z/OS

zSeries Platform Test Report

Version 1 Release 6

SA22-7997-00

z/OS

zSeries Platform Test Report

Version 1 Release 6

SA22-7997-00

Note!
Before using this information and the products it supports, be sure to read the general information under “Notices” on page 205.

First Edition, September 2004

This is a major revision of SA22-7663-11.

This edition applies to Parallel Sysplex environment function that includes data sharing and parallelism. Parallel Sysplex uses the OS/390 (5647-A01), z/OS (5694-A01), or z/OS.e (5655-G52) operating system.

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

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

IBM Corporation
Department B6ZH, Mail Station P350
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

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

IBMLink (United States customers only): IBMUSM(BIXLER)
Internet e-mail: bixler@us.ibm.com
World Wide Web: www.ibm.com/servers/eserver/zseries/zos/integtst/

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

Opening remarks

A message from our team

We changed our title from the *z/OS Parallel Sysplex Test Report* but it's still us! Same team, same testing, but we've gradually expanded our focus from Parallel Sysplex to a platform wide view of z/OS's place in the enterprise. To reflect that focus, we changed our title to be the ***zSeries Platform Test Report for z/OS and Linux Virtual Servers***.

As you read this document, keep in mind that ***we need your feedback.*** We want to hear anything you want to tell us, whether it's positive or less than positive. ***We especially want to know what you'd like to see in future editions.*** That helps us prioritize what we do in our next test phase. We will also make additional information available upon request if you see something that sparks your interest. To find out how to communicate with us, please see "How to send your comments" on page xviii.

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

Al Alexsa
Vinay Anumalla
Loraine Arnold
Ozan Baran
Ryan Bartoe
Jeff Bixler
Muriel Bixler
Dave Buehl
Jon Burke
Alex Caraballo
John Corry
Don Costello
Luis Cruz
Martha DeMarco
Tony DiLorenzo
Bob Fantom
Nancy Finn
Bobby Gardinor
Kieron Hinds
Joan Kelley
Parul Lewicke
Fred Lates
Al Lease
Sue Marcotte
Tammy McAllister
James Mitchell
Bob Muenkel
Elaine Murphy
Jim Rossi
Tom Sirc
Karen Smolar
Jeff Stokes
Jim Stutzman
Ashwin Venkatraman

Important—Currency of the softcopy edition

Each release of the *z/OS Collection* (SK3T-4269 or SK3T-4270) and *z/OS DVD Collection* (SK3T-4271) contains a back-level edition of this test report.

Because we produce our test reports toward the end of the product development cycle, just before each new software release becomes generally available (GA), we cannot meet the production deadline for the softcopy collections that coincide with the product's GA release. Therefore, there is normally a one-edition lag between the release of our latest test report edition and the softcopy collection in which it is included. That is, the test report that appears in any given softcopy collection is normally one edition behind the most current edition available on the Web.

If you obtained this document from a softcopy collection on CD-ROM or DVD, you can get the most current edition from the z/OS Integration Test Web site at:

www.ibm.com/servers/eserver/zseries/zos/integtst/

Contents

Figures

Tables

About this document

This document is a test report written from the perspective of a system programmer. The IBM zSeries Integration Test team—a team of IBM testers and system programmers simulating a customer production Parallel Sysplex environment—wants to continuously communicate directly with you, the zSeries customer system programmer. We provide this test report to keep you abreast of our efforts and experiences in performing the final verification of each system release before it becomes generally available to customers.

An overview of Integration Test

We have been producing this test report since March, 1995. At that time, our sole focus of our testing was the S/390 MVS Parallel Sysplex. With the introduction of OS/390 in 1996, we expanded our scope to encompass various other elements and features, many of which are not necessarily sysplex-oriented. In 2001, OS/390 evolved into z/OS, yet our mission remains the same to this day.

Our mission and objectives

IBM's testing of its products is and always has been extensive. ***The test process described in this document is not a replacement for other test efforts.*** Rather, it is an additional test effort with a shift in emphasis, focusing more on the customer experience, cross-product dependencies, and high availability. We simulate the workload volume and variety, transaction rates, and lock contention rates that exist in a typical customer shop, stressing many of the same areas of the system that customers stress. When we encounter a problem, our goal is to keep systems up and running so that end users can still process work.

Even though our focus has expanded over the years, our objectives in writing this test report remain as they were:

- Run a Parallel Sysplex in a production shop in the same manner that customers do. We believe that only by being customers ourselves can we understand what our own customers actually experience when they use our products.
- Describe the cross-product and integrated testing that we do to verify that certain functions in specific releases of IBM mainframe server products work together.
- Share our experiences. In short, if any of our experiences turn out to be painful, we tell you how to avoid that pain.
- Provide you with specific recommendations that are tested and verified.

We continue to acknowledge the challenges that information technology professionals face in running multiple hardware and software products and making them work together. We're taking more of that challenge upon ourselves, ultimately to attempt to shield you from as much complexity as possible. The results of our testing should ultimately provide the following benefits:

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

Our test environment

The Parallel Sysplex that forms the core of our test environment has grown and changed over the years. Today, our test environment has evolved to a highly interconnected, multi-platform e-business enterprise—just like yours.

To see what our environment looks like, see the following:

- "Our Parallel Sysplex hardware configuration" on page 3
- "Our Parallel Sysplex software configuration" on page 11
- "Our networking and application enablement configuration" on page 77
- "Our workloads" on page 14

Who should read this document

System programmers should use this book to learn more about the integration testing that IBM performs on z/OS and certain related products, including selected test scenarios and their results. We assume that the reader has knowledge of MVS and Parallel Sysplex concepts and terminology and at least a basic level of experience with installing and managing the z/OS or OS/390 operating system, subsystems, network products, and other related software. See "Where to find more information" on page xvii.

How to use this document

Use this document as a companion to—***never*** *a replacement for*—your reading of other z/OS element-, feature-, or product-specific documentation. Our configuration information and test scenarios should provide you with concrete, real-life examples that help you understand the "big picture" of the Parallel Sysplex environment. You might also find helpful tips or recommendations that you can apply or adapt to your own situation. Reading about our test experiences should help you to confidently move forward and exploit the key functions you need to get the most from your technology investment.

However, you also need to understand that, while the procedures we describe for testing various tasks (such as installation, configuration, operation, and so on) are based on the procedures that are published in the official IBM product documentation, they also reflect our own specific operational and environmental factors and are intended for illustrative purposes only. Therefore, ***do not*** use this document as your sole guide to performing any task on your system. Instead, follow the appropriate IBM product documentation that applies to your particular task.

How to find the Parallel Sysplex Test Report

We make all editions of our test reports available on our z/OS Integration Test Web site at:

www.ibm.com/servers/eserver/zseries/zos/integtst/

If you cannot get to our Web site for some reason, see Appendix C, "Availability of our test reports," on page 199 for other ways to access our test reports.

We have traditionally published our test report on a quarterly basis where each quarterly edition was cumulative for the current year. At the end of each year, we freeze the content in our last edition; we then begin with a new test report the following year. The most recent quarterly edition as well as all of the previous year-end editions are available on our Web site.

In 2003, our publication schedule changed from our traditional quarterly cycle as a result of the change in the development cycle for annual z/OS releases. Keep an

eye on our Web site for announcements about the availability of new editions of our test report. In any event, the contents of our test reports remain cumulative for any given year.

We also have a companion publication, *OS/390 Parallel Sysplex Recovery*, GA22-7286-00, which documents the Parallel Sysplex recovery scenarios we've executed in our test environment, including operating system, subsystem, and coupling facility recovery. We describe how to be prepared for potential problems in a Parallel Sysplex, what the indicators are to let you know that a problem exists, and what actions to take to recover.

Note: The recovery book was written in the OS/390 V2R4 time frame; however, many of the recovery concepts that we discuss still apply to later releases of OS/390 and z/OS.

Where to find more information

If you are unfamiliar with Parallel Sysplex terminology and concepts, you should start by reviewing the following publications:

Table 1. Parallel Sysplex planning library publications

Publication title	Order number
z/OS Parallel Sysplex Overview	SA22-7661
z/OS MVS Setting Up a Sysplex	SA22-7625
z/OS Parallel Sysplex Application Migration	SA22-7662
z/OS and z/OS.e Planning for Installation	GA22-7504

In addition, you can find lots of valuable information on the World Wide Web.

- See the Parallel Sysplex for OS/390 and z/OS Web site at: www.ibm.com/servers/eserver/zseries/pso/
- See the Parallel Sysplex Customization Wizard at: www.ibm.com/servers/eserver/zseries/zos/wizards/parallel/
- See the z/OS Managed System Infrastructure (msys) for Operations Web site at: www.ibm.com/servers/eserver/zseries/msys/msysops/

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM® messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS® elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX® System Services running OMVS).

- Your Microsoft® Windows® workstation. You can install code to access IBM message explanations on the *z/OS Collection* (SK3T-4269), using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your *z/OS Collection* (SK3T-4269), or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

How to send your comments

Your feedback is important to us. If you have any comments about this document or any other aspect of Integration Test, you can send your comments by e-mail to bixler@us.ibm.com or use the contact form on our Web site at:

www.ibm.com/servers/eserver/zseries/zos/integtst/

You can also submit the Readers' Comments form located at the end of this document.

Be sure to include the document number and, if applicable, the specific location of the information you are commenting on (for example, a specific heading or page number).

Summary of changes

We periodically update our test report with new information and experiences. If the edition you are currently reading is more than a few months old, you may want to check whether a newer edition is available (see “How to find the Parallel Sysplex Test Report” on page xvi).

This information below summarizes the changes that we have made to this document.

Summary of changes
for SA22-7997-00
September 2004

This document contains information previously presented in SA22-7663-11.

New information

- Enabling NFS recovery for system outages
- Automount enhancement for HFS to zSeries file system (zFS) migration
- Using multipliers with BPXPRMxx parameters
- Using the superkill command
- Added an ipV6 environment equivalent to our ipV4 environment. V1R6 now supports OSPF V3 for ipV6 and ipV6 support for DVIPA and Sysplex Distributor.
- Using wildcard characters in the automove system list (SYSLIST)
- Using the clear and uptime shell commands
- Enhanced latch contention detection
- Using distributed BRLM
- Using ISHELL enhancements
- zFS modify console command
- Using gskkyman support for storing a PKCS #7 file with a chain of certificates
- LDAP migration to z/OS V1R6
- Setting up a peer-to-peer replication network between an IBM Tivoli Directory Server 5.2 and a z/OS LDAP Server
- Using LDAP DB2 restart/recovery function
- Using LDAP alias support
- Using the enhanced LDAP configuration utility (LDAPCNF)
- Using LDAP change logging with TDBM
- NAS accessing SYS1.SIEALNKE
- EIM enhancements in z/OS V1R6
- Updates to our z/OS V1R5 test environment running WebSphere Application Server
- Migrating to WebSphere for z/OS V5.*X* on z/OS V1R6

Changed information

- Our sysplex hardware configuration

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of changes
for SA22-7663-11
June 2004

This document contains information previously presented in SA22-7663-10.

New information

- XCF REALLOCATE processing
- Using zSeries Application Assist Processors (zAAPs)
- IBM Health Checker for z/OS and Sysplex Version 3
- Migrating to WebSphere Business Integration Message Broker Version 5.0
- Implementing shared channels in a distributed-queuing management (DQM) environment
- Setting up a Kerberos peer trust relationship between z/OS and Windows 2000
- Using the z/OS LDAP client with the Windows 2000 Active Directory service
- Using LDAP with Kerberos authentication

Changed information

- Our sysplex hardware configuration

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of changes
for SA22-7663-10
March 2004

This document contains information previously presented in SA22-7663-09.

New information

- Migrating to z/OS V1R5 and z/OS.e V1R5
- Using DFSMS enhanced data integrity for sequential data sets
- Implementing the common service layer (CSL) and single point of control (SPOC) in IMS V8
- Using System Automation OS/390 (SA OS/390) V2R2
- Enhancements in z/OS UNIX in z/OS V1R5, including:
  - Remounting a shared HFS
  - Mounting file systems using symbolic links
  - Creating directories during z/OS UNIX initialization
  - Temporary file system (TFS) enhancements
- Setting up the LDAP server for RACF change logging
- Support for additional bind types for EIM authentication
- Using WebSphere MQ shared queues and coupling facility structures
- Migrating to WebSphere for z/OS V5.0

Changed information

- Overview of our LDAP Server configuration

Deleted information

- System-managed coupling facility structure duplexing

- Verifying use of the cryptographic hardware
- Migrating to IMS Version 8 Release 1
- Using z/OS DFSMStvs
- Setting up the CICSPlex SM Web User Interface
- Using IBM HTTP Server
- LDAP Server enhancements in z/OS V1R4
- Using IMS Connect for z/OS Version 1.2
- Implementing the System SSL started task
- Using PKI Services
- Using WebSphere Studio Workload Simulator in z/OS Integration Test

Although the above information has been deleted from this edition, it continues to be available in our December 2003 edition.

Part 1. Parallel Sysplex

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

Chapter 1. About our Parallel Sysplex environment

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

Note: Throughout this document, when you see the term ***sysplex***, understand it to mean a sysplex with a coupling facility, which is a ***Parallel Sysplex.***

Overview of our Parallel Sysplex environment

We currently run a 14-member Parallel Sysplex that consists of the following:

- Four central processor complexes (CPCs) running z/OS in 14 logical partitions (LPs).

  The CPCs consist of the following machine types:
  - One IBM @server zSeries 990 (z990) processor
  - One IBM @server zSeries 900 (z900) processor
  - One IBM @server zSeries 890 (z890) processor
  - One IBM @server zSeries 800 (z800) processor

  The z/OS images consist of the following:
  - Nine production z/OS systems
  - One production z/OS.e system
  - Three test z/OS systems
  - One z/OS system to run TPNS (Our December 1998 edition explains why we run TPNS on a non-production system.)
- Three coupling facilities:
  - One failure-independent coupling facility that runs in a LP on a standalone CPC
  - Two non-failure-independent coupling facilities that run in LPs on two of the CPCs that host other z/OS images in the sysplex
- Two Sysplex Timer external time references (ETRs)
- Other I/O devices, including ESCON- and FICON-attached DASD and tape drives.

The remainder of this chapter describes all of the above in more detail.

Outside of the Parallel Sysplex itself, we also have three LPs in which we run the following:
- One native Linux image
- One z/VM image that hosts multiple Linux guest images running in virtual machines

Our Parallel Sysplex hardware configuration

This section provides an overview of our Parallel Sysplex hardware configuration as well as other details about the hardware components in our operating environment.

Overview of our hardware configuration

Figure 1 on page 4 is a high-level, conceptual view of our Parallel Sysplex hardware configuration. In the figure, broad arrows indicate general connectivity

between processors, coupling facilities, Sysplex Timers, and other I/O devices; they do not depict actual point-to-point connections.

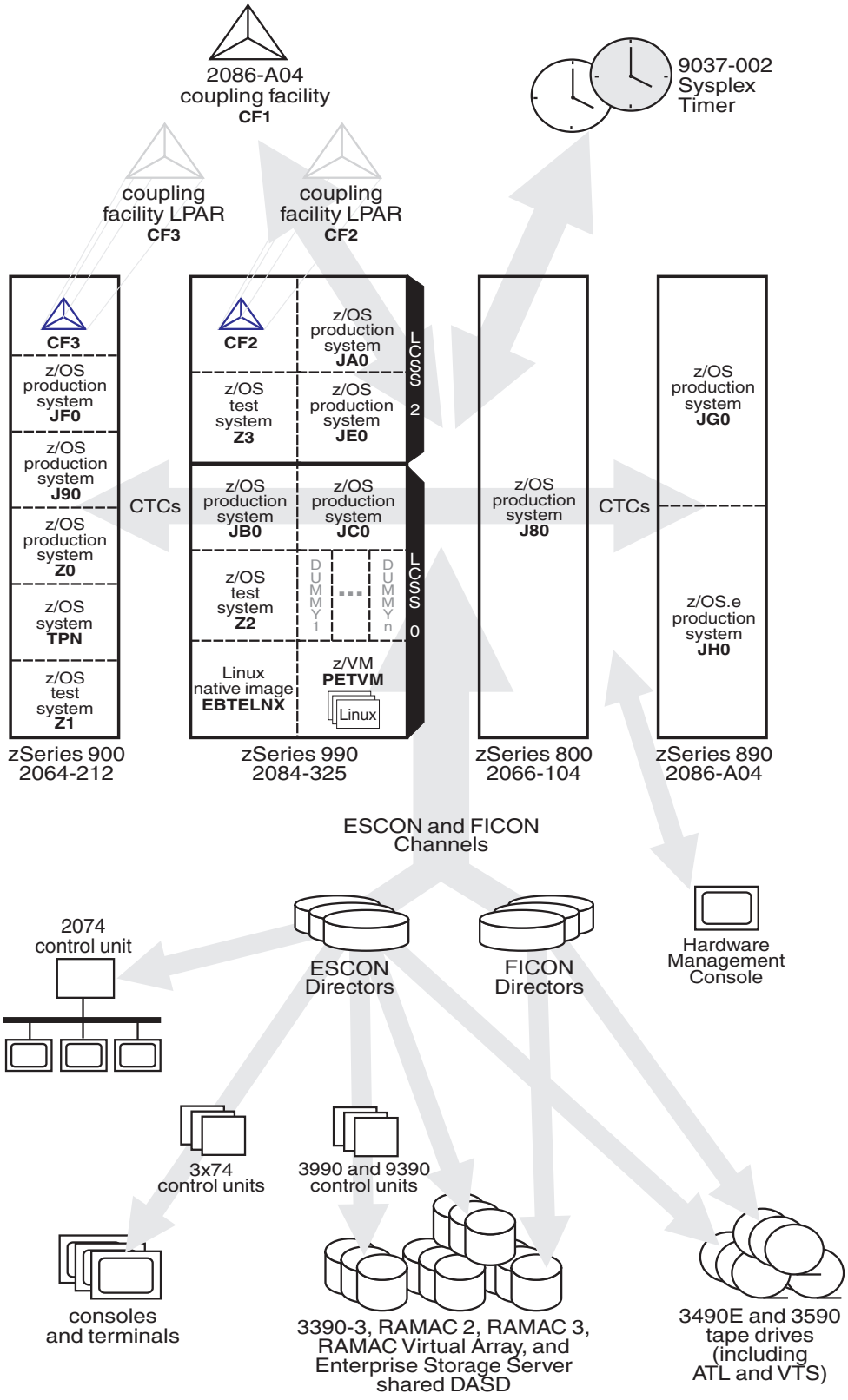


Figure 1. Our sysplex hardware configuration

Hardware configuration details

The figures and tables in this section provide additional details about the mainframe servers, coupling facilities, and other sysplex hardware shown in Figure 1 on page 4.

Mainframe server details

Table 2 provides information about the mainframe servers in our sysplex:

Table 2. Our mainframe servers

Server model (Machine type-model)	CPCs CPs	Mode LPs	HSA	Storage: Central	Expanded	LCSS	System name, usage Virtual CPs (static, managed) Initial LPAR weight
IBM @server zSeries 800 Model 104 (2066-104) (see note 2 below)	1 CPC 4 CPs	LPAR 1 LP	160M	7168M			**J80**, z/OS production system 4 CPs
IBM @server zSeries 890 Model A04 (2086-A04) (see notes 2 and 3 below)	1 CPC 4 CPs	LPAR mode 2 LPs	1344M	7168M			**JG0**, z/OS production system 4 shared CPs weight of 400
				7168M			**JH0**, z/OS.e production system 4 shared CPs weight of 400
IBM @server zSeries 900 Model 212 (2064-212) (see note 1 below)	1 CPC 16 CPs (4 ICF)	LPAR mode 6 LPs (1 LP is a coupling facility)	256M	5888M			**JF0**, z/OS production system 12 shared CPs (8, 4) weight of 285
				6144M			**J90**, z/OS production system 12 shared CPs (3, 9) weight of 285
				6144M			**Z0**, z/OS production system 12 shared CPs (7, 5) weight of 285
				4096M			**Z1**, z/OS test system 12 shared CPs (4, 8) weight of 145
				4096M			**TPN**, z/OS system for TPNS 8 shared CPs
IBM @server zSeries 990 Model 325 (2084-325)	1 CPC 32 CPs (3 ICF, 2 IFL, 2 zAAP)	LPAR mode 20 LPs[4] (1 LP is a coupling facility)	See note 5.	31G		2	**JA0**, z/OS production system 16 shared CPs
				31G		0	**JB0**, z/OS production system 16 shared CPs
				31G		0	**JC0**, z/OS production system 16 shared CPs
				31G		2	**JE0**, z/OS production system 16 shared CPs
				4096M		0	**Z2**, z/OS test system 6 shared CPs
				4096M		2	**Z3**, z/OS test system 6 shared CPs
				1024M		0	**EBTELNX**, native Linux image 1 shared CP
				1024M		0	**PETVM**, z/VM system 2 shared CPs weight of 10

Table 2. Our mainframe servers (continued)

Server model (Machine type-model)	CPCs CPs	Mode LPs	HSA	Storage: Central Expanded LCSS	System name, usage Virtual CPs (static, managed) Initial LPAR weight

Notes:

1. For our z900 server, we applied the IYP version of IOCP 1.1.0, which is available with the fix for APAR OW46633 (PTF UW90695). We also applied the fix for HCD APAR OW43131 (PTFs UW99341, UW99342, UW99343, UW99344, UW99345) and the fix for HCM APAR IR43534 (PTFs UR90329 and UR90330).
2. For our z800 server, we applied the fix for IOCP APAR OW52993. We also applied the fix for HCD APAR OW51339.
3. Since z/OS.e is engine licensed, customers must define the MSU capacity of a z/OS.e LP to be on an engine boundary. To do this, IBM recommends using the **Defined capacity** field in the activation profile on the z800 hardware management console (HMC). You must also send to IBM the Transmit System Availability Data (TSAD) for your z800 server, either by using the IBM Remote Support Facility (RSF) on the z800 or by mailing a diskette or DVD cartridge to IBM. For details, see *z/OS and z/OS.e Planning for Installation*, GA22-7504, and *z800 Software Pricing Configuration Technical Paper*, GM13-0121, available from the zSeries Library at www.ibm.com/servers/eserver/zseries/library/.
4. We added several “dummy” logical partitions on our z990 server—LPs that are defined but not activated—in order to force the number of LPs to be greater than 15. Currently, you can define up to 30 LPs on the z990.
5. On the z990 and z890 support elements (SE), you no longer specify an HSA expansion percentage in the activation profile. Instead, the HSA size is now calculated from IOCP MAXDEV value.

Coupling facility details

Table 3 provides information about the coupling facilities in our sysplex. Figure 2 on page 9 further illustrates the coupling facility channel distribution as described in Table 3.

Table 3. Our coupling facilities

Coupling facility name	Model CPCs and CPs CFLEVEL (CFCC level) Controlled by	Storage: Central Expanded	Channel distribution
CF1	zSeries 890 Model A04 (2086-A04) stand-alone coupling facility 1 CPC with 4 CPs CFLEVEL=13 (CFCC Release 13.00, Service Level 02.05) Controlled by the HMC	6G	17 TYPE=CFP channels. These are peer-mode intersystem coupling (ISC) channels[2]. There are 17 corresponding TYPE=CFP channels on the following systems (″shared″ indicates that the systems share that number of channels using MIF): • JG0/JH0: 4 shared • J80: 3 • JA0/JB0/JC0/JE0/Z2/Z3 and CF2[3]: 4 shared • JF0/J90/TPN/Z0/Z1 and CF3[3]: 6 shared
			2 TYPE=CBP channels. These are peer-mode integrated cluster bus (ICB) channels[2]. There are 2 corresponding TYPE=CBP channels on the following systems: • JG0/JH0: 1 • JA0/JB0/JC0/JE0/Z2/Z3 and CF2: 1 shared
CF2	Coupling facility LP on a zSeries 990 Model 325 (2084-325) 3 dedicated ICF CPs CFLEVEL=13 (CFCC Release 13.00, Service Level 02.05) Controlled by the HMC	6G	15 TYPE=CFP channels. These are peer-mode ISC channels[2, 4]. There are 15 corresponding TYPE=CFP channels on the following systems: • JG0/JH0: 2 shared • J80: 3 • JF0/J90/TPN/Z0/Z1 and CF3: 6 shared • CF1: 4
			7 TYPE=CBP channels. These are peer-mode ICB channels[2, 4]. There are 7 corresponding TYPE=CBP channels on the following systems: • JG0/JH0: 1 shared • JF0/J90/TPN/Z0/Z1 and CF3: 5 shared • CF1: 1
			8 TYPE=ICP channels. These are peer-mode internal coupling (IC) channels[1,2, 4]. There are 8 corresponding TYPE=ICP channels on the following systems: • JA0/JB0/JC0/JE0/Z2/Z3: 2 shared, spanning LCSS 0 and 2 • JB0/JC0/Z2: 4 shared (in LCSS 0) • JA0/JE0/Z3: 2 shared (in LCSS 2)

Table 3. Our coupling facilities (continued)

Coupling facility name	Model CPCs and CPs CFLEVEL (CFCC level) Controlled by	Storage: Central Expanded	Channel distribution
CF3	Coupling facility LP on a zSeries 900 Model 212 (2064-212) 4 dedicated ICF CPs CFLEVEL=13 (CFCC Release 13.00, Service Level 04.05) Controlled by the HMC	6G	16 TYPE=CFP channels[2, 3]. There are 16 corresponding TYPE=CFP channels on the following systems: • JG0/JH0: 3 shared • JB0/JC0/Z2: 1 shared • JA0/JE0/Z3 and CF2: 5 shared • JA0/JB0/JC0/JE0/Z2/Z3 and CF2: 1 shared • CF1: 6
			1 TYPE=CBR channels. There is 1 corresponding TYPE=CBS channel shared by systems JG0/JH0.
			9 TYPE=CBP channels[2, 3]. There are 9 corresponding TYPE=CBP channels on the following systems: • J80: 2 shared • JB0/JC0/Z2: 2 shared • JA0/JE0/Z3 and CF2: 4 shared • JA0/JB0/JC0/JE0/Z2/Z3 and CF2: 1 shared
			2 TYPE=ICP channels[1, 2, 3]. There are 2 corresponding TYPE=ICP channels shared by systems JF0/J90/TPN/Z0/Z1.

Notes:

1. Our servers that contain internal coupling facilities (CF2 and CF3) also have internal coupling (IC) channels. IC channels are logical connections between a coupling facility LP and the z/OS LPs on the same CPC. IC channels require no channel or cabling hardware (although CHPID numbers must still be defined in the IOCDS). Because they utilize the system bus, IC channels offer improved coupling performance over intersystem coupling (ISC, channel types CFS, CFR, and CFP) channels and integrated cluster bus (ICB, channel types CBS, CBR, and CBP) channels.
2. On the z800, z890, z900, and z990 servers, you can define coupling facility channels as peer channels on both sides of a coupling facility connection. A peer channel contains both sender and receiver functions; however, it is not necessary for both sides use both functions. You can define ISC, ICB, and IC channels in peer mode as channel types CFP, CBP, and ICP, respectively. You can only use peer mode between coupling facility LPs and z/OS LPs that reside on z800, z890, z900, and z990 servers. See *z/OS HCD Planning* for more information.
3. On our z900 server that contains CF3, all of the peer-mode channels are shared by all of the LPs (z/OS LPs and the coupling facility LP) on the CPC. These paths support the XCF communications both between the z/OS LPs and the other coupling facilities in the sysplex, as well as the system-managed coupling facility structure duplexing between the coupling facility LP and the other coupling facilities in the sysplex.
4. On our z990 server that contains CF2 has two logical channel subsystems (LCSSs). Coupling facility channels that are defined within a single LCSS can only be shared by the LPs (both z/OS images and coupling facility images) in that LCSS. Coupling facility channels that span multiple LCSSs can be shared by the LPs (both z/OS images and coupling facility images) in those LCSSs.

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

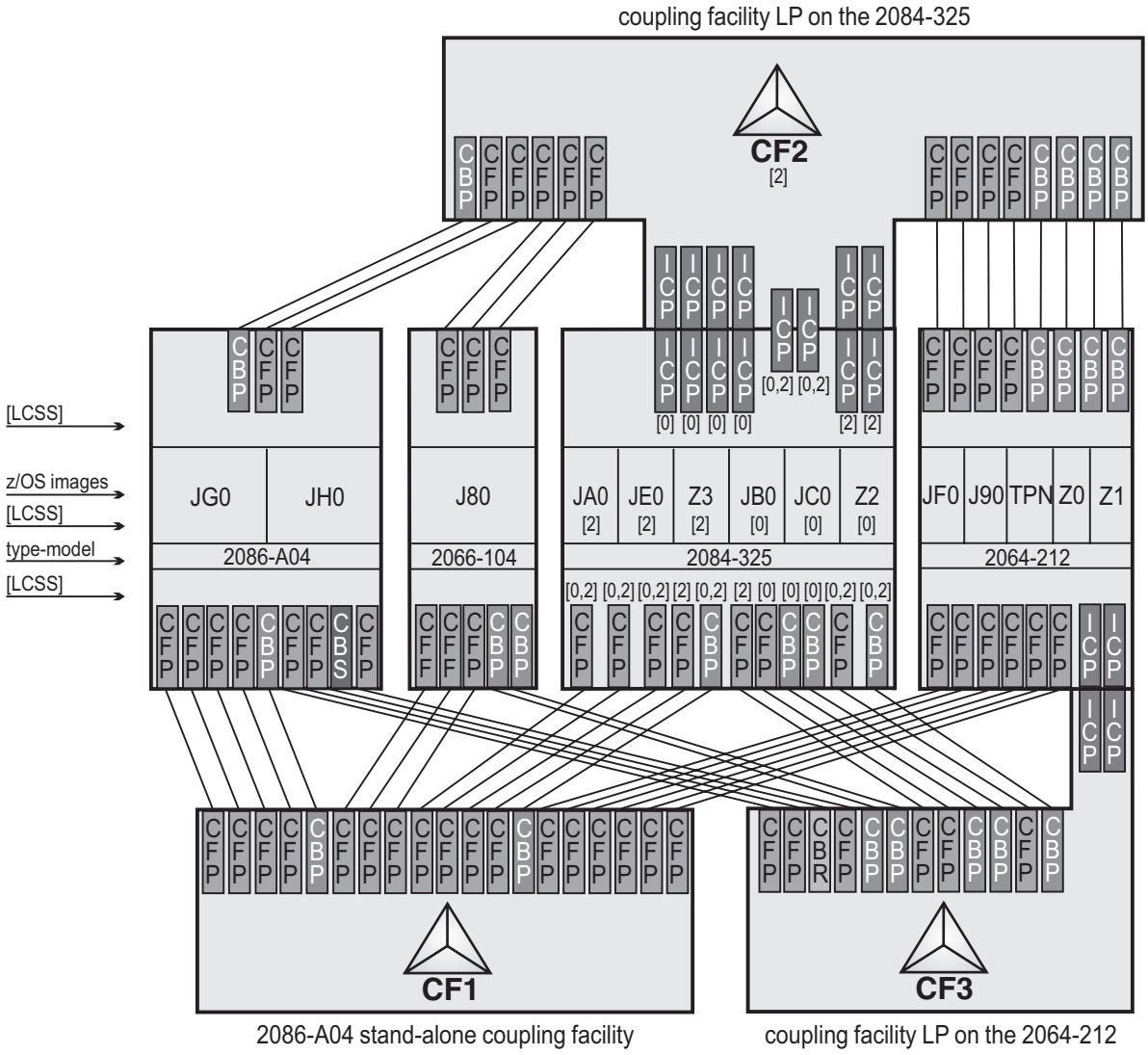


Figure 2. Our coupling facility channel configuration

Other sysplex hardware details

Table 4 highlights information about the other hardware components in our sysplex:

Table 4. Other sysplex hardware configuration details

Hardware element	Model or type	Additional information
External Time Reference (ETR)	Sysplex Timer (9037-002 with feature code 4048)	We use the Sysplex Timer with the Expanded Availability feature, which provides two 9037 control units connected with fiber optic links. We don't have any Sysplex Timer logical offsets defined for any of the LPs in our sysplex.

Table 4. Other sysplex hardware configuration details (continued)

Hardware element	Model or type	Additional information
Channel subsystem	CTC communications connections	We have CTC connections from each system to every other system. We now use FICON CTC channels on all of our CPCs. **Note:** All of our z/OS images use both CTCs and coupling facility structures to communicate. This is strictly optional. You might choose to run with structures only, for ease of systems management. We use both structures and CTCs because it allows us to test more code paths, and under some circumstances, XCF signalling using CTCs is faster than using structures. See *S/390 Parallel Sysplex Performance* for a comparison.
	Coupling facility channels	We use a combination of ISC, ICB, and IC coupling facility channels in peer mode. We use MIF to logically share coupling facility channels among the logical partitions on a CPC. We define at least two paths from every system image to each coupling facility, and from every coupling facility to each of the other coupling facilities.
	ESCON channels	We use ESCON channels and ESCON Directors for our I/O connectivity. Our connections are “any-to-any”, which means every system can get to every device, including tape. (We do not use any parallel channels.)
	FICON channels	We have FICON native (FC) mode channels from all of our CPCs to our Enterprise Storage Servers and our 3590 tape drives through native FICON switches. (See *FICON Native Implementation and Reference Guide*, SG24-6266, for information about how to set up this and other native FICON configurations.) We maintain both ESCON and FICON paths to the Enterprise Storage Servers and 3590 tape drives for testing flexibility and backup. Note that FICON channels do not currently support dynamic channel path management. We have also implemented FICON CTCs, as described in the IBM Redpaper *FICON CTC Implementation* available on the IBM Redbooks Web site.
Storage control units	3990 Storage Control Model 6 9390 (RAMAC 3) Storage Control	We have at least two paths to all DASDs. In fact, we have eight paths to all of our production workload databases.
DASD	3390 Model 3 RAMAC 2 (9392-002) RAMAC 3 (9392-003) RAMAC Virtual Array (RVA, 9393) Enterprise Storage Server (ESS, 2105-F20,800)	All volumes shared by all systems; about 90% of our data is SMS-managed. We currently have four IBM TotalStorage Enterprise Storage Servers, of which two are FICON only, and two that are attached with both ESCON and FICON. **Note:** IBM recommends against running with both ESCON and FICON channel paths from the same CPC to a control unit. We have some CPCs that are ESCON-connected and some that are FICON-connected.
Tape	3490E tape drives	16 IBM 3490 Magnetic Tape Subsystem Enhanced Capability (3490E) tape drives that can be connected to any system.
	3590 tape drives	4 IBM TotalStorage Enterprise Tape System 3590 tape drives that can be connected to any system.
Automated tape library (ATL)	3495 Model L40 with 16 additional 3490E tape drives and 12 3590 tape drives	All tape drives are accessible from all systems.
Virtual Tape Server (VTS)	3494 Model L10 with 32 virtual 3490E tape drives and 12 ESCON- and FICON-attached 3590 tape drives	All tape drives are accessible from all systems.

Our Parallel Sysplex software configuration

We run the z/OS operating system along with the following software products:

- CICS Transaction Server (CICS TS) V2R2
- IMS V8.1 (and its associated IRLM)
- DB2 UDB for z/OS and OS/390 V7 (and its associated IRLM)
- WebSphere for z/OS V4.0.1.
- WebSphere MQ for z/OS V5.3.1
- WebSphere Business Integration Message Broker V5.0

We also run z/OS.e in one partition on our z800 server. z/OS.e supports next-generation e-business workloads; it does not support traditional workloads, such as CICS and IMS. However, z/OS.e uses the same code base as z/OS and invokes an operating environment that is identical to z/OS in all aspects of service, management, reporting, and zSeries functionality. See *z/OS.e Overview*, GA22-7869, for more information.

Note that we currently only run IBM software in our sysplex.

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

A note about IBM License Manager

In z/OS V1R1, IBM introduced a new base element called IBM License Manager (ILM). IBM has since decided not to deliver the IBM License Manager tool for zSeries. Therefore, when you run z/OS on a z800, z900, or z990 server, you must ensure that the ILMMODE parameter in IEASYS*xx* is set to `ILMMODE=NONE`.

Overview of our software configuration

Figure 3 on page 12 shows a high-level view of our sysplex software configuration.

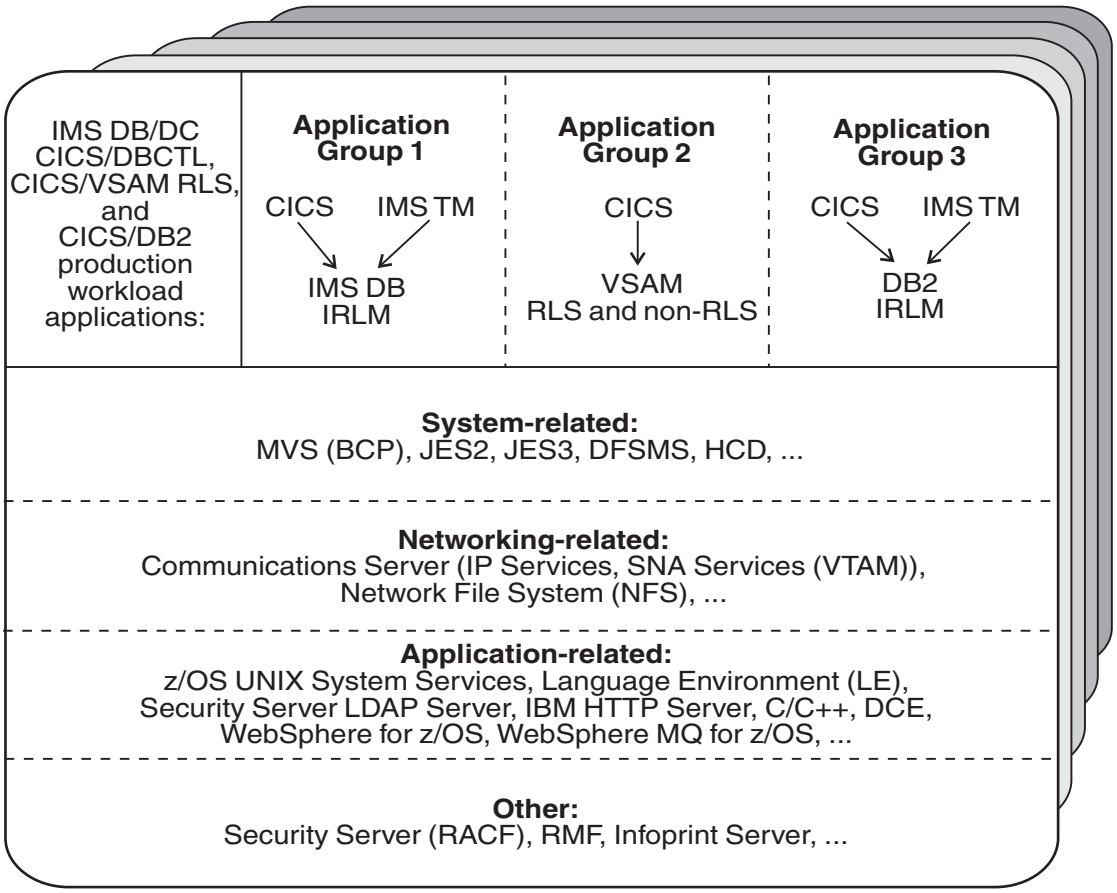


Figure 3. Our sysplex software configuration

We run three separate application groups in one sysplex and each application group spans multiple systems in the sysplex. Table 5 provides an overview of the types of transaction management, data management, and serialization management that each application group uses.

Table 5. Our production OLTP application groups

Application groups	Transaction management	Data management	Serialization management
Group 1	• CICS • IMS TM	IMS DB	IRLM
Group 2	• CICS	VSAM	VSAM record-level sharing (RLS)
Group 3	• CICS • IMS TM	DB2	IRLM

Our December 1995 edition describes in detail how a transaction is processed in the sysplex using application group 3 as an example. In the example, the transaction writes to both IMS and DB2 databases and is still valid for illustrative purposes, even though our application group 3 is no longer set up that way. For more information about the workloads that we currently run in each of our application groups, see “Database product OLTP workloads” on page 20.

About our naming conventions

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

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

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

- *g* represents the application group, and can be either 1, 2, or 3
- *r* represents the CICS region type, and can be either A for AORs, F for FORs, T for TORs, or W for WORs (Web server regions)
- *s* represents the system name, and can be 0 for system Z0, 8 for J80, 9 for J90, and A for JA0 through G for JG0
- *i* represents the instance of the region and can be A, B, or C (we have 3 AORs in each application group on each system)

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

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

Our networking configuration

For a detailed description of our networking configuration, see Chapter 6, “About our networking and application enablement environment,” on page 77.

Our VTAM configuration

Figure 4 on page 14 illustrates our current VTAM configuration.

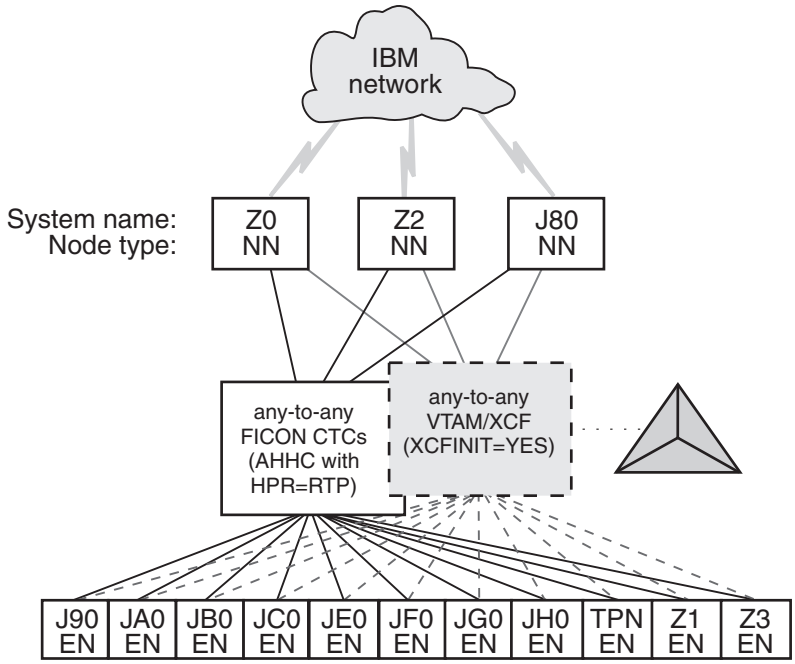


Figure 4. Our VTAM configuration

TPNS runs on our system TPN and routes CICS logons to any of the other systems in the sysplex (except JH0, which runs z/OS.e and does not support CICS).

