even abends), so use caution when running this function.

#### 12.5.1.1 $ZAPJOB operator command

A $ZAPJOB statement can be placed in the initialization deck. It is executed as an operator command at the end of JES2 initialization.

**Note:** This is a different implementation than using a ZAPJOB initialization statement, and extreme caution must be taken to *remove the command* before using the initialization statement again.

Examples of the $ZAPJOB command are as follows:

```
$ZAPJOB,JOBNAME=test1,JOBID=job8
$ZAPJOB,JOBID=STC30127
```

#### 12.5.1.2 ZAPJOB initialization statement

A ZAPJOB initialization statement removes a job before verifying the job queue.

```
ZAPJOB,JOBNAME=test1,JOBID=job8
ZAPJOB,JOBID=STC30127
```

## 12.6 JES2 Release 10 Migration considerations

Note the following when implementing OS/390 Version 2 Release 10:

1. The JES2 checkpoint must be in Release 4 mode, meaning a $ACTIVATE must be done prior to migration.
2. A COLD start is required from Release 3 and prior releases.
3. Migrate to Release 4 or higher *first,* to avoid a cold start.
4. MAS coexistence is available with Release 5, Release 7, and Release 8.
5. APAR OW42299 is needed on downlevel systems
6. A Release 4 version of APAR OW42299 is available but not officially supported in a MAS

### 12.6.1 JES2 levels supported by OS/390

Many customers migrate to a new level of JES2 at the same time they migrate to a new level of OS/390. IBM recommends that you migrate to the JES2 that comes comprehensively-tested with OS/390 at the same time you migrate to OS/390, or as soon as possible thereafter. In this way you benefit directly from the new function in the OS/390 level of JES2 and enable other elements or features to benefit from those levels.

However, IBM recognizes that under some circumstances this might not be practical. For this reason, OS/390 supports certain prior levels of JES2, allowing you to stage your migration to JES2

For OS/390 R9 and earlier releases, any level of JES2 back to SP 5.1.1 is supported.

For OS/390 Release 10 and continuing thereafter, the JES2 levels supported by a given OS/390 release are the same as the JES2 levels that can coexist in the same multi-access spool with the JES2 delivered in that OS/390 release.

That is, a given JES2 release and a given OS/390 release (running on a single system, or individual systems participating in a multisystem configuration) is a supported combination, if the given JES2 release can coexist with the JES2 delivered with that OS/390 release.

As an example:

- JES2 FMID HJE7703 was shipped in OS/390 Release 10. This level of JES2 coexists with HJE7703, HJE6608, HJE6607, and HJE6605. These levels of JES2 are supported with OS/390 Release 10.

Table 6 lists the support.

*Table 6.* JES2 levels supported by OS/390

| **OS/390 Release** | **Supported JES2 FMIDs** |
|---|---|
| Release 10 | HJE7703, HJE6608, HJE6607, HJE6605 |
| Release 9 | HJE6608, HJE6607, HJE6605, **HJE6604, HJE6603, HJE6601, HJE5520, HJE5510** |
| Release 8 | HJE6608, HJE6607, HJE6605, **HJE6604, HJE6603, HJE6601, HJE5520, HJE5510** |
| Release 7 | HJE6607, HJE6605, **HJE6604, HJE6603, HJE6601, HJE5520, HJE5510** |
| Release 6 | HJE6605, **HJE6604, HJE6603, HJE6601, HJE5520, HJE5510** |
| Release 5 | HJE6605, **HJE6604, HJE6603, HJE6601, HJE5520, HJE5510** |
| Release 4 | HJE6604, HJE6603, HJE6601, HJE5520, HJE5510 |
| Release 3 | HJE6603, HJE6601, HJE5520, HJE5510 |
| Release 2 | HJE6601, HJE5520, HJE5510 |
| Release 1 | HJE6601, HJE5520, HJE5510 |
| **JES2 Levels Supported by OS/390** | Discontinuance of service support has been announced effective:<br>January 31, 2001 for HJE5510 and HJE6601<br>March 31, 2001 for HJE5520, HJE6603, and HJE6604<br>These FMIDs are highlighted in boldface.<br><br>HJE6605 is the same JES2 FMID that came with OS/390 R6. As a result, it continues to be service supported for as long as OS/390 R6 is service supported, even though discontinuance of service support for OS/390 R5 was announced for March 31, 2001. |

### 12.6.2 JES2 coexistence

*JES2 coexistence* refers to two or more different releases of JES2 sharing the same spool.

OS/390 supports the coexistence of up to four consecutive releases. While the four-release coexistence policy applies to JES2, the fact that a JES2 installation can be staged has been taken into account in determining which are the four consecutive releases that can coexist. If a JES2 release is functionally equivalent to its predecessor (that is, its FMID did not change), then from a coexistence standpoint, it is considered the same release.