Our VTAM configuration is a pure any-to-any AHHC. Systems Z0, Z2, and J80 are the network nodes (NNs) and the remaining systems are end nodes (ENs).

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

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

Our workloads

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

Table 6 on page 15 summarizes the workloads we run during our prime shift and off shift. We describe each workload in more detail below.

Table 6. Summary of our workloads

Shift	Base system workloads	Application enablement workloads	Networking workloads	Database product workloads
Prime shift	• Automatic tape switching • Batch pipes • JES2/JES3 printer simulators	• Enterprise Identity Mapping (EIM) • IBM HTTP Server • LDAP Server • z/OS UNIX Shelltest (rlogin/telnet) • z/OS UNIX Shelltest (TSO) • WebSphere for z/OS	• AutoWEB • FTP workloads • MMFACTS for NFS • NFSWL • Silk Test NFS video stream • TCP/IP CICS sockets • TN3270	• CICS DBCTL • CICS/DB2 • CICS/QMF online queries • CICS/RLS batch • CICS/RLS online • CICS/NRLS batch • CICS/NRLS online • DB2 Connect • DB2 online reorganization • DB2/RRS stored procedure • IMS AJS • IMS/DB2 • IMS full function • IMS SMQ fast path • MQ batch stress for shared queues • MQ-CICS bridge workload • MQ connection testing • MQ/DB2 bookstore application • QMF batch queries
Off shift	• Random batch • Automatic tape switching • JES2/JES3 printer simulators	• Enterprise Identity Mapping (EIM) • IBM HTTP Server • LDAP Server • z/OS UNIX Shelltest (rlogin/telnet) • z/OS UNIX Shelltest (TSO) • WebSphere for z/OS	• FTP workloads • Silk Test NFS video stream • MMFACTS for NFS	• CICS /DBCTL • CICS/DB2 • CICS/RLS batch • CICS RLS online • CICS/NRLS batch • CICS/NRLS online • DB2 DDF • DB2 utility • IMS/DB2 • IMS utility • MQ/DB2 bookstore application • QMF online queries

Base system workloads

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

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

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

Both workloads are set up to run under OPC so that 3 to 5 job streams with hundreds of jobs are all running at the same time to all systems in the sysplex. With

WLM-managed initiators, there are no system affinities, so any job can run on any system. In this way we truly exploit the capabilities of automatic tape switching.

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

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

Application enablement workloads

We run the following application enablement workloads:

Enterprise Identity Mapping (EIM)

This workload exercises the z/OS EIM client and z/OS EIM domain controller. It consists of a shell script running on a z/OS image that simulates a user running EIM transactions.

IBM HTTP Server

These workloads are driven from AIX/RISC workstations. They run against various HTTP server environments, including the following:

- HTTP scalable server
- HTTP standalone server
- Sysplex distributor routing to various HTTP servers

These workloads access the following:

- DB2 through net.data
- MVS data sets
- FastCGI programs
- Counters
- Static html pages
- Protected pages
- SSL transactions

LDAP Server

This workload consists of a script running on a Windows NT workstation that simulates multiple users running a transaction that drives several different **ldapsearch** commands against the LDAP server on z/OS.

z/OS UNIX Shelltest (rlogin/telnet)

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

z/OS UNIX Shelltest (TSO)

In this workload, simulated users driven by the Teleprocessing Network Simulator (TPNS) logon to TSO/E and invoke the z/OS UNIX shell and issue various commands. The users perform tasks that simulate real z/OS UNIX users daily jobs, for example:

- Moving data between the HFS and MVS data sets.
- Compiling C programs.
- Running shell programs.

WebSphere for z/OS

We run a number of different Web application workloads in our test environment on z/OS. Generally, each workload drives HTTP requests to Web applications that consist of any combination of static content (such as HTML documents and images

files), Java Servlets, JSP pages, and Enterprise JavaBeans (EJB) components. These Web applications use various connectors to access data in our DB2, CICS, or IMS subsystems.

Our Web application workloads currently include the following:

- J2EE applications (including persistent (CMP and BMP) and stateless session EJB components) that:
  - Access DB2 using JDBC
  - Access CICS using the CICS Common Client Interface (CCI)
  - Access IMS using the IMS Connector for Java CCI
- Non-J2EE applications (only static resources, Servlets, and JSP pages) that:
  - Access DB2 using JDBC
  - Access CICS using CICS CTG
  - Access IMS using IMS Connect
- Other variations of the above applications, including those that:
  - Access secure HTTPS connections using SSL
  - Perform basic mode authentication
  - Use HTTP session data
  - Use connection pooling

WebSphere MQ workloads

Our WebSphere MQ environment includes one WebSphere MQ for z/OS V5.3.1 queue manager on each system in the sysplex. We have two queue sharing groups: one with three queue managers and another with seven queue managers.

Our workloads test the following WebSphere MQ features:

- CICS Bridge
- Distributed queueing with APPC, SSL, and TCP/IP channels
- Large messages
- Shared queues
- Clustering
- Transaction coordination with RRS

We use the following methods to drive our workloads (not all workloads use each method):

- Batch jobs
- Web applications driven by WebSphere Studio Workload Simulator
- TPNS TSO users running Java programs via z/OS UNIX shell scripts

The batch-driven workloads that use WebSphere MQ for z/OS include the following:

MQ batch stress for non-shared queues: This workload runs on one system and stresses WebSphere MQ for z/OS by issuing MQI calls. These calls include a variety of commands affecting local queues.

MQ batch stress for shared queues: This workload runs on one system and stresses WebSphere MQ for z/OS by issuing MQI calls. These calls include a variety of commands affecting shared queues. Workload parameters control the number of each type of call.

Communications testing: This workload tests our communications channels by kicking off an application that sends messages to a remote queue manager. A trigger monitor program running on the remote system kicks off a separate

application that sends the same message back to the originating system. The remote queue manager resides on a mainframe Linux system running WebSphere MQ V5.3.1.

We also run several Web applications to test WebSphere MQ for z/OS. These Java-based workloads use the IBM HTTP Server with the WebSphere V4.0 plugin. We recently migrated some of the WebSphere V4.0 plugin applications to WebSphere V5.0.

DQM and DQMssl: These workloads test the communication between z/OS queue managers as well as z/OS and Linux queue managers using SSL TCPIP channels and non-SSL APPC channels. The application puts messages on remote queues and waits for replies on its local queues.

MQCICS: This workload uses the MQ CICS bridge to run a transaction that updates a DB2 parts table. The CICS bridge request and reply queues are shared queues with persistent messages. We defined a separate coupling facility structure for this application.

MQLarge: This workload tests various large message sizes by creating temporary dynamic queues and putting large messages on those queues. Message sizes vary from 1MB to 100MB starting in increments of 10MB. The script running the application randomly chooses a message size and passes this to the mqLarge program. mqLarge then dynamically defines a queue using model queues that have their maxmsgl set to accommodate the message.

Websphere Business Integration Message Broker

Our Websphere Business Integration Message Broker environment consists of four message brokers: three on test systems, and one on a production system. All are running at WBIMB 5.0.1 FixPack 03. We use the following methods to drive our workloads (not all workloads use each method):

- Web applications driven by WebSphere Studio Workload Simulator
- Batch jobs
- TPNS TSO users running Java programs via z/OS UNIX shell scripts

The Web applications consist of html pages, java servlets, and WBIMB message flows to process the messages. These Java-based workloads use the IBM HTTP Server with the WebSphere V4.0 plugin, and will be configured to use Websphere Application Server 5.0 in the near future.

Retail_IMS: This workload tests message manipulation by taking a message, extracting certain fields from it, and adding an IMS header.

Retail_Info: This workload tests inserting and deleting fields from a message into a simple DB2 table.

Retail_Wh: This workload tests inserting and deleting an entire message (using a data warehouse node) into a LOB DB2 table.

We have one batch-driven workload that uses WBIMB:

Sniffer: This workload tests basic MQ and WBIMB functionality using persistent and non-persistent messages. It is based on SupportPac IP13: Sniff test and Performance on z/OS. (See http://www-306.ibm.com/software/integration/support/supportpacs/category.html#cat1)

We have one TPNS workload that uses WBIMB:

Retail_TPNS: This workload is another version of Retail_IMS, but rather than being driven by WebSphere Studio Workload Simulator, it is driven by TNPS via z/OS UNIX shell scripts.

Networking workloads

We run the following networking workloads:

FTP workloads:

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

MMFACTS for NFS: This client/server workload is designed to simulate the delivery of multimedia data streams, such as video, across the network. It moves large volumes of randomly-generated data in a continuous, real-time stream from the server (in our case, z/OS) to the client. Data files can range in size from 4 MB to 2 Gigabytes. A variety of options allow for variations in such things as frame size and required delivery rates.

NFSWL: This client/server workload consists of shell scripts that run on our AIX clients. The shell script implements reads, writes, and deletes on an NFS mounted file system. We mount both HFS and zFS file systems that reside on z/OS. This workload is managed by a front end Web interface.

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

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

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

Silk Test NFS video stream: This client/server workload is very similar to that of MMFACTS except that it sends actual video streams across the network instead of simulating them.

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

TN3270: This workload uses TPNS to simulate TN3270 clients which logon to TSO using generic resources. This workload exploits Sysplex Distributor.

Database product workloads

Database product OLTP workloads

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

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

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

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

- CICS/DBCTL
- IMS SMQ Fast Path
- IMS SMQ full function
- IMS automated job submission (AJS)

Highlights of our IMS data sharing workloads include:

- Full function, Fast Path, and mixed mode transactions
- Use of virtual storage option (VSO), shared sequential dependent (SDEP) databases, generic resources, and High Availability Large Databases (HALDB)
- Integrity checking on INSERT calls using SDEP journaling
- A batch message processing (BMP) application to do integrity checking on REPLACE calls
- A set of automatically-submitted BMP jobs to exercise the High-Speed Sequential Processing (HSSP) function of Fast Path and the reorg and SDEP scan and delete utilities. This workload continuously submits jobs at specific intervals to run concurrently with the online system. We enhanced this workload based on recent customer experiences to more closely resemble a real-world environment.

VSAM/RLS data sharing workload: In application group 2, we run one OLTP VSAM/RLS data sharing workload. This workload runs transactions that simulate a banking application (ATM and teller transactions). The workload also runs transactions that are similar to the IMS data sharing workload that runs in application group 1, except that these transactions use VSAM files.

VSAM/NRLS workload: Also in application group 2, we added two new workloads. One uses transactions similar to our VSAM/RLS workload but accessing VSAM non-RLS files. The other is a very I/O-intensive workload that simulates a financial brokerage application.

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

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

In the second of our DB2 workloads, we use the same CICS regions and the same DB2 data sharing members. However, we use different transactions and different

databases. The table space layout is also different for the databases used by the second DB2 workload—it has partitioned table spaces, segmented table spaces, simple table spaces, and partitioned indexes.

Our third workload is a derivative of the second, but incorporates large objects (LOBs), triggers, user defined functions (UDFs), identity columns, and global temporary tables.

The fourth workload uses IMS/TM executing 12 different transaction types accessing DB2 tables with LOBs. It also excercises UDFs, stored procedures and global temporary tables.

Database product batch workloads

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

- IMS Utility
- RLS batch (read-only) and TVS batch
- DB2 batch workloads

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

DB2 batch workloads: Our DB2 batch workloads include:

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

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

WebSphere MQ / DB2 bookstore application

Our multi-platform bookstore application lets users order books or maintain inventory. The user interface runs on AIX, and we have data in DB2 databases on AIX and z/OS systems. We use WebSphere MQ for z/OS to bridge the platforms and MQ clustering to give the application access to any queue manager in the cluster. See our December 2001 edition for details on how we set up this application.

Chapter 2. Migrating to and using z/OS

This chapter describes our experiences with migrating to new releases of the z/OS operating system.

Overview

The following sections describe our most recent migration activities:

- "Migrating to z/OS V1R6"
- "Migrating to z/OS.e V1R6" on page 25
- "Migrating to z/OS V1R5" on page 29
- "Migrating to z/OS.e V1R5" on page 42

We primarily discuss our sysplex-related base operating system experiences. This includes the enablement of significant new functions and, if applicable, performance aspects. Detailed test experiences with major new functions beyond migration appear in subsequent chapters.

We discuss our networking and application-enablement environment and test experiences in Part 2, "Networking and application enablement," on page 73.

You can read about our migration experiences with earlier releases of z/OS and OS/390 in previous editions of our test report, available on our Web site:

For migration experiences with...	See...
z/OS V1R4	our December 2003 edition
z/OS V1R3	our December 2002 edition
z/OS V1R1 and V1R2	our December 2001 edition

Migrating to z/OS V1R6

This section describes our migration experiences with z/OS V1R6.

z/OS V1R6 base migration experiences

In this section we only describe our experiences with our base migration to z/OS V1R6, without having implemented any new functions. It includes our high-level migration process along with other migration activities and considerations.

Our high-level migration process for z/OS V1R6

The following is an overview of our z/OS V1R6 migration process.

Before we began: We reviewed the migration information in *z/OS and z/OS.e Planning for Installation*, GA22-7504 and *z/OS Migration*.

Table 7 on page 24 shows the high-level process we followed to migrate the members of our sysplex from z/OS V1R5 to z/OS V1R6.

Table 7. Our high-level migration process for z/OS V1R6

Stage	Description
Updating parmlib for z/OS V1R6	We created SYS1.PETR16.PARMLIB to contain all the parmlib members that changed for z/OS V1R6 and we used our LOAD*xx* member for migrating our systems one at a time. (See "Using concatenated parmlib" on page 25 for more about our use of concatenated parmlib and see our December 1997 edition for an example of how we use LOAD*xx* to migrate individual systems.)
Applying coexistence service	We applied the necessary coexistence service (also known as compatibility or toleration PTFs) to position our systems for the migration. See the coexistence service requirements in *z/OS and z/OS.e Planning for Installation* and make sure you install the fixes for any APARs that relate to your configuration before you migrate.
IPLing our first z/OS V1R6 image	We brought up z/OS V1R6 on our Z1 test system and ran it there for about one week.
Updating the RACF templates	To test the RACF dynamic template enhancement, we IPLed the first z/OS V1R6 image without first running the IRRMIN00 utility with PARM=UPDATE. As expected, the following message appeared: `ICH579E RACF TEMPLATES ON DATABASE ARE DOWNLEVEL` RACF initialization still completed successfully. We then ran IRRMIN00 with PARM=UPDATE to dynamically update the templates on all six RACF data sets without the need for an IPL. (See *z/OS Security Server RACF System Programmer's Guide*, SA22-7681 for details about RACF templates.)
IPLing additional z/OS V1R6 images	We continued to bring up additional z/OS V1R6 images across our sysplex, as follows: • We brought up z/OS V1R6 on our Z2 and Z3 test systems and ran for a couple of days. • Next, we migrated one production system, JF0, and ran for about a week. • Next, we migrated an additional test system, Z1, and two production systems, JG0 and JH0, and ran for another week. • At this point, we took all of the V1R6 images back down to V1R5. This is part of our increased focus on migration testing and fallback. We ran for a full day and experienced no fallback issues. • Next, we migrated an additional test system, Z0, and three production systems, TPN, JB0, and JC0, and ran for a couple of days. • Next, we migrated two more production systems, JA0 and JE0, and ran for about a week. • We then migrated the remaining two systems, J80 and J90.

The total time for our migration was approximately a month.

More about our migration activities for z/OS V1R6

This section highlights additional details about some of our migration activities.

Running with mixed product levels: During our migration, we successfully ran our sysplex with mixed product levels, including the following:

- z/OS V1R5 and z/OS V1R6
- z/OS V1R5 and z/OS.e V1R6
- z/OS V1R5 JES2 and z/OS V1R6 JES2
- z/OS V1R5 JES3 and z/OS V1R6 JES3

Using concatenated parmlib: We continue to use concatenated parmlib support to add or update parmlib members for z/OS V1R6. Appendix A, "Some of our parmlib members," on page 193 summarizes the additions and changes we made by parmlib member. Also see our Web site for examples of some of our parmlib members.

This is a good use of concatenated parmlib because it isolates all of the parmlib changes for z/OS V1R6 in one place and makes it easier to migrate multiple systems. Rather than change many parmlib members each time we migrate another system to V1R6, we just add the PARMLIB statements at the appropriate places in SYS0.IPLPARM(LOAD*xx*) to allow that system to use SYS1.PETR16.PARMLIB.

Recompiling REXX EXECs for automation: We recompiled our SA OS/390 REXX EXECs when we migrated to z/OS V1R6. We discuss the need to recompile these REXX EXECs in our our December 1997 edition.

Installing related service: We did not need to install any service in order to migrate from z/OS V1R5 to z/OS V1R6.

Defining greater than 16 CPs per z/OS image

Beginning with z/OS V1.6, you can now define up to 24 processors in a single z/OS image. Note that the limit of 24 processors is the total of general purpose processors and zAAPs. We defined greater than 16 processors for several of the partitions on our z990 server, up to 22 general purpose processors plus 2 zAAPs. Making this change is done, as usual, from the hardware management console (HMC) by updating the CP panel on the image profiles. Enhancements have been made to several z/OS elements (WLM, etc.) in z/OS V1.6 to support this new limit. IBM intends to support up to 32 processors in a single z/OS image in 2005.

Migrating to z/OS.e V1R6

This section describes our migration experiences with z/OS.e V1R6.

z/OS.e V1R6 base migration experiences

This section describes our experiences with migrating one system image (JH0) from z/OS.e V1R5 to z/OS.e V1R6. Here we only cover our experiences with our base migration to z/OS.e V1R6, including our high-level migration process and other migration activities and considerations.

Our high-level migration process for z/OS.e V1R6

The following is an overview of our z/OS.e V1R6 migration process.

Before we began: We reviewed the information in *z/OS and z/OS.e Planning for Installation*, GA22-7504, which covers both z/OS V1R6 and z/OS.e V1R6.

Important notice about cloning and software licensing

As discussed in *z/OS and z/OS.e Planning for Installation*, you might find that sharing system libraries or cloning an already-installed z/OS or z/OS.e system is faster and easier than installing z/OS or z/OS.e with an IBM installation package such as ServerPac. Most Parallel Sysplex customers are already aware of the concept of cloning and the benefits it provides.

However, prior to sharing or cloning z/OS or z/OS.e, **you must have a license for each z/OS and z/OS.e operating system that you run.** If you don't have the appropriate license or licenses, you must contact IBM. Any sharing or cloning of z/OS or z/OS.e without the appropriate licenses is not an authorized use of such programs. On a z800 server, if you want to run both z/OS and z/OS.e, z/OS requires the appropriate license for the machine on which it runs and z/OS.e requires a license for the number of engines on which it runs.

For more information about z/OS.e licensing, see *z800 Software Pricing Configuration Technical Paper* at www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130121.pdf.

Table 8 shows the high-level process we followed to migrate our z/OS.e V1R5 system to z/OS.e V1R6.

Table 8. Our high-level migration process for z/OS.e V1R6

Stage	Description
Obtaining licenses for z/OS.e	You need a license for the appropriate number of engines on the z800 or z890 server on which you intend to run z/OS.e (and, you would also need a license to run z/OS on the z800 or z890, if you intend to install it there). We use an internal process to do this; however, you must use the official process stated in *z800 Software Pricing Configuration Technical Paper*.
Updating the z800 or z890 LPAR name	z/OS.e must run in LPAR mode and the LPAR name must be of the form `ZOSE`*xxxx*, where *xxxx* is up to 4 user-specified alphanumeric characters. The name of the LPAR in which we run z/OS.e is `ZOSEJH0`. (We used HCD to set this when we first installed z/OS.e V1R3.)
Updating parmlib for z/OS.e V1R6	z/OS.e requires the `LICENSE=Z/OSE` statement in the IEASYS*xx* parmlib member. We used the same SYS1.PETR16.PARMLIB data set that we created for z/OS V1R6. We then have separate IEASYS*xx* and IFAPRD*xx* members in SYS1.PARMLIB that we tailored specifically for z/OS.e. See "Updating system data sets for z/OS.e" on page 27 for details.
Updating our LOAD*xx* member	During our initial installation of z/OS.e V1R3, we updated the entry for our system JH0 in our LOAD*xx* member in SYS0.IPLPARM to point to our new IEASYS02 parmlib member and to reflect the new LPAR name. Therefore, we did not need to change it for V1R6.

Table 8. Our high-level migration process for z/OS.e V1R6 (continued)

Stage	Description
Updating our IEASYMPT member	During our initial installation of z/OS.e V1R3, we updated the entry for our system JH0 in our IEASYMPT member in SYS1.PETR13.PARMLIB to point to our new IFAPRD*xx* parmlib member and to reflect the new LPAR name. Therefore, when we created our new SYS1.PETR16.PARMLIB, we carried the change along for V1R6.
IPLing the z/OS.e V1R6 image	We brought up z/OS.e V1R6 on our JH0 production system.

More about our migration activities for z/OS.e V1R6

This section highlights additional details about some of our migration activities.

About our z890 LPAR environment: z/OS.e must run in LPAR mode on a zSeries 800 or 890 mainframe server; it cannot run in basic mode. In addition, the name of the LPAR in which z/OS.e runs must be of the form `ZOSE`*xxxx*, where *xxxx* is up to four user-specified alphanumeric characters. The name of our z/OS.e z890 LPAR is ZOSEJH0.

Note: You can only run z/OS.e in a partition named ZOSE*xxxx*. You cannot IPL a z/OS system in a partition named ZOSE*xxxx*.

We currently run z/OS.e (JH0) in a mixed LPAR environment alongside LPARs running z/OS (JG0) on the same z890 server.

Note: Don't let the fact that z/OS.e only runs on a z800 or z890 server confuse you. These are fully functional zSeries servers and, in addition to z/OS.e, theyt supports all of the same zSeries operating systems as a z900 or z990 server.

Updating system data sets for z/OS.e: We continue to use concatenated parmlib support to add or update parmlib members for z/OS.e V1R6. We use the same SYS1.PETR16.PARMLIB data set as we do for our z/OS V1R6 systems.

Below are examples of our parmlib customizations to accommodate z/OS.e V1R6. Appendix A, "Some of our parmlib members," on page 193 summarizes the changes we made by parmlib member.

Example: We have a separate IEASYS*xx* member, IEASYS02, which specifies the `LICENSE=Z/OSE` statement that z/OS.e requires.

The entry for our z/OS.e system (JH0) in our LOAD*xx* member in SYS0.IPLPARM points to our IEASYS02 parmlib member and specifies the name of our z/OS.e LPAR, as follows:

```
⋮
HWNAME   z800name
LPARNAME ZOSEJH0
PARMLIB  SYS1.PETR16.PARMLIB
```

```
SYSPARM   02
:
```

Example: We have a separate IFAPRD*xx* member, IFAPRD02, which specifies the product ID value 5655-G52 for z/OS.e. There is no change to the product name value for z/OS.e (the product name value remains Z/OS).

Below is an example of one of the entries from our IFAPRD02 member:

```
:
PRODUCT OWNER('IBM CORP')
        NAME(Z/OS)
        ID( 5655-G52 )
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(Z/OS)
        STATE(ENABLED)
:
```

We also have an entry for our system JH0 in our IEASYMPT member in SYS1.PETR16.PARMLIB to point to our new IFAPRD02 parmlib member and to reflect the z/OS.e LPAR name, as follows:

```
:
SYSDEF HWNAME(z800name)
       LPARNAME( ZOSEJH0 )
       SYSNAME(JH0)
       SYSCLONE(JH)
:
       SYMDEF(&PROD= '02' )
:
```

Using current z/OS.e levels of JES2 and LE: As required, we are using the level of JES2 and Language Environment (LE) that comes with z/OS.e V1R6. z/OS.e does not permit the use of a lower level JES2 (or JES3) or LE.

Updating the ARM policy: You must ensure that your automation policies, such as ARM, do not try to use a z/OS.e image to start products that z/OS.e does not support. For example, do not identify a z/OS.e image as a restart target in a Parallel Sysplex that contains a mix of z/OS.e and z/OS images where the z/OS images run IMS, CICS, or DB2 with a requirement for CICS. CICS, IMS, or DB2 that uses CICS cannot restart on a z/OS.e image, but must restart on a z/OS image. If, for example, a CICS region attempts to start on z/OS.e, the region will start but the applications will fail with a U4093 abend.

Back when we installed z/OS.e V1R3, we removed our z/OS.e image, JH0, as a restart target for the unsupported subsystems mentioned above.

Removing z/OS.e from participation in MNPS: In our environment, CICS is the only exploiter of multiple node persistent sessions (MNPS) support. Because CICS cannot run on z/OS.e, there is no reason for the VTAM on z/OS.e to connect to the MNPS structure, ISTMNPS. We removed our z/OS.e image from participating in MNPS by coding the STRMNPS=NONE statement in our VTAM start member, ATCSTR*xx*, in SYS1.VTAMLST.

Removing z/OS.e from participation in TSO generic resource groups: Since TSO on z/OS.e only allows a maximum of eight concurrent sessions, we removed our z/OS.e image from participating in TSO generic resource groups. You can do this by coding the GNAME=NONE parameter—either in a separate TSOKEY*xx* member in parmlib or on the START command that starts the terminal control address space (TCAS).

In our case, we use a single TSOKEY*xx* member that has a symbolic value for the GNAME parameter. We then set that symbol to `NONE` for our JH0 image in our IEASYMPT member.

Other experiences with z/OS.e V1R6

Our testing of z/OS.e V1R6 included the following workloads or scenarios:

- z/OS UNIX System Services
- DB2 UDB
- IBM HTTP Server in scalable server mode
- WebSphere Application Server for z/OS
- CICS Transaction Gateway (CTG) to access CICS regions running in z/OS images on the same CPC and other CPCs
- DB2 access from Linux guests under z/VM on the same CPC
- our Bookstore application transactions

Migrating to z/OS V1R5

This section describes our migration experiences with z/OS V1R5.

z/OS V1R5 base migration experiences

In this section we only describe our experiences with our base migration to z/OS V1R5, without having implemented any new functions. It includes our high-level migration process along with other migration activities and considerations.

Our high-level migration process for z/OS V1R5

The following is an overview of our z/OS V1R5 migration process.

Before we began: We reviewed the migration information in *z/OS and z/OS.e Planning for Installation*, GA22-7504 and *z/OS Migration*.

Table 9 shows the high-level process we followed to migrate the members of our sysplex from z/OS V1R4 to z/OS V1R5.

Table 9. Our high-level migration process for z/OS V1R5

Stage	Description
Updating parmlib for z/OS V1R5	We created SYS1.PETR15.PARMLIB to contain all the parmlib members that changed for z/OS V1R5 and we used our LOAD*xx* member for migrating our systems one at a time. (See “Using concatenated parmlib” on page 30 for more about our use of concatenated parmlib and see our December 1997 edition for an example of how we use LOAD*xx* to migrate individual systems.)
Applying coexistence service	We applied the necessary coexistence service (also known as compatibility or toleration PTFs) to position our systems for the migration. See the coexistence service requirements in *z/OS and z/OS.e Planning for Installation* and make sure you install the fixes for any APARs that relate to your configuration before you migrate.
IPLing our first z/OS V1R5 image	We brought up z/OS V1R5 on our Z1 test system and ran it there for about one week.

Table 9. Our high-level migration process for z/OS V1R5 (continued)

Stage	Description
Updating the RACF templates	To test the RACF dynamic template enhancement, we IPLed the first z/OS V1R5 image without first running the IRRMIN00 utility with PARM=UPDATE. As expected, the following message appeared: `ICH579E RACF TEMPLATES ON DATABASE ARE DOWNLEVEL` RACF initialization still completed successfully. We then ran IRRMIN00 with PARM=UPDATE to dynamically update the templates on all six RACF data sets without the need for an IPL. (See *z/OS Security Server RACF System Programmer's Guide*, SA22-7681 for details about RACF templates.)
IPLing additional z/OS V1R5 images	We continued to bring up additional z/OS V1R5 images across our sysplex, as follows: • We brought up z/OS V1R5 on our Z2 test system and ran for a couple of days. • Next, we migrated three production systems, JA0, JG0, and JH0 (z/OS.e), to z/OS V1R5 and ran a couple of days. • At this point, we took all of the V1R5 images back down to V1R4 feature 2. This is part of our increased focus on migration testing and fallback. We ran for a full day and experienced no fallback issues. • We migrated systems Z1, Z2, JA0, JG0, and JH0 back to V1R5, along with production systems JF0, TPN, and Z0. We ran this way for about one week. • We then migrated the remaining six systems, JB0, JC0, JE0, J80, J90, and Z3, to z/OS V1R5.

The total time for our migration was approximately three weeks.

More about our migration activities for z/OS V1R5

This section highlights additional details about some of our migration activities.

Running with mixed product levels: During our migration, we successfully ran our sysplex with mixed product levels, including the following:

- z/OS V1R4 and z/OS V1R5
- z/OS V1R4 and z/OS.e V1R5
- z/OS V1R4 JES2 and z/OS V1R5 JES2
- z/OS V1R4 JES3 and z/OS V1R5 JES3

Using concatenated parmlib: We continue to use concatenated parmlib support to add or update parmlib members for z/OS V1R5. Appendix A, "Some of our parmlib members," on page 193 summarizes the additions and changes we made by parmlib member. Also see our Web site for examples of some of our parmlib members.

This is a good use of concatenated parmlib because it isolates all of the parmlib changes for z/OS V1R5 in one place and makes it easier to migrate multiple systems. Rather than change many parmlib members each time we migrate another system to V1R5, we just add the PARMLIB statements at the appropriate places in SYS0.IPLPARM(LOAD*xx*) to allow that system to use SYS1.PETR15.PARMLIB.

Recompiling REXX EXECs for automation: We recompiled our SA OS/390 REXX EXECs when we migrated to z/OS V1R5. We discuss the need to recompile these REXX EXECs in our our December 1997 edition.

Installing related service: We needed to install the fix for IMS APAR PQ81018 to fix a problem that prevented us from running transactions that accessed Fast Path databases (DEDBs). The APAR resolves an incompatibility between IMS V8 DEDB processing and z/OS V1R5 which results in an S0C4 abend in module DBFMMIT0 during DEDB area open processing.

Using DFSMS enhanced data integrity for sequential data sets

Data integrity for shared, sequential data sets (DISP=SHR) has been enhanced in z/OS DFSMS V1R5. When you activate the enhanced data integrity function, you can prevent accidental data loss by setting the sharing specifications for sequential data sets that are opened for output or update. When the function is active, DFSMS reports a violation if a program attempts to open a sequential data set for writing when the data set is already open for writing.

Parameters to control the enhanced data integrity function reside in a new, customer-created parmlib member, IFGPSEDI. In this member, you can specify whether the enhanced data integrity function is to operate in WARN mode or ENFORCE mode when a violation occurs. You can also specify a list of data sets to exclude from enhanced data integrity processing.

Table 10 summarizes how the enhanced data integrity function behaves in each operating mode.

Table 10. Summary of the modes of operation for the DFSMS enhanced data integrity function for sequential data sets

When a violation occurs and the enhanced data integrity mode is...	The OPEN request proceeds as follows:
MODE(WARN)	The program issues a warning message (IEC984I or IEC985I) when an application attempts to open for input or output a shared, sequential data set that is already open for output, even if the data set is in the exclude list. You can use this information to analyze the violations and determine which data sets to exclude from data integrity processing or which applications to modify.
MODE(ENFORCE)	The program abends when an application attempts to open for output a shared, sequential data set that is already open for output, unless the data set is in the exclude list. No warning messages are issued.
MODE(DISABLE) or the IFGPSEDI member does not exist	The enhanced data integrity function is inactive. Application processing continues as before.

We are currently running with enhanced data integrity in WARN mode. We have entries in the exclude list for the DFSMShsm journal, DFSMSrmm journal, IMS, and JES3 checkpoint data sets.

Example: The following is an example of the contents of our SYS1.PARMLIB(IFGPSEDI) member:

```
MODE(WARN)                /* PS ENHANCED DATA INTEGRITY - WARNING MODE
DSN(D10.PET.DFRMM.JRN)    /* EXCLUDE LIST
DSN(HSM.PET.JRNL)
DSN(DBS%.IMS%.*)
```

```
DSN(DBS%.DSW4.**)
DSN(SYS1.PET.JES3*)
DSN(SYS1.CDS1%)
DSN(SYS1.*.CDS1%)
```

For more information about setting up this new function, see *z/OS Migration*.

Using the new XCF REALLOCATE process

After a coupling facility structure is allocated, the system makes no attempt to optimize the placement of that structure with respect to the installation's wishes (as expressed in the active CFRM policy). Yet, there are many reasons why, over time, structures that were initially allocated in their installation-desired coupling facilities can and do move to other sub-optimal (from the installation's standpoint) coupling facilities. The use of existing commands to restore structures to their desired locations is a complex and often error-prone process, especially in cases where structures support user-managed or system-managed duplexing and the installation has three or more coupling facilities.

APAR OA03481 addresses this problem by providing a new XCF REALLOCATE process which is controlled by enhanced SETXCF commands. The lowest release that supports this enhancement is z/OS V1R4. In order to exploit the enhancement, the PTF for OA03481 must be installed prior to IPLing the system into the sysplex. All systems in the sysplex must support the REALLOCATE process in order to exploit this enhancement.

The REALLOCATE process will **not** be started when XCF discovers an active system in the sysplex which does not have support for the REALLOCATE process installed. If a REALLOCATE process is in progress and a system without support for the REALLOCATE process joins the sysplex, processing will terminate immediately, allowing the structure that is currently in rebuild processing to complete the current process but no additional processing (for example, reduplexing of the structure) will occur.

The intended use of the REALLOCATE process is to simplify coupling facility-related procedures, such as the following:

- To move structures out of a coupling facility following a CFRM policy change that deletes or changes that coupling facility (for example, in preparation for a coupling facility upgrade)
- To move structures back into a coupling facility following a CFRM policy change that adds or restores the coupling facility (for example, following a coupling facility upgrade or addition)
- To clean up pending CFRM policy changes that may have accumulated for whatever reason, even in the absence of a need to actually relocate any structures
- To clean up simplex or duplexed structures that were allocated in or moved into the "wrong" coupling facilities for whatever reason (for example, the "right" coupling facility was inaccessible at the time of allocation)
- To clean up duplexed structures that have their primary and secondary instances ″reversed″ due to a prior condition that resulted in stopping duplexing with KEEP=NEW and the structure reduplexed

The REALLOCATE process uses existing XCF structure allocation algorithms to recognize the need to relocate structure instances by comparing each structure's current location with the location selected by allocation criteria using either the active or pending CFRM policy. When the locations differ or a policy change is

pending, the REALLOCATE process uses the structure rebuild process to make the necessary adjustments. The following types of structure rebuild processes are supported:

- user-managed rebuild
- user-managed duplexing rebuild
- system-managed rebuild
- system-managed duplexing rebuild

The following SETXCF operator commands have been enhanced to support the REALLOCATE process:

- **SETXCF START,REALLOCATE**

  This command starts the REALLOCATE process. Each allocated structure (simplex or duplexed) is evaluated. Once selected as the target of the REALLOCATE process, predetermined steps are used to relocate the instances or activate a pending policy change. Messages IXC543I, IXC544I, IXC545I, and IXC546I are issued for tracking the REALLOCATE process. Existing messages IXC52*n*I and IXC57*n*I are issued for structure rebuild processing. Message IXC574I is written to the log with the evaluation information for a structure.
- **SETXCF STOP,REALLOCATE**

  This command stops the REALLOCATE process after processing of the current target structure completes.
- **SETXCF STOP,REALLOCATE,FORCE**

  This commands immediately stops the REALLOCATE process. The structure rebuild processing for the target structure is allowed to complete but, for a duplexed structure, the steps to relocate or reduplex the structure may not be done. The FORCE option should be used when structure rebuild processing for the target structure is not making progress.

The remainder of this topic shows several examples of using the enhanced SETXCF commands to control the REALLOCATE process and the resulting messages.

Example: The following is an example of issuing the SETXCF START,REALLOCATE command from a system with APAR OA03481 installed that is in a sysplex with at least one down-level system (that is, a system at a release level lower than z/OS V1R4 or that does not have APAR OA03481 installed).

```
12:59:14.93 MARTHA   00000200  SETXCF START,REALLOCATE
12:59:14.97 MARTHA   01000000  IXC543I THE REQUESTED START,REALLOCATE WAS REJECTED
                 395 01000000  AT LEAST ONE SYSTEM DOES NOT SUPPORT THE REALLOCATE PROCESS
```

Example: The following is an example of the response from issuing the SETXCF START,REALLOCATE command from a system at a release level lower than z/OS V1R4 or that does not have APAR OA03481 installed.

```
13:19:24.75 MARTHA   01000000  SETXCF SYNTAX ERROR, COULD NOT RECOGNIZE: REALLOCATE.
                 839 01000000  ONE OF THE FOLLOWING WAS EXPECTED:
                 839 01000000    PATHIN PATHOUT CLASSDEF POLICY
                 839 01000000    REBUILD ALTER
```

Example: When you issue the SETXCF START,REALLOCATE command in a sysplex in which all systems have APAR OA03481 installed, REALLOCATE processing begins. The following log excerpts show the results for the case where changes were made to the preference list in the CFRM policy and the REALLOCATE process was used to process and complete the pending policy changes. Commentary has been inserted as appropriate to describe the progress of the REALLOCATE processing.

```
SETXCF START,REALLOCATE
IXC543I THE REQUESTED START,REALLOCATE WAS ACCEPTED.
```

(Duplexed structure DSNDB1G_LOCK1 is evaluated.)

```
IXC574I EVALUATION INFORMATION FOR REALLOCATE PROCESSING 486
OF STRUCTURE DSNDB1G_LOCK1
  DUPLEXED STRUCTURE ALLOCATED IN COUPLING FACILITY: CF2      CF3
  ACTIVE POLICY INFORMATION USED.
 CFNAME     STATUS/FAILURE REASON
 --------   --------------------
 CF2        PREFERRED CF 1
                                  INFO110: 00000064 CC007B00 0000000D
 CF3        PREFERRED CF 2
                                  INFO110: 00000064 CC007B00 0000000D
 CF1        PREFERRED CF ALREADY SELECTED
                                  INFO110: 00000064 CC007B00 0000000D
```

(The structure is not selected as a target structure.)

```
IXC544I REALLOCATE PROCESSING FOR STRUCTURE DSNDB1G_LOCK1 487
WAS NOT ATTEMPTED BECAUSE
STRUCTURE IS ALLOCATED IN PREFERRED CF

IXC574I EVALUATION INFORMATION FOR REALLOCATE PROCESSING 488
OF STRUCTURE DSNDB1G_GBP0
  DUPLEXED STRUCTURE ALLOCATED IN COUPLING FACILITY: CF3      CF2
  ACTIVE POLICY INFORMATION USED.
 CFNAME     STATUS/FAILURE REASON
 --------   --------------------
 CF3        PREFERRED CF 1
                                  INFO110: 00000064 CC007800 0000000D
 CF2        PREFERRED CF 2
                                  INFO110: 00000064 CC007800 0000000D
 CF1        PREFERRED CF ALREADY SELECTED
                                  INFO110: 00000064 CC007800 0000000D
IXC544I REALLOCATE PROCESSING FOR STRUCTURE DSNDB1G_GBP0 489
WAS NOT ATTEMPTED BECAUSE
STRUCTURE IS ALLOCATED IN PREFERRED CF

IXC574I EVALUATION INFORMATION FOR REALLOCATE PROCESSING 490
OF STRUCTURE DSNDB1G_GBP1
  DUPLEXED STRUCTURE ALLOCATED IN COUPLING FACILITY: CF2      CF3
  ACTIVE POLICY INFORMATION USED.
 CFNAME     STATUS/FAILURE REASON
 --------   --------------------
 CF2        PREFERRED CF 1
                                  INFO110: 0000008C AC007800 0000000D
 CF3        PREFERRED CF 2
                                  INFO110: 0000008C AC007800 0000000D
 CF1        PREFERRED CF ALREADY SELECTED
                                  INFO110: 0000008C AC007800 0000000D
IXC544I REALLOCATE PROCESSING FOR STRUCTURE DSNDB1G_GBP1 491
WAS NOT ATTEMPTED BECAUSE
THE STRUCTURE HAS NO ACTIVE CONNECTORS

IXC574I EVALUATION INFORMATION FOR REALLOCATE PROCESSING 492
OF STRUCTURE DSNDB1G_GBP2
  DUPLEXED STRUCTURE ALLOCATED IN COUPLING FACILITY: CF3      CF2
  ACTIVE POLICY INFORMATION USED.
 CFNAME     STATUS/FAILURE REASON
 --------   --------------------
 CF3        PREFERRED CF 1
                                  INFO110: 00000046 CC007800 0000000D
 CF2        PREFERRED CF 2
                                  INFO110: 00000046 CC007800 0000000D
```

```
 CF1        PREFERRED CF ALREADY SELECTED
                                    INFO110: 00000046 CC007800 0000000D
IXC544I REALLOCATE PROCESSING FOR STRUCTURE DSNDB1G_GBP2 493
WAS NOT ATTEMPTED BECAUSE
STRUCTURE IS ALLOCATED IN PREFERRED CF
   ⋮
```

(Additional log output omitted to condense the display.)

```
IXC574I EVALUATION INFORMATION FOR REALLOCATE PROCESSING 509
OF STRUCTURE DSNDB1G_SCA
  DUPLEXED STRUCTURE ALLOCATED IN COUPLING FACILITY: CF2      CF3
  PENDING POLICY INFORMATION USED.
```

(Structure DSNDB1G_SCA is evaluated using the pending policy. The REALLOCATE process will cause the pending policy change to happen.)

```
 CFNAME     STATUS/FAILURE REASON
 --------   --------------------
 CF2        PREFERRED CF 1
                                    INFO110: 00000064 CC007B00 0000000D
 CF3        PREFERRED CF 2
                                    INFO110: 00000064 CC007B00 0000000D
```

(Structure DSNDB1G_SCA becomes the target structure.)
(Step 1 — Stop duplexing.)

```
IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 510
DSNDB1G_SCA IS BEING STOPPED
TO FALL BACK TO THE OLD STRUCTURE DUE TO
REQUEST FROM AN OPERATOR
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 511
DSNDB1G_SCA HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
AND IS ENTERING THE QUIESCE FOR STOP PHASE.
 TIME: 06/16/2004 09:59:04.236728
 AUTO VERSION: BB51F5FD F1E7C385
   ⋮
```

(Additional log output omitted....)

```
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 946
DSNDB1G_SCA HAS COMPLETED THE QUIESCE FOR STOP PHASE
AND IS ENTERING THE STOP PHASE.
 TIME: 06/16/2004 09:59:06.102207
 AUTO VERSION: BB51F5FD F1E7C385
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 947
BEEN STOPPED FOR STRUCTURE DSNDB1G_SCA
STRUCTURE NOW IN COUPLING FACILITY CF2
 PHYSICAL STRUCTURE VERSION: BB47CF06 640AC8B5
 LOGICAL STRUCTURE VERSION: BB47CF06 640AC8B5
 AUTO VERSION: BB51F5FD F1E7C385
   ⋮
```

(Additional log output omitted....)

```
IXC579I PENDING DEALLOCATION FOR STRUCTURE DSNDB1G_SCA IN 948
        COUPLING FACILITY  002064.IBM.02.00000002A48A
                           PARTITION: 04    CPCID: 00
HAS BEEN COMPLETED.
PHYSICAL STRUCTURE VERSION: BB51F601 0CC58370
```

```
INFO116: 13088068 01 6A00 00000014
   :
```

(Additional log output omitted....)

(Step 2 — Rebuild the structure. As the structure exists in the preferred CF, rebuild is in place to cause the policy change to take effect.)

```
IXC521I REBUILD FOR STRUCTURE DSNDB1G_SCA 950
HAS BEEN STARTED
IXC526I STRUCTURE DSNDB1G_SCA IS REBUILDING FROM 935
COUPLING FACILITY CF2 TO COUPLING FACILITY CF2.
REBUILD START REASON: OPERATOR INITIATED
INFO108: 00000064 00000064.
IXL014I IXLCONN REBUILD REQUEST FOR STRUCTURE DSNDB1G_SCA 936
WAS SUCCESSFUL.  JOBNAME: DB91MSTR ASID: 017D
 CONNECTOR NAME: DB2_DB91 CFNAME: CF2
IXL015I REBUILD NEW STRUCTURE ALLOCATION INFORMATION FOR 937
STRUCTURE DSNDB1G_SCA, CONNECTOR NAME DB2_DB91
 CFNAME     ALLOCATION STATUS/FAILURE REASON
 --------   --------------------------------
 CF2        STRUCTURE ALLOCATED CC007B00
 CF3        PREFERRED CF ALREADY SELECTED CC007B00
   :
```

(Additional log output omitted....)

```
IXC521I REBUILD FOR STRUCTURE DSNDB1G_SCA 140
HAS BEEN COMPLETED
   :
```

(Additional log output omitted....)

```
SCA STRUCTURE DSNDB1G_SCA REBUILD SUCCESSFUL.
IXC579I PENDING DEALLOCATION FOR STRUCTURE DSNDB1G_SCA IN 141
        COUPLING FACILITY  002084.IBM.00.00000001B52A
                           PARTITION: 23    CPCID: 00
HAS BEEN COMPLETED.
PHYSICAL STRUCTURE VERSION: BB47CF06 640AC8B5
INFO116: 13088068 01 6A00 00000001
TRACE THREAD: 00029DA7.
   :
```

(Additional log output omitted....)

(Step 3 — Re-duplex the structure.)

```
IXC570I SYSTEM-MANAGED DUPLEXING REBUILD STARTED FOR STRUCTURE 148
DSNDB1G_SCA IN COUPLING FACILITY CF2
 PHYSICAL STRUCTURE VERSION: BB60B035 01AF4B24
 LOGICAL STRUCTURE VERSION: BB60B035 01AF4B24
 START REASON:  OPERATOR-INITIATED
 AUTO VERSION: BB60B046 8526F7A7
   :
```

(Additional log output omitted....)

(Simplex structure IXCPLEX_PATH4 is evaluated.)

```
IXC574I EVALUATION INFORMATION FOR REALLOCATE PROCESSING 656
OF STRUCTURE IXCPLEX_PATH4
  SIMPLEX STRUCTURE ALLOCATED IN COUPLING FACILITY: CF1
  ACTIVE POLICY INFORMATION USED.
 CFNAME     STATUS/FAILURE REASON
 --------   --------------------
```

```
 CF1        PREFERRED CF 1
                                  INFO110: 0000008C CC007800 0000000D
 CF3        PREFERRED CF ALREADY SELECTED
                                  INFO110: 0000008C CC007800 0000000D
 CF2        PREFERRED CF ALREADY SELECTED
                                  INFO110: 0000008C CC007800 0000000D
```

(The structure is not selected as a target structure.)

```
IXC544I REALLOCATE PROCESSING FOR STRUCTURE IXCPLEX_PATH4 657
WAS NOT ATTEMPTED BECAUSE
STRUCTURE IS ALLOCATED IN PREFERRED CF
   ⋮
```

(Additional log output omitted....)

(Simplex structure IRLMLOCKT is evaluated.)

```
 IXC574I EVALUATION INFORMATION FOR REALLOCATE PROCESSING 997
 OF STRUCTURE IRLMLOCKT
   SIMPLEX STRUCTURE ALLOCATED IN COUPLING FACILITY: CF1
   PENDING POLICY INFORMATION USED.
```

(Structure IRLMLOCKT is evaluated using the pending policy. Based on the pending policy change, the structure is not allocated in the most preferred CF; therefore, reallocation is required.)

```
  CFNAME     STATUS/FAILURE REASON
  --------   --------------------
  CF3        PREFERRED CF 1
                                   INFO110: 0000008C AC007800 0000000D
  CF2        PREFERRED CF ALREADY SELECTED
                                   INFO110: 0000008C AC007800 0000000D
```

(Structure IRLMLOCKT becomes the target structure as it is currently allocated in a less-preferred CF.)
(Start rebuild for the structure.)

```
 IXC570I SYSTEM-MANAGED REBUILD STARTED FOR STRUCTURE 998
 IRLMLOCKT IN COUPLING FACILITY CF1
  PHYSICAL STRUCTURE VERSION: BB3DBE1C C419E1AC
  LOGICAL STRUCTURE VERSION: BB3DBE1C C419E1AC
  START REASON:  OPERATOR-INITIATED
  AUTO VERSION: BB60B0A4 24906247
 IXC578I SYSTEM-MANAGED REBUILD SUCCESSFULLY ALLOCATED 999
 STRUCTURE IRLMLOCKT.
OLD COUPLING FACILITY: CF1
OLD PHYSICAL STRUCTURE VERSION: BB3DBE1C C419E1AC
NEW COUPLING FACILITY: CF3
NEW PHYSICAL STRUCTURE VERSION: BB60B0A4 5DDCFD25
LOGICAL STRUCTURE VERSION: BB3DBE1C C419E1AC
 AUTO VERSION: BB60B0A4 24906247
IXC574I ALLOCATION INFORMATION FOR SYSTEM-MANAGED REBUILD 000
OF STRUCTURE IRLMLOCKT
 AUTO VERSION: BB60B0A4 24906247
 CFNAME     STATUS/FAILURE REASON
 --------   --------------------
 CF3        STRUCTURE ALLOCATED
                                  INFO110: 0000008C AE007800 00000000
 CF2        PREFERRED CF ALREADY SELECTED
                                  INFO110: 0000008C AE007800 0000000D
IXC571I SYSTEM-MANAGED REBUILD FOR STRUCTURE 001
IRLMLOCKT HAS COMPLETED THE ALLOCATION PHASE
AND IS ENTERING THE COPY PHASE.
 TIME: 06/16/2004 10:01:17.269945
 AUTO VERSION: BB60B0A4 24906247
IXC572I SYSTEM-MANAGED REBUILD FOR STRUCTURE 610
IRLMLOCKT HAS COMPLETED THE INITIALIZATION
```

```
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
ATTACH SUBPHASE.
 TIME: 06/16/2004 10:01:17.424153
 AUTO VERSION: BB60B0A4 24906247
IXC572I SYSTEM-MANAGED REBUILD FOR STRUCTURE 611
IRLMLOCKT HAS COMPLETED THE ATTACH
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
LIST SUBPHASE.
 TIME: 06/16/2004 10:01:17.438925
 AUTO VERSION: BB60B0A4 24906247
IXC572I SYSTEM-MANAGED REBUILD FOR STRUCTURE 171
IRLMLOCKT HAS COMPLETED THE LIST
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
EXIT SUBPHASE.
 TIME: 06/16/2004 10:01:17.813450
 AUTO VERSION: BB60B0A4 24906247
   ⋮
```

(Additional log output omitted....)

```
IXC579I NORMAL DEALLOCATION FOR STRUCTURE IRLMLOCKT IN 612
        COUPLING FACILITY  002086.IBM.02.00000002AE1A
                           PARTITION: 01    CPCID: 00
HAS BEEN COMPLETED.
PHYSICAL STRUCTURE VERSION: BB3DBE1C C419E1AC
INFO116: 130AA000 FF FFFF 00000010
TRACE THREAD: 0009C98B.
IXC577I SYSTEM-MANAGED REBUILD HAS 613
BEEN COMPLETED FOR STRUCTURE IRLMLOCKT
STRUCTURE NOW IN COUPLING FACILITY CF3
 PHYSICAL STRUCTURE VERSION: BB3DBE1C C419E1AC
 LOGICAL STRUCTURE VERSION: BB3DBE1C C419E1AC
 AUTO VERSION: BB60B0A4 24906247
IXC580I SYSTEM-MANAGED REBUILD OF STRUCTURE IRLMLOCKT 614
AUTO VERSION: BB60B0A4 24906247
RESULTED IN THE FOLLOWING STRUCTURE ATTRIBUTES:
  PHYSICAL STRUCTURE VERSION: BB3DBE1C C419E1AC
  LOGICAL STRUCTURE VERSION : BB3DBE1C C419E1AC
  CURRENT SIZE              :       8192 K
  CURRENT ENTRY COUNT       :      11531
  CURRENT ELEMENT COUNT     :          0
  CURRENT EMC COUNT         :          0
   ⋮
```

(Additional log output omitted....)

(Summary of REALLOCATE processing. Note that some of the structure processing referenced in the summary was omitted from this example display.)

```
IXC545I REALLOCATE PROCESSING RESULTED IN THE FOLLOWING: 527
        10  STRUCTURE(S) REALLOCATED - SIMPLEX
        11  STRUCTURE(S) REALLOCATED - DUPLEXED
         0  STRUCTURE(S) POLICY CHANGE MADE - SIMPLEX
        17  STRUCTURE(S) POLICY CHANGE MADE - DUPLEXED
        48  STRUCTURE(S) ALREADY ALLOCATED IN PREFERRED CF - SIMPLEX
        43  STRUCTURE(S) ALREADY ALLOCATED IN PREFERRED CF - DUPLEXED
         5  STRUCTURE(S) NOT PROCESSED
       167  STRUCTURE(S) NOT ALLOCATED
        19  STRUCTURE(S) NOT DEFINED
  --------
       320  TOTAL

         0  ERROR(S) ENCOUNTERED DURING PROCESSING
```

(REALLOCATE processing complete.)

```
IXC543I THE REQUESTED START,REALLOCATE WAS COMPLETED.
```

Example: The following is an example of issuing the DISPLAY XCF,CF,CFNAME=ALL command during REALLOCATE processing :

```
000200  D XCF,CF,CFNAME=ALL
000000  IXC362I  09.48.50  DISPLAY XCF 519
000000  THE REALLOCATE PROCESS IS IN PROGRESS.
000000  CFNAME: CF1
000000    COUPLING FACILITY     :   002086.IBM.02.00000002AE1A
000000                              PARTITION: 01   CPCID: 00
000000    POLICY DUMP SPACE SIZE:   6000 K
000000    ACTUAL DUMP SPACE SIZE:   N/A
000000    STORAGE INCREMENT SIZE:   N/A
000000
000000    NO SYSTEMS ARE CONNECTED TO THIS COUPLING FACILITY
000000
000000    NO STRUCTURES ARE IN USE BY THIS SYSPLEX IN THIS COUPLING FACILITY
000000  CFNAME: CF2
000000    COUPLING FACILITY     :   002084.IBM.00.00000001B52A
000000                              PARTITION: 23   CPCID: 00
000000    POLICY DUMP SPACE SIZE:   6000 K
000000    ACTUAL DUMP SPACE SIZE:   6144 K
000000    STORAGE INCREMENT SIZE:   256 K
000000
000000    CONNECTED SYSTEMS:
000000    JA0      JB0      JC0      JE0      JF0      JG0      JH0
000000    J80      J90      TPN      Z0       Z1       Z2       Z3
000000
000000    STRUCTURES:
000000    COUPLE_CKPT1(NEW)      CQS_FF_LOGSTR(NEW)     CQS_FP_LOGSTR(OLD)
000000    DFHCFLS_CFDTA(NEW)     DFHNCLS_DFHNC001(NEW)  DSNDB1G_LOCK1(OLD)
000000    DSNDB1G_SCA            DSNDB2G_LOCK1(NEW)     DSNDB2G_SCA(NEW)
000000    IGWLOCK00(NEW)         ISTGENERIC(NEW)        ISTGENERIC_TEST(NEW)
000000    ISTMNPS(NEW)           LOGGER_OPERLOG(NEW)    SYSARC_DFHSM_RCL(NEW)
000000    SYSZWLM_ACBA2086(NEW)  SYSZWLM_A48A2064(NEW)  SYSZWLM_B52A2084(NEW)
000000    SYSZWLM_WORKUNIT(NEW)  SYSZWLM_1CA52066(NEW)
000000  CFNAME: CF3
000000    COUPLING FACILITY     :   002064.IBM.02.00000002A48A
000000                              PARTITION: 04   CPCID: 00
000000    POLICY DUMP SPACE SIZE:   6000 K
000000    ACTUAL DUMP SPACE SIZE:   6144 K
000000    STORAGE INCREMENT SIZE:   256 K
000000
000000    CONNECTED SYSTEMS:
000000    JA0      JB0      JC0      JE0      JF0      JG0      JH0
000000    J80      J90      TPN      Z0       Z1       Z2       Z3
000000
000000    STRUCTURES:
000000    COUPLE_CKPT1(OLD)      CQS_FF_LOGSTR(OLD)     CQS_FP_LOGSTR(NEW)
   ⋮
```

(Additional log output omitted....)

Example: The following is an example of issuing the DISPLAY XCF,STR,STRNAME=*,CONNAME=ALL command during REALLOCATE processing.

```
000200  D XCF,STR,STRNAME=*,CONNAME=ALL
000000  IXC360I  09.49.07  DISPLAY XCF 532
000000  THE REALLOCATE PROCESS IS IN PROGRESS.
000000  STRNAME: APPCLOG
000000   STATUS: NOT ALLOCATED
000000   POLICY INFORMATION:
   ⋮
```

(Additional log output omitted....)

```
000000  STRNAME: CSLRMGR_PROD
000000   STATUS: ALLOCATED
000000           REALLOCATE EVALUATION PENDING
000000   TYPE: SERIALIZED LIST
000000   POLICY INFORMATION:
000000    POLICY SIZE    : 32000 K
000000    POLICY INITSIZE: 20000 K
000000    POLICY MINSIZE : 15000 K
000000    FULLTHRESHOLD  : 60
000000    ALLOWAUTOALT   : YES
000000    REBUILD PERCENT: N/A
000000    DUPLEX         : ALLOWED
000000    PREFERENCE LIST: CF2      CF1      CF3
000000    ENFORCEORDER   : NO
000000    EXCLUSION LIST IS EMPTY
000000   ACTIVE STRUCTURE
000000   ----------------
000000    ALLOCATION TIME: 05/17/2004 21:37:02
000000    CFNAME         : CF3
000000    COUPLING FACILITY: 002064.IBM.02.00000002A48A
000000                       PARTITION: 04   CPCID: 00
000000    ACTUAL SIZE    : 20224 K
000000    STORAGE INCREMENT SIZE: 256 K
000000    PHYSICAL VERSION: BB3B9434 0EEDFB40
000000    LOGICAL  VERSION: BB3B9434 0EEDFB40
000000    SYSTEM-MANAGED PROCESS LEVEL: 9
000000    XCF GRPNAME    : IXCLO0E1
000000    DISPOSITION    : KEEP
000000    ACCESS TIME    : NOLIMIT
000000    MAX CONNECTIONS: 32
000000    # CONNECTIONS  : 1
   :
   :
```

(Additional log output omitted....)

Example: The following is an example of issuing the DISPLAY XCF,STR command during REALLOCATE processing.

```
IXC359I  13.49.12  DISPLAY XCF 517
THE REALLOCATE PROCESS IS IN PROGRESS.
STRNAME           ALLOCATION TIME   STATUS
APPCLOG         02/25/2004 17:08:37 ALLOCATED
                                    REALLOCATE EVALUATION PENDING
```

(The REALLOCATE process evaluates structures in a serial manner and has not yet reached this structure.)

```
CICS_USERJRN_001 03/03/2004 13:28:14 ALLOCATED
                                     REALLOCATE EVALUATION PENDING
COUPLE_CKPT1     03/02/2004 20:40:48 DUPLEXING REBUILD NEW STRUCTURE
                                     DUPLEXING REBUILD
                                       METHOD     : SYSTEM-MANAGED
                                     REBUILD PHASE: DUPLEX ESTABLISHED
                                     REALLOCATE EVALUATION PENDING
COUPLE_CKPT1     02/24/2004 23:40:51 DUPLEXING REBUILD OLD STRUCTURE
                                     DUPLEXING REBUILD
COUPLE_CKPT2          --       --    NOT ALLOCATED
CQS_FF_LOGSTR    03/03/2004 08:26:14 DUPLEXING REBUILD NEW STRUCTURE
                                     DUPLEXING REBUILD
                                       METHOD     : SYSTEM-MANAGED
                                     REBUILD PHASE: DUPLEX ESTABLISHED
                                     REALLOCATE EVALUATION PENDING
CQS_FF_LOGSTR    03/03/2004 08:26:06 DUPLEXING REBUILD OLD STRUCTURE
                                     DUPLEXING REBUILD
CQS_FF_LOGTEST   03/03/2004 08:24:08 ALLOCATED
                                     REALLOCATE EVALUATION PENDING
```

```
CQS_FP_LOGSTR    03/03/2004 08:26:10 DUPLEXING REBUILD NEW STRUCTURE
                                     DUPLEXING REBUILD
                                       METHOD     : SYSTEM-MANAGED
                                     REBUILD PHASE: DUPLEX ESTABLISH
                                     REALLOCATE EVALUATION PENDING
CQS_FP_LOGSTR    03/03/2004 08:26:03 DUPLEXING REBUILD OLD STRUCTURE
                                     DUPLEXING REBUILD
CQS_FP_LOGTEST   03/03/2004 08:24:10 ALLOCATED
                                     REALLOCATE EVALUATION PENDING
CSLRMGR_PROD     03/03/2004 08:26:00 ALLOCATED
                                     REALLOCATE EVALUATION PENDING
CSLRMGR_TEST     02/02/2004 15:59:24 ALLOCATED
                                     REALLOCATE EVALUATION PENDING
DFHCFLS_CFDTA    03/02/2004 20:44:21 DUPLEXING REBUILD NEW STRUCTURE
                                     DUPLEXING REBUILD
                                        METHOD     : SYSTEM-MANAGED
                                      REBUILD PHASE: DUPLEX ESTABLISHED
                                      REALLOCATE EVALUATION PENDING
  ⋮
```

(Additional log output omitted....)

```
WAS5P1_ERRLOG          --       --    NOT ALLOCATED
WAS5T1_ERRLOG          --       --    NOT ALLOCATED
WAS50TST_ERRLOG        --       --    NOT ALLOCATED
```

Example: The following is an example of issuing the DISPLAY XCF,STR,STRNAME=COUPLE_CKPT1 command during REALLOCATE processing.

```
STRNAME: COUPLE_CKPT1
 STATUS: REASON SPECIFIED WITH REBUILD START:
           POLICY-INITIATED
         DUPLEXING REBUILD
           METHOD        : SYSTEM-MANAGED
             AUTO VERSION: BADC074A A28C45A1
         REBUILD PHASE: DUPLEX ESTABLISHED
         REALLOCATE EVALUATION PENDING
 TYPE: SERIALIZED LIST
 POLICY INFORMATION:
  POLICY SIZE    : 50560 K
  POLICY INITSIZE: N/A
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 95
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: N/A
  DUPLEX         : ENABLED
```

Example: The following is an example of issuing the SETXCF STOP,REALLOCATE command.

```
14:00:39.68 TP0D1    00000200  SETXCF STOP,REALLOCATE
14:00:39.79          00000000  IXC543I THE REQUESTED STOP,REALLOCATE WAS ACCEPTED
```

Example: The following is an example of issuing the SETXCF STOP,REALLOCATE,FORCE command.

```
17:43:19.08 IBMUSR8  00000200  SETXCF STOP,REALLOCATE,FORCE
SETXCF STOP,REALLOCATE,FORCE
IXC543I THE REQUESTED STOP,REALLOCATE,FORCE WAS ACCEPTED. 684
IXC545I REALLOCATE PROCESSING RESULTED IN THE FOLLOWING: 685
         0  STRUCTURE(S) REALLOCATED - SIMPLEX
         1  STRUCTURE(S) REALLOCATED - DUPLEXED
         0  STRUCTURE(S) POLICY CHANGE MADE - SIMPLEX
         0  STRUCTURE(S) POLICY CHANGE MADE - DUPLEXED
         0  STRUCTURE(S) ALREADY ALLOCATED IN PREFERRED CF - SIMPLEX
         2  STRUCTURE(S) ALREADY ALLOCATED IN PREFERRED CF - DUPLEXED
         0  STRUCTURE(S) NOT PROCESSED
         0  STRUCTURE(S) NOT ALLOCATED
```

```
        3  STRUCTURE(S) NOT DEFINED
  --------
        6  TOTAL
        0  ERROR(S) ENCOUNTERED DURING PROCESSING
IXC543I THE REQUESTED STOP,REALLOCATE,FORCE WAS COMPLETED. 686
```

Migrating to z/OS.e V1R5

This section describes our migration experiences with z/OS.e V1R5.

z/OS.e V1R5 base migration experiences

This section describes our experiences with migrating one system image (JH0) from z/OS.e V1R4 to z/OS.e V1R5. Here we only cover our experiences with our base migration to z/OS.e V1R5, including our high-level migration process and other migration activities and considerations.

Our high-level migration process for z/OS.e V1R5

The following is an overview of our z/OS.e V1R5 migration process.

Before we began: We reviewed the information in *z/OS and z/OS.e Planning for Installation*, GA22-7504, which covers both z/OS V1R5 and z/OS.e V1R5.

Important notice about cloning and software licensing

As discussed in *z/OS and z/OS.e Planning for Installation*, you might find that sharing system libraries or cloning an already-installed z/OS or z/OS.e system is faster and easier than installing z/OS or z/OS.e with an IBM installation package such as ServerPac. Most Parallel Sysplex customers are already aware of the concept of cloning and the benefits it provides.

However, prior to sharing or cloning z/OS or z/OS.e, **you must have a license for each z/OS and z/OS.e operating system that you run.** If you don't have the appropriate license or licenses, you must contact IBM. Any sharing or cloning of z/OS or z/OS.e without the appropriate licenses is not an authorized use of such programs. On a z800 server, if you want to run both z/OS and z/OS.e, z/OS requires the appropriate license for the machine on which it runs and z/OS.e requires a license for the number of engines on which it runs.

For more information about z/OS.e licensing, see *z800 Software Pricing Configuration Technical Paper* at www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130121.pdf.

Table 11 shows the high-level process we followed to migrate our z/OS.e V1R4 system to z/OS.e V1R5.

Table 11. Our high-level migration process for z/OS.e V1R5

Stage	Description
Obtaining licenses for z/OS.e	You need a license for the appropriate number of engines on the z800 server on which you intend to run z/OS.e (and, you would also need a license to run z/OS on the z800, if you intend to install it there). We use an internal process to do this; however, you must use the official process stated in *z800 Software Pricing Configuration Technical Paper*.

Table 11. Our high-level migration process for z/OS.e V1R5 (continued)

Stage	Description
Updating the z800 LPAR name	z/OS.e must run in LPAR mode and the LPAR name must be of the form ZOSE*xxxx*, where *xxxx* is up to 4 user-specified alphanumeric characters. The name of the LPAR in which we run z/OS.e is ZOSEJH0. (We used HCD to set this when we first installed z/OS.e V1R3.)
Updating parmlib for z/OS.e V1R5	z/OS.e requires the LICENSE=Z/OSE statement in the IEASYS*xx* parmlib member. We used the same SYS1.PETR15.PARMLIB data set that we created for z/OS V1R5. We then have separate IEASYS*xx* and IFAPRD*xx* members in SYS1.PARMLIB that we tailored specifically for z/OS.e. See “Updating system data sets for z/OS.e” for details.
Updating our LOAD*xx* member	During our initial installation of z/OS.e V1R3, we updated the entry for our system JH0 in our LOAD*xx* member in SYS0.IPLPARM to point to our new IEASYS02 parmlib member and to reflect the new LPAR name. Therefore, we did not need to change it for V1R5.
Updating our IEASYMPT member	During our initial installation of z/OS.e V1R3, we updated the entry for our system JH0 in our IEASYMPT member in SYS1.PETR13.PARMLIB to point to our new IFAPRD*xx* parmlib member and to reflect the new LPAR name. Therefore, when we created our new SYS1.PETR15.PARMLIB, we carried the change along for V1R5.
IPLing the z/OS.e V1R5 image	We brought up z/OS.e V1R5 on our JH0 production system.

More about our migration activities for z/OS.e V1R5

This section highlights additional details about some of our migration activities.

About our z800 LPAR environment: z/OS.e must run in LPAR mode on a zSeries 800 mainframe server; it cannot run in basic mode. In addition, the name of the LPAR in which z/OS.e runs must be of the form ZOSE*xxxx*, where *xxxx* is up to four user-specified alphanumeric characters. The name of our z/OS.e LPAR is ZOSEJH0.

Note: You can only run z/OS.e in a partition named ZOSE*xxxx*. You cannot IPL a z/OS system in a partition named ZOSE*xxxx*.

We currently run z/OS.e (JH0) in a mixed LPAR environment alongside LPARs running z/OS (JG0) and z/VM (PETVM2) on the same z800 server.

Note: Don’t let the fact that z/OS.e only runs on a z800 server confuse you. A z800 is a fully functional zSeries server and, in addition to z/OS.e, it supports all of the same zSeries operating systems as a z900 or z990 server.

Updating system data sets for z/OS.e: We continue to use concatenated parmlib support to add or update parmlib members for z/OS.e V1R5. We use the same SYS1.PETR15.PARMLIB data set as we do for our z/OS V1R5 systems.

Below are examples of our parmlib customizations to accommodate z/OS.e V1R5. Appendix A, “Some of our parmlib members,” on page 193 summarizes the changes we made by parmlib member.

Example: We have a separate IEASYS*xx* member, IEASYS02, which specifies the `LICENSE=Z/OSE` statement that z/OS.e requires.

The entry for our z/OS.e system (JH0) in our LOAD*xx* member in SYS0.IPLPARM points to our IEASYS02 parmlib member and specifies the name of our z/OS.e LPAR, as follows:

```
⋮
HWNAME   z800name
LPARNAME  ZOSEJH0
PARMLIB  SYS1.PETR15.PARMLIB
SYSPARM   02
⋮
```

Example: We have a separate IFAPRD*xx* member, IFAPRD02, which specifies the product ID value `5655-G52` for z/OS.e. There is no change to the product name value for z/OS.e (the product name value remains `Z/OS`).

Below is an example of one of the entries from our IFAPRD02 member:

```
⋮
PRODUCT OWNER('IBM CORP')
        NAME(Z/OS)
        ID( 5655-G52 )
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(Z/OS)
        STATE(ENABLED)
⋮
```

We also have an entry for our system JH0 in our IEASYMPT member in SYS1.PETR15.PARMLIB to point to our new IFAPRD02 parmlib member and to reflect the z/OS.e LPAR name, as follows:

```
⋮
SYSDEF HWNAME(z800name)
       LPARNAME( ZOSEJH0 )
       SYSNAME(JH0)
       SYSCLONE(JH)
⋮
       SYMDEF(&PROD= '02' )
⋮
```

Using current z/OS.e levels of JES2 and LE: As required, we are using the level of JES2 and Language Environment (LE) that comes with z/OS.e V1R5. z/OS.e does not permit the use of a lower level JES2 (or JES3) or LE.

Updating the ARM policy: You must ensure that your automation policies, such as ARM, do not try to use a z/OS.e image to start products that z/OS.e does not support. For example, do not identify a z/OS.e image as a restart target in a Parallel Sysplex that contains a mix of z/OS.e and z/OS images where the z/OS images run IMS, CICS, or DB2 with a requirement for CICS. CICS, IMS, or DB2 that uses CICS cannot restart on a z/OS.e image, but must restart on a z/OS image. If, for example, a CICS region attempts to start on z/OS.e, the region will start but the applications will fail with a U4093 abend.

Back when we installed z/OS.e V1R3, we removed our z/OS.e image, JH0, as a restart target for the unsupported subsystems mentioned above.

Removing z/OS.e from participation in MNPS: In our environment, CICS is the only exploiter of multiple node persistent sessions (MNPS) support. Because CICS cannot run on z/OS.e, there is no reason for the VTAM on z/OS.e to connect to the MNPS structure, ISTMNPS. We removed our z/OS.e image from participating in MNPS by coding the `STRMNPS=NONE` statement in our VTAM start member, ATCSTR*xx*, in SYS1.VTAMLST.

Removing z/OS.e from participation in TSO generic resource groups: Since TSO on z/OS.e only allows a maximum of eight concurrent sessions, we removed our z/OS.e image from participating in TSO generic resource groups. You can do this by coding the `GNAME=NONE` parameter—either in a separate TSOKEY*xx* member in parmlib or on the START command that starts the terminal control address space (TCAS).

In our case, we use a single TSOKEY*xx* member that has a symbolic value for the GNAME parameter. We then set that symbol to `NONE` for our JH0 image in our IEASYMPT member.

Other experiences with z/OS.e V1R5

Our testing of z/OS.e V1R5 included the following workloads or scenarios:

- z/OS UNIX System Services
- DB2 UDB
- IBM HTTP Server in scalable server mode
- WebSphere Application Server for z/OS
- CICS Transaction Gateway (CTG) to access CICS regions running in z/OS images on the same CPC and other CPCs
- DB2 access from Linux guests under z/VM on the same CPC
- our Bookstore application transactions

Using the IBM Health Checker for z/OS and Sysplex

The IBM Health Checker for z/OS and Sysplex is a tool that checks the current, active z/OS and sysplex settings and definitions for an image and compares their values to those either suggested by IBM or defined by the installation as the criteria. The objective of the Health Checker is to identify potential problems before they impact system availability or, in the worst cases, cause outages.

We are using Version 3 of the IBM Health Checker for z/OS and Sysplex, which we downloaded from the z/OS downloads page at www.ibm.com/servers/eserver/zseries/zos/downloads/. The documentation, *z/OS and Sysplex Health Checker User's Guide*, SA22-7931, is also available on this Web page.

Using the default USERPARM member supplied with the Health Checker, we created and customized several new members to perform various types of checking. For instance, we use one member with one set of parameters to perform XCF checks, another member with different parameters to perform APF and LINKLST checks, and so on. This also makes the reports easier to look at and use. It also allows us to isolate the checks that examine values that have a sysplex scope so that we can run them on only one z/OS image, rather than on every image in the sysplex—thereby eliminating redundant information from the reports for each image.

z/OS performance

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

The following describes what we do in each phase of our testing, and what we plan to periodically report to you in our test reports:

- Monitor our performance in terms of our service level agreements

  Our goal for our sysplex workloads continues to be 90% CP utilization across the systems in the sysplex, with WLM goals such as 80% of CICS transactions completed in less than 0.6 seconds on those images where CICS runs. We fill in the remaining 10% with batch work and various additional types of users, such as z/OS UNIX users (such as WebSphere for z/OS), TSO users, and workstation clients.

  Note: This is not formal performance testing for purposes of publishing performance statistics for z/OS. It is a way for us to establish and report on reasonable goals for response times and transaction rates for the various types of workloads we run, just as a customer would do to create a service level agreement (SLA).
- Identify performance problems in our environment, find solutions to those problems, and report the information to you.
- Provide you with periodic performance snapshots of our environment, in the form of RMF reports, to provide pertinent information such as how many transactions we process per second and what our response times are for various workloads. You can find those reports in Appendix B, “Some of our RMF reports,” on page 195.

Chapter 3. Implementing the IMS Common Service Layer and the Single Point of Control

The IMS Common Service Layer (CSL) is a collection of IMS manager address spaces that provide the necessary infrastructure for systems management tasks. The IMS CSL reduces the complexity of managing multiple IMS systems by providing you with a single-image perspective in an IMSplex. That is, you can now manage multiple IMS subsystems in an IMSplex as if they were one system.

An IMS single point of control (SPOC) is a program with which you can manage operations of all IMS systems within an IMSplex.

We used the following documentation to help us implement the CSL and SPOC in our production IMSplex:

- *IMS Version 8: Common Service Layer Guide and Reference*, SC27-1293
- *IMS Version 8: Common Queue Server Guide and Reference*, SC27-1292
- *IMS Version 8: Installation Volume 2: System Definition and Tailoring*, GC27-1298

Setting up the Common Service Layer

The CSL address spaces, or CSL managers, include the operations manager (OM), resource manager (RM), and structured call interface (SCI). The CSL managers perform the following functions:

- **Operations manager (OM):** Helps control the operations of all IMS systems in an IMSplex. The OM receives processing control when an OM request (an IMS command, for example) is received by the OM application programming interface (API). All commands and responses to those commands must come through the OM API.
- **Resource manager (RM):** Helps manage resources that are shared by multiple IMS systems in an IMSplex. The RM provides the infrastructure for managing global resource information and coordinating IMSplex-wide processes.
- **Structured call interface (SCI):** Allows IMSplex members to communicate with one another. Communication between IMSplex members can occur within a single z/OS image or among multiple images. The individual IMSplex members do not need to know where the other members reside or what communication interface to use.

See Figure 5 on page 51 for a depiction of these address spaces.

Steps for setting up the CSL

We performed the following steps to set up the CSL on our z/OS production systems:

1. Added PGMNAME(BPEINI00) to the PPT (SCHED00):

```
PPT PGMNAME(BPEINI00)       /* PROGRAM NAME = BPEINI00              */
                 CANCEL     /* PROGRAM CAN BE CANCELED              */
                 KEY(7)     /* PROTECT KEY ASSIGNED IS 7            */
                 NOSWAP     /* PROGRAM IS NON-SWAPPABLE             */
                 NOPRIV     /* PROGRAM IS NOT PRIVILEGED            */
                 DSI        /* REQUIRES DATA SET INTEGRITY          */
                 PASS       /* CANNOT BYPASS PASSWORD PROTECTION    */
                 SYST       /* PROGRAM IS A SYSTEM TASK             */
                 AFF(NONE)  /* NO CPU AFFINITY                      */
                 NOPREF     /* NO PREFERRED STORAGE FRAMES          */
```

Note: BPEINI00 can also be used to start CQS, so the CQSINIT0 entry is no longer needed. Once we migrated all of our production systems to IMS V8.1, we removed the entry for CQSINIT0.

---

2. Added the IMSPLEX parameter to the CQSIP*xxx* member on each system:

```
CQSGROUP=SQGRP,
IMSPLEX(NAME=PROD),
SSN=CQSA,
STRDEFG=ALL,
STRDEFL=PEA
```

---

3. Added the RSRCSTRUCTURE parameter to the CQSSGALL member:

```
STRUCTURE(
 STRNAME=FFMSGQ_STR,
 OVFLWSTR=FFOVFLO_STR,
 STRMIN=0,
 SRDSDSN1=CQS.FF.SRDS1,
 SRDSDSN2=CQS.FF.SRDS2,
 LOGNAME=CQS.FF.LOGSTRM,
 OBJAVGSZ=1024,
 OVFLWMAX=50
 )
STRUCTURE(
 STRNAME=FPMSGQ_STR,
 OVFLWSTR=FPOVFLO_STR,
 STRMIN=0,
 SRDSDSN1=CQS.FP.SRDS1,
 SRDSDSN2=CQS.FP.SRDS2,
 LOGNAME=CQS.FP.LOGSTRM,
 OBJAVGSZ=512,
 OVFLWMAX=50
)
RSRCSTRUCTURE(STRNAME=CSLRMGR_PROD)
```

Note: A resource structure is not needed if only one RM is used in the IMSplex. For availability reasons, we chose to start two RMs and defined a resource structure.

---

4. Created a DFSCG*xxx* member:

```
CMDSEC=N,
IMSPLEX=PROD,
LEOPT=N,
NORSCCC=(),
OLC=LOCAL
```

---

5. Added the CSLG parameter to the DFSPB*xxx* member on each system:

```
DLINM=DLIGRP81,DBRCNM=DBRCGP81,
AUTO=N,
GRSNAME=IMSPETGR,
SUF=1,
CRC=#,LHTS=512,NHTS=512,UHTS=512,
CMDMCS=Y,
CSLG=PET,
APPC=N,AOIS=S,
FIX=HP,VSPEC=81,PRLD=DB,SPM=02,
RSRMBR=,GRNAME=NATIVE2,
...
```

---

6. Created the initialization proclib members for the CSL manager address spaces:

   Example: The following is the SCI initialization member:

```
*-------------------------------------------------------------------*
* Sample SCI Initialization Proclib Member.                         *
*-------------------------------------------------------------------*

ARMRST=N,                   /* ARM should restart OM on failure  */
SCINAME=SCI1,               /* SCI Name (SCIID = SCI1SCI)        */
IMSPLEX(NAME=PROD)          /* IMSplex Name                      */
```

   Example: The following is the OM initialization member:

```
*-------------------------------------------------------------------*
* Sample OM Initialization Proclib Member.                          *
*-------------------------------------------------------------------*

ARMRST=N,                   /* ARM should restart OM on failure  */
CMDLANG=ENU,                /* Use English for Commmand Desc     */
CMDSEC=N,                   /* No Command Security               */
CMDTEXTDSN=IMS810.SDFSDATA, /*                                   */
OMNAME=OM1,                 /* OM Name (OMID = OM1OM)            */
IMSPLEX(NAME=PROD)          /* IMSplex Name (CSLPLEX1)           */
```

   Example: The following is the RM initialization member:

```
*-------------------------------------------------------------------*
* Sample RM Initialization Proclib Member.                          *
*-------------------------------------------------------------------*

ARMRST=N,                   /* ARM should restart RM on failure  */
CQSSSN=CQSC,
IMSPLEX(NAME=PROD,          /* IMSPLEX NAME                      */
RSRCSTRUCTURE(STRNAME=CSLRMGR_PROD)),
RMNAME=RMC                  /* RM Name (RMID = RM1RM)            */
```

---

7. Created the CSL startup procedures:

   Example: The following is our SCI startup procedure, CSLSCI:

```
//CSLSCI   PROC RGN=3000K,SOUT=A,
//             IMSVAR=&IMSVAR,
//             BPECFG=BPECFG00,
//             SCIINIT=000,
//             PARM1=(SCINAME=&SCINAME)
//*
//SCIPROC  EXEC PGM=BPEINI00,REGION=&RGN,
//  PARM='BPECFG=&BPECFG,BPEINIT=CSLSINI0,SCIINIT=&SCIINIT,&PARM1'
//*
//STEPLIB  DD  DISP=SHR,DSN=IMS810.&IMSVAR..SDFSRESL
//         DD  DSN=SYS1.CSSLIB,DISP=SHR
//PROCLIB  DD  DSN=D10.PETDSW4.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
```

   Example: The following is our OM startup procedure, CSLOM:

```
//CSLOM    PROC RGN=3000K,SOUT=A,
//             IMSVAR=&IMSVAR,
//             BPECFG=BPECFG00,
//             OMINIT=000,
//           PARM1=(OMNAME=&OMNAME)
//*
//OMPROC   EXEC PGM=BPEINI00,REGION=&RGN,
//  PARM='BPECFG=&BPECFG,BPEINIT=CSLOINI0,OMINIT=&OMINIT,&PARM1'
```

```
//*
//STEPLIB  DD  DSN=IMS810.&IMSVAR..SDFSRESL,DISP=SHR
//         DD  DSN=SYS1.CSSLIB,DISP=SHR
//PROCLIB  DD  DSN=D10.PETDSW4.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
```

Example: The following is our RM startup procedure, CSLRM:

```
//CSLRM    PROC RGN=0M,SOUT=A,
//             IMSVAR=&IMSVAR,
//             BPECFG=BPECFG00,
//             INIT=CSLRINI0,
//             RMINIT=&RMINIT
//*
//RMPROC   EXEC PGM=BPEINI00,REGION=&RGN,
//   PARM='BPEINIT=&INIT,BPECFG=&BPECFG,RMINIT=&RMINIT'
//STEPLIB  DD  DSN=IMS810.&IMSVAR..SDFSRESL,DISP=SHR
//         DD  DSN=SYS1.CSSLIB,DISP=SHR
//PROCLIB  DD  DSN=D10.PETDSW4.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
```

---

8. Updated the SCI registration exit routine (DSPSCIX0) and placed it into an authorized library:

```
PTBLEYEC DS    0H                    TABLE EYECATCHER
         DC    C'PLEXTABL'
PLEXTABL DS    0H
         DC    CL(DSNL)'RECON1.PROD'             RECON NAME
         DC    CL(PNL)'PROD '                    IMSplex name
         DC    XL(RCL)'00000000'           RC00 = use the IMSplex name
         DC    CL(DSNL)'RECON2.PROD'             RECON NAME
         DC    CL(PNL)'PROD '                    IMSplex name
         DC    XL(RCL)'00000000'           RC00 = use the IMSplex name
         DC    CL(DSNL)'RECON3.PROD'             RECON NAME
         DC    CL(PNL)'PROD '                    IMSplex name
         DC    XL(RCL)'00000000'           RC00 = use the IMSplex name
```

---

9. Defined the resource manager coupling facility structure in the CFRM policy:

```
STRUCTURE NAME(CSLRMGR_PROD)
          SIZE(32000)
          INITSIZE(20000)
          ALLOWAUTOALT(YES)
          FULLTHRESHOLD(60)
          DUPLEX(ALLOWED)
          PREFLIST(CF2,CF1,CF3)
```

Our CSL and SPOC configuration

Figure 5 on page 51 illustrates our the CSL and SPOC configuration in our IMSplex:

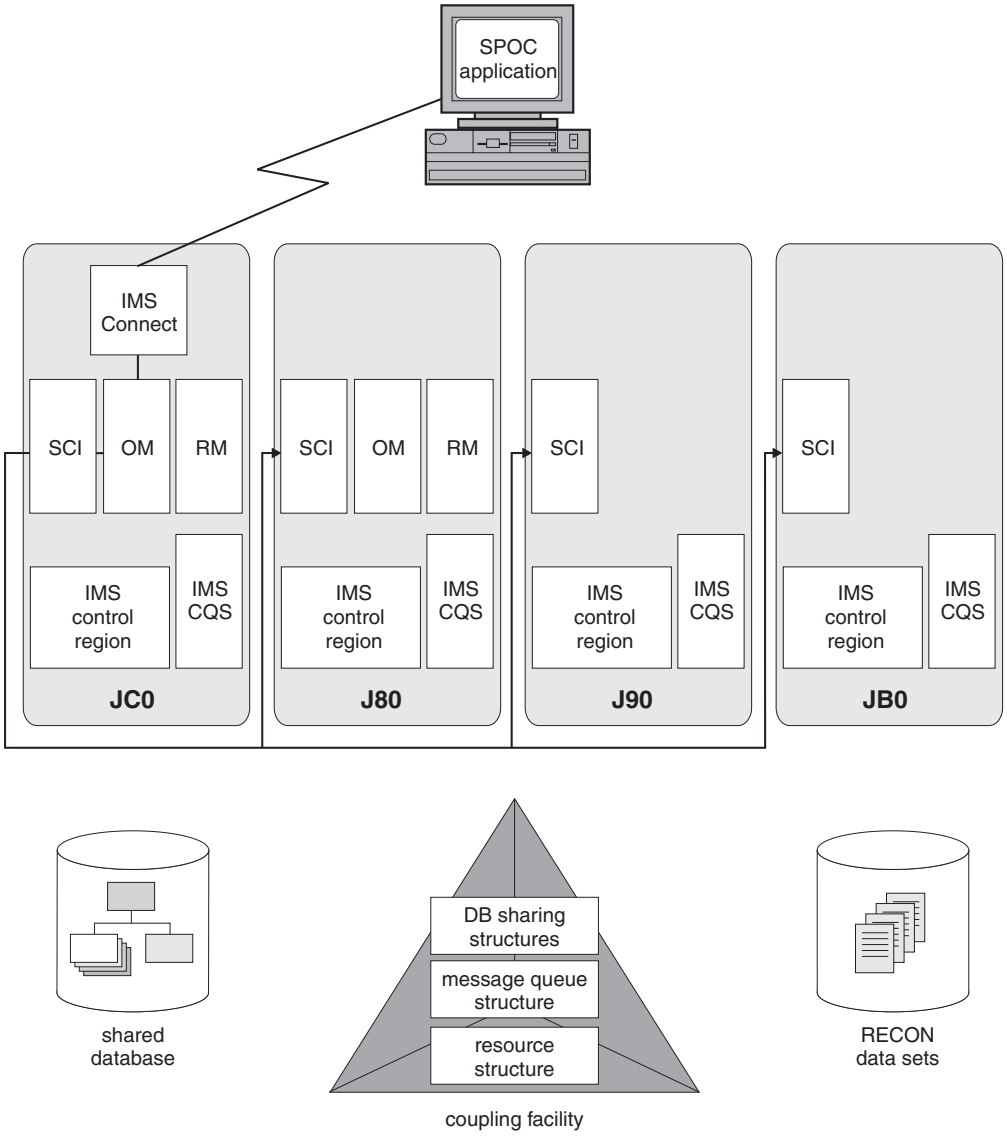


Figure 5. Our IMS CSL and SPOC configuration

Note the following about our configuration:

- We run one SCI address space on each z/OS system where IMS subsystems run.
- We only run OM and RM address spaces on systems J80 and JC0.
- We use automations to start the CSL address spaces following system IPLs. However, we found that if the RM doesn't detect the required CQS address space within 10 minutes, it terminates with a U0010-00000508 user abend. To avoid this, we added the CQS startup to automations, instead of allowing IMS to automatically start the CQS address space. This ensures that the CQS address space is available when the RM expects it.