Table 7 lists the JES2 levels that can coexist:

*Table 7. JES2 level coexistence*

| **Highest JES2 FMID running in a multisystem configuration** | **JES2 FMIDs that can coexist with the JES2 FMID identified in Column 1** |
|---|---|
| HJE7703 | HJE7703, HJE6608, HJE6607, HJE6605 |
| HJE6608 | HJE6608, HJE6607, HJE6605, **HJE6604, HJE5520, HJE5501** |
| HJE6607 | HJE6607, HJE6605, **HJE6604, HJE6603, HJE6601, HJE5520, HJE5510** |
| HJE6605 | HJE6605, **HJE6604, HJE6603, HJE6601, HJE5520, HJE5510** |
| HJE6604 | HJE6604, HJE6603, HJE6601, HJE5520, HJE5510 |
| HJE6603 | HJE6603, HJE6601, HJE5520, HJE5510 |
| HJE6601 | HJE6601, HJE5520, HJE5510 |
| HJE5520 | HJE5520, HJE5510 |
| HJE5510 | HJE5510 |
| JES2 levels that can coexist | Discontinuance of service support has been announced effective:<br>January 31, 2001 for HJE5510 and HJE6601<br>March 31, 2001 for HJE5520, HJE6603, and HJE6604<br>These FMIDs are highlighted in boldface.<br><br>HJE6605 is the same JES2 FMID that came with OS/390 R6. As a result, it continues to be service supported for as long as OS/390 R6 is service supported, even though discontinuance of service support for OS/390 R5 was announced for March 31, 2001. |

# Chapter 13. OS/390 installation overview

This chapter gives you an overview of packaging changes and installation improvements in OS/390 Release 10. It covers the following items:

- Changed base elements
- New and changed optional features
- Removed elements and features
- Installation changes
- System requirements
- Installation improvements

## 13.1 Changed base elements

Figure 95 lists all OS/390 Release 10 base elements. Those elements shown in *italics underscore* are changed in Release 10, and those shown in **bold** are new and exclusive in OS/390.

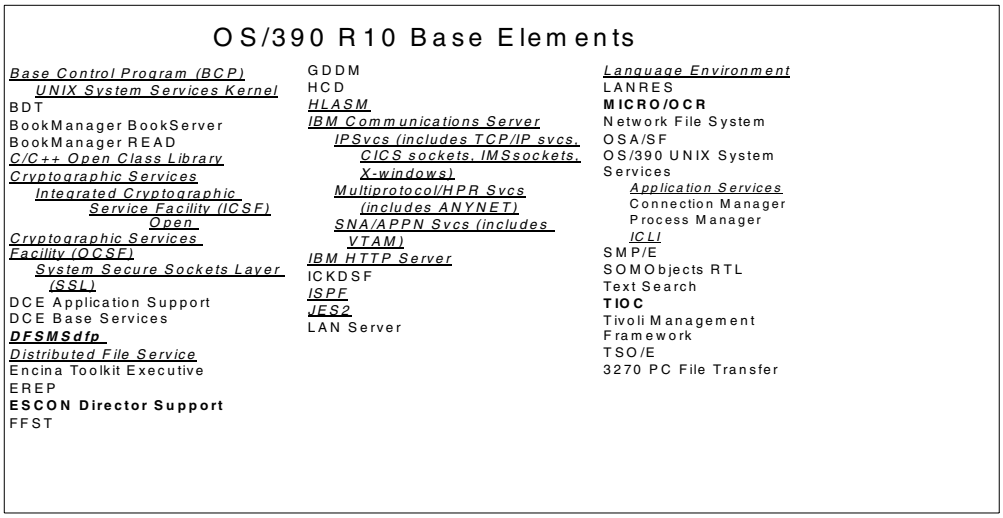



*Figure 95. OS/390 Release 10 base elements*

## 13.2 New and changed optional features

Figure 96 on page 188 lists all OS/390 Release 10 optional features. Highlighting of elements has the following meaning:

- The $ identifies optional priced features.
- *Italics underscore* identifies features that are changed in Release 10.
- Plain underscore identifies new features in Release 10.
- **Bold** identifies new exclusive elements.

Combinations are allowed.