IMS performance considerations for CSL

For the CLS address spaces, IBM recommends using the SYSSTC service class or a service class with higher importance (that is, a lower-numbered value) than the CNTL and CQS address spaces and all dependent regions. Since WLM provides five levels of importance, a general guideline would be to group resources into

service classes with the following relative importance:

IMPORTANCE Level	Value	Address spaces
higher	N	IRLM
	N+1	VTAM, APPC, DBRC, SCI, OM, RM
	N+2	CNTL, CQS
	N+3	DLIS
lower	N+4	dependent regions

Thus, when CPU resources are constrained, the following rules would apply:

- All dependent regions should have the lowest dispatching priority among the other IMS address spaces.
- CNTL and CQS, the address spaces with the next largest CPU consumption, should have a lower dispatching priority than the CSL address spaces.

Setting up the single point of control

A SPOC communicates with one OM address space; the OM then communicates with all of the other IMS address spaces in the IMSplex, through the SCI, as required for operations.

Steps for setting up the single point of control

We performed the following steps to set up the single point of control:

1. Verified that IMS service PQ69527 (PTF UQ73719) is installed.

2. Verified that IMS Connect is installed.

   We are currently running IMS Connect V2.1. However, note that if you are running IMS Connect V1.2, you must install PQ62379 (PTF UQ69902) and PQ70216 (PTF UQ74285).

3. Updated the HWS configuration member to add the EXIT and IMSPLEX parameters:

```
HWS
 (ID=HWSC,RACF=N)
TCPIP
 (HOSTNAME=TCPIP,RACFID=RACFID,PORTID=(9999,9998,9997,9996,9995,9994,
 9993,9992,9991,9990),
 EXIT=(HWSCSLO0,HWSCSLO1),
 MAXSOC=51,TIMEOUT=9999)
DATASTORE
 (ID=IMSC,GROUP=NATIVE2,MEMBER=HWSC,TMEMBER=IMSPETJC)
IMSPLEX
 (MEMBER=IMSPLEXC,TMEMBER=PROD)
```

4. Installed DB2 UDB V8.1 (DB2 Control Center) on the workstation.

   We are currently running at the FixPak 4 service level. You can get the latest FixPaks from the DB2 Technical Support Web site at www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/index.d2w/report.

Steps for setting up DB2 Control Center for the IMS SPOC

We performed the following steps to set up DB2 Control Center for the IMS SPOC:

1. Started the Control Center, then clicked **Selected → Add** from the menu bar.

2. In the **Add System** dialog box:
   a. Selected the **IMS** button
   b. In the **System name** box, typed the name of our IMSplex: PROD
   c. In the **Host name** box, typed the IP address of the system where the IMS Connect subsystem is located
   d. In the **Port number** box, typed the IMS Connect port number we wanted to use: 9995
   e. Clicked **OK**

   Example: Figure 6 is an example of the **Add System** dialog box:

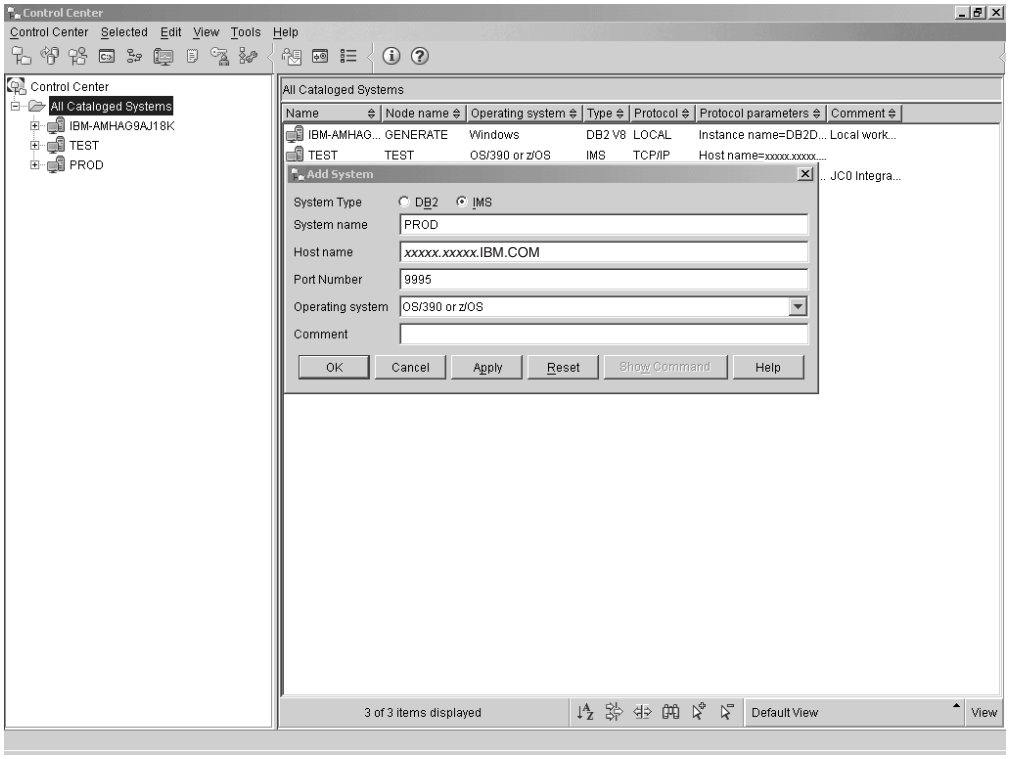


Figure 6. Example of the Control Center Add System dialog

3. Started the Command Center by clicking **Tools → Command Center** from the menu bar.

4. In the Command Center:
   a. In the **Command type** field, selected **IMS commands** from the pull-down menu
   b. In the **IMS sysplex** field, selected PROD (the name of our IMSplex) from the **Select Connection** dialog
   c. When prompted, entered the user ID and password that we had set during the installation of DB2 Control Center.

Example: Figure 7 is an example of the initial setup of the Command Center: **Add System** dialog box:

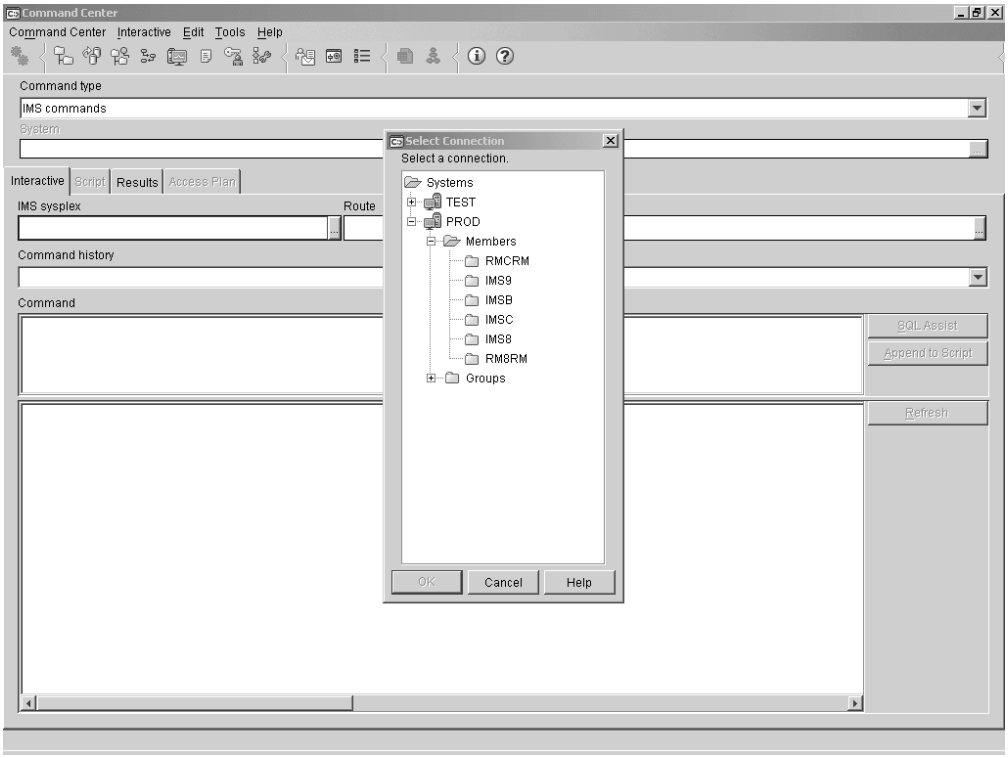


Figure 7. Example of the Command Center initial setup

---

5. To issue an IMS command:
   a. In the **Route** field, selected the IMSplex member to which the command is to be issued
   b. In the **Command** field, entered the IMS command to be issued
   c. Clicked the **Execute** icon at the far left side of the icon bar, in the upper left corner

   Example: Figure 8 on page 55 is an example of using the Command Center to issue the DIS QCNT LTERM MSGAGE 0 command to member IMSC in our PROD IMSplex:

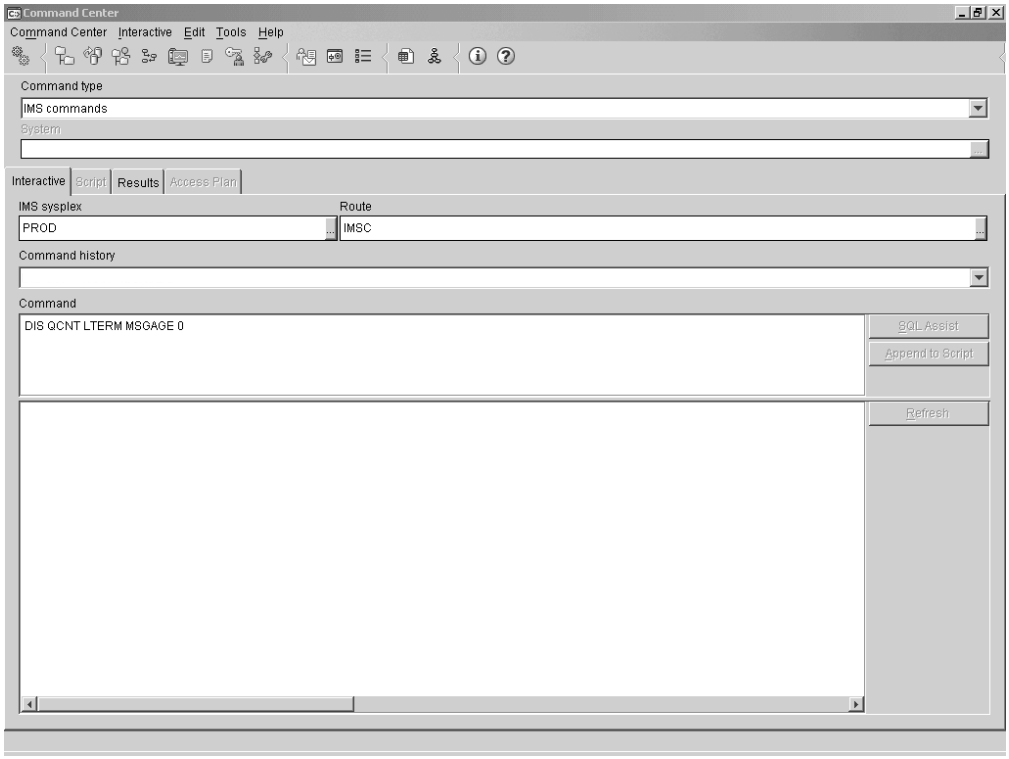


Figure 8. Example of issuing an IMS command to IMSplex member IMSC

Result: Figure 9 on page 56 is an example of the response to the DIS QCNT LTERM MSGAGE 0 command that was issued to member IMSC. Note that the response appears on a separate tab, **Results**, in the Control Center display.

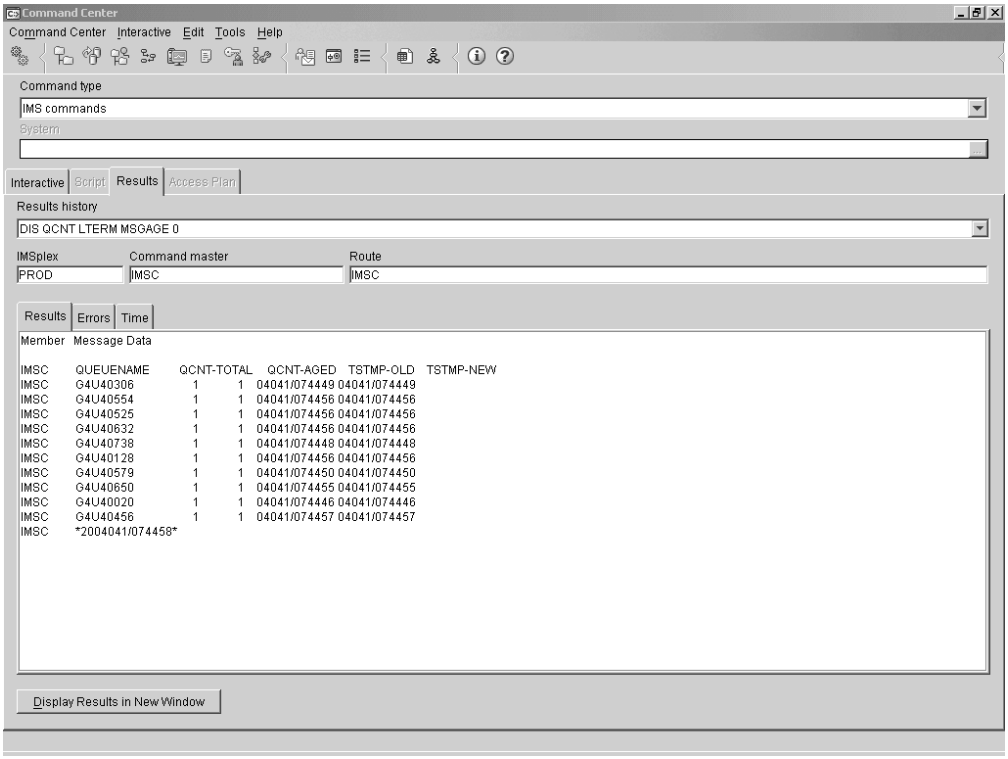


Figure 9. Example of the response to an IMS command that was issued to IMSplex member IMSC

Example: Figure 10 on page 57 is an example of issuing the DIS LINE 1 command to all members of the IMSplex by selecting `All_Members` in the **Route** field:

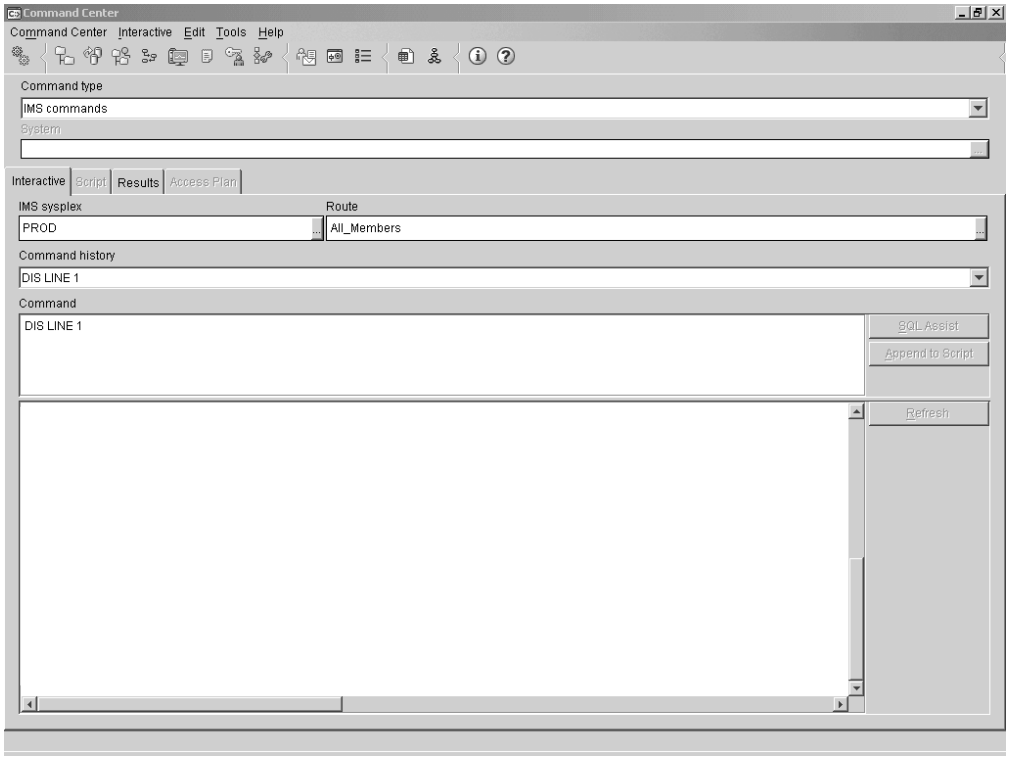


Figure 10. Example of issuing an IMS command to all members of the IMSplex

Result: Figure 11 on page 58 is an example of the response to a command that was issued to all members of the IMSplex:

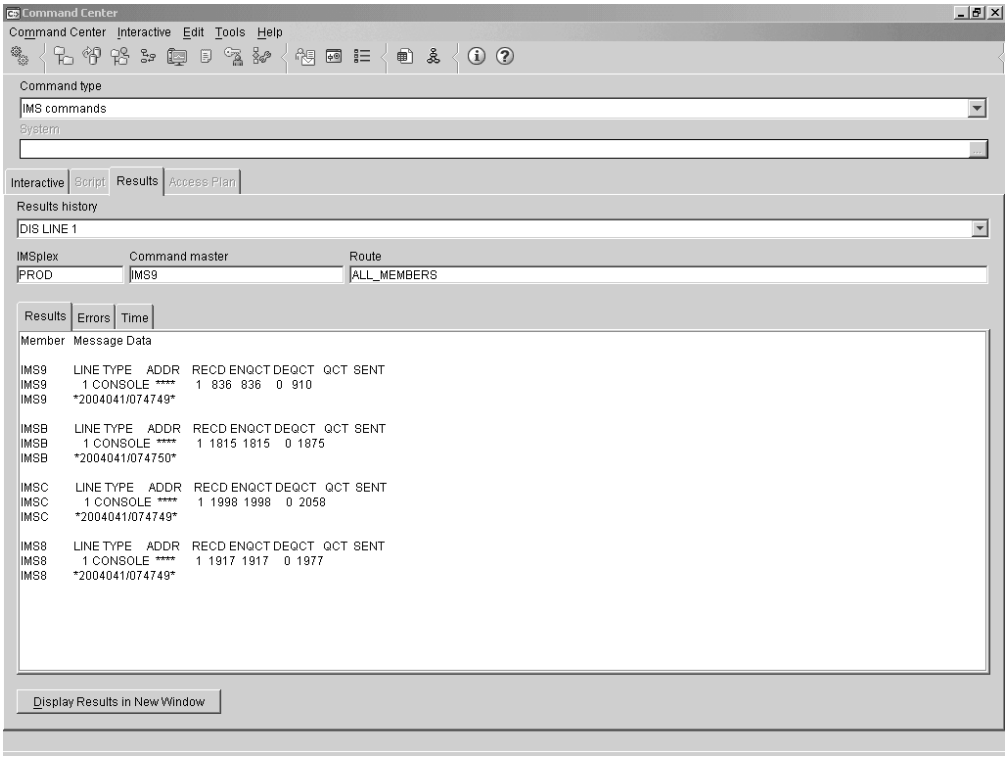


Figure 11. Example of the response to an IMS command that was issued to all members of the IMSplex

Chapter 4. Using zSeries Application Assist Processors (zAAPs)

As promised, we've updated our most recent testing of zAAP since the June 2004 report.

IBM @server zSeries 890 (z890) and zSeries 990 (z990) servers make available a new, optional feature—the zSeries Application Assist Processor (zAAP)—which provides a strategic z/OS Java execution environment for customers who desire the powerful integration advantages and traditional qualities of service of the zSeries platform.

A zAAP is similar in concept to a System Assist Processor (SAP). Unlike CPs, ICFs, and IFLs, zAAPs can do nothing on their own; they cannot perform an IPL and cannot run an operating system. zAAPs must operate along with general purpose CPs within logical partitions running z/OS; however, they are designed to operate asynchronously with the general purpose CPs to execute Java programming under control of the IBM Java Virtual Machine (JVM).

This chapter describes what we did to configure and to prepare to exercise the zAAP feature on our z990 server.

Prerequisites for zAPP

The following are prerequisites for zAAP and execution of the JVM processing cycles:

- the IBM Software Developer's Kit (SDK) for z/OS
- z/OS 1.6 (or z/OS.e 1.6)
- Java 2 Technology Edition V1.4.1 with PTF for APAR PQ86689
- and the Processor Resource/Systems Manager ((PR/SM) must be enabled).

Subsystems and applications using SDK 1.4 that exploit zAAPs

The following subsystems and applications using SDK 1.4 exploit zAAPs:

- WebSphere Application Server 5.1
- CICS/TS 2.3
- DB2 V7, DB2 V8 (we tested with both)
- IMS V7, IMS V8, and IMS V9 (we tested with IMS V8)
- Websphere MQ 5.3.1
- Websphere Business Integration Message Broker 5.0

Setting up zAAP

First we executed the recommended zAAP Projection Tool for Java 2 Technology Edition SDK 1.3.1 against the available Java workload we had at the early stage of our testing. This provided us a baseline or a starting point for defining the minimum number of zAAP processors we would require on our z990 and z890 processors. Please reference the following for more details pertaining to the Projection Tool:

- "Installation of the zAAP Projection Tool Instrumented SDK in WebSphere for z/OS Version 5" available at:

  `http://www-1.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100431`

- z/OS Performance: Capacity Planning Considerations for zAAP Processors available at:

  `http://www-1.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100417`

As mentioned in the intro we recommend that you contact your hardware support for the latest hardware and software requirements. We licensed 1 CP on our z890 and 2 CPs on our z990 as zAAPs.

Configuring zAAPs

We configured two zAAPs on all our z/OS images on our z990 server and we configured one zAAP on our z890 server. When you configure the z/OS logical partitions, you simply specify how many logical zAAPs you want to configure for each partition, just as you do the number of standard CPs. When you IPL the system, z/OS determines how many zAAPs are configured and manages an additional dispatcher queue for zAAP-eligible work.

We did the following to configure the zAAPs:

1. Updated the image profiles for the Z2 and Z3 partitions to define two zAAPs to each partition.

   Example: Figure 12 shows an example of the image profile for our Z2 image with two zAAPs defined.

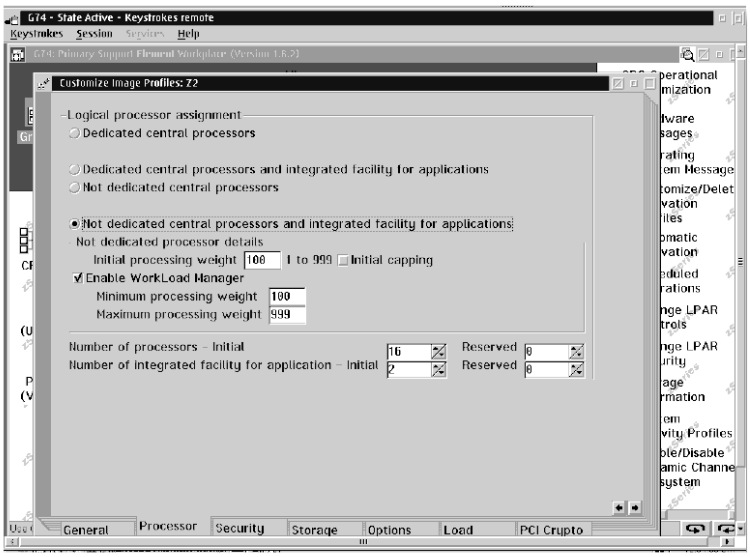


Figure 12. Example of the image profile for our Z2 image with two zAAPs defined.

2. Updated parmlib member IEAOPT*xx* for the z/OS partitions to specify the following options:

IFACROSSOVER=YES
zAAP-eligible work may execute on zAAPs or it can “cross over” and execute on standard processors.

IFAHONORPRIORITY=YES
Standard processors execute both Java and non-Java work in order of dispatching priority.

---

3. Deactivated and reactivated the z/OS partitions to bring the zAAPs online.

   You can use the D M=CPU command to display the status of the zAAPs. The zAAPs appear as assist processors in the response to the D M=CPU command.

   Example: The following is an example of the response to the D M=CPU command on system Z2:

```
IEE174I 15.34.28 DISPLAY M
PROCESSOR STATUS
ID  CPU                 SERIAL
00  +                   02B52A2084
01  +                   02B52A2084
02  +                   02B52A2084
03  +                   02B52A2084
04  +A                  02B52A2084
05  +A                  02B52A2084

CPC ND = 002084.D32.IBM.00.00000001B52A
CPC SI = 2084.325.IBM.00.000000000001B52A
CPC ID = 00
CPC NAME = G74
LP NAME = Z2         LP ID =  2
CSS ID  = 0
MIF ID  = 2
```

   Example: The following is an example of the response to the D M=CPU command on system Z3:

```
IEE174I 15.38.30 DISPLAY M
PROCESSOR STATUS
ID  CPU                 SERIAL
00  +                   24B52A2084
01  +                   24B52A2084
02  +                   24B52A2084
03  +                   24B52A2084
04  +A                  24B52A2084
05  +A                  24B52A2084

CPC ND = 002084.D32.IBM.00.00000001B52A
CPC SI = 2084.325.IBM.00.000000000001B52A
CPC ID = 00
CPC NAME = G74
LP NAME = Z3         LP ID = 24
CSS ID  = 2
MIF ID  = 4
```

Monitoring zAAP utilization

There is support in RMF (supplied by APAR OA05731) to provide information about zAAP utilization. This information is useful to determine if and when you need to add additional zAAP capacity.

SMF is another source of information. SMF type 72 records contain information about zAAP utilization. There are also new fields in SMF type 30 records to indicate

the amount of time spent on zAAP work as well as the amount of time spent executing zAAP-eligible work on standard processors for both crossover and honor priority execution modes.

Here is an example of our RMF Monitor III displaying the use of the zAAPs on our z990 processor highlighted in **bold**:

```
                        RMF V1R5   CPC Capacity                  Line 1 o
Command ===>                                                  Scroll ===>
Switched to option set WLMPOL01 on JA0.
Samples: 120     System: JA0   Date: 09/14/04  Time: 11.09.00  Range: 120

Partition:   JA0        2084 Model 325
CPC Capacity:    1114   Weight % of Max: 10.0       4h MSU Average:   175
Image Capacity:  1069   WLM Capping %:   ****       4h MSU Maximum:   205

Partition  --- MSU ---  Cap  Proc    Logical Util %   - Physical Util % -
             Def   Act  Def   Num    Effect   Total   LPAR  Effect  Total

*CP                                                    2.1    46.9   49.0
EBTELNX        0     1  NO    2.0       1.3     1.3   0.0     0.1    0.1
JA0            0   191  NO    7.0      60.3    61.2   0.3    16.9   17.1
JC0            0   158  NO    7.0      49.8    50.6   0.2    13.9   14.2
JE0            0   137  NO    8.0      37.9    38.6   0.2    12.1   12.3
Z2             0    27  NO    7.0       8.5     8.7   0.1     2.4    2.4
Z3             0    17  NO    4.0       9.3     9.5   0.0     1.5    1.5
PHYSICAL                                               1.3            1.3

*ICF                                                   1.2    60.6   61.8
CF2                         3.0      98.9    98.9    0.0    42.4   42.4
JA0                    NO   2.0      46.7    46.9    0.1    13.3   13.4
JC0                    NO   2.0       0.0     0.0    0.0     0.0    0.0
JE0                    NO   2.0       0.0     0.0    0.0     0.0    0.0
PETVM                  NO   2.0      15.0    15.6    0.2     4.3    4.4
Z2                     NO   2.0       2.0     2.1    0.0     0.6    0.6
Z3                     NO   2.0       0.1     0.1    0.0     0.0    0.0
PHYSICAL                                               1.0            1.0
```

Preparing our workloads to exercise the zAAP feature

Initially, we selected several of our current MQ Web workloads and rewrote them as base Java applications (non-Web) to exercise the zAAP feature. We also installed WebSphere Business Integration Message Broker 5.0 for zAAP testing. This product itself uses Java and will increase the utilization of the zAAP feature in addition to the workloads. .

In the near future, we plan to use the following MQ-based workloads (which run via TPNS scripts and TSO users) to test the zAAP feature:

- MQLARGE — This workload currently runs primarily on system JG0. Its purpose is to create temporary MQ queues and put large messages on them. We added a compute module to increase the application's CPU usage, as MQ itself does not consume much CPU resource.
- MQCICS — This workload runs on system JB0 and puts a request message on a CICS bridge queue to run a DB2 transaction.
- MQDQM — This is a communications test workload that puts and gets messages between a local MQ queue manager and a remote system's queue manager.
- MQDQLSSL — This workload is similar to the MQDQM workload but uses SSL channels to test security.

- RetailTPNS — This workload simulates a retail type of application and uses WebSphere MQ Integrator to put messages on a queue for the broker to process. We run a TPNS version in Java that will use the zAAP feature.

Other workloads we support are:

- DB2— In the past, Integration Test had implemented the NST (Native Stress Test) Version 6 workload for DB2, which is a TPNS driven, CICS based workload comprised of COBOL programs, stored procedures and user defined functions (UDFs). For z/OS 1.6 with the introduction of eServer zSeries Application Assist Processor, several of the stored procedures originally written in COBOL were converted to Java, and a new Workload Manager (WLM) address space to run the Java stored procedures was defined.
- IMS— The IMS Java workload is based on the IMS Java Dealership IVP application, which is documented in the ″IMS Java User's Guide″ (SC27-1296-00). The environment is IMS V8 running SDK 1.4.2. This application consists of one database and one transaction (with multiple methods). We modified the IMS V8 Dealership IVP application by:
  - Creating MFS screen formats so the transactions can be set up as an OLTP workload
  - Coded TPNS scripts to drive different dealership application methods.

We are currently running the following methods: FindACar, ListModels and ShowModelDetails.

We executed our zAAP testing with different configurations.The majority of our testing was completed with our zAAPs configured online. We also performed extensive testing with the zAAP processors configuring them online and offline with Java workload running.

Chapter 5. Parallel Sysplex automation

In this chapter, we typically describe how we use automation to more efficiently operate our sysplex from a single point of control, and to automate the startup, shutdown, and restart of many of our major subsystems and applications.

Our early experiences with automation

We began writing about Parallel Sysplex automation in our 1997 test reports. At that time, we were just beginning to use NetView and System Automation for OS/390 (then called SA/MVS) to more efficiently operate our sysplex. We were running NetView V3R1 and SA/MVS V1R2.

We eventually migrated to Tivoli NetView for OS/390 V1R4 and System Automation for OS/390 (SA OS/390) V1R3, including the sysplex automation enhancements that IBM delivered in October, 1999, as an SPE in APAR OW39485. For information about our use of those products, see our December 2001 edition.

Automation with msys for Operations

Managed System Infrastructure for Operations (msys for Operations), which was introduced as a new base element in z/OS V1R2, simplifies the day-to-day operation of z/OS and z/OS.e Parallel Sysplex configurations by automating typical operator tasks and events. msys for Operations actually includes parts of two licensed standalone products: Tivoli NetView for OS/390 and System Automation for OS/390. We tested and ran msys for Operations along with Tivoli Netview for OS/390 V5R1 and SA OS/390 V1R3, prior to migrating to SA OS/390 V2R2.

Migrating to System Automation for OS/390 Version 2 Release 2

We have now completed our migration from msys for Operations to SA OS/390 V2R2. We followed the instructions in *System Automation for z/OS Planning and Installation*, SC33-7038, in the appendix about migrating to SA OS/390 from msys for Operations. Also, for basic information, we found *System Automation for z/OS User's Guide*, SC33-7040, to be very useful. You can find the SA OS/390 publications on the z/OS Internet Library at www.ibm.com/servers/eserver/zseries/zos/bkserv/ or under the **Library** link on the System Automation Web site at www.ibm.com/servers/eserver/zseries/software/sa/.

We continue to run Tivoli Netview for OS/390 V5R1. We also continue to run the NetView focal points, as we have discussed in previous years' test reports, on our current release of NetView. (See our December 2001 edition for more information about our NetView focal points.)

Applying service for SA OS/390: We needed to apply the fixes for the following SA OS/390 APARs: OA04046, OA04152, and OA05945.

Migrating the contents of our AOFCUST member: Our migration included using the INGCUST dialog to migrate the contents of our AOFCUST member to automation control file (ACF) fragments. As we reported on previously, we had customized our AOFCUST member to define a set of spare, local page data sets that can automatically be added when we experience an auxiliary storage shortage (see our December 2003 edition).

Starting the automation manager: In our sysplex, we use a startup procedure called HSAMPROC to start the automation manager. In our environment, we always start automations using a cold start. We generally do not use automations to control DB2, CICS, and IMS because the nature of our testing often requires that we vary the start and stop times for those subsystems, rather than leave them on a regular schedule.

Using SA OS/390

Our operations staff invokes the INGPLEX command in full mode to access the main menu of sysplex-related functions in SA OS/390.

Using the DRAIN and ENABLE subcommands

From the INGPLEX main menu, we make frequent use of the DRAIN subcommand for clearing off our coupling facilities for maintenance. We find that this function saves us considerable time in our coupling facility maintenance activities.

Currently, when the DRAIN function completes, the coupling facility status changes to DRAINED NOHWACC because we do not have the BCP internal interface working. We then use the HMC to manually deactivate and re-activate the drained coupling facility.

After re-activating the coupling facility, we use the ENABLE subcommand to repopulate it with structures.

Refreshing the automation manager

Each time you make changes to the ACF, you must refresh the automation manager. Failure to refresh the automation manager following an ACF build results in the following message:

```
AOF618I NO VALID ACF FOUND FOR sysname - ACF TOKEN MISMATCH
```

This means that the ACF does not have the same token as the automation manager's configuration file.

We perform the following steps to refresh the automation manager after making changes to the current ACF:

1. From a NetView agent session, issue the following command:

   ```
   INGAMS
   ```

   Result: The following is an example of the dialog panel that appears:

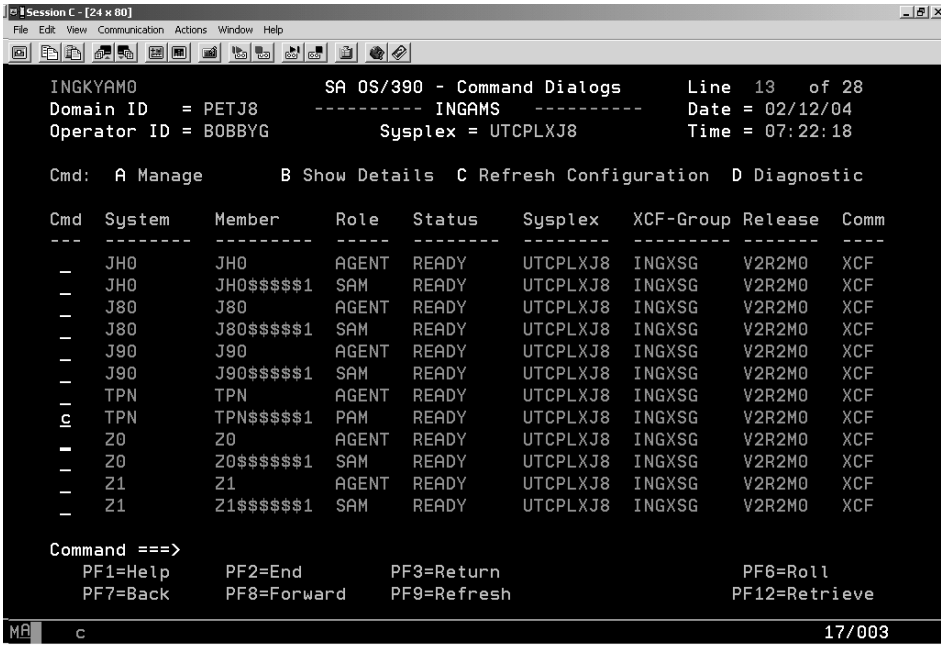


Figure 13. Example of the INGAMS dialog

2. Locate the entry for the system that is acting as the primary automation manager (PAM). The entry will indicate PAM in the Role column.
   In the above example dialog, system TPN is acting as the PAM.

3. Enter the `C` (refresh configuration) line command next to the PAM system and press Enter.
   Result: The following is an example of the dialog panel that appears:

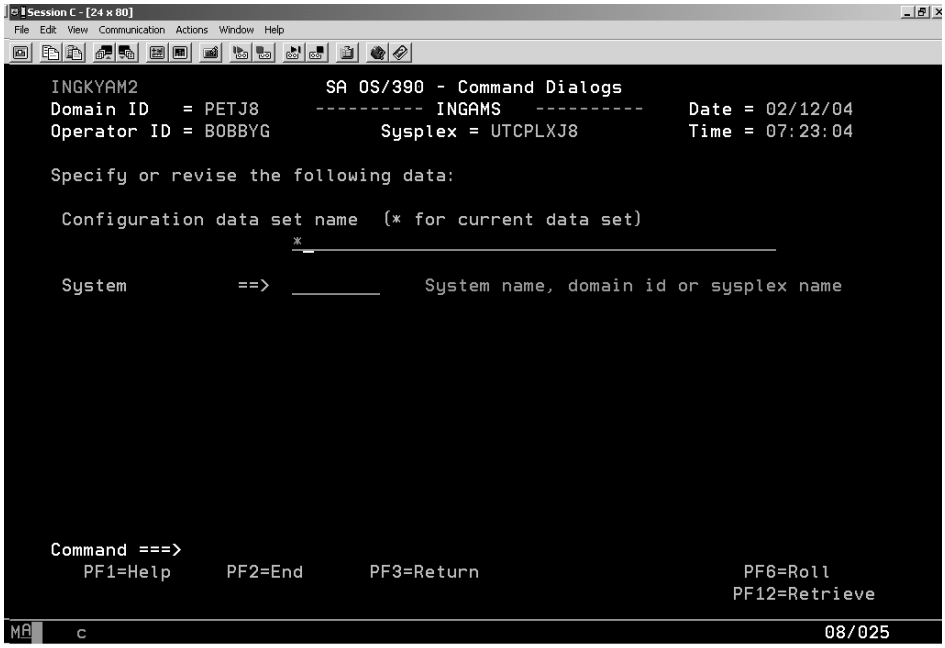


Figure 14. Example of the Refresh Configuration dialog

4. Enter an asterisk (*) for the configuration data set name and press Enter.

This refreshes the automation manager's configuration from the updated ACF.

Turning off the automation flag for a resource

During our testing, we often need to turn off automations for various resources so that we can manually control them. This turns the automation flag off in the automation manager, not the automation agent. The automation manager will remember the flag's setting unless all automation managers are brought down and restarted COLD.

We perform the following steps to turn off the automation flag for a resource:

1. From the NetView console, enter the following command:

   DS *subsystem*

   Example: To display the LDAP servers, we would issue the following command:

   DS LDAP*

   Result: The following is an example of the dialog panel that appears:

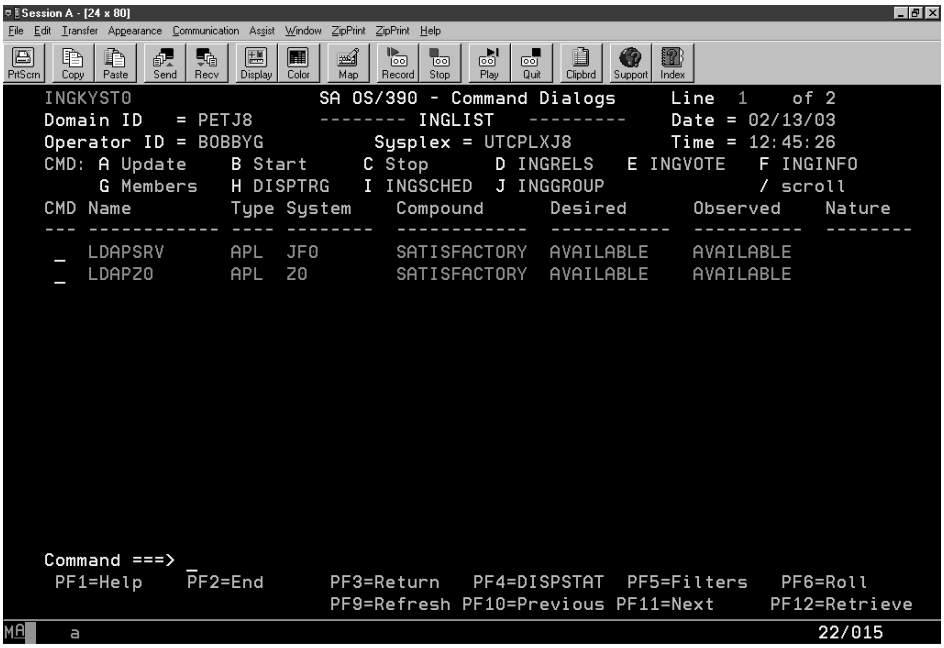


Figure 15. Example display from the DS LDAP command*

---

2. Enter the A (update) line command next to the desired resource and press Enter.

   Example: To update the automation settings for the LDAPSRV server, enter an A next to that resource.

   Result: The following is an example of the dialog panel that appears:

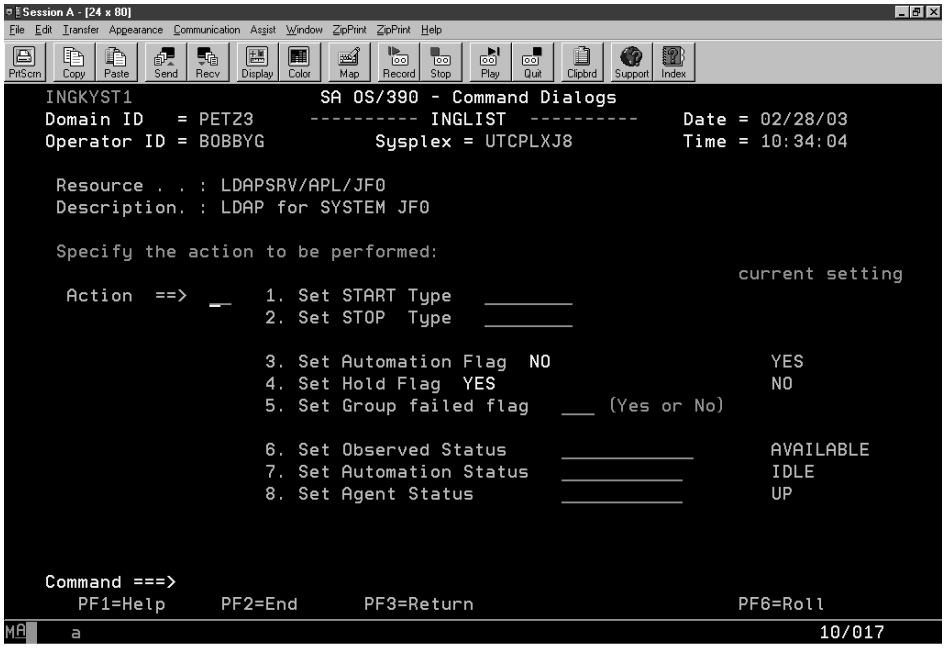


Figure 16. Example of the automation settings dialog for the LDAPSRV server (automation flag is on)

Note that the value under the **current setting** column for the automation flag is YES. This means that automation is turned on for this resource.

---

3. To turn off the automation flag, enter 3 (Set Automation Flag NO) on the **Action** line and press Enter.

   Result: The following is an example of the dialog panel that appears:

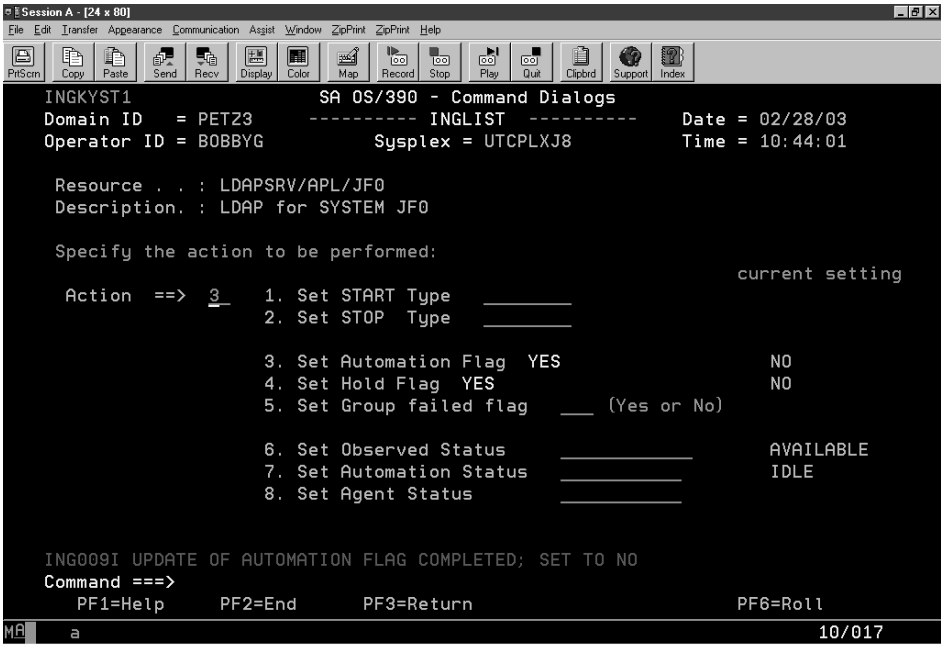


Figure 17. Example of the automation settings dialog for the LDAPSRV server (automation flag is off)

Note that the value under the **current setting** column for the automation flag is now NO. This means that automation is turned off for this resource.

The resource can now be manually started and stopped.

Part 2. Networking and application enablement

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

Chapter 6. About our networking and application enablement environment

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

Our networking and application enablement configuration

Figure 18 illustrates at a high level our networking and application enablement configuration. In the figure, the broad arrows indicate general network connectivity of a given type, rather than specific, point-to-point, physical connections.

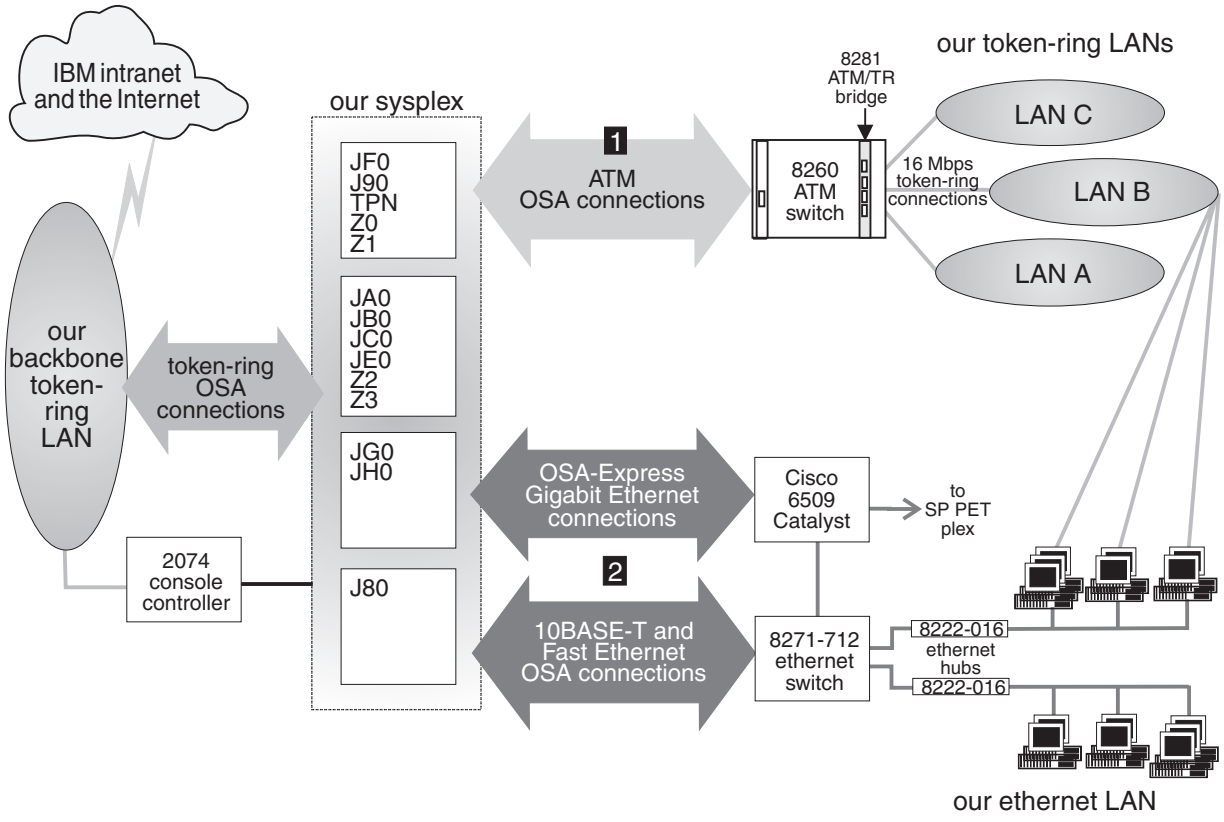


Figure 18. Our networking and application enablement configuration

Note the following about Figure 18:

- We use the following OSA features to connect our systems to our LANs:
  - OSA-2
    ENTR (Ethernet/Token Ring)
  - OSA-Express
    ATM (Asynchronous Transfer Mode)
    FENET (Fast Ethernet)
    Gigabit Ethernet (GbE)

- All ATM connections (1) use ATM LAN emulation; we have no native ATM connections. Although not shown, some of our CPCs that do not have an ATM connection instead use OSA-2 ENTR to directly connect to each of our token-ring LANs.
- Host system Ethernet connections (2) use either OSA-2 ENTR 10BASE-T, OSA-Express FENET, or OSA-Express Gigabit Ethernet features depending on the CPC model and the type of adapter it supports.
- Although not shown, all RS/6000s on LAN A also have a direct connection to the backbone token ring.
- We recently replaced our Gigabit Ethernet switch with a Cisco 6509 Catalyst switch. This new switch handles all of our Gigabit Ethernet and some of our Ethernet connections. Eventually, all of our Ethernet traffic will flow through the Cisco 6509 and we'll remove the 8271-712 Ethernet switch. (For technical details about the Cisco 6500 Catalyst family, go to http://www.cisco.com.)
- We also added Gigabit Ethernet connectivity between our sysplex and a remote AIX cluster owned by the SP PET team. We use this connectivity for the AIX portion of our bookstore application.

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

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

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

Our Ethernet LAN configuration

Our network configuration includes an Ethernet LAN. We primarily use it for FTP testing from Windows 95 and Windows NT clients and for VIPA testing. (For more information about VIPA, see our December '99 edition.) Many of our Ethernet client workstations also contain a token-ring adapter that connects the workstations to our token-ring LAN B as well. We use an OS/2 LAN Server on LAN B to drive the FTP testing on the Ethernet clients. You can read more about this setup in "What's happening in LAN B?" on page 83

Our systems' Ethernet connectivity includes a combination of 10BASE-T, Fast Ethernet, and Gigabit Ethernet connections using OSA-2 ENTR, OSA-Express FENET, and OSA-Express Gigabit Ethernet features, respectively.

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

The OSA-Express FENET feature operates at either 10 or 100 Mbps in half- or full-duplex mode and supports auto-negotiation with its attached Ethernet hub, router, or switch. We used the latest edition of *zSeries OSA-Express Customer's Guide and Reference* and the OSA/SF GUI for Windows to install and configure the OSA-Express FENET feature. (See our December 1999 edition for our experiences installing the OSA/SF GUI for Windows.) We also recommend that you check with your IBM support representative to ensure that you have the latest microcode level for this feature.

Our ATM configuration

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

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

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

Our ipV6 Environment Configuration

With z/OS V1R6, we now have an ipV6 environment equivalent to our ipV4 environment. V1R6 now supports OSPF V3 for ipV6 and ipV6 support for DVIPA and Sysplex Distributor.

We used the following manuals as guides in setting up ipV6.

- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS Communications Server: IP Configuration Reference*, SC31-8776
- *z/OS Communications Server: IPv6 Network and Application Design Guide*, SC31-8885

To configure a z/OS image for ipV6 the following changes have to be made:

Note: This is not meant to be an all inclusive guide for ipV6 setup.

1. Add new NETWORK, AF_INET6 to BPXPRMxx statement.
2. Add New INTERFACE to TCPIP profile for ipV6 'device'.
3. Add support for DYNAMIC XCF
4. Create DVIPA for ipV6
5. Add INTERFACE to OMPROUTE profile.
6. Make appropriate additions to Nameserver.

z/OS UNIX System Services changes and additions

The following are the changes and additions we made to z/OS UNIX System Services:

1. Changing BPXPRMxx to add ipV6 support

We made the following changes to BPXPRMxx to add ipV6 support:

```
NETWORK DOMAINNAME(AF_INET6)
DOMAINNUMBER(19)
MAXSOCKETS(60000)
TYPE(INET)
```

Note: INADDRANYPORT and INADDRANYCOUNT values are used for both ipV4 and ipV6 when the BPXPRMxx is configured for ipV4 and ipV6 support. If AF_INET is specified, it is ignored and the values from the NETWORK statement for AF_INET are used if provided. Otherwise, the default values are used.

2. Adding NETWORK statements to have a stack that supports ipV4 and ipV6

   We added the following two NETWORK statements to have a stack that supports ipV4 and ipV6:

```
FILESYSTYPE TYPE(CINET) ENTRYPOINT(BPXTCINT)
NETWORK DOMAINNAME(AF_INET)
DOMAINNUMBER(2)
MAXSOCKETS(2000)
TYPE(CINET)
INADDRANYPORT(20000)
INADDRANYCOUNT(100)
NETWORK DOMAINNAME(AF_INET6)
DOMAINNUMBER(19)
MAXSOCKETS(3000)
TYPE(CINET)
SUBFILESYSTYPE NAME(TCPCS) TYPE(CINET) ENTRYPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPCS2) TYPE(CINET) ENTRYPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPCS3) TYPE(CINET) ENTRYPOINT(EZBPFINI)
```

TCPIP Profile changes

We made the following additions to our IPv6 INTERFACE statements:

```
INTERFACE OSA9E0V6
DEFINE IPAQENET6
PORTNAME GBPRT9E0
IPADDR FEC0:0:0:1:x:xx:xx:xxx  ;(Site-Local Address)
3FFE:0302:0011:2:x:xx:xx:xxx ; (Global Address)
```

Note: In order to configure a single physical device for both IPv4 and IPv6 traffic, you must use DEVICE/LINK/HOME for the IPv4 definition and INTERFACE for the IPv6 definition, so that the PORTNAME value on the INTERFACE statement matches the device_name on the DEVICE statement.

Dynamic XCF addition

We made the following addition for our Dynamic XCF:

```
IPCONFIG6 DYNAMICXCF FEC0:0:0:1:0:168:49:44
```

Dynamic VIPA additions

The following statement was added to our VIAPDYNAMIC section:

Note: V6Z2FTP is the INTERFACE name for this VIPA.

```
VIPADEFINE V6Z2FTP 2003:0DB3:1::2
VIPADISTRIBUTE SYSPLEXPORTS V6Z2FTP PORT 20 21
DESTIP FEC0:0:0:1:0:168:49:37
```

OMPROUTE addition

Setting up OMPROUTE only requires adding the INTERFACE name to the OMPROUTE profile for the basic setup that we used.

```
IPV6_OSPF_INTERFACE
Name = OSA9E0V6;
```

Note: During testing we encountered the following message:

```
EZZ7954I  IPv6 OSPF adjacency failure, neighbor 192.168.25.33, old state 128,
 new state 4, event 10
```

The neighbor id in the message is the ROUTERID from the OMPROUTE profile. It will not show an ipV6 address.

NAMESERVER changes

We created seperate ipV6 names for each LPAR. To keep things simple for the system name, we used the existing LPAR name with IP6 as the suffix. For the ipV6 ip addresses, we used a common prefix and used the ipV4 address as the suffix. This made it easier to identify for diagnosing problems.

Forward file changes

The following change was made to our forward file:

```
J80IP6                IN AAAA 3FFE:302:11:2:9:12:20:150
```

Reverse file entry addition

We added the following for the reverse file entry:

```
$TTL 86400
$ORIGIN 2.0.0.0.1.1.0.0.2.0.3.0.E.F.F.3.IP6.ARPA.
@       IN SOA  Z0EIP.PDL.POK.IBM.COM.  ALEXSA@PK705VMA (
                    012204  ;DATE OF LAST CHANGE TO THIS FILE
                    21600   ;REFRESH VALUE FOR SECONDARY NS (IN SECS)
                    1800    ;RETRY VALUE FOR SECONDARY NS (IN SECS)
                    48384   ;EXPIRE DATA WHEN REFRESH NOT AVAILABLE
                    86400 ) ;MINIMUM TIME TO LIVE VALUE (SECS)
@                    IN NS   Z0EIP.PDL.POK.IBM.COM. ; PRIMARY DNS
0.5.1.0.0.2.0.0.2.1.0.0.9.0.0.0 IN PTR J80IP6.PDL.POK.IBM.COM.
```

Our token ring LAN configuration

As Figure 18 illustrates, we have a total of four token-ring LANs: a backbone ring and three test LANs that use various:

- Communications protocols (TCP/IP, SNA, NetBIOS, and Internet Packet Exchange (IPX))
- Workstation operating systems (AIX, Linux, OS/2, PC DOS, and Microsoft Windows NT, Windows 95, and Windows 2000)
- Workstation types (RS/6000s and various types of PCs)

LANs A, B, and C in Figure 18 use only the token-ring LAN protocol. The three LANs connect to our host systems through the 8281 LAN bridge and 8260 ATM switch as described above. (You can read about our ATM experiences in our December 1998 edition.) For host systems running on CPCs that do not have an ATM connection, we instead use OSA-2 ENTR features to provide direct token-ring connections to each of the three LANs (these connections are not shown in Figure 18).

Note that we also have an OS/2 LAN Server with a CLAW protocol channel adapter that connects to system JE0 for LAN Server. This is not shown in Figure 18; see Figure 21 on page 85 for an illustration of this.

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

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

More about our backbone token ring

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

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

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

What's happening in LAN A?

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

Figure 19 on page 83 depicts our z/OS UNIX DCE test configuration in LAN A, including the connections from the RS/6000s to the backbone (which, for clarity, are not shown in Figure 18 above).

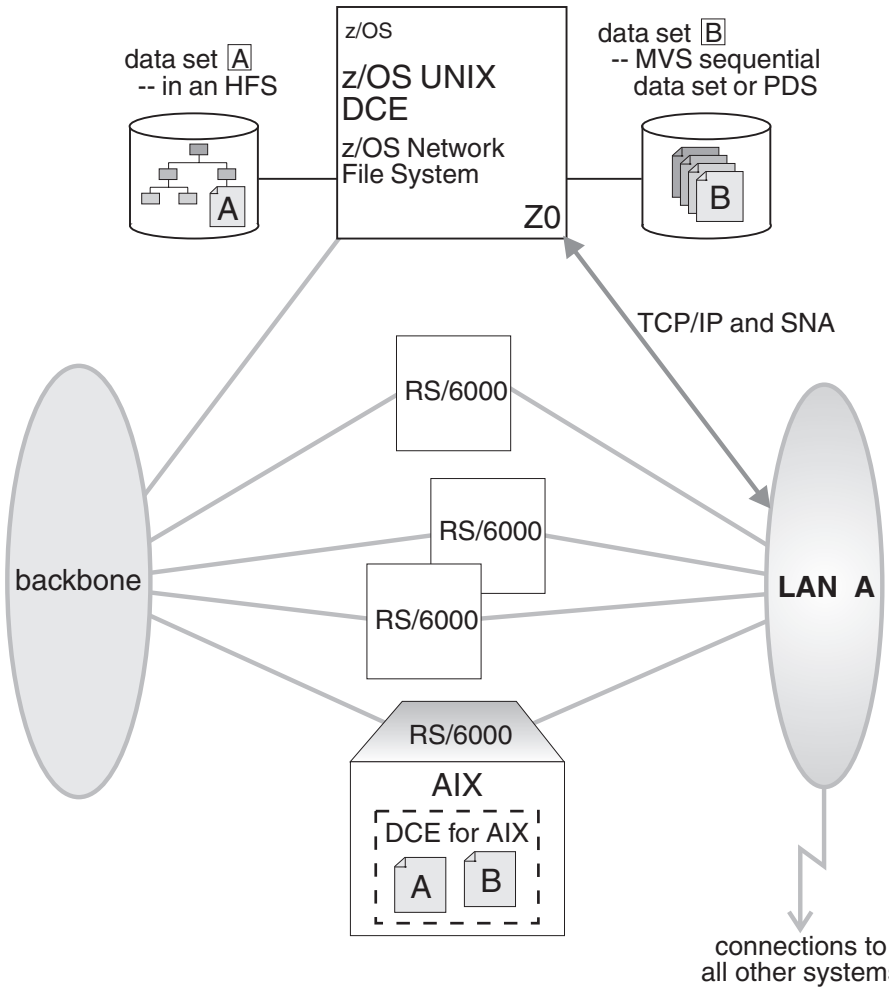


Figure 19. Our token-ring LAN A

The RS/6000 workstations on LAN A all run the AIX operating system. We use them to exercise the z/OS UNIX, DCE, and DFS functions. See “Our workloads” on page 14 for a more detailed description of these workloads. For more information about DCE and DFS, see “DCE and DFS Publications” in Appendix D, “Useful Web sites,” on page 201.

What’s happening in LAN B?

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

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

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

The NFS function communicates with z/OS NFS using TCP/IP, and both the control data and the workload data travel over LAN B. (See “Comparing the network file systems” on page 86 for a description of the different types of NFSs we use.)

Figure 20 depicts our NFS and FTP test configuration in LAN B. For more information about NFS, see “Network File System Publications” in Appendix D, “Useful Web sites,” on page 201.

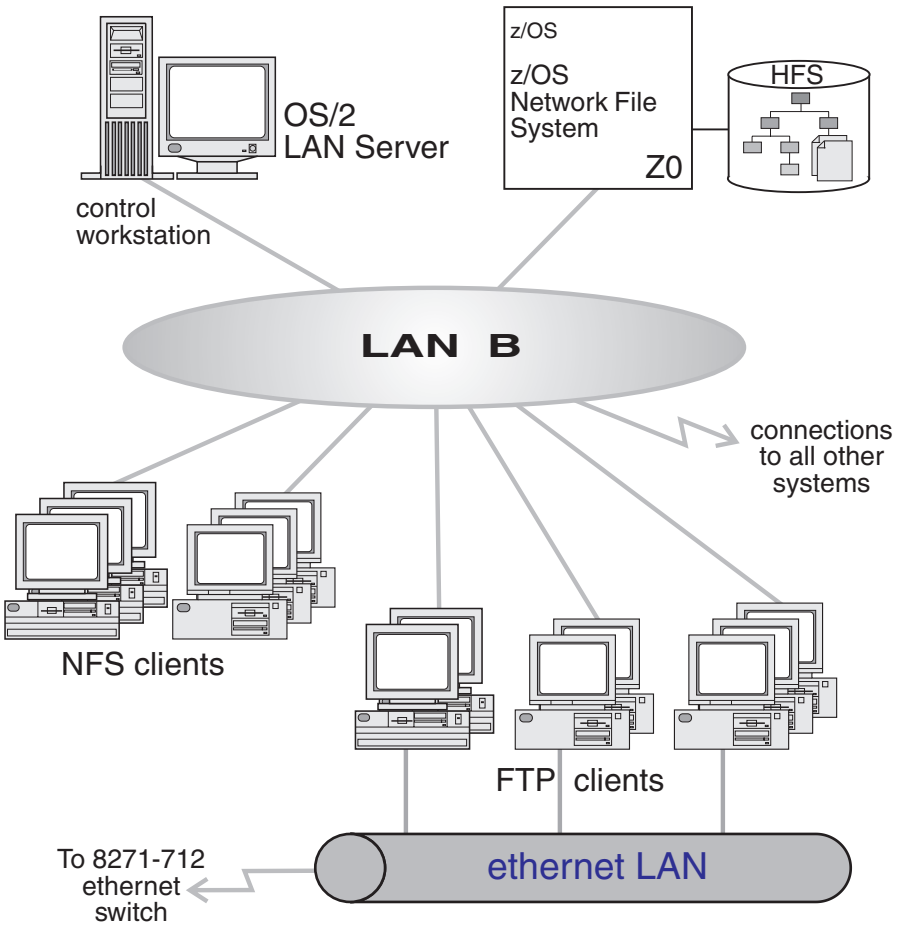


Figure 20. Our token-ring LAN B

What’s happening in LAN C?

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

First, we have an OS/2 LAN Server acting as a FEP (**A**) with a SNA connection to LAN Server in system Z0. We could conceivably connect the OS/2 LAN Server to

any z/OS system, but we currently happen to be using Z0. We use Communications Manager/2 for the SNA connection and APPC communications.

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

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

Figure 21 depicts our LAN Server test configuration in LAN C. For more information about LAN Server, see “LAN Server Publications” in Appendix D, “Useful Web sites,” on page 201.

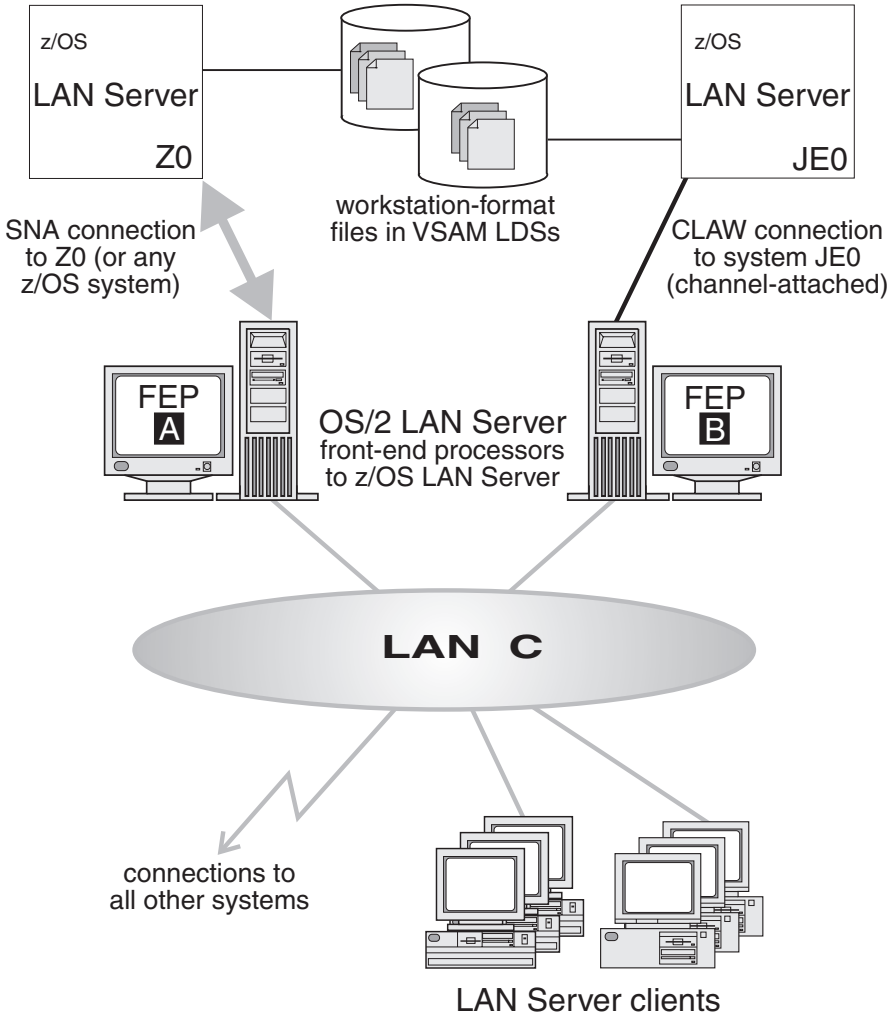


Figure 21. Our token-ring LAN C

Comparing the network file systems

If you are a faithful reader of our test report, you might have noticed that we have changed our Network File System (NFS) approach a number of times, depending on the circumstances at the moment. Currently, we have the z/OS NFS (called DFSMS/MVS NFS in OS/390 releases prior to R6) on system Z0.

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

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

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

Migrating to the z/OS NFS: We plan to implement some of the new functions available in z/OS NFS, such as file locking over the z/OS NFS server and file extension mapping support. You can read descriptions of these new functions in *z/OS Network File System Customization and Operation*, SC26-7417 and *z/OS Network File System User's Guide*. In addition, you can read about WebNFS support in our December 1999 edition, and the use of the LAN Server NFS in our June 2004 edition. We hope to have additional experiences with these new functions to share with you in a future test report.

In the meantime, we'd like to highlight one aspect of the migration to the z/OS NFS. ***Pay attention to the following words*** in the section on allocating the mount handle data sets in *z/OS Network File System Customization and Operation*, SC26-7417: "Delete and allocate the mount handle data sets before running any new versions of the Network File System. If an old mount handle data set is used, the server issues a message and shuts down." We somehow missed this and attempted to migrate without deleting our old data sets and recreating them. When the server shut down, we had a difficult time figuring out why.

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

Networking and application enablement workloads

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

Enabling NFS recovery for system outages

In z/OS V1R6, we improved NFS recoverability and availability by using Automatic Restart Management (ARM) and dynamic virtual IP address (DVIPA) with our NFS server. With these enhancements, the NFS server is automatically moved to another MVS image in the sysplex during a system outage.

Note: We are running a shared HFS environment.

We used the following documentation to help us implement ARM for NFS recovery.

- Automatic Restart Management
  - ARMWRAP as described in the IBM Redpaper *z/OS Automatic Restart Manager* available on the IBM Redbooks Web site.
  - *z/OS MVS Setting Up a Sysplex*, SA22-7625
- Dynamic VIPA(DVIPA)
  - *z/OS Communications Server: IP Configuration Guide*, SC31-8775

Setting up the NFS environment for ARM and DVIPA

Part 1 of Figure 22 on page 88: illustrates how the NFS server on MVS A acquires DVIPA 123.456.11.22. The AIX clients issue a hard mount specifying DVIPA 123.456.11.22. Before the enhancements, the AIX clients specified a static IP address for MVS A. A system outage would result in the mounted file systems being unavailable from the AIX client's perspective until MVS A was restarted.

Part 2 of Figure 22 on page 88 : illustrates that when an outage of MVS A occurs, ARM automatically moves the NFS server to MVS B. The NFS Server on MVS B acquires the DVIPA 123.456.11.22. From the AIX client's perspective the mounted file systems become available once the NFS server has successfully restarted on MVS B. The original hard mount persists.

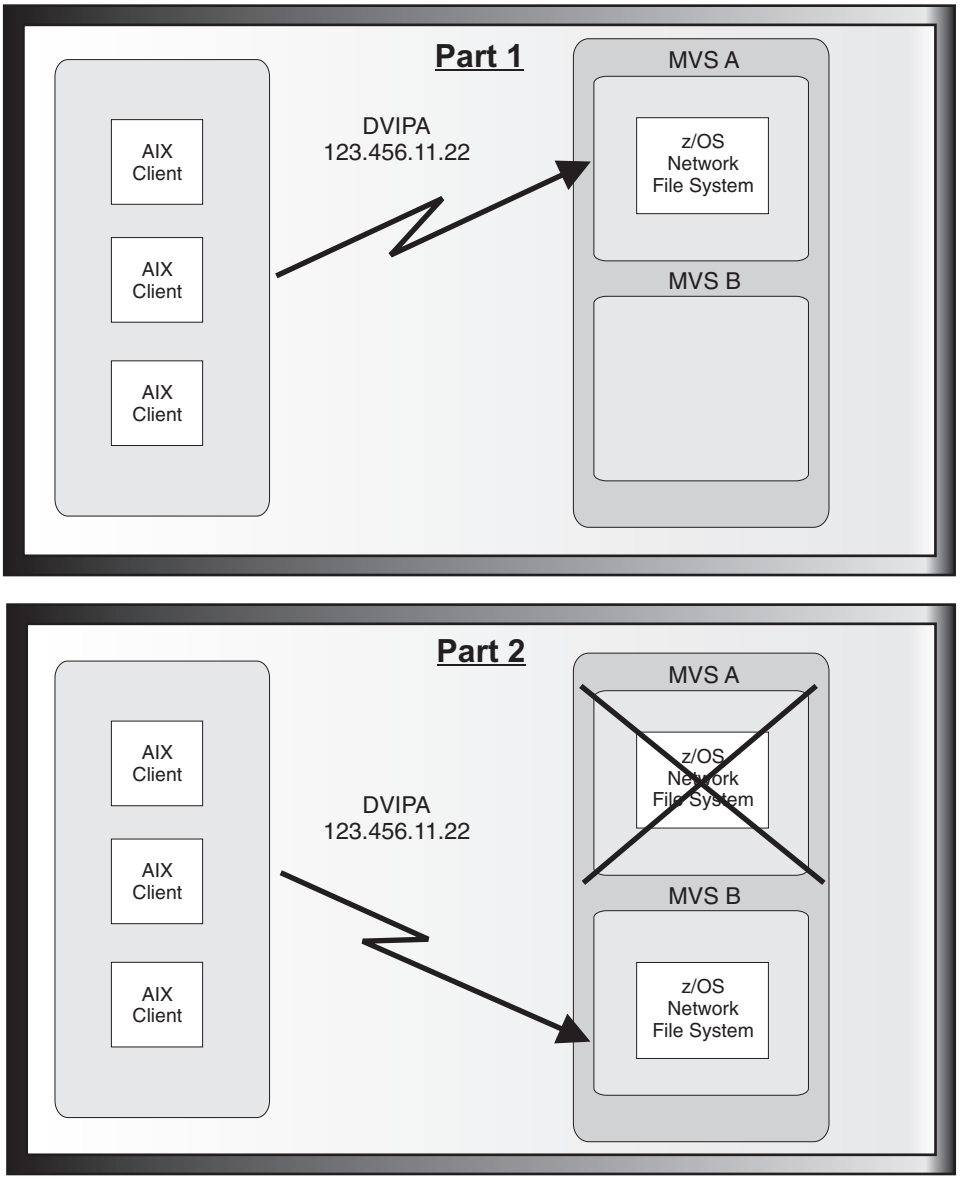


Figure 22. NFS configuration

Note: An ARM enabled NFS will not automatically move back to MVS A after MVS A recovers.

Step for setting up our NFS environment

We performed the following steps to set up our NFS environment for ARM and DVIPA:

1. Acquiring dynamic VIPA:

   We added the following statement in the TCP/IP profiles for MVSA and MVSB to allow NFS to acquire dynamic VIPA:

   ```
   VIPARANGE DEFINE 255.255.255.255 123.456.11.22 ; NFS VIPA
   ```

   We recycled TCPIP on MVSA and MVSB to activate the above changes.

Note: You could also use the `VARY TCPIP,,OBEYFILE` command with a data set that contains VIPARANGE statement.

---

2. Defining the NFS element:

   We added the following statement to our ARM policy member (ARMPOLxx) in SYS.PARMLIB member to define the NFS element:

```
RESTART_GROUP(NFSGRP)
  TARGET_SYSTEM(MVSB)
  FREE_CSA(600,600)
    ELEMENT(NFSSELEM)
      RESTART_ATTEMPTS(3,300)
      RESTART_TIMEOUT(900)
      READY_TIMEOUT(900)
```

---

3. Loading the ARM policy:

   We ran the IXCMIAPU utility to load ARMPOLxx and then activated the policy:

```
setxcf start,policy,type=arm,polname=armpolxx
```

---

4. Registering NFS using an ARM policy:

   We used ARMWRAP, the ARM JCL Wrapper with the following parameters to register NFS as ARM element:

```
//*********************************************************************
//*REGISTER ELEMENT 'NFSSELEM' ELEMENT TYPE 'SYSTCPIP' WITH ARM
//*REQUIRES ACCESS TO SAF FACILITY IXCARM.SYSTCPIP.NFSSELEM
//ARMREG EXEC PGM=ARMWRAP,
//       PARM=('REQUEST=REGISTER,READYBYMSG=N,',
//             'TERMTYPE=ALLTERM,ELEMENT=NFSSELEM,',
//             'ELEMTYPE=SYSTCPIP')
//* ---------------------------------------------------------------- *
//* DELETE VIPA FOR NFS SERVER                                       *
//* ---------------------------------------------------------------- *
//DELVIPA  EXEC PGM=EZBXFDVP,
//            PARM='POSIX(ON) ALL31(ON) /-p TCPIP -d &VIPA'
//SYSPRINT DD SYSOUT=*
//* ---------------------------------------------------------------- *
//* ACQUIRE VIPA FOR NFS SERVER                                      *
//* ---------------------------------------------------------------- *
//DEFVIPA  EXEC PGM=EZBXFDVP,
//            PARM='POSIX(ON) ALL31(ON) /-p TCPIP -c &VIPA'
//SYSPRINT DD SYSOUT=*
```

---

5. Terminating the address space:

   The following example shows what is executed when the address space is terminated:

```
//* ---------------------------------------------------------------- *
//* DELETE VIPA FOR NFS SERVER                                       *
//* ---------------------------------------------------------------- *
//DELVIPA  EXEC PGM=EZBXFDVP,
//            PARM='POSIX(ON) ALL31(ON) /-p TCPIP -d &VIPA'
//SYSPRINT DD SYSOUT=*
//*********************************************************************
//*FOR NORMAL TERMINATION,DEREGISTER FROM ARM
```

```
//*FOR NORMAL TERMINATION,DEREGISTER FROM ARM
//*******************************************
//ARMDREG EXEC PGM=ARMWRAP,
//        PARM=('REQUEST=DEREGISTER')
```

Chapter 7. Using z/OS UNIX System Services

In this chapter, we cover the following z/OS UNIX System Services topics:

z/OS UNIX enhancements in z/OS V1R5

z/OS UNIX made several enhancements in z/OS V1R5. In this section, we cover the following topics:

We used the information in *z/OS UNIX System Services Planning* to help us plan and implement the above enhancements.

Remounting a shared HFS

Remounting a mounted shared HFS is a new function in z/OS V1R5 that allows you to remount an HFS or zFS file system within a directory tree so as to change the access mode. This function is also available for z/OS V1R4 by applying the fix for APAR OA02584.

We have utilized this new function to remount the version HFS from READ mode to RDWR mode. Our normal configuration is to have the version HFS mounted read-only, as IBM recommends. In our environment, we have a requirement to be able to make changes to /usr/lpp and other directories in the version HFS which, until z/OS V1R5, we could only accomplish during our STAGE3 processing (see the discussion of our CUSTHFS procedure in bour December 2001 edition). While we still prepare the HFS for our configuration needs prior to implementing it into production, this new function provides us with the ability to perform additional changes while the HFS is being used in the production environment.

You can perform the remount from the ISHELL panel or by using the TSO unmount command from the server or any of the client systems within the same shared HFS sysplex. The following is an example of the TSO command for remounting our version HFS in read/write mode:

```
unmount filesystem('OMVSSPN.PETPA1.ROOT.FS') remount(rdwr)
```

Mounting file systems using symbolic links

z/OS V1R5 introduces support for using MVS system symbols in symbolic links. This provides the ability to mount different file systems at a logical mount point that resolves to a different path name on different systems. This function uses two new, special identifiers in a symlink, followed by an MVS standard symbolic string template. The special identifier indicates that the text that follows it requires symbolic substitution. We tested this function using HFS and zFS file systems.

The following are the new identifiers:

- $SYSSYMR/*template*

Results in a relative path name. The path name lookup proceeds from its current position in the path name.

- $SYSSYMA/*template*

  Results in an absolute path name. The path name lookup starts over from the root.

Note: When coding the **ln** command, the new identifier ends with a forward slash (/) and the system symbol starts with a backslash (\).

To test the relative and absolute path name lookups, we used one of our existing system symbols, &SYSCLONE. The value of &SYSCLONE. is “Z0” on our system Z0. The following examples illustrate the difference between the relative and absolute path name lookups.

Example: We issued the following **ln** command in the OMVS shell on system Z0:

```
ln -s \$SYSSYMR/\&SYSCLONE./testdir /pet5/rdir
```

On system Z0, we mounted OMVSSPN.Z0.SYMBOLIC.TEST at /pet5/rdir. The HFS was linked with a relative path name and was mounted at /pet5/Z0/testdir.

Example: We issued the following **ln** command in the OMVS shell on system Z0:

```
ln -s \$SYSSYMA/\&SYSCLONE./testdir /pet5/dir
```

On system Z0, we mounted OMVSSPN.Z0.SYMBOLIC.TEST at /pet5/dir. The HFS was linked with an absolute path name and was mounted at /Z0/testdir.

Creating directories during z/OS UNIX initialization

The z/OS UNIX parmlib member, BPXPRM*xx*, now supports a new, optional keyword, MKDIR, on the existing ROOT and MOUNT statements. The MKDIR keyword allows one or more directories to be created in the mounted file system as part of the mount process during z/OS UNIX initialization. You can specify multiple MKDIR keywords on each ROOT or MOUNT statement; the directories are created in the order in which the MKDIR keywords occur. Such directories can serve as mount points that can be used in subsequent MOUNT statements.