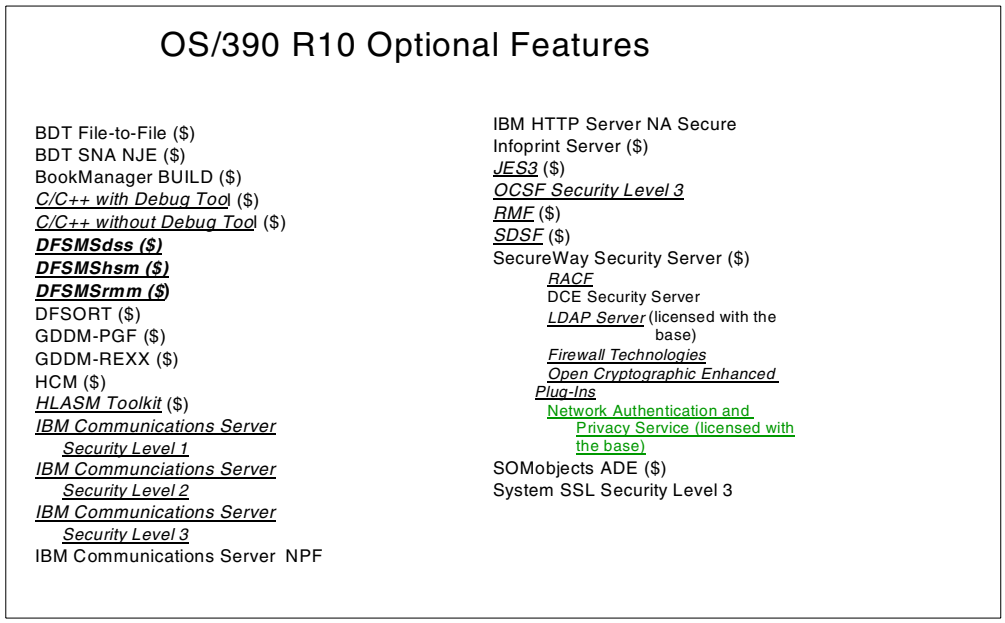
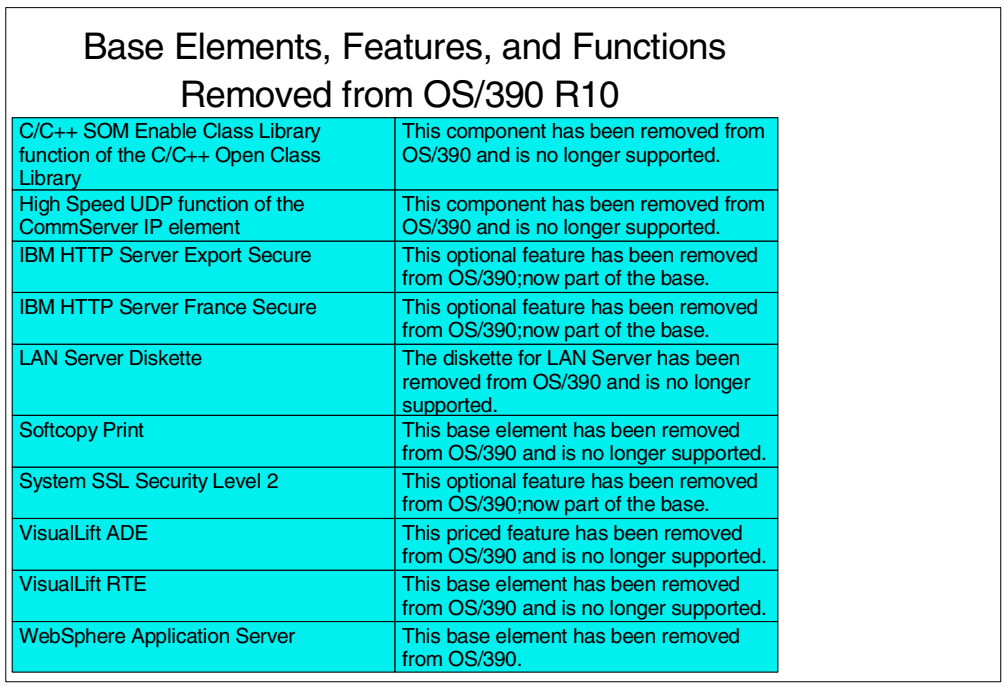



*Figure 96. OS/390 Release 10 optional features*

## 13.3 Removed elements and features

Figure 97 lists the elements of OS/390 that are removed from OS/390 Release 10.



Base Elements, Features, and Functions Removed from OS/390 R10

| | |
|---|---|
| C/C++ SOM Enable Class Library function of the C/C++ Open Class Library | This component has been removed from OS/390 and is no longer supported. |
| High Speed UDP function of the CommServer IP element | This component has been removed from OS/390 and is no longer supported. |
| IBM HTTP Server Export Secure | This optional feature has been removed from OS/390;now part of the base. |
| IBM HTTP Server France Secure | This optional feature has been removed from OS/390;now part of the base. |
| LAN Server Diskette | The diskette for LAN Server has been removed from OS/390 and is no longer supported. |
| Softcopy Print | This base element has been removed from OS/390 and is no longer supported. |
| System SSL Security Level 2 | This optional feature has been removed from OS/390;now part of the base. |
| VisualLift ADE | This priced feature has been removed from OS/390 and is no longer supported. |
| VisualLift RTE | This base element has been removed from OS/390 and is no longer supported. |
| WebSphere Application Server | This base element has been removed from OS/390. |

*Figure 97. Elements and features removed from OS/390 Release 10*

For the following functions, OS/390 Release 9 was the last release where these functions were shipped as part of OS/390. We recommend that you make transitions from those functions to newer ones.

These functions are:

- LAN Server

  This component is no longer available because the prerequisite software on OS/2 (on diskette) is also no longer available. The LAN Server FMIDs are still shipped in Release 10 for those customers doing a migration off to an alternative solution. Alternatives are described in the IBM Redbook *File Server Consolidation on S/390*, SG24-5330, and also in this book in Chapter 6, "DCE DFS and SMB support" on page 91.

- High speed UDP function of the Communications Server IP element

  Starting with Release 10 High speed UDP can no longer be configured. You can configure the TCP/IP stack to obtain equivalent functions.

- Visual Lift Runtime environment (RTE)
- Visual Lift
- Softcopy Print

  This component provided printing functions for BookManager books on AFP printers. An alternative is to print the PDF versions of the OS/390 books that are shipped on CD with the OS/390 manuals.

- WebSphere Application Server for OS/390

  This element is removed from OS/390 Release 10. If you plan to use the new Release, you have to order WebSphere Application Server Release 3.02 (5655-A98) as a separate program product.

- Cumulative service tapes (CUM)

  Starting with Release 10 of OS/390, these CUM tapes are no longer shipped due to their limited worldwide availability and downlevel contents.

## 13.4 System requirements

There are some hardware and software requirements for the installation of OS/390, which we describe in the following sections.