The MKDIR keyword has the following syntax:

```
MKDIR('pathname')
```

where *pathname* specifies a relative path name of a directory to be dynamically created after the file system has been successfully mounted. The path name must not start with a slash (/) and must be enclosed in single quotes.

The path name is relative to the file system’s mount point (specified by the MOUNTPOINT keyword) and can contain intermediate directory components but each component must already exist in the file system hierarchy. You can use multiple MKDIR keywords to create the necessary intermediate directories. Note that the length of the MKDIR path name plus the length of the MOUNTPOINT path name must not exceed the value of the PATH_MAX configuration variable.

The directory to be created must reside in a file system that is mounted in RDWR mode. The directory will have permission bits of 755 and will inherid the UID and GID from its parent directory. These attributes will be overlaid when this directory is actually used as a mount point.

Note the following about the usage of the MKDIR keyword:

- Failure to create a directory does not cause the mount to fail. A message is written to the system log if there is a problem creating a directory. No message is written if the directory already exists.
- MKDIR is only supported in the BPXPRM*xx* parmlib member. There is no downlevel support for MKDIR; therefore, only use MKDIR in a common BPXPRM*xx* member when all sharing systems are at z/OS V1R5 or higher.
- File system reinitialization using the MODIFY BPXOINIT,FILESYS=REINIT command does not support the use of MKDIR in the BPXPRM*xx* member.
- The OMVS restart function (that is, MODIFY OMVS,SHUTDOWN followed by MODIFY OMVS,RESTART) does support the use of MKDIR in the BPXPRM*xx* member.
- MKDIR should not be used for file systems that mount asynchronously, such as the network file system (NFS). In such cases, the creation of the directory cannot be guaranteed. Message BPXF025I is issued to the system log when a file system is to be mounted asynchronously.
- Do not use MKDIR with the SYSNAME keyword when SYSNAME identifies a remote system to perform the mount, as the results are unpredictable. MKDIR will not process on a file system that is already mounted on a remote system.

Note also that, in addition to checking statement syntax, the SETOMVS SYNTAXCHECK=(*xx*) command also checks the MVS catalog for the existence of the HFS or zFS data set names used in each ROOT and MOUNT statement in the specified BPXPRM*xx* member. Messages are written to the syslog if any errors are found. Although mount points are not verified, this can help to ensure that mounts will succeed.

Testing the MKDIR keyword

We made the following changes to the mounts for the sysplex root (/) file system, each system-specific /tmp file system, and a zFS /pet3 file system in our common parmlib member, SYS1.PARMLIB(BPXPRM00):

```
ROOT FILESYSTEM('OMVSSPN.SYSPLEX.ROOT.FS') TYPE(HFS)
   MODE(RDWR) MKDIR('mkdirrootv1r5')                                  A

MOUNT FILESYSTEM('OMVSSPN.&SYSNAME..TMP.FS') TYPE(HFS)
   MODE(RDWR) MOUNTPOINT('/&SYSNAME./tmp') UNMOUNT
   PARM('FSFULL(90,5)') MKDIR('mkdirtmpv1r5')                         B

MOUNT FILESYSTEM('OMVSSPN.PET3.ZFS.FS') TYPE(ZFS) MODE(RDWR)
   MOUNTPOINT('/pet3') AUTOMOVE(I,Z0,Z1,Z2,Z3) MKDIR('mkdirpet3v1r5')  C
```

We issued the SETOMVS SYNTAXCHECK=(00) command to perform syntax and data set name checking on the member. The following message appeared:

```
IEE252I MEMBER BPXPRM00 FOUND IN SYS1.PARMLIB
```

We then checked the syslog to verify that there were no error messages.

We observed the following results the next time the systems were initialized:

A — For the sysplex root (/) file system: We had to wait until we could unmount the sysplex root (/) file system before the directory on the MKDIR keyword could be created the next time the root was mounted during z/OS UNIX initialization. The MKDIR on the ROOT statement will not process as long as the root file system is mounted on any system in the sysplex. Therefore, we tested this by taking down all of the systems in the sysplex and re-IPLing. The first system to join the sysplex mounted the root file system and successfully

processed the MKDIR keyword to create the /mkdirrootv1r5 directory. The directory had permission bits of 755 and had the same UID and GID as the parent directory.

B **— For the system-specific /tmp file systems:** Each system's /tmp file system is unmounted when the system is removed from the sysplex. Thus, as we IPLed each system, it mounted the /tmp file system and successfully processed the MKDIR keyword to create the /&SYSNAME./tmp/mkdirtmpv1r5 directory.

C **— For the /pet3 file system:** When we had the /pet3 file system remotely mounted in the sysplex, the MKDIR keyword did not process on that file system. When we unmounted the /pet3 file system, since this file system is defined in the common BPXPRM00 member, the next system to initialize z/OS UNIX mounted the /pet3 file system and successfully processed the MKDIR keyword to create the /pet3/mkdirpet3v1r5 directory.

All processing messages were written to the system log, not to the console.

We also tested to make sure that multiple MKDIR keywords worked properly on the ROOT and MOUNT statements. We added the following MKDIR keywords to our BPXPRM00 member (following the MKDIR keyword that we had previously added):

```
ROOT FILESYSTEM('OMVSSPN.SYSPLEX.ROOT.FS') TYPE(HFS)
   MODE(RDWR) MKDIR('mkdirrootv1r5')
              MKDIR('mkdirrootv1r5/mkdirrootv1r5dir2')                A
              MKDIR('mkdirrootv1r52')                                 B
              MKDIR('mkdirrootv1r52/mkdirrootv1r52dir2')              C

MOUNT FILESYSTEM('OMVSSPN.&SYSNAME..TMP.FS') TYPE(HFS)
   MODE(RDWR) MOUNTPOINT('/&SYSNAME./tmp') UNMOUNT
   PARM('FSFULL(90,5)') MKDIR('mkdirtmpv1r5')
              MKDIR('mkdirtmpv1r5/mkdirtmpv1r5dir2')
              MKDIR('mkdirtmpv1r52')
              MKDIR('mkdirtmpv1r52/mkdirtmpv1r52dir2')

MOUNT FILESYSTEM('OMVSSPN.PET3.ZFS.FS') TYPE(ZFS) MODE(RDWR)
   MOUNTPOINT('/pet3') AUTOMOVE(I,Z0,Z1,Z2,Z3) MKDIR('mkdirpet3v1r5')
              MKDIR('mkdirpet3v1r5/mkdirpet3v1r5dir2')
              MKDIR('mkdirpet3v1r52')
              MKDIR('mkdirpet3v1r52/mkdirpet3v1r52dir2')
```

We added three new MKDIR keywords to each of the above file system mounts. Using the first one (ROOT) as an example, we added MKDIR keywords, as follows:

A — To create a new directory under an existing directory

B — To create a new directory upon the next mount activity

C — To create a new directory under the directory created in **B**

The file systems were mounted and the directories were successfully created in the same manner as previously described. Further, we did notice that when a mount contains a series of MKDIR keywords, if one of the MKDIRs in the series fails, it does not prevent the subsequent MKDIRs from being attempted.

Testing the SYNTAXCHECK keyword

After making the above changes to the BPXPRM00 parmlib member, we issued the following command:

```
SETOMVS SYNTAXCHECK=(00)
```

Error messages, if any, are only written to the system log.

When we ran this command on a z/OS V1R5 system, it reported no errors for the MKDIR keyword. When we ran it on a z/OS V1R4 system and specified our V1R5 BPXPRMxx member, it reported errors for the MKDIR keyword.

Early in our testing, we experienced a problem such that if the syntax check encountered an uncataloged file system data set, it would then flag all file system data sets after it as being uncataloged, whether they were or were not. z/OS UNIX APAR OA05966 resolved this problem.

Temporary file system (TFS) enhancements

The temporary file system (TFS) is an in-memory physical file system that supports in-storage mountable file systems. A TFS can run in the z/OS UNIX kernel address space but, for 64-bit exploitation, it is preferable to run it in a logical file system (LFS) colony address space.

Overview of the TFS enhancements that we tested

Other enhancements to TFS in z/OS V1R5 include the following:

- "New parameters for mounting a TFS"
- "STOP and MODIFY command support for TFS colony address spaces" on page 96
- "Access control list (ACL) support" on page 97

New parameters for mounting a TFS: Each TFS mount in 64-bit mode consumes space, as requested, above the 2G bar. In addition, TFS allocates some control blocks and a buffer cache below the bar. The default buffer cache size is 1M bytes.

The **mount** command for a TFS supports the following parameters:

Parameter	Description
-s *size*	*size* is the number of megabytes for the file system (default=1). If the specified value is larger than the size that can be supported, the maximum size will be used.
-b *block*	*block* specifies the blocking factor used to set the size of a TFS block. Range is 0-4 (default=0). The blocking factor relates to the TFS block size as follows:

Blocking factor	Resulting TFS block size
0	4K
1	8K
2	16K
3	32K
4	64K

-c *cache*	*cache* is the amount of buffer storage, in megabytes, that TFS will use in the 31-bit address range to support a 64-bit file system. This parameter is ignored for file systems allocated in the 31-bit range. Range is 1-64; the default value is calculated based on the TFS block size such that the number of cache buffers is 256. Out-of-range values will be set to the closest range boundary.

-u *uid*	*uid* is the numeric UID to be assigned to the file system's root directory (default=0).
-g *group*	*group* is the numeric GID to be assigned to the file system's root directory (default=0).
-p *perm*	*perm* is the permission bits, in octal, to be assigned to the file system's root directory (default=0777).
-3	Specifies that TFS is to allocate the file system in 31-bit storage, regardless of system capabilities.

TFS dynamically determines if 64-bit addressing is enabled and, if so, places file systems above the bar. The -3 parameter on the mount command forces TFS to place a file system in 31-bit storage even if 64-bit addressing is enabled. Because locating a file system above the bar still requires a buffer pool to reside below the bar, TFS will only locate a file system above the bar when it is at least 5M bytes large and the amount of below-the-bar storage needed is less than the size of the file system above the bar.

The maximum file size that TFS can support is a function of the TFS block size. The following are the approximate maximum file sizes based on the blocking factor:

Blocking factor	Approximate maximum file size
0	2 gigabytes (G)
1	25G
2	240G
3	2 terabytes (T)
4	17T

The maximum file system size is also a function of the TFS block size, but is always limited to a maximum of 2**31–1 (X'7FFFFFFF') blocks. Using the default block size of 4K, this yields a maximum file system size of about 2**43 bytes. Increasing the block size to 64K yields a maximum file system size of about 2**47 bytes.

For information about our test experiences, see “Testing TFS colony startup and mounting the file system” on page 97.

STOP and MODIFY command support for TFS colony address spaces: TFS, running in a colony address space, will now respond to the MVS STOP and MODIFY commands. (The STOP and MODIFY commands are not supported when TFS runs in the kernel address space.) When you issue the STOP command, TFS will stop if no file systems are mounted. The following MODIFY commands are also available to stop TFS or to force it to stop or terminate even if file systems are mounted:

MODIFY parameter	Description
STOP	This is the same as the STOP command. It causes TFS to exit if no file systems are mounted. A WTOR message is issued allowing TFS to be restarted.
TERM	Causes TFS to exit if no file systems are mounted and does not

issue a WTOR to restart TFS. The SETOMVS RESET=(*xx*) command can be used to start another TFS.

FORCESTOP Similar to STOP, but TFS will terminate even if there are mounted file systems.

FORCETERM Similar to TERM, but TFS will terminate even if there are mounted file systems.

For information about our test experiences, see “Testing TFS colony STOP and MODIFY commands” on page 98.

Access control list (ACL) support: TFS now supports ACLs. There are no unique external interfaces other than ACL limits. The number of ACL entries that TFS supports is limited by the block size. Each ACL uses one TFS block. For example, if the TFS block size is set to 4K (the default: -b0), it will limit the number of entries in any ACL to about 500 entries.

For information about our test experiences, see “Testing TFS colony access control list support” on page 99.

Testing the TFS enhancements

The following are some of our experiences with testing the TFS enhancements in z/OS V1R5:

Testing TFS colony startup and mounting the file system: We did the following to start TFS in a colony address space and mount the file system:

1. Created the following TFS startup procedure in *hlq*.PROCLIB(TFS):

```
//TFS    PROC REGSIZE=0M
//TFSGO  EXEC PGM=BPXVCLNY,REGION=&REGSIZE,TIME=1440
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSOUT   DD DUMMY
//CEEDUMP  DD DUMMY
//*     PEND
```

2. Modified *hlq*.PARMLIB(BPXPRM00) (our common z/OS UNIX parameter member) to specify that the TFS file system is to start in a colony address space using the TFS start up procedure:

```
FILESYSTYPE TYPE(TFS)
ENTRYPOINT(BPXTFS)
ASNAME(TFS,'SUB=MSTR')
```

   Note: We chose to start the TFS physical file system outside of JES (SUB=MSTR) which imposes some restrictions for SYSOUT (see *z/OS UNIX System Services Planning* for more information). Otherwise, to start TFS under JES, the ASNAME parameter would simply be `ASNAME(TFS)`.

We currently do not use a TFS for the /tmp file system. We have a /tmp/tfs directory in the /tmp file system on one of our 64-bit systems (Z0) and we mount a TFS at this mount point.

3. Mounted the TFS file system in *hlq*.PARMLIB(BPXPRMZ0), which is a system-specific z/OS UNIX parameter member (we use both BPXPRM00 and BPXPRMZ0 to initialize OMVS on system Z0):

```
MOUNT FILESYSTEM('/Z0/TMP/TFS') TYPE(TFS) MODE(RDWR)
   MOUNTPOINT('/tmp/tfs') PARM('-s 10') UNMOUNT
```

Following the next IPL of system Z0, the TFS colony started in 64-bit mode outside of JES. The TFS was successfully mounted at the /tmp/tfs mount point.

4. We tested cancelling TFS and restarting it. We issued the following command to cancel TFS:

   ```
   CANCEL TFS
   ```

   Result: The following messages appeared:

   ```
   BPXF063I FILE SYSTEM /Z0/TMP/TFS 121
   WAS SUCCESSFULLY UNMOUNTED.
   *nnnn BPXF032D FILESYSTYPE TFS TERMINATED.  REPLY 'R' WHEN READY TO
   RESTART. REPLY 'I' TO IGNORE.
   ```

   We replied `R` to the WTOR message and TFS successfully restarted.

---

5. We tested unmounting and remounting the TFS file system using various combinations of parameters on the **mount** command. We also tested the -3 parameter, which forced the TFS to be mounted in 31-bit mode, even though it was running on a 64-bit system. The messages associated with a TFS mount go to the syslog.

   Example: The following are examples of the TFS mount messages that appear in the syslog:

   ```
   BPXTF006I TFS MOUNTED /Z0/TMP/TFS
   BPXTF007I FILESYSTEM SIZE=1,048,576  MAX FILE SIZE=2,147,483,648
   ```

During our testing, we also observed the following:

- Double messages appeared in the syslog for the TFS mounts. This was resolved by z/OS UNIX APAR OA05417.
- When we tried using some incorrect parameters or out-of-range parameter values on the mount, certain displays (such as the **df** shell command and the D OMVS,F system command) still showed the incorrect information without indicating any error. In the case of an out-of-range parameter value, the parameter's default value was automatically used in place of the out-of-range value. This behavior is documented in z/OS UNIX documentation APAR OA06175.

Testing TFS colony STOP and MODIFY commands: We tested various combinations of the STOP and MODIFY commands on an active TFS colony.

1. Using the STOP TFS or MODIFY TFS,STOP command while no file system was mounted, TFS stopped, issued the WTOR message to restart, and did not allow any TFS mounts in the interim.

   Example: `STOP TFS` or `MODIFY TFS,STOP`

   Result:

   ```
   *nnnn BPXF032D FILESYSTYPE TFS TERMINATED.
   REPLY 'R' WHEN READY TO RESTART. REPLY 'I' TO
   IGNORE.
   ```

   When we attempted to mount the file system, we received the following error, as expected:

   ```
   RETURN CODE 0000007A, REASON CODE 052C00B6. THE MOUNT FAILED FOR FILE SYSTEM /Z0/TMP/TFS.
   ```

We then replied R to the WTOR message. TFS restarted and we successfully mounted the file system.

---

2. Using the STOP TFS or MODIFY TFS,STOP command while a file system was mounted, TFS did not stop until the file system was unmounted (or FORCESTOP was issued, as below).

   Example: `STOP TFS` or `MODIFY TFS,STOP`

   Result:

   ```
   BPXTF002I TFS TERMINATION REQUEST FAILED DUE TO ACTIVE MOUNTS
   ```

   Once we unmounted the file system, the STOP and MODIFY TFS,STOP commands functioned as in case 1, above.

   Note: When we issued the MODIFY TFS,FORCESTOP command with and without a file system mounted, TFS did stop each time.

---

3. Using the MODIFY TFS,TERM command while a file system was mounted, TFS did not stop until the file system was unmounted.

   Example: `MODIFY TFS,TERM`

   Result:

   ```
   BPXTF002I TFS TERMINATION REQUEST FAILED DUE TO ACTIVE MOUNTS
   ```

   When we issued the same command with no active file system mounts, TFS was terminated and did not issue a WTOR message to restart:

   ```
   BPXTF001I TFS TERMINATION REQUEST ACCEPTED
   ```

   To restart TFS, we issued the SETOMVS RESET=(00) command. TFS successfully started and we successfully mounted the file system.

---

4. Using the MODIFY TFS,FORCETERM command (both with and without a file system mounted), TFS was terminated and did not issue a WTOR message to restart.

   Example: `MODIFY TFS,FORCETERM`

   Result:

   ```
   BPXTF003I TFS UNCONDITIONAL TERMINATION REQUEST ACCEPTED
   ```

   To restart TFS, we issued the SETOMVS RESET=(00) command. TFS successfully started and we successfully mounted the file system.

Testing TFS colony access control list support: Support for access control lists in TFS uses the same interfaces as for the HFS and zFS file systems. We tested the **setfacl** and **getfacl** commands using files and directories in the TFS and they successfully functioned the same way they do for HFS and zFS.

z/OS UNIX enhancements in z/OS V1R6

z/OS UNIX made several enhancements in z/OS V1R6. In this section, we cover the following topics:

- "Shells and utilities support for 64-bit virtual addressing" on page 105
- "Using distributed BRLM" on page 113
- "Using ISHELL enhancements" on page 115

We used the information in *z/OS UNIX System Services Planning* to help us plan and implement these enhancements.

Using multipliers with BPXPRMxx parameters

The z/OS UNIX parmlib member, BPXPRM*xx*, now allows the use of multipliers with certain parameters. For example, if you currently use MAXFILESIZE(1073741824); this function will let you to enter MAXFILESIZE(1M).

The BPXPRM*xx* parmlib parameters that accept the multiplier function are:

- MAXFILESIZE
- MAXCORESIZE
- MAXASSIZE
- MAXMAPAREA
- MAXSHAREPAGES
- IPCSHMMPAGES

Each parameter has specific limits, which are found in the BPXPRM*xx* chapter of *z/OS MVS Initialization and Tuning Reference*.

The following character abbreviations are used:

Table 12. Character Parameter Limit Multipliers

Denomination value	Character abbreviation	Bytes
Null		1
Kilo	K	1,024
Mega	M	1,048,576
Giga	G	1,073,741,824
Tera	T	1,099,511,627,776
Peta	P	1,125,899,906,842,624

Testing the multipliers

We used the SETOMVS command to test the new multiplier function. The following commands were issued:

- SETOMVS MAXSHAREPAGES=3M
- SETOMVS MAXMMAPAREA=130K
- SETOMVS MAXCORESIZE=8M
- SETOMVS MAXCORESIZE=100M
- SETOMVS IPCSHMMPAGES=16283G
- SETOMVS MAXFILESIZE=1G

Using the superkill option

The superkill option allows you to force the ending of a process or job. The superkill option is available in the following environments:

- kill command

  The superkill option was added to the kill command (-K).

You must issue a kill signal to the process you want to superkill first.

Examples:

1. We issued superkill without issuing kill first.

```
kill -K 84017224
kill: FSUMF344 84017224: Cannot superkill without prior KILL signal to process
```

2. We attempted unsuccessfully to superkill all processes.

```
kill -K -- -1
kill: FSUMF342-1: Cannot superkill pid-1 (all processes)
```

3. We attempted unsuccessfully to superkill a process group.

```
kill -K -- -2
kill: FSUMF343-2: Cannot superkill a process group
```

See *z/OS UNIX System Services Command Reference*.

- console support

  The superkill option was added to the MODIFY BPXOINIT console command (SUPERKILL=pid). You are not required to send a KILL signal to the process before issuing the superkill from the console support.

  Example: To obtain the Process Identification (PID) of the process you want to work with, issue a command such as `D OMVS,A=ALL` or (in the shell) `ps -ef`.

```
F BPXOINIT,SUPERKILL=50462791
BPXM027I COMMAND ACCEPTED.
IEA989I SLIP TRAP ID=X422 MATCHED. JOBNAME=ALEASE2, ASID=0171.
   <<Note that we have the S422 dump suppressed, otherwise it would have
     produced a dump for a X'422' abend, reason code X'0109'>>
BPXB0181 THREAD 2675698000000002, IN PROCESS 50462791, ENDED 153
WITHOUT BEING UNDUBBED WITH COMPLETION CODE OF 03422000, AND REASON CODE
0D2C0109.
```

  See *z/OS MVS System Commands*.

- ISHELL support

  There is TSO ISHELL support for superkill.

  Logon to TSO —> ISPF —> ISH —> Tools —> 1. Work with processes (ps) —>

  After a list of processes is displayed, and you enter an ″s″ (S=Signal) action code, the superkill signal number will be ″99″.

  Note that you must enter a sigkill (signal number 9), at least 3 seconds prior to entering the superkill, or you will be presented with a message with return code of Errno=79x (the parameter is incorrect), and Reason=0D1005D8 (JRSigkillNotSent).

The following set of restrictions applies:

- You cannot do a superkill via pthread_kill or sigqueue(). The superkill option is ignored for these services.
- You cannot do a superkill to a group or specify a PID of –1 (kill everyone).
- Superkills will be deferred in the case where a target process has blocked all signals via the BPX1SDD service. The 'defer signal' function was created for conditions which cannot deal with USS abends on their system task. For this reason superkills will also have no effect on these processes. The kill() will not fail but will simply be ignored by the target process.
- A regular SIGKILL must be sent to a process before it can be superkilled. If not, the attempt will result in EINVAL/JRSigkillNotSent. This is analogous to the required 'cancel' before a 'force arm'. However, if using the console form of superkill, the sigkill (cancel) before does not apply.

If the environment is valid then the target process will be abended with a X'422' abend, reason code X'0109'.

Using wildcard characters in the automove system list (SYSLIST)

We tested the new wildcard character support for the automove system list (SYSLIST) on our systems. This function lets you use wildcards in certain situations when you specify the automove system list. Before this enhancement, if an AUTOMOVE INCLUDE with the SYSLIST function was used for a file system mount, only the list of systems specified were eligible to become owners of the file system. If none of the systems specified were active, the system would then unmount the file system. Now, however, using the wildcard support, you can both specify your preferred ownership system candidate systems and use a wildcard to allow any of the remaining systems in the shared HFS sysplex eligible to become the file system owner to prevent the system from unmounting the file system.

We tested this support by mounting a file system using the INCLUDE statement, and specifying a subset of the systems in our 14-way sysplex as well as a wildcard character (*). In the example below shows a MOUNT statement for filesystem OMVSSPN.PET1.FS with an AUTOMOVE system list:

```
MOUNT FILESYSTEM('OMVSSPN.PET1.FS') TYPE(HFS) MODE(RDWR)
  MOUNTPOINT('/pet1') AUTOMOVE(I,Z1,Z2,Z3,*)
```

Display of mounted filesystem showing automove attributes follows:

```
HFS            15 ACTIVE                      RDWR
  NAME=OMVSSPN.PET1.FS
  PATH=/pet1
  OWNER=J80       AUTOMOVE=I CLIENT=Y
  INCLUDE SYSTEM LIST:  Z1        Z2       Z3       *
```

Note that system J80 is the owning system for the file system. In the event that either someone issues a file system shutdown on system J80 or if system J80 leaves the sysplex, the system transfers ownership of the file system to system Z1, if it's active. If Z1 is inactive, Z2 gets ownership, and so forth down the list. If system Z3 isn't active then any remaining active image in the sysplex will be randomly selected to become the owner.

You can also use either AUTOMOVE(I,*) or AUTOMOVE(YES) to specify that **any** active system in the sysplex is eligible to take ownership of the file system. For example, the following two MOUNT statements would yield the same result that any active system can take over OMVSSPN.PET5.ZFS:

1

```
MOUNT FILESYSTEM('OMVSSPN.PET5.ZFS') TYPE(ZFS) MODE(RDWR)
 MOUNTPOINT('/pet5') AUTOMOVE(I,*)
```

2

```
MOUNT FILESYSTEM('OMVSSPN.PET5.ZFS') TYPE(ZFS) MODE(RDWR)
 MOUNTPOINT('/pet5') AUTOMOVE(YES)
```

In all methods of issuing a MOUNT, the wildcard in AUTOMOVE must **always** be the last item in the system list. This includes PARMLIB, TSO, shell, ishell, C program, assembler program and REXX program methods.

Note that the system does not validate the system identifiers specified in the system list (Z1, for example) until ownership of a file system is being transferred. However, using the wildcard at the end of the list ensures that the file system can always find a new owner even if you spell every single system identifier incorrectly. This feature will prevent your file system from being unmounted unexpectedly. The wildcard feature is very useful for installations that want to:

- Make sure a file system is always mounted
- Do not want to list every system in the system list

Using the clear and uptime shell commands

We tested the new V1R6 clear and uptime shell commands:

- "Using the clear command"
- "Using the uptime command"

For more information about these commands, see the *z/OS UNIX System Services Command Reference*.

Using the clear command

The clear shell command clears the screen of all previous output and places the prompt at the top of the page. Similarly to the `tput clear` command, you cannot use this command if you've accessed the shell using a 3270 window. The clear utility has no options, and if you try to enter any options, the system issues a usage message.

The following is our experience with the clear utility:

- **From OMVS (TSO)**: From the shell, we entered `clear`. The command cleared the screen, leaving a blank line and the prompt at the top of the screen. We were able to use the PF7 key to return to previous pages. We also entered more display commands, ls –al for example, and then entered the clear command again. Again, the command cleared the screen, put the prompt on the top. We used PF7 (BACKSCR) and PF8(SCROLL) to navigate.
- **From rlogin or telnet (AIX)**: We accessed from AIX using the rlogin command and then entered `clear`. The command cleared the screen and the prompt was displayed at the top of the screen. We also entered more display commands, ls –al for example, and then entered the clear command again. Again, the command cleared the screen, put the prompt on the top. Note that from rlogin, the clear command actually removes the data on the screen, so that even if you scroll back, the data will no longer be displayed. When we did scroll back, the command prompt went back to the top of the screen when we began typing again.
- **From telnet (3270 window)**: We accessed the shell using telnet on TSO then entered `clear`. The command **did not** clear the screen. The prompt is simply redisplayed.

Using the uptime command

The uptime command gives a one-line display with the following information:

- Current time
- How long the system has been running
- Number of users who are currently logged into z/OS UNIX and the system load averages for the past 1, 5, and 15 minutes. Load averages are not supported on z/OS UNIX, and are displayed as 0.00

We tested the uptime command from telnet, rlogin, and TSO (OMVS) sessions. Note that the uptime command has no documented parameters, and when we tried to use parameters, we received a usage message.

The following example shows how we entered the uptime command and the output from it:

```
# uptime
06:49PM  up 5 day(s), 01:55,  1 users,  load average: 0.00, 0.00, 0.00
```

The load average value is always 0.00 on our systems because we are running on z/OS UNIX.

Enhanced latch contention detection

In z/OS V1R6 changes were implemented to eliminate outages caused by address spaces that terminate without cleaning up the ownership of the acquired USS global resource serialization latch. The kernel now detects latch contention on a timed basis and initiates one of the following:

- Attempts to correct the problem
- Issues a message that contention exists.

An action is taken if the latch that is causing contention is held for an excessive amount of time. If the kernel detects that the oldest latch holder's address space or process no longer exist, it corrects the latch contention problem. If the kernel detects a latch holder's address space and process still exist, the following eventual action message is displayed indicating that additional actions might be required:

BPXM056E UNIX SYSTEM SERVICES LATCH CONTENTION DETECTED

If the message does not get DOMed after a reasonable amount of time, your installation should have an automation script or an operator to react to this message and eventually issue the **F BPXOINIT,RECOVER=LATCHES** command to attempt to resolve the contention.

The abnormal termination is caused by a 422-1A5 abend that generates a system dump. The dump is generated because of the indication of an internal system problem. If more than one latch is in contention, multiple tasks can abend and result in multiple dumps being requested. However, prior to issuing the recovery command, it is advisable to use the `D GRS` command to determine the resources and address spaces that are involved with the contention. For example, use `D GRS,C`.

Once the latch contention is resolved, message BPXM056E will be DOMed. This command can take several minutes to resolve the latch contention. If the command cannot resolve the latch contention within a reasonable amount of time, the following eventual action message is displayed and message BPXM056E is DOMed: *BPXM057E UNIX SYSTEM SERVICES LATCH CONTENTION NOT RESOLVING*.

Note: A new 422-1A5 abend is introduced that can terminate a user task holding a USS latch for an excessive amount of time.

The following new message indicates the entry of an unsupported operand when using the **F BPXOINIT,RECOVER=** command: *BPXM058I MODIFY BPXOINIT RECOVER COMMAND REJECTED*.

Testing contention recovery

The following example shows what occurred when we used **F BPXOINIT,RECOVER=LATCHES**:

The SYS.BPX.AP00.PRTB1.PPRA.LSN type latches were held for approximately 6 hours on one system. After issuing **F BPXOINIT,RECOVER=LATCHES**, these latches were relieved (received abend 422/1A5), and the BPXM056E message was DOMed. In our installation no dump was taken because we have S422 abends suppressed (SLIP SET,C=422,ID=Y422,A=NODUMP,E).

Example:

```
D GRS,C
ISG343I 10.34.16 GRS STATUS 167
S=STEP    SYSZBPX  PROCINIT
SYSNAME   JOBNAME            ASID       TCBADDR   EXC/SHR    STATUS
JC0       U078023            0242       008E79C0 EXCLUSIVE    OWN
JC0       U078023            0242       008E7B58 EXCLUSIVE    WAIT
NO REQUESTS PENDING FOR ISGLOCK STRUCTURE
LATCH SET NAME:  SYS.BPX.AP00.PRTB1.PPRA.LSN
CREATOR JOBNAME: OMVS      CREATOR ASID: 000F
  LATCH NUMBER:  908
    REQUESTOR  ASID  EXC/SHR    OWN/WAIT
    U078025    0221  EXCLUSIVE  OWN
    U078023    0242  EXCLUSIVE  WAIT

F BPXOINIT,RECOVER=LATCHES
BPXM027I COMMAND ACCEPTED.
IEA989I SLIP TRAP ID=Y422 MATCHED.  JOBNAME=U078025 , ASID=0221.
BPXP018I THREAD 26A6D5F000000002, IN PROCESS 524404, ENDED 171
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0F422000, AND REASON CODE
000001A5.
BPXM067I UNIX SYSTEM SERVICES LATCH CONTENTION RESOLVED

D GRS,C
ISG343I 10.36.39 GRS STATUS 188
NO ENQ RESOURCE CONTENTION EXISTS
NO REQUESTS PENDING FOR ISGLOCK STRUCTURE
NO LATCH CONTENTION EXISTS
```

There were instances when we issued the **F BPXOINIT,RECOVER=LATCHES** and received the BPXM067I message with no S422/1A5 abends. This indicates that there is no latch contention for this function to attempt recovery from.

Example:

```
F BPXOINIT,RECOVER=LATCHES
BPXM027I COMMAND ACCEPTED.
BPXM067I UNIX SYSTEM SERVICES LATCH CONTENTION RESOLVED
```

Shells and utilities support for 64-bit virtual addressing

We tested several z/OS UNIX shell utilities that were enhanced to support 64-bit virtual addressing. These enhanced utilities support the creation and use of 64-bit applications.

Overview of 64-bit support

The utilities that we tested fall into one of three groups:

- Utilities that report information about an executable or object file that has been compiled in 64-bit mode (such as **nm** and **file**): These utilities recognize 64-bit executables and symbols within executables and object files and libraries.

  For example, the **file** utility indicates whether a file is an executable and, if so, whether it is a 64-bit executable.

- Utilities that report information about 64-bit processes (such as **ps** and **ipcs**): These utilities report information about memory usage by 64-bit processes and threads.

  For example, **ps** reports memory usage above the bar by a running process. The **ipcs** utility reports information on shared memory that can be allocated and attached above the bar by 64-bit programs.

- Utilities that manage object files or executables (such as **ar**, **cp** and **mv**, and the **lex** and **yacc** libraries). These utilities support 64-bit executables and object files.

  For example, the **ar** utility, which creates and manages object libraries, now stores the addressing mode symbols. The **cp** and **mv** utilities, when used to

move executables from MVS data sets to the z/OS UNIX file system, re-binds 64-bit executables. Also, **lex** and **yacc** provide object libraries with 64-bit versions of functions to support 64-bit code created using **lex** or **yacc**.

The shell utilities themselves are not compiled as 64-bit applications.

Examples of the utilities that we tested

The following are some examples that highlight our testing of the 64-bit virtual addressing support in various shell utilities.

The file utility: The **file** utility determines the format of each file by inspecting the attributes and, for a regular file, by reading an initial part of the file. If the **file** is an executable, file determines its addressing mode for output. If the file is not an executable, **file** compares it to templates found in a magic file to determine the file type.

A word about the magic file: The magic file defines the following output text for an executable:

```
OpenEdition MVS executable
```

This wording is outdated; therefore, you may wish to update your /etc/magic file using the new /samples/magic file. The updated /samples/magic file now defines the following output text for an executable:

```
z/OS UNIX executable
```

Examples of testing the file utility: The following examples highlight our experiences testing the **file** utility.

Example: `file /bin/makedepend`

Result: The file utility does not support the display of information for external links and issues the following response:

```
FSUM8718 /bin/makedepend:       cannot open: EDC5129I No such file or directory.
```

The output from the **ls -al** command shows that /bin/makedepend is an external link:

```
erwxrwxrwx   1 ALEASE1  OMVSGRP        8 Jun 22 09:01 /bin/makedepend -> CCNEMDEP
```

Example: `file /bin/*`

Result: The following excerpt of the command response shows both AMODE 31 and AMODE 64 executables:

```
⋮
/bin/dbx24:.z/OS Unix executable
/bin/dbx31:.z/OS Unix executable
/bin/dbx31vdbg:.z/OS Unix executable (amode=31)
/bin/dbx64:.z/OS Unix executable
/bin/dbx64vdbg:.z/OS Unix executable (amode=64)
/bin/dce_err:.z/OS Unix executable (amode=31)
/bin/dce_login:.z/OS Unix executable (amode=31)
/bin/dcecf_postproc:.commands text  - Bourne or POSIX shell script
⋮
```

The nm utility: The man pages for the **nm** utility indicate that the –M option inserts three columns preceding the symbol name in the command output. The three columns have the following format:

```
rmode amode compiler_options
```

The rmode and amode columns display one of the following:
- 24-bit mode
- 31-bit mode
- 64-bit mode
- ANY mode
- MIN mode
- Undetermined or not applicable

The compiler options column dispays a character for each compiler option in effect, or a dash if no options are in effect, as follows:

I Symbol is compiled with IPA. (IPA will not be seen when running nm against an executable as that information is no longer available.)

X Symbol is compiled with XPlink.

Examples of testing the nm utility: The following examples highlight our experiences testing the **nm** utility.

As in the example of the **file** utility above, the **nm** utility does not support the display of information for external links and issues the same error message as above.

Example: `nm -M /bin/dbx64vdbg`

Result:

```
560 T  64  64 -    BPX4PTR
560 U --- --- -    BPX4PTR
224 T MIN MIN X    BPXISD64
208 T  64  64 X    BPXISD64#C
  0 U --- --- -    B_IMPEXP
  0 U --- --- -    B_LIT
  0 U --- --- -    B_PRV
  0 U --- --- -    B_PRV
  0 U --- --- -    B_PRV
  0 U --- --- -    B_PRV
  0 U --- --- -    B_PRV
  0 U --- --- -    B_PRV
208 U --- --- -    B_TEXT
  0 U --- --- -    B_TEXT
584 U --- --- -    B_TEXT
760 U --- --- -    B_TEXT
560 U --- --- -    B_TEXT
952 U --- --- -    B_TEXT
840 U --- --- -    B_TEXT
608 T MIN MIN -    CEELLIST
760 T  64  64 -    CELQETBL
760 U --- --- -    CELQETBL
584 U --- --- -    CELQLLST
584 U --- --- -    CELQLLST
584 T  64  64 -    CELQLLST
  0 T  64  64 -    CELQSTRT
  0 U --- --- -    CELQSTRT
960 T MIN MIN X    CELQTLOC
952 T  64  64 -    CELQTLOE
840 T  64  64 -    CELQTRM
840 U --- --- -    CELQTRM
224 T MIN MIN X    CELQVDBG
```

```
0 U --- --- -    C_DATA64
0 T ANY ANY -    IEWBCIE
0 T ANY ANY -    IEWBLIT
0 U --- --- -    IEWBLIT
```

The ps utility: The **ps** utility now shows the memlimit and and amount of storage in use above the 2-gigabyte bar. Two new output specifiers (vszlmt64 and vsz64) are also available.

The following are the pertinent fields of the **ps** display:

vsz — Displays the amount of memory (virtual storage) that the process is using, as a decimal number of kilobytes.

vszlmt64 — Displays the maximum amount of virtual storage above the 2-gigabyte bar allowed for the current process. In the display, each value is followed by a multiplier indicating the units represented: (space) = no multiplier; K = Kilo; M = Mega; G = Giga; T = Tera; P = Peta.

vsz64 — Displays the virtual storage used above the 2-gigabyte bar. In the display, each value is followed by a multiplier indicating the units represented: (space) = no multiplier; K = Kilo; M = Mega; G = Giga; T = Tera; P = Peta.

Example of testing the ps utility: The following example highlights our experiences testing the **ps** utility.

1. We issued the following command to create a 64-bit compiled module for a private program called brlmfcntl.c using the new xlc compile invocation utility available in z/OS V1R6. (We used the default configuration file at /usr/lpp/cbclib/xlc/etc/xlc.cfg and the utility command at /usr/lpp/cbclib/xlc/bin.)

   ```
   /usr/lpp/cbclib/xlc/bin/xlc_64 -o brlmfcntl.out64 brlmfcntl.c
   ```

   This creates an executable named brlmfcntl.out64.

---

2. Our first attempt to run the new program failed because we had used the default MEMLIMIT value, which is zero. The following is an example of how we invoked the program and the resulting error message:

   ```
   brlmfcntl.out64 brlmf1 120
   ```

   Result:

   ```
   1 + Done(137) brlmfcntl.out64 /
    17695742      Killed  ./brlmfcntl.out64
   ```

---

3. To resolve the problem in step 2, we issued the following MVS operator command to set the MEMLIMIT for the process under which we were running the program. In this case, that process was the shell session. We set the MEMLIMIT to 10M.

   ```
   SETOMVS PID=68026746,MEMLIMIT=10M
   ```

   To verify this change, we issued the following operator command and checked the MEMLIMIT value:

   ```
   DISPLAY OMVS,L,PID=068026746
   ```

Note: When setting the MEMLIMIT on your systems, carefully consider factors such as a system-wide setting, override capability, and process-specific settings.

---

4. Our brlmfcntl program takes as the second parameter the number of seconds to remain active. We forked three programs to hold for two minutes (120 seconds) each, as follows:

```
brlmfcntl.out64 brlmf1 120 &
brlmfcntl.out64 brlmf2 120 &
brlmfcntl.out64 brlmf3 120 &
```

---

5. We issued the ps command to display the vsz64 and vszlmt64 fields, as follows:

```
ps -e -f -o jobname,pid,ppid,vsz,vsz64,vszlmt64,args
```

Result:

```
JOBNAME        PID       PPID     VSZ       VSZ64   VSZLMT64 COMMAND
⋮
ALEASE19   68026754     917883    4804    8388608        10M brlmfcntl.out64 brlmf1 120
ALEASE11     917891     917883    4804    8388608        10M brlmfcntl.out64 brlmf2 120
ALEASE12     917892     917883    4804    8388608        10M brlmfcntl.out64 brlmf3 120
⋮
```

The above display indicates that the programs are using 8M of storage above the bar (VSZ64) and that the limit is 10M (VSZLMT64).

The ipcs utility: The **ipcs** utility writes information to the standard output stream about active inter-process communication facilities. The PGSZ and SEGSZ fields are added to the list of fields under the –x option. The SEGSZPG field is now displayed with the –x option. These fields are defined as follows:

SEGSZPG
: The size, in pages, of the associated shared memory segment.

PGSZ
: The page size of the associated shared memory segment.

SEGSZ
: The size, in bytes, of the associated shared memory segment.

Note that the new PGSZ field is not properly defined in the z/OS V1R6 documentation or the man pages for the **ipcs** command. See documentation APAR OA08772 for more information.

Examples of testing the ipcs utility: The following examples highlight our experiences testing the **ipcs** utility.

We ran the command using the three options (–a, –b, and –x) that display the SEGSZPG, PGSZ, and SEGSZ fields on a system where we had some shared memory activity. The following examples show excerpts of the displays.