### 13.4.1 Driving system hardware requirements

The hardware requirements are the same regardless of whether you choose to install via ServerPac, SystemPac or CBPDO. In addition to required DASD space required to install the software, you need the following:

- A CPU that can run at least MVS/ESA SP V5.1 or any level of OS/390
- A TSO terminal, color preferred for the ServerPac dialogs
- A tape device that can be 3480, 3490 (ServerPac, SystemPac only), 3590 (SystemPac only), 6250, or 4mm (for PC server system/390)
- At least 94 MB of central storage for ServerPac and SystemPac dump-by-data-set installations

### 13.4.2 Driving system software requirements

In the case that you install OS/390 Release 10 via ServerPac or dump-by-data-set SystemPac, there are two alternatives to do this. Depending on

the alternative you choose, there are different software requirements for the driving system.

#### 13.4.2.1 Installation with HFS unload from the driving system

This is the recommended way to proceed, and you must ensure the following is in place:

- A minimum of OS/390 Release 4 with PTFs as the driving system
- An activated OMVS address space, which implies that SMS is active and security is set up
- An HFS containing the OS/390 UNIX pax utility
- A user ID that is a superuser (UID=0) for installation
- A user ID that has read access to RACF facility classes BPX.FILEATTR.APF and BPX.FILEATTR.PROGCTL (or BPX.FILEATTR.*)

You should unload both HFS and non-HFS data sets from your driving system. Your unload and IPL scenario is then:

1. Unload non-HFS data sets from your driving system.
2. Unload HFS data sets from your driving system.
3. IPL your target system.

The advantage of this scenario is that you have UNIX access and IP connectivity for your first IPL.

#### 13.4.2.2 Installation with HFS unload from the target system

In this type of installation, you have to have *either* of the following:

- A minimum of MVS/ESA V5.1 and DFSMS/MVS V1R4 with PTFs installed as your driving system
- Any level of OS/390 and DFSMS/MVS V1R4 as a minimum with PTFs installed as your driving system

In this case, your unload and IPL scenario looks like this:

1. Unload non-HFS data sets from your driving system.
2. IPL the target system.
3. Unload HFS data sets from the IPLed target system, which then enables a UNIX environment.

### 13.4.3 Target system hardware requirements

If you plan to install OS/390 Release 10 on your processor, you have to ensure that you have installed, at minimum, *one* of the following:

- An S/390 Parallel Enterprise Server (except R1 models, which means 9672-R1x)
- An S/390 Mulitiprise (all models)
- An S/390 Application StarterPak 3000 (all models)
- A PC Server System/390 or an RS/6000 with System/390 Server-on-Board model
- An S/390 Integrated Server

In Table 8 you can see how much DASD space you need for the OS/390 libraries. This includes:

- All base elements
- All optional features that support dynamic enablement
- All other optional features you ordered

It does *not* include:

- Any non-OS/390 IBM product
- Any non-IBM product
- Any user customization

*Table 8. OS/390 Release 10 DASD space requirements*

| Type of library | Space in cylinders of 3390 device |
|---|---|
| Target libraries | 4,274 |
| Distribution libraries | 4,672 |
| HFS files | 1,005 (root: 955, /etc: 50) |
| SMP/E libraries | size depends on deliverable |
| SMPLTS | 973 |

### 13.4.4 Target system software requirements

If you plan to install OS/390 Release 10 on your processor, you also have to ensure that all your IBM non-OS/390 products are at a current release and PTF level. You can find more information in *OS/390 Planning for Installation*, GC28-1726, level -09.

If you have non-IBM products installed, you should also get in contact with your software vendor or have a look, on the Internet at *Vendor product compatibility with OS/390,* which is at:

```
http://www.s390.ibm.com/os390/os390vend.html
```

From that page, you can proceed to the vendor's home pages, where you can find the following information regarding their products:

- Are they OS/390 Release 10 compatible?
- Are they SMP/E-installable?
- Are they able to install in a separate SMP/E zone from OS/390?
- Do they provide installation verification procedures (IVP)?
- Do they require the recustomization of OS/390 elements?
- Are they currently installed in customer production environments?
- Do they supply books online?

If you plan to use the new function of SDSF in a sysplex environment, you also have to install and customize MQ Series V2R1. This is described more detailed in Chapter 4, "System Display and Search Facility (SDSF)" on page 41.

### 13.4.5 OS/390 Release 10 coexistence requirements

Coexistence support for OS/390 Release 10 exists for the following lower-level OS/390 releases:

- OS/390 R6

  This is a deviation from the normal four-release coexistence policy, but is still in use because the Y2K actions took many customers only up to Release 6.

- OS/390 R7
- OS/390 R8
- OS/390 R9

For further details on these topics, a list of APARs and PTFs for the OS/390 elements, and instructions on how to prepare for fallback service to older OS/390 releases when you install OS/390 Release 10, refer to OS/390 *Planning for Installation*, GC28-1726 (level-09).

## 13.5 Installation improvements

One major enhancement in OS/390 Release 10 is that you now have the option of ordering your SystemPac and the selective follow-on services on 3590 media. This medium is available for the dump-by-data-set format. This reduces the number of tapes shipped to your location dramatically.

The number of solution developer (formerly known as independent software vendor, or ISV) products that can be integrated into a SystemPac has also increased. You can find more information at the following Web site:

```
http://www.can.ibm.com/custompac
```

### 13.5.1 Web-based wizards

Web-based wizards provide a simplified, step-by-step approach to complete a number of OS/390 tasks, such as planning, installation, and configuration.