Example: `ipcs -a`

Result:

```
⋮
Shared Memory:
T         ID     KEY       MODE       OWNER    GROUP  CREATOR    CGROUP   NATTCH   SEGSZPG PGSZ     SEGSZ     ATIME    DTIME    CTIME        CPID      LPID
m      40004 0x023625dc --rw-rw----  LORAIN0    IMWEB  LORAIN0    IMWEB         1     1      4K       1793      20:33:41 00:00:00 20:33:41     327706    327706
```

```
m      40005 0x033625dc --rw-rw----  LORAIN0    IMWEB  LORAIN0    IMWEB       1     10240  4K       41943040  20:33:41 00:00:00 20:33:41     327706    327706
m      40006 0x043625dc --rw-rw----  LORAIN0    IMWEB  LORAIN0    IMWEB       1     1      4K       99        20:33:41 00:00:00 20:33:41     327706    327706
⋮
```

Example: `ipcs -b`

Result:

```
⋮
Shared Memory:
T          ID      KEY       MODE       OWNER      GROUP   SEGSZPG PGSZ     SEGSZ
m       40004 0x023625dc --rw-rw----  LORAIN0    IMWEB    1       4K       1793
m       40005 0x033625dc --rw-rw----  LORAIN0    IMWEB    10240   4K       41943040
m       40006 0x043625dc --rw-rw----  LORAIN0    IMWEB    1       4K       99
⋮
```

Example: `ipcs -x`

Result:

```
⋮
Shared Memory:
T     KEY      OWNER    GROUP   SEGSZPG PGSZ     SEGSZ     ATPID     ATADDR                INFO
m 0x023625dc  LORAIN0   IMWEB    1       4K       1793      327706    0x00000000273f6000
m 0x033625dc  LORAIN0   IMWEB    10240   4K       41943040  327706    0x0000000027500000     M
m 0x043625dc  LORAIN0   IMWEB    1       4K       99        327706    0x00000000273fd000
⋮
```

The ar utility: You can use the **ar** utility to store multiple versions of the same object file within one archive library. This is useful if you are providing an archive library which may be used to resolve references from code compiled with various compiler options. These options cause differences in the object files which must be matched with the archive library member attributes. Attributes for **ar** are AMODE, XPLINK, and IPA. The **ar** utility stores the attribute information for each entry in the symbol table. The linkage editor uses the attribute information to resolve external references with the appropriate archive library member. Because the names of archive library members consist of only the final component of the path name, the member names must be unique for the different object file versions. It's a good idea to establish a naming convention for the object files and to implement build procedures to generate the correct names.

To display the attributes of the symbols within an object file or an archive library of object files, use the **nm** command, which displays the symbol table of object, library, or executable files.

Examples of testing the ar utility: The following example highlights our experiences testing the **ar** utility.

We performed the following steps to create an archive library, add members to it, delete members from it, and display its contents:

1. Using the **xlc** utility, we compiled a source file (displayfs.c) using various compiler options—31-bit, 31-bit with XPLINK, and 64-bit (which also forces XPLINK)—as follows:

   ```
   /usr/lpp/cbclib/xlc/bin/xlc    -o displayfs.out31 displayfs.c
   /usr/lpp/cbclib/xlc/bin/xlc_x  -o displayfs.out31x displayfs.c
   /usr/lpp/cbclib/xlc/bin/xlc_64 -o displayfs.out64 displayfs.c
   ```

   ---

2. We created an archive library called libdisplayfs.a containing the three members:

```
ar  -ruv libdisplayfs.a displayfs.out31 displayfs.out31x displayfs.out64
```

---

3. The following command displays the contents of the archive library:

   ```
   ar  -tv libdisplayfs.a
   ```

   Result: The archive library contains three members:

   ```
   rwxrwxrwx 0/0   77824 Jul 28 13:17 2004 displayfs.out31
   rwxr-xr-x 0/0   81920 Jul 28 13:19 2004 displayfs.out31x
   rwxr-xr-x 0/0   61440 Jul 28 13:17 2004 displayfs.out64
   ```

---

4. We created another 64-bit compiled object (as in step 1) named displayfs.out64b and then added it to the archive library using the following command:

   ```
   ar  -rcv libdisplayfs.a displayfs.out64b
   ```

   Result:

   ```
   a - displayfs.out64b
   ```

---

5. Displayed the contents once again:

   ```
   ar  -tv libdisplayfs.a
   ```

   Result: The archive library now contains four members:

   ```
   rwxrwxrwx 0/0   77824 Jul 28 13:17 2004 displayfs.out31
   rwxr-xr-x 0/0   81920 Jul 28 13:19 2004 displayfs.out31x
   rwxr-xr-x 0/0   61440 Jul 28 13:17 2004 displayfs.out64
   rwxr-xr-x 0/0   61440 Jul 28 13:32 2004 displayfs.out64b
   ```

---

6. We then deleted the last member that we added (displayfs.out64b):

   ```
   ar  -dsv libdisplayfs.a displayfs.out64b
   ```

   Result:

   ```
   d - displayfs.out64b
   ```

---

7. The following command attempts to display the member we just deleted:

   ```
   ar  -tv  libdisplayfs.a displayfs.out64b
   ```

   Result:

   ```
   ar: displayfs.out64b not found
   ```

---

8. Displayed the contents one more time:

   ```
   ar  -tv libdisplayfs.a
   ```

   Result:

   ```
   rwxrwxrwx 0/0   77824 Jul 28 13:17 2004 displayfs.out31
   rwxr-xr-x 0/0   81920 Jul 28 13:19 2004 displayfs.out31x
   rwxr-xr-x 0/0   61440 Jul 28 13:17 2004 displayfs.out64
   ```

The cp and mv utilities: As of z/OS V1R6, the **cp** and **mv** utilities can now copy or move 64-bit executables between MVS and the z/OS UNIX file systems and

correctly rebind executables, just as these utilities do for 24- and 31-bit applications. There are no external changes to the way you use these commands.

The ulimit utility: The **ulimit** utility sets or displays resource limits on processes created by the user. There are two new options available, as follows:

–A Set or display the maximum address space size for the process, in units of 1024 bytes. If the limit is exceeded, storage allocation requests and automatic stack growth will fail. An attempt to set the address space size limit lower than the size that is already in use will fail.

–M Set or display the amount of storage above the 2-gigabyte bar (MEMLIMIT), in one megabyte increments, that a process is allowed to have allocated and unhidden.

You can specify unlimited as the new limit. Using these options without specifying a limit value will simply display the current setting. These new limits also appear in the display using the `–a` option.

Examples of testing the ulimit utility: The following examples highlight our experiences testing the **nm** utility.

Example: `ulimit -a`

Result:

```
core file         15842b
cpu time          unlimited
data size         unlimited
file size         unlimited
stack size        unlimited
file descriptors  65535
address space     72808k
memory above bar  10m
```

Example: `ulimit -M`

Result:

```
10
```

Example: `ulimit -A`

Result:

```
72808
```

The limit and unlimit (tcsh) utilities: The **limit** utility limits the consumption of resources by the current process and each process it creates so that, individually, those processes cannot exceed a maximum-use value for a specified resource. If no maximum-use value is specified on the command, then the current limit is displayed. If no resource is specified, then the limitations for all resources are displayed.

The tcsh **limit** command includes two new resources, memlimit and addressspace, as follows:

addressspace The maximum address space size for the process, measured in kilobytes. If a process exceeds the limit, functions such as malloc()

and mmap() will fail. Also, automatic stack growth will fail. An attempt to set the address space size limit lower than the size that is already in use will fail.

memlimit — The amount of storage, in megabytes, above the 2-gigabyte bar that a process is allowed to have allocated and unhidden at any given time.

The **unlimit** utility removes the limitation on the specified resource. If no resource is specified on the command, then all resource limitations are removed.

Examples of testing the limit and unlimit utilities: The following examples highlight our experiences testing the **limit** and **unlimit** utilities.

Example: `limit`

Result:

```
cputime         unlimited
filesize        unlimited
datasize        unlimited
stacksize       unlimited
coredumpsize    7921 kbytes
descriptors     65535
addressspace    72808 kbytes
memlimit        10 megabytes
```

Example:

```
limit memlimit 8
limit memlimit
```

Result:

```
memlimit        8 megabytes
```

Example:

```
limit memlimit 8m
limit memlimit
```

Result:

```
memlimit        8 megabytes
```

Using distributed BRLM

With V1R6, byte range lock manager (BRLM) has changed to support distributed BRLM. Instead of a single, central BRLM, distributed BRLM means that each system in the sysplex runs a seperate BRLM, which is responsible for locking files in the file systems owned and mounted on that system. This means that a file system can be moved while byte range locks are held for files in the file system. When a file system changes owners, the corresponding locking history changes BRLM servers at the same time. (Note that this is not the case when a system failure occurs.)

For this reason, distributed BRLM is now the only supported method when all systems are at the V1R6 level and distributed BRLM is automatically activated on every system for an all z/OS V1R6 sysplex. Each system runs a BRLM and is responsible for handling lock requests for files whose filesystems are mounted and owned locally on that system.

If you are already running with a z/OS UNIX couple data set (CDS) indicating that distributed BRLM is enabled (DISTBRLM set to 1), there is no change required to activate distributed BRLM for V1R6. Likewise, if your sysplex only has systems at the V1R6 level, there is no change required, because distributed BRLM is the default. V1R6 systems ignore the z/OS UNIX CDS DISTBRLM setting.

However, if you migrate to V1R6 by running mixed levels in a sysplex, you should enable distributed BRLM before IPLing the V1R6 system because a V1R6 system may attempt to activate distributed BRLM when the central BRLM server leaves the sysplex, regardless of the z/OS UNIX CDS setting. The inconsistency between distributed BRLM being active and central BRLM being defined in the z/OS UNIX CDS can cause an EC6-BadOmvsCds abend on downlevel systems. This is a notification-only abend indicating that the CDS should be updated. z/OS UNIX will still operate normally, and distributed BRLM will be active in the sysplex. See *z/OS Migration* for more information.

We used the following display command on each system to display whether distributed BRLM is enabled and active:

```
F BPXOINIT,FILESYS=DISPLAY,GLOBAL
```

We got the following output from the display command:

```
BPXM027I COMMAND ACCEPTED.
BPXF040I MODIFY BPXOINIT,FILESYS PROCESSING IS COMPLETE.
BPXF041I 2004/08/10 14.10.09 MODIFY BPXOINIT,FILESYS=DISPLAY,GLOBAL
SYSTEM   LFS VERSION ---STATUS---------------------- RECOMMENDED ACTION
Z0         1.  6.  0 VERIFIED                       NONE
JA0        1.  6.  0 VERIFIED                       NONE
TPN        1.  6.  0 VERIFIED                       NONE
Z1         1.  6.  0 VERIFIED                       NONE
J90        1.  6.  0 VERIFIED                       NONE
JF0        1.  6.  0 VERIFIED                       NONE
JB0        1.  6.  0 VERIFIED                       NONE
JC0        1.  6.  0 VERIFIED                       NONE
JE0        1.  6.  0 VERIFIED                       NONE
Z2         1.  6.  0 VERIFIED                       NONE
Z3         1.  6.  0 VERIFIED                       NONE
J80        1.  6.  0 VERIFIED                       NONE
JG0        1.  6.  0 VERIFIED                       NONE
JH0        1.  6.  0 VERIFIED                       NONE
CDS VERSION=  2        MIN LFS VERSION=  1.  6.  0
BRLM SERVER=N/A        DEVICE NUMBER OF LAST MOUNT=    9266
MAXIMUM MOUNT ENTRIES=     800   MOUNT ENTRIES IN USE=     699
MAXIMUM AMTRULES=           51   AMTRULES IN USE=            9
DISTBRLM ENABLED=YES             DISTBRLM ACTIVE=YES
```

We're planning to use the enhancement allowing a filesystem to be moved even when it contains locked files. Prior to z/OS V1R6, you would receive an enomove return code error results if you issued filesystem move commands while a file was open and was byte range locked. In z/OS V1R6, this error code now only occurs if you have a pre-z/OS R6 system in the sysplex. In a pre-z/OS V1R6 system, an enomove return code prevents the filesystem move. In order to move filesystems that contain locked files, all systems in the sysplex must be at the z/OS V1R6 level.

Note that the system does not necessarily report an enomove return code.

Restriction: With distributed BRLM, certain cross-system deadlock scenarios may not be detected. Locking applications must ensure that they do not cause deadlocks. See *z/OS UNIX System Services Planning*.

Using ISHELL enhancements

In z/OS R6, we tested the following enhancements to the ISPF shell (ISHELL) commands. For additional ISHELL information, use the help panels that come with the product.

Wild card support for the command filter: UNIX System Services (USS) now supports the command **filter** on the directory list. If you enter the ISHELL command without any argument, a panel will be displayed to enter the new filter enhancements. You can use any characters with the wild card character (*) in the filter, and the wild card character * can match any number of characters or no characters. For example, the command filter *.c will show only files that end with .c. Filter *a* will show only file names that contain an a. The filter is case sensitive. If there is not a match for the filter then the entire directory list is given and the filter is disabled. The following example shows a series filters issued, and the output displayed:

```
filter *.c
EUID=0  *.c    /u/lates/
   Type  Filename
 _ File  hello.c
 _ File  rlimit.c
 _ File  wstatvfs.c

filter *a*
EUID=0  *a*    /u/lates/
  Type  Filename
_ File  a.out
_ Dir   aaa
_ File  chkosname.jar
_ File  copymap

filter w*
 EUID=0  w*    /u/lates/
   Type  Filename
 _ File  wstatvfs
 _ File  wstatvfs.c
```

Command line position panel option: To test the new command line position enhancements from an ISHELL panel we selected OPTIONS/ADVANCED. This gives us 3 command line selections; top, botttom and inherit, as follows:

```
                      Advanced Options
 Select options

 _  Bypass delete confirmations
 _  Bypass exit confirmation
 _  No auto-skip on action panels
 _  Always start initial panel with current directory

 Command line position:
 _  1.  Top
    2.  Bottom
    3.  Inherit
```

We tested all three command line positions successfully - they each positioned the command line as expected.

Options panel for displaying Permissions: To test the new permissions display from the ISHELL panel, we selected the OPTIONS/DIRECTORY list. This panel lists selections for displaying the permissions listed:

```
                          Directory List Options

Selected options and fields to be displayed with /

_  File type   ( 4 columns)
_  Permissions ( 4 columns, octal)
_  Permissions (10 columns, rwx)
_  Change time (16 columns)
_  Owner       ( 9 columns)
_  File size   (10 columns)

_  View/change sort options...
_  View/change file name highlighting...
/   Verbose directory list panel
_  Null Enter refreshes list
_  Stop processing multiple selections after a message
```

We selected ″Permissions (4 columns, octal)″ and displayed directory /pet6 as follows:

```
EUID=0   /pet6/
  Type  Perm  Filename
_ Dir    750  .
_ Dir    755  ..
_ File   644  a
_ File   644  A
_ File   644  b
_ File   644  B
_ File   644  c
_ File   644  C
_ File   644  d
_ File   644  D
_ File   644  e
_ File   644  E
_ File   644  f
```

Next we selected ″Permissions (10 columns, rwx)″ and displayed directory /pet6 as follows:

```
EUID=0   /pet6/
   Type  Permission  Filename
 _ Dir    rwxr-x---  .
 _ Dir    rwxr-xr-x  ..
 _ File   rw-r--r--  a
 _ File   rw-r--r--  A
 _ File   rw-r--r--  b
 _ File   rw-r--r--  B
 _ File   rw-r--r--  c
 _ File   rw-r--r--  C
 _ File   rw-r--r--  d
 _ File   rw-r--r--  D
 _ File   rw-r--r--  e
 _ File   rw-r--r--  E
 _ File   rw-r--r--  f
```

We then selected **both** permission selections to get the following display:

```
EUID=0   /pet6/
  Type  Perm  Permission  Filename
_ Dir    750   rwxr-x---  .
_ Dir    755   rwxr-xr-x  ..
_ File   644   rw-r--r--  a
_ File   644   rw-r--r--  A
_ File   644   rw-r--r--  b
_ File   644   rw-r--r--  B
_ File   644   rw-r--r--  c
_ File   644   rw-r--r--  C
```

```
_ File   644   rw-r--r--  d
_ File   644   rw-r--r--  D
_ File   644   rw-r--r--  e
_ File   644   rw-r--r--  E
_ File   644   rw-r--r--  f
```

Option for preserving extended attributes on copy: To test this function, we first updated file 'a' in /pet6 to have extended attributes using the following command:

```
extattr +aps a
```

We then used the ISHELL panel to do a copy (**c**) of file 'a' to file 'aaaa':

```
EUID=0   /pet6/
  Type  Perm  Owner      ------Size  Filename
_ Dir    750  LORAIN0         1952  .
_ Dir    777  LORAIN0        24576  ..
c File   644  LORAIN0            0  a
_ File   644  LORAIN0            0  A
_ File   644  LORAIN0            0  b
_ File   644  LORAIN0            0  B
```

We got the following panel, where we selected **1** to copy our file into another file:

```
                        Copy from a File

Copying from file:
/pet6/a

Destination for copy:
1   1.  File...
    2.  Data set...

Select additional options for data set copy:
_  Binary copy
_  Conversion...
```

We were prompted to enter the destination file name (aaa), as follows:

```
                     Enter the Pathname

Change this to the pathname of the target file:
                                                          More:     +
/pet6/aaaa
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
```

We were also prompted to enter the permissions for the destination file:

```
            Enter File Permissions

Permissions  . . 644  (3 digits, each 0-7)
```

Now we get to the extended attributes panel, where we selected to copy the extended attributes to the destination file:

```
                         Extended Atrributes

The selected file contains extended attributes.  Select the option below
to copy all of the extended attributes to the new file.  Note that
authority may be needed to set some extended attributes.

s  Copy extended attributes
```

When this is done, we get the following panel:

```
EUID=0   /pet6/
  Type  Perm  Owner      ------Size  Filename
_ Dir    750  LORAIN0          1952  .
_ Dir    777  LORAIN0         24576  ..
_ File   644  LORAIN0             0  a
_ File   644  LORAIN0             0  A
_ File   644  LORAIN0             0  aaaa
_ File   644  LORAIN0             0  b
```

Finally, we listed the file attributes using the directory panel and confirmed the extended attributes for the new file were copied from the source file as follows:

```
           Display File Attributes

Pathname : /pet6/aaaa
                                        More:   -
Major device  . . . . : 0
Minor device  . . . . : 0
File format . . . . . : NA
Shared AS . . . . . . : 1
APF authorized  . . . : 1
Program controlled  . : 1
Shared library  . . . : 0
Char Set ID/Text flag : 00000 OFF
Directory default ACL : 0
File default ACL  . . : 0
Seclabel  . . . . . :
```

Using the hierarchical file system (HFS)

We provided extensive coverage of our strategy for managing the z/OS UNIX hierarchical file system (HFS), including shared HFS, in our December 2001 edition. Refer to that edition for more information.

Automount enhancement for HFS to zSeries file system (zFS) migration

We tested a new automount enhancement that eases the migration from HFS to zFS file systems. Prior to the new function, you could not use a generic automount policy to automount both HFS and zFS file systems - all the file systems had to be the same type for a given automount managed mountpoint. The enhanced HFS to zFS automount migration function allows a single automount policy to mount both HFS and zFS file systems. This will help if you want to migrate your file systems over time rather than all at once, and so have a mixture of HFS and zFS file systems in your installation.

It works like this: the automount function has changed so that when you specify either HFS or ZFS as the file system type in an automount policy, the system re-checks the data set at mount time to determine what type of data set it really is, and then directs the mount to the appropriate file system type. However, to use this function, the naming conventions of the file systems for both HFS and zFS must be the same.

The example below shows a zFS policy that we implemented to mount both HFS and zFS file systems. This policy will mount both pre-existing HFS or zFS type filesystems, but only allocates new filesystems as zFS file systems. This is the recommended policy for easing the migration to zFS file systems.

```
name *
type ZFS
filesystem OMVSSPN.<uc_name>.FS
```

```
lowercase no
allocuser  space(3,2)   cyl    storclas(SMSOE)
mode rdwr
duration 30
delay 10
```

The next automount policy example will mount both pre-existing HFS or zFS file types as well, but will only allocate new file systems as HFS file systems:

```
name *
type HFS
filesystem OMVSSPN.<uc_name>.FS
lowercase no
mode rdwr
allocuser space(3,1) cyl storclas(SMSOE)
duration 30
delay 10
```

Using the zSeries file system (zFS)

We provided extensive coverage of our strategy for setting up and managing a z/OS DFS zSeries file system (zFS) in our December 2003 edition. Refer to that edition for more information.

zFS enhancements in z/OS V1R6

The following topics describe some of the new zFS functions in z/OS V1R6 which we implemented and tested.

- "zFS parmlib search"
- "zFS performance monitoring with zfsadm (query and reset counters)" on page 120
- "HANGBREAK, zFS modify console command" on page 122

zFS parmlib search

zFS implemented a new logical parmlib search capability. We tested the following options:

Using IOEPRM00: If an IOEZPRM DD statement for specifying zFS configuration parameters is not in the started proc, the zFS will look in SYS1.PARMLIB for the existence of an IOEPRM00 member. If that member is not found, then zFS uses default settings. We created member IOEPRM00 and populated it with the settings that we use in our sysplex:

```
user_cache_size=256m
debug_setting_dsn=sys1.&SYSNAME..zfs.debug(file1)
trace_dsn=sys1.&SYSNAME..zfs.trace
trace_table_size=128m
```

Specifying IOEPRMxx: Another option is to specify one or more IOEPRMxx members of parmlib to use. The members are identified in the zFS FILESYSTYPE statement of the BPXPRMxx parmlib member. The following example shows how to specify that we want to use members IOEPRM01 and IOEPRM02 for our zFS configuration settings.

```
FILEYSTYPE TYPE(ZFS) ENTRYPOINT(IOEFSCM) ASNAME(ZFS,'SUB=MSTR') PARM('PRM=(01,02)')
```

Using the SYSCLONE symbolic: Another option allows us to have a unique IOEPRMxx for each image by using the SYSCLONE symbolic. The following example illustrates how parmlib members would be selected if we were to start zFS on system Z0. Members IOEPRM97, IOEPRM98, IOEPRM99, and IOEPRMZ0

would be used. If a parmlib member is not found, the search for the configuration option will continue with the next parmlib member.

The maximum number of suffixes for IOEPRMxx that can be specified on the FILESYSTYPE statement is 32.

zFS performance monitoring with zfsadm (query and reset counters)

The **zfsadm query** command displays and resets zFS internal performance statistics counters and timers.

Format:

```
zfsadm query [-locking] [-reset] [-storage] [-usercache] [-iocounts]
               [-iobyaggregate] [-iobydasd] [-level] [-help]
```

Options:

-locking Specifies that the locking statistics report should be displayed.

-reset Specifies the report counters should be reset to zero. Should be specified with a report type.

-storage Specifies that the storage report should be displayed.

-usercache Specifies that the user cache report should be displayed.

-iocounts Specifies that the I/O count report should be displayed.

-iobyaggregate
Specifies that the I/O count by aggregate report should be displayed.

-iobydasd Specifies that the I/O count by Direct Access Storage Device (DASD) report should be displayed.

-level Prints the level of the zfsadm command. This is useful when you are diagnosing a problem. All other valid options specified with this option are ignored.

-help Prints the online help for this command. All other valid options specified with this option are ignored.

Note that the **-reset** option will reset the counters AFTER, not before, the display that is displayed when the option is used. For example, **zfsadm query -locking -reset** would display the locking statistics report, then reset the counters. Subsequent locking display will show statistics from counter reset.

Example: `zfsadm query -locking -reset`

Result:

```
Locking Statistics


Untimed sleeps:      13575   Timed Sleeps:          0   Wakeups:      13574

Total waits for locks:              22319606
Average lock wait time:             1.906 (msecs)

Total monitored sleeps:             13522
Average monitored sleep time:       5.692 (msecs)

                Top 15 Most Highly Contended Locks
```

```
  Thread     Async       Spin
   Wait      Disp.       Resol.    Pct.      Description
----------  -------   ----------  -----     --------------
 23683110       0      983063    99.583%    Log system map lock
        3   37549           6     0.151%    Volser I/O queue lock
    11852       0       20564     0.130%    Async global device lock
    10321       0         192     0.42%     Vnode-cache access lock
     2113       0        3705     0.23%     Transaction-cache complete list lock

     1134    1116        3425     0.22%     Transaction-cache main lock
     2446       0         187     0.10%     Anode bitmap allocation handle lock

     1405       0         269     0.6%      Anode fileset quota lock
     1437       0           4     0.5%      Async IO device lock
      202     425         531     0.4%      User file cache main segment lock
      829       0          81     0.3%      Anode fileset handle lock
      609       0          29     0.2%      Metadata-cache buffer lock
      352       0         169     0.2%      Anode file zero lock
      420       0          39     0.1%      Anode file notify lock
      280       0          21     0.1%      Transaction-cache active list lock
Total lock contention of all kinds:   24769331

                Top 5 Most Common Thread Sleeps

   Thread Wait             Pct.             Description
   -----------             -----            -----------
      13521              99.992%           Transaction allocation wait

          1                 0.7%           OSI cache item cleanup wait

          0                 0.0%           Directory Cache Buffer Wait

          0                 0.0%           User file cache Page Wait

          0                 0.0%           User file cache File Wait
```

Example: `zfsadm query -locking`

Result:

```
Locking Statistics
Untimed sleeps:         10   Timed Sleeps:          0   Wakeups:         10

Total waits for locks:              15898
Average lock wait time:             2.174 (msecs)

Total monitored sleeps:                10
Average monitored sleep time:       5.622 (msecs)

                Top 15 Most Highly Contended Locks

  Thread      Async    Spin
   Wait       Disp.    Resol.     Pct.    Description
----------   -------   -------   -----    --------------
     17009       0      679      99.718%  Log system map lock
         0      19        0       0.107%  Volser I/O queue lock
         7       0        7       0.78%   Async global device lock
         6       0        0       0.33%   Vnode-cache access lock
         6       0        0       0.33%   Anode bitmap allocation handle lock

         3       0        0       0.16%   Transaction-cache complete list lock
         1       0        0       0.5%    Vnode lock
         1       0        0       0.5%    Async IO device lock
         0       0        0       0.0%    Async IO set free list lock
         0       0        0       0.0%    Async IO event free list lock
         0       0        0       0.0%    LVM global lock
```

```
          0       0        0       0.0%    OSI Global process lock
          0       0        0       0.0%    Main volume syscall lock
          0       0        0       0.0%    User file cache all file lock
          0       0        0       0.0%    User file cache main segment lock

Total lock contention of all kinds:      17738

                Top 5 Most Common Thread Sleeps

Thread Wait     Pct.       Description
-----------   --------     -------------
       10      100.0%      Transaction allocation wait

        0        0.0%      OSI cache item cleanup wait

        0        0.0%      Directory Cache Buffer Wait

        0        0.0%      User file cache Page Wait

        0        0.0%     User file cache File Wait
```

Corresponding pfsctl Application Programming Interface (APIs) are also provided to retrieve these performance statistics.

- **Statistics iobyaggr Information** – The statistics iobyaggr information subcommand call contains information about the number of reads and writes and the number of bytes transferred for each aggregate.
- **Statistics iobydasd Information** – The statistics iobydasd information subcommand call contains information about the number of reads and writes and the number of bytes transferred for each DASD volume.
- **Statistics iocounts Information** – The statistics iocounts information subcommand call contains information about how often zFS performs I/O for various circumstances and how often it waits on that I/O.
- **Statistics Locking Information** – The statistics locking information subcommand call is a performance statistics operation that returns locking information.
- **Statistics Storage Information** – The statistics storage information subcommand call is a performance statistics operation that returns storage information.
- **Statistics User Cache Information** — The statistics user cache information subcommand call is a performance statistics operation that returns user cache information.

For more information on these APIs, see *z/OS Distributed File Service zSeries File System Administration*.

HANGBREAK, zFS modify console command

The following new modify console command for zFS attempts recovery for specific hang conditions: **modify procname,hangbreak**.

The **hangbreak** command causes zFS to post a failure to any requests in zFS that are waiting. This can allow the hang condition to be broken and resolved. This should only be used if you suspect that there is a hang involving zFS. The modify **zfs,query,threads** operator command is used to determine if one or more requestor threads remain in the same wait over several queries. If this command does not successfully break the hang, you need to stop or cancel zFS. If you suspect that zFS is in an infinite loop, you need to cancel zFS.

Example:

```
F ZFS,HANGBREAK
IOEZ00025I zFS kernel: MODIFY command - HANGBREAK completed successfully.
```

Chapter 8. Using the IBM HTTP Server

This chapter describes our experiences with IBM HTTP Server V5.3.

For the most current debugging and tuning hints and tips for all supported releases of IBM HTTP Server and IBM WebSphere Application Server, see the *WebSphere Troubleshooter* (www.ibm.com/software/webservers/appserv/troubleshooter.html).

Using gskkyman support for storing a PKCS #7 file with a chain of certificates

In z/OS V1R6, the System SSL component changed the processing of the gskkyman utility to generate and manage certificates. We tested the new enhancements, which are covered in *z/OS Crytographic Services System Secure Sockets Layer Programming*. In this section, we'll describe one small issue we encountered in using the new gskkyman support for storing certificates and their chains in a PKCS #7 file.

To test the gskkyman support for storing certificates and their chains in a PKCS #7 file, we created a certificate authority file containing the entire chain, and exported it to a PKCS #7 file with extension .p7b. So far, so good, but when we downloaded the file to our browser, we received the following message:

```
This is an invalid Security Certificate file
```

It turned out that we needed to update our HTTP server configuration file before we could download the file. We added the following AddType line:

```
AddType  .p7b      application/x-x509-ca-ra-cert-chain    ebcdic  1.0
```

After adding this directive, we restarted the HTTP Server and exported the Certificate Authority to a PKCS #7 file with an extension of .p7b. Then, we were able to download the .p7b file to our browser.

Chapter 9. Using LDAP Server

LDAP Server is a component of z/OS Security Server which uses the Lightweight Directory Access Protocol (LDAP) standard, an open industry protocol for accessing information in a directory.

This chapter contains the following sections: b

Overview of our LDAP configuration

We have a multiplatform LDAP configuration and we use both replication and referral. Figure 23 shows a high-level view of our LDAP multiplatform configuration:

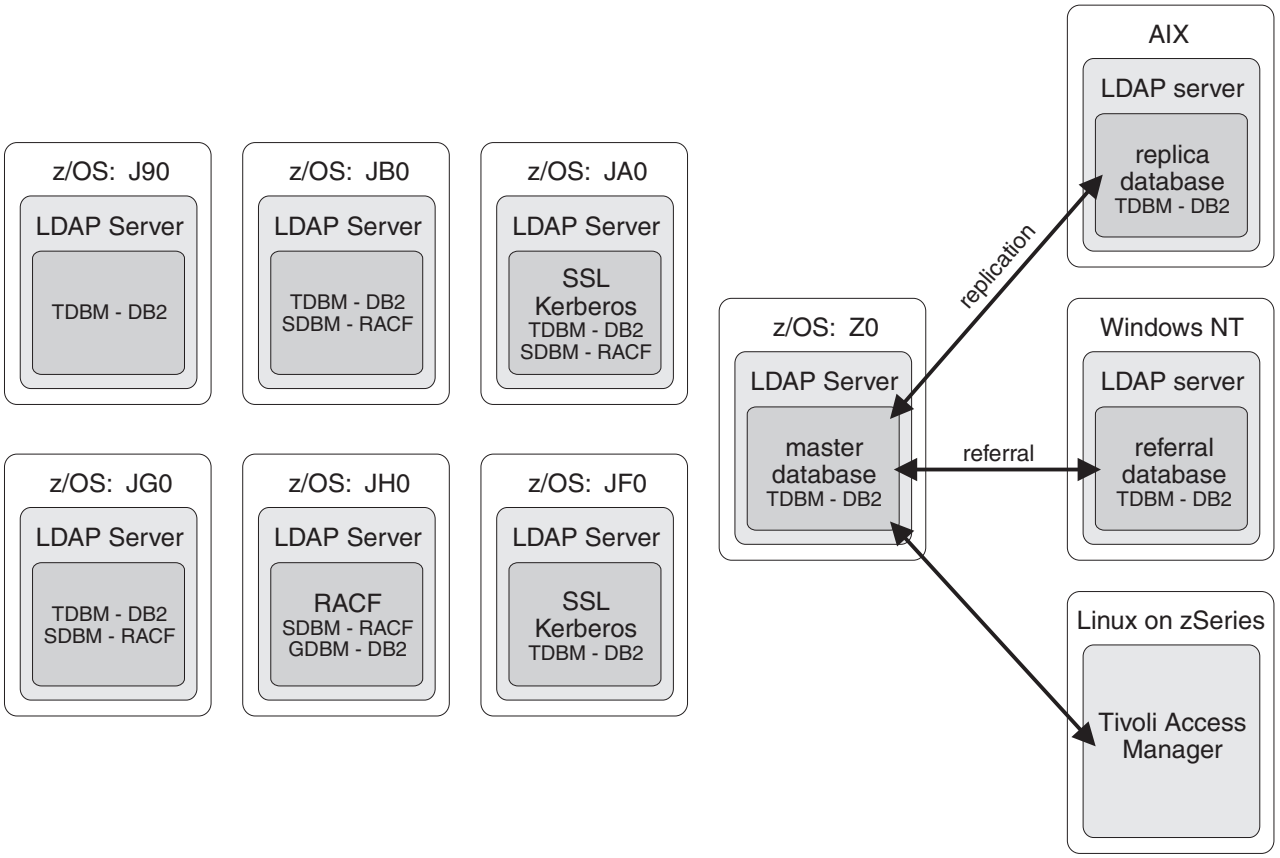


Figure 23. Overview of our LDAP configuration

Our LDAP environment includes the following features:

- **Master LDAP server:** Our master server on z/OS system Z0 has a DB2 backend database, TDBM.

- **Secure LDAP servers:** We have two LDAP servers in our sysplex (on systems JA0 and JF0) that are set up for SSL secure transactions and Kerberos authentication. The servers have a TDBM backend and listen on port 636.
- **LDAP server for our RACF backend:** We have an LDAP server with an SDBM backend that connects to the RACF directory for our sysplex.
- **LDAP referral from Windows NT to z/OS:** We manage an LDAP server on Windows NT using a Web-based management tool at:

  `http://NT_server_IP_address/ldap/index.html`

  This LDAP server has a TDBM backend and has a general referral in its configuration file that points to our master LDAP server on z/OS. This allows a user to issue an **ldapsearch** command from the Windows NT LDAP server for an entry that is not found in that directory but that might be found in the directory on the master server on z/OS. The **ldapsearch** command returns all matching entries from both directories.
- **LDAP referral from z/OS to Windows NT:** We maintain an LDAP server referral database on Windows NT using the Directory Management Tool for Windows. This server is set up to have referral processing enabled between the master LDAP server on z/OS and the LDAP server on Windows NT.
- **LDAP replica server on AIX:** We manage an LDAP replica server on AIX using a Web-based management tool at:

  `http://AIX_server_IP_address/ldap/index.html`

  This LDAP server has a TDBM backend and was configured through the Web-based management tool to be a replica of the master LDAP server on z/OS.
- **Tivoli Access Manager running on Linux on zSeries:** We set up Tivoli Access Manager on a SUSE Linux image running on zSeries to enable cross-platform testing between Linux and z/OS. Tivoli Access Manager uses the z/OS LDAP Server as a backend to store user ID information that is used to authorize access for users of Tivoli Access Manager. We perform our testing by running shell scripts on Linux that create a workload to stress the master LDAP server on z/OS.

Setting up the LDAP server for RACF change logging

During our z/OS V1R5 testing, we set up and configured support for change logging in LDAP Server. This new function is delivered in Security Server LDAP APAR OA03857 and applies to z/OS V1R3 and higher.

Change logging provides the following functions:

- Provides a log of changes made to user profiles in RACF, including password changes
- Allows a client to search the log of changes
- Allows retrieval of an enveloped version of a RACF password

Log entries are stored in a new type of backend called GDBM.

Change logging also requires that the SDBM backend be configured, that LDAP Program Callable (PC) support be enabled. Support for the exploitation of this new function by RACF is provided by RACF APAR OA03853 and SAF APAR OA03854. For details and a link to the updated documentation on the Web, see LDAP APAR OA03857.

This section describes our experiences setting up and configuring change logging in our environment.

Activating change notification in RACF

We did the following to enable RACF to provide notification of changes for logging in the LDAP change log:

1. Define the RACFEVNT class profile named NOTIFY.LDAP.USER:

   ```
   RDEFINE RACFEVNT NOTIFY.LDAP.USER
   ```

   A generic profile can also be used.

2. Activated the RACFEVNT class:

   ```
   SETROPTS CLASSACT(RACFEVNT)
   ```

For more information, see the documentation in RACF APAR OA03853.

Setting up the GDBM backend for the LDAP server

Note: You cannot use the LDAP configuration utility, ldapcnf, to configure the GDBM backend. For more information, see the documentation in LDAP APAR OA03857.

At a minimum, the GDBM backend section of the LDAP configuration file requires the following:

```
database GDBM GLDBGDBM [name]
dbuserid dbowner
servername string
```

Other options are also available to specify such things as the maximum age of a change log entry, the maximum number of entries that the change log can contain, and whether change logging is on or off (default is on).

Also, Program Callable (PC) support must be enabled in the global section of the configuration file, as follows:

```
listen ldap://:pc
```

We already had PC support enabled in our configuration file.

We did the following to set up the GDBM backend:

1. Loaded the change log schema into the LDAP server from the ChangeLog.ldif file:

   ```
   ldapmodify -h ip_addr -D "cn=LDAPxxxxx" -w pw -f /path/etc/ldap/ChangeLog.ldif
   ```

2. Added the following GDBM configuration options to the slapd.conf configuration file. (Although SDBM was already set up on this server, we have included those options here as well).

   ```
   #  GDBM-specific CONFIGURATION SETTINGS
   # ------------------------------------------------------------------------
   # ------------------------------------------------------------------------
   database gdbm GLDBGDBM
   servername USIBMT6PETDB2
   dbuserid GLDSRV
   ```

```
dsnaoini GLD.CNFOUT.JB0(DSNAOINI)
attroverflowsize 500
######################################################################
# sdbm database definitions
######################################################################
database sdbm GLDBSDBM
suffix "sysplex=UTCPLXJ8,o=IBM,c=US"
```

3. Started the LDAP server:

   START LDAP*xx*

   Result: We viewed the SLAPDOUT output to verify that the server startup was successful and that change logging was enabled. SLAPDOUT contained the following:

```
******************************** TOP OF DATA *********************************
GLD0022I z/OS Version 1 Release 4 Security Server LDAP Server
 Starting slapd.
GLD0010I Reading configuration file /etc/ldap/slapd.conf.
GLD3135I Grant/Deny ACL support is enabled below suffixes: "CN=CHANGELOG".
GLD0244I Change logging is enabled
         Logging started status (0 = off, 1 = on): 1
         Limit in seconds on age of change log entries (0 = no limit): 0
         Limit on the number of change log entries (0 = no limit): 0
         Current number of change log entries: 0
         First change number in use: 0
         Last change number in use: 0
GLD0163I Backend capability listing follows:
GLD0166I Backend type: sdbm, Backend ID: SDBM BACKEND
GLD0207I SDBM BACKEND manages the following suffixes:
GLD0208I Backend suffix: SYSPLEX=UTCPLXJ8
GLD0209I End of suffixes managed by SDBM BACKEND.
GLD0165I Capability: LDAP_Backend_ID     Value: SDBM BACKEND
GLD0165I Capability: LDAP_Backend_BldDateTime     Value: 2003-10-21-17.46.55.000
GLD0165I Capability: LDAP_Backend_APARLevel     Value: OA03857
GLD0165I Capability: LDAP_Backend_Release     Value: R 4.0
GLD0165I Capability: LDAP_Backend_Version     Value: V 1.0
GLD0165I Capability: LDAP_Backend_Dialect     Value: DIALECT 1.0
GLD0165I Capability: LDAP_Backend_BerDecoding     Value: STRING
GLD0165I Capability: LDAP_Backend_ExtGroupSearch     Value: YES
GLD0165I Capability: LDAP_Backend_krbIdentityMap     Value: YES
GLD0165I Capability: supportedControl     Value: 2.16.840.1.113730.3.4.2
GLD0165I Capability: supportedControl     Value: 1.3.18.0.2.10.2
GLD0167I End of capability listing for Backend type: sdbm, Backend ID: SDBM BACKEND
GLD0166I Backend type: gdbm, Backend ID: GDBM BACKEND
GLD0207I GDBM BACKEND manages the following suffixes:
GLD0208I Backend suffix: CN=CHANGELOG
GLD0209I End of suffixes managed by GDBM BACKEND.
GLD0165I Capability: LDAP_Backend_ID     Value: GDBM BACKEND
GLD0165I Capability: LDAP_Backend_BldDateTime     Value: 2003-10-21-17.47.40.000000
GLD0165I Capability: LDAP_Backend_APARLevel     Value: OA03857
GLD0165I Capability: LDAP_Backend_Release     Value: R 4.0
GLD0165I Capability: LDAP_Backend_Version     Value: V 1.0
GLD0165I Capability: LDAP_Backend_Dialect     Value: DIALECT 1.0
GLD0165I Capability: LDAP_Backend_BerDecoding     Value: BINARY
GLD0165I Capability: LDAP_Backend_ExtGroupSearch     Value: NO
GLD0165I Capability: LDAP_Backend_krbIdentityMap     Value: NO
GLD0165I Capability: supportedControl     Value: 2.16.840.1.113730.3.4.2
GLD0167I End of capability listing for Backend type: gdbm, Backend ID: GDBM BACKEND
GLD0164I Backend capability listing ended.
GLD0002I Configuration file successfully read.
GLD0189I Nonsecure communication is active for IP: INADDR_ANY, nonsecure port: 389
GLD0202I Program Call communication is active.
GLD0122I Slapd is ready for requests.
******************************** BOTTOM OF DATA ********************************
```

The LDAP server GDBM backend was successfully set up and enabled for change logging.

Testing the change logging function and the GDBM database

With change logging active, we made several changes to a RACF user ID and then tested functions to search the GDBM database, set the maximum number of change log entries, search the GDBM database anonymously, and delete change log entries.

Searching the GDBM database

Before adding any change log entries, we issued the following command to verify that the GDBM database could properly be searched:

```
ldapsearch -h ip_addr -D "racfid=XXXXX,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US"
     -w pw -b "cn=changelog" "objectclass=*"
```

Result: The search displayed the following output:

```
cn=changelog
objectclass=top
objectclass=container
cn=changelog
```

Since there were no change log entries in the database yet, the search returned only the base entry of the database, as expected.

Testing the maximum number of change log entries

We did the following to test the option to limit the maximum number of change log entries:

1. Added the changeLogMaxEntries option to the slapd.conf file and specified a value of 1000, as shown:

   ```
   #  GDBM-specific CONFIGURATION SETTINGS
   # ---------------------------------------------------------------------
   # ---------------------------------------------------------------------
   database gdbm GLDBGDBM
   servername USIBMT6PETDB2
   dbuserid GLDSRV
   dsnaoini GLD.CNFOUT.JB0(DSNAOINI)
   changeLogging on
   changeLogMaxEntries 1000
   changeLogMaxAge 86400
   attroverflowsize 500
   ```

   ---

2. Made eight changes to a RACF user ID, USER01, which creates eight change log entries (a total of nine, including the cn=changelog root entry) in the GDBM backend.