When the wizard asks you for information, you can insert your own environment-related information. The wizard then takes you to the next step.

These wizards are based on Internet pages. They do not perform the tasks on your system, but they generate things like tailored instructions, job streams, policies, or parmlib members that you can transfer to your OS/390 environment.

New wizards in OS/390 Release 10 are:

- OS/390 Installation SDSF Configuration Assistant

  This wizard supports a Parallel Sysplex environment. It collects user input and generates results like JCL or RACF commands that you can put into MVS data sets via cut and paste functions on your workstation.

- Enhanced OS/390 UNIX Configuration Assistant

  Using this wizard, you can build BPXPRMxx parmlib numbers with system processing parameters, and do initial RACF security setup for OS/390 UNIX and TCP/IP definitions.

You should also investigate the other Web-based wizards that are available:

- OS/390 Installation Planning Assistant
- S/390 ServerPac Ordering assistant
- S/390 Parallel Sysplex Configuration Assistant
- OS/390 Planning Assistant for e-business

You can find these wizards at the following Web page:

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

# Appendix A. MQ definitions for MQSeries Queue Managers

The following figures show an example of MQ definitions required to establish the communications between two MQSeries Queue Managers:

```
DEFINE NOREPLACE                -
 CHANNEL('TO.MQSA')             -
 CHLTYPE(CLUSSDR)               -
 CLUSTER('SDSF')                -
 CLUSNL(' ')                    -
 TRPTYPE(TCP)                   -
 CONNAME('WTSC65(1414)')        -
 DESCR(' ')                     -
 DISCINT(6000)                  -
 SHORTRTY(10)                   -
 SHORTTMR(60)                   -
 LONGRTY(999999999)             -
 LONGTMR(1200)                  -
 SCYEXIT(' ')                   -
 SCYDATA(' ')                   -
 MSGEXIT(' ')                   -
 MSGDATA(' ')                   -
 SENDEXIT(' ')                  -
 SENDDATA(' ')                  -
 RCVEXIT(' ')                   -
 RCVDATA(' ')                   -
 SEQWRAP(999999999)             -
 MCAUSER(' ')                   -
 CONVERT(NO)                    -
 BATCHINT(0)                    -
 BATCHSZ(50)                    -
 MAXMSGL(4194304)               -
 HBINT(300)                     -
 NPMSPEED(FAST)
```

*Figure 98. MQ Manager definition on System SC64 - Part 1*

```
DEFINE NOREPLACE                -
 CHANNEL('TO.MQSB')             -
 CHLTYPE(CLUSRCVR)              -
 CLUSTER('SDSF')                -
 CLUSNL(' ')                    -
 TRPTYPE(TCP)                   -
 CONNAME('wtsc64(1414)')        -
 DESCR(' ')                     -
 DISCINT(6000)                  -
 SHORTRTY(10)                   -
 SHORTTMR(60)                   -
 LONGRTY(999999999)             -
 LONGTMR(1200)                  -
 SCYEXIT(' ')                   -
 SCYDATA(' ')                   -
 MSGEXIT(' ')                   -
 MSGDATA(' ')                   -
 SENDEXIT(' ')                  -
 SENDDATA(' ')                  -
 RCVEXIT(' ')                   -
 RCVDATA(' ')                   -
 PUTAUT(DEF)                    -
 SEQWRAP(999999999)             -
 MCAUSER(' ')                   -
 CONVERT(NO)                    -
 BATCHINT(0)                    -
 NETPRTY(0)                     -
 MCATYPE(THREAD)                -
 BATCHSZ(50)                    -
 MAXMSGL(4194304)               -
 HBINT(300)                     -
 NPMSPEED(FAST)
```

*Figure 99. MQ Manager definition on System SC64 - Part 2*

```
DEFINE NOREPLACE                -
 QALIAS('ISF.CLIENT.SDSF.SC64.REQUESTQ') -
 CLUSTER('SDSF')                -
 CLUSNL(' ')                    -
 DESCR('SDSF Server Request Queue Alias')-
 PUT(ENABLED)                   -
 DEFPRTY(5)                     -
 DEFPSIST(NO)                   -
 GET(DISABLED)                  -
 TARGQ('ISF.SERVER.SDSF.SC64.REQUESTQ')  -
 DEFBIND(OPEN)
DEFINE NOREPLACE                -
 QALIAS('SYSTEM.DEFAULT.ALIAS.QUEUE')    -
 CLUSTER(' ')                   -
 CLUSNL(' ')                    -
 DESCR(' ')                     -
 PUT(ENABLED)                   -
 DEFPRTY(0)                     -
 DEFPSIST(NO)                   -
 GET(ENABLED)                   -
 TARGQ(' ')                     -
 DEFBIND(OPEN)
```

*Figure 100. MQ Manager definition on System SC64 - Part 3*

```
DEFINE NOREPLACE                -
 CHANNEL('TO.MQSA')             -
 CHLTYPE(CLUSRCVR)              -
 CLUSTER('SDSF')                -
 CLUSNL(' ')                    -
 TRPTYPE(TCP)                   -
 CONNAME('WTSC65(1414)')        -
 DESCR(' ')                     -
 DISCINT(6000)                  -
 SHORTRTY(10)                   -
 SHORTTMR(60)                   -
 LONGRTY(999999999)             -
 LONGTMR(1200)                  -
 SCYEXIT(' ')                   -
 SCYDATA(' ')                   -
 MSGEXIT(' ')                   -
 MSGDATA(' ')                   -
 SENDEXIT(' ')                  -
 SENDDATA(' ')                  -
 RCVEXIT(' ')                   -
 RCVDATA(' ')                   -
 PUTAUT(DEF)                    -
 SEQWRAP(999999999)             -
 MCAUSER(' ')                   -
 CONVERT(NO)                    -
 BATCHINT(0)                    -
 NETPRTY(0)                     -
 MCATYPE(THREAD)                -
 BATCHSZ(50)                    -
 MAXMSGL(4194304)               -
 HBINT(300)                     -
 NPMSPEED(FAST)
```

*Figure 101. MQ Manager definition on System SC65 - Part 1*

```
DEFINE NOREPLACE                -
 CHANNEL('TO.MQSB')             -
 CHLTYPE(CLUSSDR)               -
 CLUSTER('SDSF')                -
 CLUSNL(' ')                    -
 TRPTYPE(TCP)                   -
 CONNAME('WTSC64(1414)')        -
 DESCR(' ')                     -
 DISCINT(6000)                  -
 SHORTRTY(10)                   -
 SHORTTMR(60)                   -
 LONGRTY(999999999)             -
 LONGTMR(1200)                  -
 SCYEXIT(' ')                   -
 SCYDATA(' ')                   -
 MSGEXIT(' ')                   -
 MSGDATA(' ')                   -
 SENDEXIT(' ')                  -
 SENDDATA(' ')                  -
 RCVEXIT(' ')                   -
 RCVDATA(' ')                   -
 SEQWRAP(999999999)             -
 MCAUSER(' ')                   -
 CONVERT(NO)                    -
 BATCHINT(0)                    -
 BATCHSZ(50)                    -
 MAXMSGL(4194304)               -
 HBINT(300)                     -
 NPMSPEED(FAST)
```

*Figure 102. MQ Manager definition on System SC65 - Part 2*

```
DEFINE NOREPLACE                -
 QALIAS('ISF.CLIENT.SDSF.SC65.REQUESTQ')  -
 CLUSTER('SDSF')                -
 CLUSNL(' ')                    -
 DESCR('SDSF Server Request Queue Alias')  -
 PUT(ENABLED)                   -
 DEFPRTY(5)                     -
 DEFPSIST(NO)                   -
 GET(DISABLED)                  -
 TARGQ('ISF.SERVER.SDSF.SC65.REQUESTQ')   -
 DEFBIND(OPEN)
DEFINE NOREPLACE                -
 QALIAS('ISF.CLIENT.SDSF65.SC65.REQUESTQ') -
 CLUSTER(' ')                   -
 CLUSNL(' ')                    -
 DESCR('SDSF Server Request Queue Alias')  -
 PUT(ENABLED)                   -
 DEFPRTY(5)                     -
 DEFPSIST(NO)                   -
 GET(DISABLED)                  -
 TARGQ('ISF.SERVER.SDSF64.SC64.REQUESTQ')  -
 DEFBIND(OPEN)
 DEFINE NOREPLACE               -
  QALIAS('SYSTEM.DEFAULT.ALIAS.QUEUE')    -
  CLUSTER(' ')                  -
  CLUSNL(' ')                   -
  DESCR(' ')                    -
  PUT(ENABLED)                  -
  DEFPRTY(0)                    -
  DEFPSIST(NO)                  -
  GET(ENABLED)                  -
  TARGQ(' ')                    -
  DEFBIND(OPEN)
```

*Figure 103. MQ Manager definition on System SC65 - Part 3*

# Appendix B. Special notices

This publication is intended to help customer Systems Programmers and IBM support personnel to understand the changes to OS/390 in Version 2 Release 10. The information in this publication is not intended as the specification of any programming interfaces that are provided by the OS/390 Release 10 product. See the PUBLICATIONS section of the IBM Programming Announcement for the OS/390 Version 2 Release 10 product for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

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

CICS
DB2
DFSMShsm
Domino
e (logo)® 
FICON
IBM ®
Infoprint
Language Environment
Lotus
MQSeries
Notes
OS/2
OS/390
Parallel Sysplex
RACF
Redbooks
Redbooks Logo 
RMF
S/390
SecureWay

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

# Appendix C. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## C.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 203.

- *IBM Communications Server for OS/390 TCP/IP 2000 Update Technical Presentation Guide*, SG24-6162
- *IBM Communications Server for OS/390 V2R10 TCP/IP Implementation Guide Volume 1: Configuration and Routing*, SG24-5227 (-02 version)
- *DFSMShsm Primer*, SG24-5272 (-01 version)
- *DFSMS Release 10 Technical Update*, SG24-6120
- *IBM Communications Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 2: UNIX Applications*, SG24-5228 (-02 version)

## C.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at http://www.redbooks.ibm.com/ for information about all the CD-ROMs offered, updates and formats.

| CD-ROM Title | Collection Kit Number |
|---|---|
| System/390 Redbooks Collection | SK2T-2177 |
| Networking and Systems Management Redbooks Collection | SK2T-6022 |
| Transaction Processing and Data Management Redbooks Collection | SK2T-8038 |
| Lotus Redbooks Collection | SK2T-8039 |
| Tivoli Redbooks Collection | SK2T-8044 |
| AS/400 Redbooks Collection | SK2T-2849 |
| Netfinity Hardware and Software Redbooks Collection | SK2T-8046 |
| RS/6000 Redbooks Collection (BkMgr Format) | SK2T-8040 |
| RS/6000 Redbooks Collection (PDF Format) | SK2T-8043 |
| Application Development Redbooks Collection | SK2T-8037 |
| IBM Enterprise Storage and Systems Management Solutions | SK3T-3694 |

# How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

| | **e-mail address** |
|---|---|
| In United States or Canada | pubscan@us.ibm.com |
| Outside North America | Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl |

- **Telephone Orders**

| | |
|---|---|
| United States (toll free) | 1-800-879-2755 |
| Canada (toll free) | 1-800-IBM-4YOU |
| Outside North America | Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl |

- **Fax Orders**

| | |
|---|---|
| United States (toll free) | 1-800-445-9269 |
| Canada | 1-403-267-4455 |
| Outside North America | Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl |

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

**IBM Intranet for Employees**

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

## IBM Redbooks fax order form

**Please send me the following:**

| Title | Order Number | Quantity |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| | | |
|---|---|---|
| First name | Last name | |
| Company | | |
| Address | | |
| City | Postal code | Country |
| Telephone number | Telefax number | VAT number |

☐ Invoice to customer number

☐ Credit card number

| | | |
|---|---|---|
| Credit card expiration date | Card issued to | Signature |

**We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.**

# Index

## Symbols


$ZAPJOB 183
&BLKSIZE 76
&MSDEST 87
&MSPARM 87
&MSPOLICY 87
&MSPOOL 87
&UNIT 78
.EDGJLOPC 86


## Numerics


3480 189
3490 189
3494 89
3590 189
4mm 189
6250 189
64-bit large real storage 19


## A


ABARS 76
Access Method Services Changes 23
  EXCP 23
  EXCPVR 23
ADDVOLUME 88
aggregates 93
Application StarterPak 3000 190
ARCHBACK 79
Architecture
  ARCHLVL 16
  Dual architecture support 30
  ESA/390 28
  ESAME 28
ARCHLVL 16
ARCINBACK 79
ASM 13
audit support 89
Auxiliary Storage Manager 13


## B


BACKDS 79
BPX.FILEATTR.APF 190
BPX.FILEATTR.PROGCTL 190
BSAM 76


## C


C/C++ 132
  Enhancements 132
CBPDO 189
CHANGEVOLUME 88
Channel program changes 21
coexistence 192
Communication Server 133
  Security
    network access 142
    port access 143
    Stack access 144
Communications Server 133
  Performance 137
    Sysplex workload distribution 139
  Security
    Network traffic access 140
  Telnet 133
Concurrent copy 79
CONTBDY 24
coupling facility 35
CQS 39
Cross System Extended Services (XES) 38
CUM tapes 189


## D


DADSM rename 77
DASD log data sets 35
DASD-only log streams 36
DCE 91
DCE LFS 91
DELETEVOLUME 88
DFS 91, 96, 97
DFS clients 91
DFS server 91
dfscntl 96
dfskern 92, 96
DFSMS 71, 72, 77
DFSMSdfp 71
DFSMSdss 71
DFSMShsm 71, 78, 79, 81, 83, 84
DFSMShsm ABARS 76
DFSMSrmm 71, 85, 87, 88, 89
DFSORT 76
Distributed File Service 91
DLL 93
Driving system 189
Dual architecture support 30
Dynamic Link Library 93
Dynamic Virtual IP Addressing (VIPA) 138


## E


ECS 78
EDGCVRSX 87
EDGJHKPA 86
EDGRMMxx 85
EDGUTIL 88
EDGUX100 87
enhanced catalog sharing (ECS) 78
enlarged file size 93
envar 97
ESAME 30, 31
EXCPVR 23
export 96
Extended addressability 72

## F

facility classes 190
FDSMSrmm 86, 87
File Transfer Protocol (FTP) 134
  functionality 136
  Security 134
  user exit 137
filesets 93

## G

General Purpose Registers 31
GETMAIN 24
  CONTBDY 24
  STARTBDY 24

## H

HBACKDS 79
HOSTMODE 84

## I

IDAW 21
IEANUC0x 16
IEANUC1x 16
IEANUC2x 16
IEBGENER 76
IMS/ESA Common Queue Services (CQS) 39
installation 187, 189
Integrated Server 190
Interactive Problem Control System (IPCS) 26
IXCL1DSU 36
IXGBRWSE 37, 39, 40

## J

JES2 177
  Migration 184
  Multi-system dumps 182
  restart
    job purging 183
  SPOOL 178
  volume fencing 179

## L

LAN Server 189
Language Environment 127
LANMAN 98
Large tape block sizes 76
large tape block sizes 76
latching performance 37
LE 127
  Downward compatibility 127
LFS 93
  blocksize 93
  fragment size 93
library manager 89
Local File System 91
log stream 35
Logger 36, 37
LOGR couple data set 35

## M

Macros
  GETMAIN 24
    STORAGE 24
  STORAGE 24
Magstar 76
migration 81
Mulitiprise 190
MULTIBLOCK keyword 37
Multi-layering 74
MVS/ESA SP V5.1 189
MVS/ESA V5.1 190
MVS/ESA Version 5.2 35

## N

NetBIOS 92
NT LM 0.12 94
Nucleus 16
  IEANUC0x 16
  IEANUC1x 16
  IEANUC2x 16

## O

OCSF 98
Online 104
OPC 86
Open Cryptographic Services Facility (OCSF) 98
OPERLOG 36
OPTION MOVEBY 85
OPTION RETAINBY 85
OS/2 WARP Version 4 92
OS/390 91, 92, 187, 192
  base elements 187
  DASD space 191
  libraries 191
  optional features 187
  Removed elements 188
  UID=0 190
OS/390 UNIX 190

## P

Parallel Enterprise Server 190
Partial release 72
pax 190
PC Server System/390 190
PC server system/390 189
PRIMARY 84

## Q

QSAM 76

## R

RACF 77, 79, 190
Real Storage Manager (RSM) 14

Rebuild ENQ contention reduction 37
Record File System (RFS) 91, 94
RFS 91, 94, 96
rfstab 96
root directory 96
RSM 14
  Changes in ESAME mode 20

## S

S/390 190
SDSF 41
  Sysplex System Management 55
    MQSeries 56
SDSF Configuration Assistant 192
SEARCHVOLUME 88
Server Message Block 91
ServerPac 189
SETSYS MIGRATIONCLEANUPDAYS 83
SETSYS TAPEMIGRATION 81
Slip Trap 27
SMB 91, 94
SMB server 92
SMB_CREATE_ANDX 94
SRM 14
  Changes in ESAME mode 20
stacked volumes 88
STARTBDY 24
STGADMIN.ADR.DUMP.CNCURRNT 79
STGADMIN.IGG.DPRN.dsn 77
STORAGE
  CONTBDY 24
  STARTBDY 24
Storage 9, 19
  components 11
    expanded 12
    main 11
    real 11
    virtual 12
  frame 14
  overview 10
  page 13
storage protection 163
STRIPE-COUNT 75
superuser 190
SYS1.LOGREC 36
SYS1.SAMPLIB 86, 87
System Display and Search Facility 41
System Logger 35
System requirements 189
System Resource Manager (SRM) 14
System/390 Server-on-Board 190
SystemPac 189
SYSZTIOT 83

## T

tape block IDs 89
tape library 89
Target system 190
TCDB 89
TCP/IP 137, 138
  Dynamic Virtual IP Addressing (VIPA) 138
  Performance
    Service policy 137
Telnet 133
  Autologon support 133
  Keepopen support 133
  Mapping support 133
  Network Qualified Name (NQN) 133
  Secure Sockets Layer (SSL) 133
Tivoli OPC 86

## U

UDP 189
UNIT=AFF 77
UNIX Configuration Assistant 193

## V

VFS 93
VIPA 138
Virtual File system 93
Virtual Storage Manager 13
VMA 76
Volume mount analyzer (VMA) 76
VSAM 104
VSAM striping 72, 74, 75
  DFSMSdss support 75
VSM 13
  Changes in ESAME mode 24

## W

WebSphere Application Server for OS/390 189
Windows 3.11 92
Windows 95 92
Windows 98 92
Windows NT 4.0 92
wizards 192, 193

## X

XES 38
XES auto alter 37
XPLINK 128
  Compiling and Linking 129
  Debugging 130

## Z

z/Architecture 10, 17
ZAPJOB 183

# IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 845 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

| | |
|---|---|
| **Document Number**<br>**Redbook Title** | SG24-5976-00<br>OS/390 Version 2 Release 10 Implementation |
| **Review** | |
| **What other subjects would you like to see IBM Redbooks address?** | |
| **Please rate your overall satisfaction:** | O Very Good O Good O Average O Poor |
| **Please identify yourself as belonging to one of the following groups:** | O Customer<br>O Business Partner<br>O Solution Developer<br>O IBM, Lotus or Tivoli Employee<br>O None of the above |
| **Your email address:**<br>The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities. | O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction. |
| **Questions about IBM's privacy policy?** | The following link explains how we protect your personal information.<br>**ibm.com**/privacy/yourprivacy/ |

IBM Redbooks

# OS/390 Version 2 Release 10 Implementation

(0.2”spine)
0.17”<->0.473”
90<->249 pages

# OS/390 Version 2 Release 10 Implementation

Redbooks

**64-bit Real Storage Support, Language Environment and C/C++ Support**

**DFSMS, WLM Enhancements, Telnet, FTP**

**JES2, SDSF, XES Auto Alter, System Logger, RMF**

This IBM Redbook contains information related to many of the changes made in OS/390 Version 2 Release 10. You can use it to help you install, tailor and configure Release 10.

This redbook gives a broad understanding of a new architecture for 64-bit real storage addressing. Other topics discussed are:

- Changes to the System Logger
- New SDSF enhancements
- Enhancements to DFSMS
- Native Windows SMB support for accessing files and printers
- RMF enhancements
- Language Environment and C++ enhancements
- Changes to VSAM
- Communication Server enhancements
- Support for automatic tuning of coupling facility structures
- Workload Manager enhancements to support migrations to goal mode
- JES2 Release 10 enhancements

**INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

**BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by IBM's International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**
**ibm.com**/redbooks

SG24-5976-00 ISBN 073841865X