   ---

3. Searched the database to verify that the change log entries were present:

   ```
   ldapsearch -h ip_addr -D "racfid=XXXXX,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US"
        -w pw -b "cn=changelog" "objectclass=*"
   ```

   Result: The search displayed the following output:

   ```
   cn=changelog
   objectclass=top
   objectclass=container
   cn=changelog

   CHANGENUMBER=401,CN=CHANGELOG
   objectclass=CHANGELOGENTRY
   ```

```
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=401
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144917.638873Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=402,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=402
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144921.623237Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=403,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=403
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144925.306248Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=404,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=404
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144929.050827Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=405,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=405
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144933.085019Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=406,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
```

```
changenumber=406
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144938.965894Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=407,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=407
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144942.378976Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=408,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=408
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144945.559614Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US
```

---

4. Changed the value of the changeLogMaxEntries option in the slapd.conf file from 1000 to 5, as shown:

```
#  GDBM-specific CONFIGURATION SETTINGS
# ---------------------------------------------------------------------
# ---------------------------------------------------------------------
database gdbm GLDBGDBM
servername USIBMT6PETDB2
dbuserid GLDSRV
dsnaoini GLD.CNFOUT.JB0(DSNAOINI)
changeLogging on
changeLogMaxEntries 5
changeLogMaxAge 86400
attroverflowsize 500
```

---

5. Recycled the LDAP server.

---

6. Issued the LDAP search again:

```
ldapsearch -h ip_addr -D "racfid=XXXXX,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US"
     -w pw -b "cn=changelog" "objectclass=*"
```

Result: The search displayed the following output:

```
cn=changelog
objectclass=top
objectclass=container
cn=changelog

CHANGENUMBER=405,CN=CHANGELOG
```

```
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=405
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144933.085019Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=406,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=406
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144938.965894Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=407,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=407
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144942.378976Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=408,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=408
targetdn=RACFID=USER01,PROFILETYPE=USER,SYSPLEX=
UTCPLXJ8,O=IBM,C=US
changetime=20040127144945.559614Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETY
PE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US
```

Note that change log entries 401 through 404 have been deleted. There were now a total of five entries (including the cn=changelog root entry) in the database, as specified by the changeLogMaxEntries value.

The option to limit the maximum number of change log entries worked successfully.

Searching the GDBM database anonymously

We did the following to enable and test the ability to anonymously search the GDBM backend:

1. Performed an anonymous search to make sure this feature was not already enabled:

   ```
   ldapsearch -h ip_addr -b "cn=changelog" "objectclass=*"
   ```

Result: The search displayed no output and simply returned to the command line.

---

2. Performed an administrative search to make sure the database could properly be searched:

```
ldapsearch -h ip_addr -D "racfid=XXXXX,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US"
     -w pw -b "cn=changelog" "objectclass=*"
```

Result: The search displayed the following output:

```
cn=changelog
objectclass=top
objectclass=container
cn=changelog

CHANGENUMBER=501,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=501
targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changetime=20040203154225.041549Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=502,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=502
targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changetime=20040203154229.595015Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=503,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=503
targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changetime=20040203154237.589303Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=504,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=504
targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changetime=20040203154242.123712Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US
```

---

3. Created a file, cl.ldif, with the following contents to modify the ACL to allow anonymous searches:

```
1  dn: cn=changelog
2  changetype: modify
3  replace: aclentry
4  aclentry:cn=Anybody:normal:rsc:sensitive:rsc:critical:rsc:system:rsc
```

---

4. Loaded the new information from the cl.ldif file into the database:

   ```
   ldapmodify -h ip_addr -D "racfid=XXXXX,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US"
        -w pw -f cl.ldif
   ```

   Result: The changes were successfully loaded.

---

5. Performed an anonymous search again:

   ```
   ldapsearch -h ip_addr -b "cn=changelog" "objectclass=*"
   ```

   Result: This time, the search displayed the database contents:

```
cn=changelog
objectclass=top
objectclass=container
cn=changelog

CHANGENUMBER=501,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=501
targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changetime=20040203154225.041549Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=502,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=502
targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changetime=20040203154229.595015Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=503,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=503
targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changetime=20040203154237.589303Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US

CHANGENUMBER=504,CN=CHANGELOG
objectclass=CHANGELOGENTRY
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=504
targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changetime=20040203154242.123712Z
changetype=MODIFY
ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
=US
```

Anonymous searches of the database worked successfully.

Deleting change log entries

We did the following to test the ability to delete unwanted change log entries from the database:

1. Performed a search for a specific change log entry:

   ```
   ldapsearch -h ip_addr -b "changenumber=604, cn=changelog" "objectclass=*"
   ```

   Result: The search displayed the following:

   ```
   CHANGENUMBER=604,CN=CHANGELOG
   objectclass=CHANGELOGENTRY
   objectclass=IBM-CHANGELOG
   objectclass=TOP
   changenumber=604
   targetdn=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
   changetime=20040209151933.319305Z
   changetype=MODIFY
   ibm-changeinitiatorsname=RACFID=XXXXX,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C
   =US
   ```

2. Issued the following command to delete the specific change log entry:

   ```
   ldapdelete -h ip_addr -D "racfid=XXXXX,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US"
        -w pw "changenumber=604, cn=changelog"
   ```

   Result: The command completed successfully.

3. Searched again for the specific change log entry to verify that it was gone:

   ```
   ldapsearch -h ip_addr -b "changenumber=604, cn=changelog" "objectclass=*"
   ```

   Result: The search displayed the following:

   ```
   ldap_search: No such object
   ldap_search: matched: cn=changelog
   ldap_search: additional info: R004026 Entry changenumber=604, cn=changelog not
   found in the database. (tdbm_search.c|1.74.3.2|841)
   ```

The change log entry was successfully deleted from the database.

Using the z/OS LDAP client with the Windows 2000 Active Directory service

We had a request from our colleagues in z/OS LDAP development to test the z/OS LDAP client with the Microsoft Windows 2000 Active Directory service using a Kerberos authentication bind. This was to help validate a fix they were working on involving the use of Kerberos authentication to bind to IBM Directory Server. In order to do this, we first wanted to ensure that we could perform a search using a simple LDAP bind between the z/OS LDAP client and Active Directory.

We did not need to do anything special to set up or enable this process. We already had access to a Windows 2000 server that had Active Directory enabled and had data loaded. The hardest part was determining the format of the Active Directory distinguished name. Fortunately, we were able to get some advice from a colleague who has some experience with Active Directory. We were eventually able to successfully perform a search from z/OS against the Active Directory, as shown in the following example:

Example: We issued the following **ldapsearch** command (as a single line) from the z/OS UNIX shell command line against the Windows 2000 Active Directory:

```
ldapsearch -h win2k_ip_addr
   -D "CN=Sue Marcotte,CN=Users,DC=kerberos,DC=xxx,DC=yyy,DC=ibm,DC=com" -w password
   -b "CN=Sue Marcotte,CN=Users,DC=kerberos,DC=xxx,DC=yyy,DC=ibm,DC=com" "objectclass=*" name
```

Result: As expected, we received the following response:

```
CN=Sue Marcotte,CN=Users,DC=kerberos,DC=xxx,DC=yyy,DC=ibm,DC=com
name=Sue Marcotte
```

Using LDAP with Kerberos authentication

We have implemented a new LDAP workload that exploits binding with Kerberos authentication. (Our December 2002 edition describes how we enabled the z/OS LDAP Server for Kerberos authentication.) This workload has uncovered a couple of problems that we would like to note.

We used the following documentation to help us investigate and diagnose these problems:

- *z/OS Integrated Security Services LDAP Client Programming*, SC24-5924
- *z/OS Integrated Security Services LDAP Server Administration and Use*, SC24-5923
- *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926

Problems we experienced with our workload

The following are the problems that we experienced with our LDAP workload using Kerberos authentication:

Abend 0C6 in LDAP Server

We experienced an abend 0C6 in LDAP Server under the following conditions:

- LDAP Server successfully started but failed to configure a TDBM backend. (There is an SDBM backend.)
- An **ldapsearch** command to the TDBM backend is issued to the server using a Kerberos authentication bind.

This caused LDAP Server to generate a CEEDUMP and fail with an abend 0C6. Below is a portion of the traceback that helped to identify the error.

```
Traceback:
  DSA Addr  Program Unit  PU Addr   PU Offset  Entry         E Addr    E  Offset   Statement  Load Mod  Service  Status
  26EE74B8  CEEHDSP       26955430  +00003F42  CEEHDSP       26955430  +00003F42              CEEPLPKA  D1515    Call
  26EE69C8                27CF9D68  +2870A3DB  addAltDN(_Connection*,_strbuf*,_Backend*)
                                                           27CF9D68  +2870A3DB              GLDNM003           Exception
  26EE68A8                27CF9378  +000002D0  storeBindInfo(_Connection*,_Operation*)
                                                           27CF9378  +000002D0              GLDNM003           Call
  26EE6788                27CF8548  +00000612  krbBind(_Connection*,_Operation*,berval*)
                                                           27CF8548  +00000612              GLDNM003           Call
  26EE66A0                2686AB68  +0000015A  SaslBindGssapi::doBindPart1(_strbuf*)
                                                           2686AB68  +0000015A              GLDNM005           Call
  26EE6550                2680A430  +000010A2  do_bind(_Connection*,_Operation*)
                                                           2680A430  +000010A2              GLDNM005           Call
  26EE63D8                26838550  +00000F38  process_request
                                                           26838550  +00000F38              GLDNM005           Call
  26EE62A0                2683F908  +0000086E  caMReceiveCB(void*,void*,int,int)
                                                           2683F908  +0000086E              GLDNM005           Call
  26EE61A8                26849E00  +0000020E  caSslAsyncBerGetNextCB(sockbuf*)
                                                           26849E00  +0000020E              GLDNM005           Call
  26EE60E8                05F50398  +00000164  asyncBerGetNext
                                                           05F50398  +00000164              GLDNMC03           Call
  26EE6008                05F513E0  +00000086  asyncBerGetNextCB(sockbuf*)
                                                           05F513E0  +00000086              GLDNMC03           Call
  26EE5F30                05F50D48  +00000136  async_get_tag(sockbuf*,unsigned long*,void(*)(sockbuf*))
```

```
                                                  05F50D48  +00000136              GLDNMC03           Call
  26EE5E68                05F51308  +0000005E  async_get_tagCB(sockbuf*)
                                                  05F51308  +0000005E              GLDNMC03           Call
  26EE5D80                05F50A38  +0000026C  async_BerRead(sockbuf*,char*,long,long*,void(*)(sockbuf*))
                                                  05F50A38  +0000026C              GLDNMC03           Call
  26EE5CB8                05F51220  +00000072  async_BerRead_filbufCB(sockbuf*)
                                                  05F51220  +00000072              GLDNMC03           Call
  26EE5BF0                05F51118  +0000008A  async_ber_filbufCB(sockbuf*,int)
                                                  05F51118  +0000008A              GLDNMC03           Call
  26EE5AF0                2684A6A0  +00000174  caSslLowerReceiveCB(void*,void*,int,int)
                                                  2684A6A0  +00000174              GLDNM005           Call
  26EE59E8                26856468  +000003D8  caInetReceiveCB
                                                  26856468  +000003D8              GLDNM005           Call
  26EE5930                26AC8CD8  +0000001A  @@GETFN       26AC8C30  +000000C2              CEEEV003           Call
  26EE57F0                05FEE3F0  +000009A4  async_service_thread
                                                  05FEE3F0  +000009A4              GLDNM035           Call
  26EE5718                05FE34A8  +0000022E  osi_thread_first
                                                  05FE34A8  +0000022E              GLDNM035           Call
  7F599E78  CEEOPCMM      0000E4D0  +00000914  CEEOPCMM      0000E4D0  +00000914              CEEBINIT  D1515    Call

 Condition Information for Active Routines
   Condition Information for  (DSA address 26EE69C8)
     CIB Address: 26EE7DF8
     Current Condition:
       CEE0198S The termination of a thread was signaled due to an unhandled condition.
     Original Condition:
       CEE3206S The system detected a specification exception (System Completion Code=0C6).
     Location:
       Program Unit:  Entry: addAltDN(_Connection*,_strbuf*,_Backend*)
       Statement:     Offset: +2870A3DB
```

APAR OA07015 addresses this problem.

Abend 0C4 in gss_release_buffer in z/OS LDAP client

We experienced an intermittent problem in which the z/OS LDAP client, while binding to the z/OS LDAP server using Kerberos authentication, generated a CEEDUMP with an exception in gss_release_buffer and failed with an abend 0C4. The CEEDUMP file appeared in the HFS under the directory from which the client was running.

Below is a portion of the traceback that helped to identify the error.

```
Traceback:
  DSA Addr  Program Unit  PU Addr   PU Offset  Entry         E Addr    E  Offset   Statement  Load Mod  Service  Status
  29D63140                29E41900  +00000234  eim_snap_dump
                                                  29E41900  +00000234          77  *PATHNAM  HIT7708  Call
  29D63028                29E41C28  +000006A6  eim_exc_handler
                                                  29E41C28  +000006A6         294  *PATHNAM  HIT7708  Call
  29D62F70                29A51838  +0000001A  @@GETFN       29A51790  +000000C2              CEEEV003           Call
  29D5FE30  CEEHDSP       298EBA50  +000024D8  CEEHDSP       298EBA50  +000024D8              CEEPLPKA  D1515    Call
  29D5F3C0                078001C0  +0000016A  gss_release_buffer
                                                  078001C0  +0000016A              EUVFKDLL           Exception
  29D5F2E8                08F96798  +000000B6  ldap_gss_release_buffer(gss_buffer_desc_struct*)
                                                  08F96798  +000000B6              GLDNMC03           Call
  29D5F1A0                08EF0858  +00000550  ldap_krb5_authenticate(ldap*,berval*,_LDAPControl**,_LDAPCon
                                                  08EF0858  +00000550              GLDNMC01           Call
  29D5EE68                08EEFB50  +00000AA6  ldap_sasl_bind_krb5_s_direct
                                                  08EEFB50  +00000AA6              GLDNMC01           Call
  29D5EDA0                08EC6AB8  +00000118  ldap_sasl_bind_s_direct
                                                  08EC6AB8  +00000118              GLDNMC01           Call
  29D5ECA8                08EC58E0  +00000326  ldap_sasl_bind_s
                                                  08EC58E0  +00000326              GLDNMC01           Call
  29D5EBA0                29E03EF8  +000000E2  eimHandleInt::kerberosBind(ldap*,unsigned long*,eimErr*)
                                                  29E03EF8  +000000E2        2936  *PATHNAM  HIT7708  Call
  29D5E9F8                29E01498  +000005A0  eimHandleInt::connect2(eimLdapInfo*,EimConnectInfo*,eimClean
                                                  29E01498  +000005A0        2435  *PATHNAM  HIT7708  Call
  29D5E8A8                29E03248  +000002A8  eimHandleInt::setMaster2(char*,int,eimErr*)
                                                  29E03248  +000002A8        2072  *PATHNAM  HIT7708  Call
  29D5E7C8                29E03130  +0000008C  eimHandleInt::connectToMaster(EimConnectInfo*,eimErr*)
                                                  29E03130  +0000008C        1433  *PATHNAM  HIT7708  Call
  29D5E6F8                29E39430  +00000082  qsy_eimConnectToMaster(eimHandleInt*,EimConnectInfo*,eimErr*
                                                  29E39430  +00000082         714  *PATHNAM  HIT7708  Call
  29D5E530                29E281C8  +00000410  eimConnectToMaster
                                                  29E281C8  +00000410        3050  *PATHNAM  HIT7708  Call
  29D5E368                29809840  +00000D28  connectEIM    29809840  +00000D28         924  *PATHNAM  HIT7708  Call
```

```
  29D5E210               2980D1C0  +000002FC  main          2980D1C0  +000002FC         184  *PATHNAM  HIT7708  Call
  29D5E0F8               29C2A146  +000000B4  EDCZMINV      29C2A146  +000000B4              CEEEV003           Call
  29D5E030  CEEBBEXT     298BBAA0  +000001A6  CEEBBEXT      298BBAA0  +000001A6              CEEPLPKA  D1515    Call

Condition Information for Active Routines
  Condition Information for  (DSA address 29D5F3C0)
    CIB Address: 29D60770
    Current Condition:
      CEE3204S The system detected a protection exception (System Completion Code=0C4).
    Original Condition:
      CEE3204S The system detected a protection exception (System Completion Code=0C4).
    Location:
      Program Unit:  Entry: gss_release_buffer
      Statement:     Offset: +0000016A
    Machine State:
      ILC..... 0004    Interruption Code..... 0004
      PSW..... 078D1400 8780032E
      GPR0..... 29D99CB0  GPR1..... 29D5F458  GPR2..... 29D99CB0  GPR3..... 078001FA
      GPR4..... 0000E748  GPR5..... 00000000  GPR6..... 29D99D00  GPR7..... 00000000
      GPR8..... 29D5F3BC  GPR9..... 29D8E010  GPR10.... 29E6B130  GPR11.... 29D8A840
      GPR12.... 2983B7A8  GPR13.... 29D5F3C0  GPR14.... 8780032A  GPR15.... 0003032A
```

APAR OA07090 addresses this problem.

LDAP Server enhancements in z/OS V1R6

The following topics describe some of the new LDAP Server functions in z/OS V1R6 that we implemented and tested.

- "LDAP migration to z/OS V1R6"
- "Setting up a peer-to-peer replication network between an IBM Tivoli Directory Server 5.2 and a z/OS LDAP Server" on page 141
- "Using DB2 restart/recovery function" on page 147
- "Using alias support" on page 148
- "Using the enhanced LDAP configurgation utility (LDAPCNF)" on page 149
- "Using change logging with TDBM" on page 150

LDAP migration to z/OS V1R6

Accessing SYS1.SIEALNKE: All Integrated Security Server products are now placing their load modules in SYS1.SIEANLKE instead of maintaining their own load module data set. This is a new data set for the Integrated Security Server products. We used *z/OS Integrated Security Services LDAP Server Administration and Use* and *z/OS Migration* to migrate to this new level and use this new data set.

Prior to starting any LDAP servers, verify that the SYS1.SIEALNKE data set is in the LNKLST concatenation. If it is not in link list, then you must use STEPLIB to locate the data set. When SYS1.SIEALNKE is not in the LNKLST concatenation, the LDAP server will not start and the following error is seen in the JES log.

```
IEF403I  LDAPSRV - STARTED - TIME=10.17.38
CSV003I  REQUESTED MODULE GLDSLAPD NOT FOUND
CSV028I  ABEND806-04  JOBNAME=LDAPSRV  STEPNAME=LDAPSRV
IEA995I  SYMPTOM DUMP OUTPUT
SYSTEM COMPLETION CODE=806  REASON CODE=00000004
```

Using enhanced dynamic, nested, or expanded static group data: In order to use the enhanced support for static, dynamic, and nested groups of users, your DB_Version level must be 3.0 or higher. When we first brought up our z/OS V1R6 LDAP server, our DB_Version was below the required level. We received the following message from the LDAP server:

```
GLD3148I Dynamic, nested, or expanded static group data is present in the TDBM backend but ignored since the DB_VERSION
```

To resolve this problem, you must update the DB_Version level from SPUFI (SQL Processor Using File Input) with the following statement

```
UPDATE dbuserid.DIR_MISC SET DB_VERSION='3.0'
```

dbuserid is the z/OS user ID that will be the owner of the DB2 tables, a value assigned during initial LDAP backend setup. For example if you defined the *dbuserid* as LDAPSRV when you set up the backend, the SPUFI DB_Version update statement would be:

```
UPDATE LDAPSRV.DIR_MISC SET DB_VERSION='3.0'
```

Once you have updated the DB_Version level to 3.0 or higher, you can use the enhanced dynamic, nested, or static group data. See *z/OS Integrated Security Services LDAP Server Administration and Use* for more information.

Setting up a peer-to-peer replication network between an IBM Tivoli Directory Server 5.2 and a z/OS LDAP Server

The procedures documented here are intended to give an LDAP administrator a set of instructions on how to set up a peer-to-peer replication network between two IBM directory servers, IBM Tivoli Directory Server 5.2 and z/OS LDAP Server on z/OS V1R6.

The procedures assume that you have installed and can use the Web Administration Tool for the IBM Tivoli Directory Server. See the *IBM Tivoli Directory Server Version 5.2 Installation Guide* for information about installing the Web Administration Tool. Another assumption is that you have decided which suffix to replicate and that the entries under that suffix are loaded in both directories.

There are two configuration options presented here. For each option, the procedure starts by creating a master/slave replication network and then promoting that to a peer-to-peer replication network.

Configuration Option 1

This option shows you how to setup a master/slave replication network with IBM Tivoli Directory Server 5.2 as the MASTER and z/OS LDAP Server on z/OS V1R6 as the SLAVE.

PART 1 consists of the following steps:

Creating the Master Server: The servers must be running to perform this task.

This task designates an entry as the root of an independently replicated subtree and creates an **ibm-replicasubentry** representing this server as the single master for the subtree. To create a replicated subtree, you must designate the subtree that you want the server to replicate.

1. Use the Web Administration Tool to log on to the master server.
2. Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage topology**.
3. Click **Add subtree**.
4. Enter the DN of the subtree that you want to replicate or click **Browse** to expand the entries to select the entry that is to be the root of the subtree.

   Note: If you are not using a suffix, there are other requirements. See *IBM Tivoli Directory Server Version 5.2 Administration Guide*.

5. The master server referral URL is displayed in the form of an LDAP URL. For example,

   ```
   ldap://<myservername>.<mylocation>.<mycompany>.com
   ```

   Note: The master server referral URL is optional. It is used only:
   - If the server contains (or will contain) any read-only subtrees.
   - To define a referral URL that is returned for updates to any read-only subtree on the server.
6. Click **OK**.
7. The new server is displayed on the Manage topology panel under the heading **Replicated subtrees**.

Creating Credentials: Credentials identify the method and required information, such as a DN and password, which the supplier uses in binding to the consumer.

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage credentials**.
3. Select **cn=replication,cn=IBMpolicies** to store the credentials from the list of subtrees.
4. Click **Add**.
5. Enter the name for the credentials you are creating. For example, **mycreds,** cn= is already filled in the field for you.
6. Select **Simple bind** as the type of authentication and click **Next**.
   - Enter the DN that the server uses to bind to the replica. For example, cn=any.

     Note: This DN cannot be the same as your server administration DN.
   - Enter the password the server uses when it binds to the replica. For example, secret.
   - Enter the password again to confirm that here are no typographical errors.
   - If you want, enter a brief description of the credentials
   - Click **Finish**.

     Note: You might want to record the credential's bind DN and password for future reference. You will need this password when you create the replica agreement.

Creating a replica server: The servers must be running to perform this task.

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage topology**.
3. Select the subtree that you want to replicate and click **Show topology**.
4. Click the arrow next to the **Replication topology** selection to expand the list of supplier servers.
5. Select the supplier server and click **Add replica**.
6. On the **Server** tab of the **Add replica** window:
   - Enter the host name of the replica server. Do not change the default non-SSL port (389).

- Leave the **Enable SSL** check box unchecked.
- Enter the replica name or leave this field blank to use the host name.
- Enter the replica ID If the server on which you are creating the replica is running, click **Get replica ID** to automatically fill this field. For a replica, which is a z/OS LDAP server, this section will be filled as **UNKNOWN**. Enter a description of the replica server.

7. Click the **Additional** tab.
   - Specify the credentials that the replica uses to communicate with the master:
     – Click **Select**.
     – Click the radio button next to **cn=replication,cn=IBM policies**.
     – Click **Show credentials**.
     – Expand the list of credentials and select **mycreds**.
     – Click **OK**.

     See “Creating Credentials” on page 142 for additional information on agreement credentials.
   - Keep the **Specify a replication schedule or enter DN (optional)** set to **None**. This sets the default as immediate replication.
   - Do not deselect any capabilities.
   - Click **OK** to create the replica. A message is displayed noting that additional actions must be taken.
   - Click **OK**.
8. Next, the supplier information must be added to the replica. Open the **slapd.conf** configuration file of the z/OS LDAP server. Find the TDBM backend definitions and add the following configuration file options under the **suffix** that is to be replicated: **masterserver, masterserverdn, masterserverpw**.

   Example:

   ```
   masterserver ldap://<MasterServerIP>:<MasterServerPort>/
   masterserverdn cn=any
   masterserverpw password
   ```

   For masterserverdn and masterserverpw use the credentials you created in “Creating Credentials” on page 142.

   Restart the replica.

Starting replication: The replica is in a suspended state and no replication is occurring. After you have finished setting up your replication topology, on the master you must:

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage queues**.
3. Select the new replica.
4. Click **Suspend/resume** to start receiving replication updates for that server.

Master/slave replication setup is complete.

PART 2, in which you promote the master/slave replication network to a peer-to-peer replication network, consists of the following steps:

Changing the master to a peer: Refer to the **current** master as peer1 and the **current** replica as peer2. To define peer2 to peer1, we add peer2's definition to peer1's configuation file **ibmslapd.conf**.

For example:

```
dn: cn=Master Server,cn=Configuration
cn: Master Server
ibm-slapdMasterDN: cn=peer
ibm-slapdMasterPW: secret
objectclass: ibm-slapdReplication
objectclass: top
```

Promoting the replica to a peer: Stop peer2.

Open the **slapd.conf** configuration file for peer2 and delete **masterserver**, **masterserverDN**, and **masterserverPW** configuration file options.

Restart peer2.

Next, define peer1 to peer2. Create an LDIF file as in the example below and add it to peer2's directory, using the **ldapadd** utility.

For example:

```
dn: cn=myReplica,o=Your Company,c=US
objectclass: top
objectclass: replicaObject
cn: myReplica
replicaHost: <ip address>
replicaBindDn: cn=peer
replicaCredentials: secret
```

Stop peer2.

Open peer2's **slapd.conf** configuration file, find the TDBM backend definitions and add **peerserverDN peerserverPW** configuration file options under the **suffix** that is being replicated.

Example:

```
peerServerDN cn=peer
peerServerPW secret
```

Starting peer-to-peer replication: Restart servers peer1 and peer2. Peer-to-peer replication network setup between an IBM Tivoli Directory Server 5.2 and a z/OS LDAP Server on z/OS V1R6 is complete.

Configuration Option 2

This option shows you how to setup a master/slave replication network with z/OS LDAP Server on z/OS V1R6 as the MASTER and IBM Tivoli Directory Server 5.2 as the SLAVE.

PART 1 consists of the following steps:

Setting up IBM Tivoli Directory Server 5.2 as the Slave: Before setting up IBM Tivoli Directory Server 5.2 as the slave, we set it up as the master.

Follow the instructions in "Configuration Option 1" on page 141 up until 8 on page 143 to set up IBM Tivoli Directory Server 5.2 as the master and z/OS LDAP Server on z/OS V1R6 as the slave.

Next, delete the replication agreement that was just built between the master and the slave. That action leaves behind some information in the IBM Tivoli Directory Server that we will use later on.

1. Use the Web Administration Tool to log on to the master server.
2. Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage topology**.
3. Select the subtree that you picked to replicate earlier and click **Show topology**.
4. Click the arrow next to the **Replication topology** selection to expand the list of all the servers within the replication network. Keep doing this until both servers are listed.
5. Delete the replica server from the topology. (It is the bottom one in the list.)

Follow the directions below to look into the IBM Tivoli Directory Server 5.2 under ″ibm-replicaGroup=default,<subtreedn>″. You should find an ibm-replicaSubEntry object with the attribute ibm-replicationServerIsMaster=TRUE. Change this attribute value to FALSE.

1. Use the Web Administration Tool to log on to the master server.
2. Expand the **Directory management** category in the navigation area of the Web Administration Tool and click **Manage entries**.
3. Select the subtree that is being replicated and then click **Find**.
4. Pick **ibm-replicaSubEntry** objectclass from the **Find Entries with Following Objectclasses** drop down menu and click **OK**.
5. Click **Edit Attributes**.
6. Pick **FALSE** under **ibm-replicationServerIsMaster** section.
7. Click **OK**, **OK**, and **CANCEL** to go back to the **Manage entries** window.

Next, we need to define the new master server, z/OS V1R6 LDAP Server, to its replica, IBM Tivoli Directory Server 5.2. We do this by adding a description of the master to IBM Tivoli Directory Server's **ibmslapd.conf** configuration file. Use the description below as an example:

```
dn: cn=Master Server,cn=Configuration
cn: Master Server
ibm-slapdMasterDN: cn=peer
ibm-slapdMasterPW: secret
ibm-slapdMasterReferral: ldap://<MasterServerIP>:<MasterServerPort>/
objectclass: ibm-slapdReplication
objectclass: ibm-slapdConfigEntry
objectclass: top
```

Finally, go back to the Web Administration Tool and click on **Replication Management**, then click on **Manage Topology**. Pick the subtree that is being replicated and click on **Edit Subtree**. Change the current referral address to the master server's (z/OS V1R6 LDAP Server) address and click **OK**.

Setting up z/OS V1R6 LDAP Server as the Master: Now, define the replica to its master. Create an LDIF file similar to the example below, which includes a replicaObject representing the replica server. Then, add the LDIF file to the master's directory, using the **ldapadd** utility.

```
dn: cn=myReplica,o=Your Company,c=US
objectclass: top
objectclass: replicaObject
cn: myReplica
replicaHost: <ip address>
replicaBindDn: cn=peer
relicaCredentials: secret
```

Starting replication: Restart both servers.

The master/slave replication between an IBM Tivoli Directory Server 5.2 and a z/OS LDAP Server on z/OS V1R6 is complete.

PART 2, in which we promote the master/slave replication network to a peer-to-peer replication network, consists of the following steps:

Changing the master to a peer: Stop the master server.

Open the **slapd.conf** configuration file. Find the TDBM backend definitions. Add **peerServerDN** and **peerServerPW** configuration file options under the **suffix** that is to be replicated.

Example:

```
peerServerDN cn=peer
peerServerPW secret
```

Changing the replica to a peer:

1. Use the Web Administration Tool to log on to the master server.
2. Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage topology**.
3. Select the subtree that you want to replicate and click **Edit subtree**.
4. Scroll right and click on **Make server a master**. A message will pop up. Click **OK**, click **OK** on the next screen.
5. Click **Show topology**.
6. Click the arrow next to the **Replication topology** selection to expand the list of supplier servers. Select the supplier server and click **Add replica**.
7. On the **Server** tab of the **Add replica** window:
   - Enter the host name of the replica server. Do not change the default non-SSL port (389).
   - Leave the **Enable SSL** check box unchecked.
   - Enter the replica name or leave this field blank to use the host name.
   - Enter the replica ID If the server on which you are creating the replica is running, click **Get replica ID** to automatically fill this field. For a replica, which is a z/OS LDAP server, this section will be filled as **UNKNOWN**. Enter a description of the replica server.
8. Click the **Additional** tab.
   - Specify the credentials that the replica uses to communicate with the master:
     – Click **Select**.
     – Click the radio button next to **cn=replication,cn=IBM policies**.
     – Click **Show credentials**.
     – Expand the list of credentials and select **mycreds**.
     – Click **OK**.

     See “Creating Credentials” on page 142 for additional information on agreement credentials.
   - Keep the **Specify a replication schedule or enter DN (optional)** set to **None**. This sets the default as immediate replication.
   - Do not deselect any capabilities.
   - Click **OK** to create the replica. A message is displayed noting that additional actions must be taken.

- Click **OK**.

Starting peer-to-peer replication: Restart IBM Tivoli Directory Server 5.2 and then restart z/OS LDAP Server.

Peer-to-peer replication network is complete.

Reference information

We used the following documentation to set up these procedures:

- *z/OS Integrated Security Services LDAP Server Administration and Use*
- *A Simplified Approach to IBM Tivoli Directory Server V5.2 Replication*
- *IBM Tivoli Directory Server Administration Guide*

Using DB2 restart/recovery function

DB2 Restart/Recovery is a new feature for the z/OS V1R6 LDAP server that allows a TDBM and/or GDBM configured LDAP server to remain up and running even if DB2 shuts down. Then, once DB2 is restored, the LDAP server can function as normal. In the past, you had to restart the LDAP server in order to reconnect to DB2 once the connection was lost.

We used the following setup and tested DB2 restart/recovery:

1. Installed the fix for APAR PQ87724, which is required for this function.

---

2. Updated the slapd.conf file as follows:

```
# ------------------------------------------------------------------
database tdbm GLDBTDBM
suffix "o=xxx"
suffix "o=xxx"
servername xxxxxxxxx
dbuserid xxxxxx
databasename xxxxxx
dsnaoini GLD.CNFOUT.xx(DSNAOINI)
attroverflowsize 500
pwEncryption  none
schemaReplaceByValue on
extendedgroupsearching on
db2terminate restore

#  GDBM-specific CONFIGURATION SETTINGS
# ------------------------------------------------------------------
# ------------------------------------------------------------------
database gdbm GLDBGDBM
servername xxxxxxxxx
dbuserid xxxxxx
dsnaoini GLD.CNFOUT.xxx(DSNAOINI)
changeLogging on
changeLogMaxEntries 1000
changeLogMaxAge 86400
attroverflowsize 500
schemaReplaceByValue on
db2terminate restore
```

Note that db2terminate restore is the default, so that if you do not specify a value, you'll get restore. If you specify db2terminate terminate, the LDAP server will shut down when DB2 shuts down.

---

3. We tested DB2 restart/recovery function by bringing DB2 down and up again, and verifying that LDAP stayed up throughout:
   - After we brought DB2 down, the system issued the following LDAP server message:

     ```
     GLD0252E DB2 termination detected, database access unavailable.
     ```

     To verify that DB2 was down, we issued an ldapsearch command:

     ```
     ldapsearch -h <hostname> -b "o=IBM" "objectclass=*"
     ```

     The system returned a message saying that no such object was present, verifying that DB2 was down.
   - We brought DB2 up again, and received the following LDAP server message:

     ```
     GLD0253I DB2 restart detected, database access available.
     ```

     Again, we issued the ldapsearch command, this time to verify that DB2 was up:

     ```
     ldapsearch -h <hostname> -b "o=IBM" "objectclass=*"
     ```

     The system displayed the output for the search query.

Using alias support

Alias support provides a means for a TDBM directory entry to point to another entry in the same TDBM directory. If a distinguished name encountered during a search operation with dereferencing contains an alias, the alias is replaced by the value it points to and search continues using the new distinguished name. This support is designed to allow a user to make directory information available even if the entry is moved. It allows the user to point to a well-known name that would always lead to the entry.

We did the following to setup and test alias support:

1. Created an ldif file to exercise alias support. New file **alias.ldif** contains the following:

   ```
   dn: cn=js, o=IBM
   objectclass:person
   objectclass:aliasObject
   sn: Smith
   aliasedobjectname: cn=Joe Smith, ou=My Team, ou=Test Team, o=IBM
   ```

---

2. Next we loaded the alias.ldif file into the TDBM database with the following ldapmodify command:

   ```
   ldapmodify - h <hostname> -D "cn=LDAP Administrator" -w xxxxx -f /sysname/etc/ldap/alias.ldif
   ```

---

3. To make sure that alias.ldif was properly loaded, we issued the following ldapsearch command:

   ```
   ldapsearch - h <hostname> -b "o=js, o=IBM" "objectclass=*"
   ```

   This command displayed the following output, showing the new alias entry, which showed that alias.ldif was loaded properly:

```
dn: cn=js, o=IBM
objectclass:person
objectclass:aliasObject
sn: Smith
aliasedobjectname: cn =Joe Smith, ou=My Team, ou=Test Team, o=IBM
```

---

4. We tested alias support by issuing the following ldapsearch command with new parameter **-a always** that specifies that aliases are always dereferenced:

   ```
   ldapsearch -h <hostname> -a always -b "cn=js, o=IBM" "objectclass=*
   ```

   This command displays the actual entry that the alias represents:

   ```
   cn=Joe Smith, ou=My Team, ou=Test Team, o=IBM
   ```

   See *z/OS Integrated Security Services LDAP Client Programming* for information on the -a *deref* parameter.

Using the enhanced LDAP configurgation utility (LDAPCNF)

The LDAP configuration utility, ldapcnf, has been enhanced so that configuring TDBM is no longer required and to support configuring the change log. We tested the enhanced LDAPCNF utility using information from *z/OS Integrated Security Services LDAP Server Administration and Use*. We did the following to setup and test ldapcnf::

1. Created a new LDAP server for testing ldapcnf by doing the following:
   - Copied the following files from **/usr/lpp/ldap/etc** to **/sysname/etc/ldap**:

     ldap.profile
     ldap.slapd.profile
     ldap.db2.profile
     ldap.racf.profile
     slapd.conf
   - Next, we updated the profile files specific to our new LDAP server. We made the following changes to profile files:
     - **ldap.profile** and **ldap.slapd.profile**- We made system specific changes, see *z/OS Integrated Security Services LDAP Server Administration and Use* for information.
     - **ldap.db2.profile** - We enabled the LDAP server for TDBM and GDBM. You can choose to configure one or both.
     - **ldap.racf.profile** - We made system specific changes, see z/OS Integrated Security Services LDAP Server Administration and Use for information..

---

2. Started the LDAP configuration utility, ldapcnf, using the following command:

   ```
   ldapcnf -i ldap.profile
   ```

   The ldapcnf utility ran successfully with a return code of 0 and the utility created the following:
   - JCL jobs
   - SLAPDCNF (LDAP server configuration file)
   - SLAPDENV (LDAP server environment variable file)
   - PROG member needed for APF authorization
   - Procedure needed to start the LDAP server
   - DSNAOINI configuration file for DB2 CLI

- SPUFI DB2 SQL Statements for TDBM and GDBM

---

3. Next we copied the LDAP started task procedure to the procedure library for our new LDAP server and copied the PROGxx member to the target system's PARMLIB.

---

4. We then submitted the APF member and DBCLI member following JCL jobs. Our DB2 administrators submitted the DBSPUFI members for TDBM and GDBM using the DB2 SPUFI tool.

---

5. Finally, we started up the LDAP server as follows:

   ```
   s user_id
   ```

   We received the following message:

   ```
   GLD0122I Slapd is ready for requests.
   ```

   The LDAP Server was running with both TDBM and GDBM configured successfully. We successfully ran a variety of LDAP commands to test this server.

Using change logging with TDBM

For the z/OS V1R6 LDAP server, change logging has been enabled for changes made to entries in the TDBM backend. When a change is made to an entry in the TDBM backend, a record of the change will be created and stored in the GDBM backend.

To set up and test TDBM change logging, we did the following:

1. Because this support includes a new backend section (GDBM) in the configuration file, we first enabled the GDBM back end for the LDAP server in the configuration file:

   ```
   #  GDBM-specific CONFIGURATION SETTINGS
    # ----------------------------------------------------------------------
    # ----------------------------------------------------------------------
    database gdbm GLDBGDBM
    servername xxxxxxx
    dbuserid xxxxxx
    #databasename xxxxxx
    dsnaoini GLD.CNFOUT.xxx(DSNAOINI)
    changeLogging on
    changeLogMaxEntries 1000
    changeLogMaxAge 86400
    attroverflowsize 500
    schemaReplaceByValue on
   ```

   Once the GDBM backend is configured, the change logging support is automatically enabled for the corresponding TDBM back end. Note that **changeLogging on** is the default setting.

---

2. After this setup step, the slapd.conf file looks as follows (the bold sections show requirements for change logging):

```
# * This file is the LDAP Server configuration file for z/OS.
 # ************************************************************/
 #global section

 timelimit       3600
 sizelimit       500

 adminDN "cn=LDAP Administrator"
 adminPW xxxxxx


 listen ldap://:389
 listen ldap://:pc
 #  TDBM-specific CONFIGURATION SETTINGS
 # ----------------------------------------------------------------------
 # ----------------------------------------------------------------------
 changeLogging on
 database tdbm GLDBTDBM
 suffix "o=IBM"
 suffix "o=Your Company"
 servername xxxxxxxxxx
 dbuserid xxxxxx
 databasename xxxxxxx
 dsnaoini GLD.CNFOUT.xx(DSNAOINI)
 attroverflowsize 500
 pwEncryption  none
 schemaReplaceByValue on
 extendedgroupsearching on
 #  GDBM-specific CONFIGURATION SETTINGS
 # ----------------------------------------------------------------------
 # ----------------------------------------------------------------------
 database gdbm GLDBGDBM
 servername xxxxxxxxxx
 dbuserid xxxxxx
 dsnaoini GLD.CNFOUT.xx(DSNAOINI)
 changeLogging on
 changeLogMaxEntries 1000
 changeLogMaxAge 86400
 attroverflowsize 500
 schemaReplaceByValue on
 ########################################################################
 # sdbm database definitions
 ########################################################################
 database sdbm GLDBSDBM
 suffix "sysplex=xxxxxxxx,O=IBM,C=US"
```

---

3. Finally, we restarted the LDAP server. SLAPDOUT displays the following, showing that ChangeLogging is enabled:

```
GLD0244I Change logging is enabled
         Logging started status (0 = off, 1 = on): 1
         Limit in seconds on age of change log entries (0 = no limit): 86400
         Limit on the number of change log entries (0 = no limit): 1000
         Current number of change log entries: 0
         First change number in use: 0
         Last change number in use: 0
GLD3151I The backend containing the following suffix is participating in change logging: 'CN=CHANGELOG'.
GLD3151I The backend containing the following suffix is participating in change logging: 'o=IBM'.
GLD0202I Program Call communication is active.
GLD0122I Slapd is ready for requests.
```

---

4. Now we're ready to start testing the TDBM change logging. First, we created an ldif file, test.ldif, that would make a change to an entry in the TDBM backend:

```
dn: cn=John Doe, ou=My Team, ou=Test Team, o=IBM
changetype:modify
replace:x
title:ICSF
```

---

5. We then issued the following ldapmodify command to make the changes in the ldif file:

   ```
   ldapmodify -h <hostname> -D "cn=LDAP Administrator" -w xxxxx -f test.ldif
   ```

   The modifications were made successfully.

---

6. I then issued an ldapsearch against the change log to verify that a record was created for the change:

   ```
   ldapsearch -h <hostname> -b "cn=changelog" "objectclass=*"
   ```

   The following output was displayed, showing that TDBM change logging was working successfully:

   ```
   cn=changelog
   objectclass=top
   objectclass=container
   cn=changelog

   CHANGENUMBER=1201,CN=CHANGELOG
   objectclass=CHANGELOGENTRY
   objectclass=IBM-CHANGELOG
   objectclass=TOP
   changenumber=1201
   targetdn=cn=John Doe, ou=My Team, ou=Test Team, o=IBM

   changetime=20040701185439.759260Z
   changetype=MODIFY
   changes=replace:title
   title: ICSF
   -
   add:ibm-entryuuid
   ibm-entryuuid: B95FA000-5DEF-10E4-B0B9-40208401B52A
   -
   ibm-changeinitiatorsname=CN=LDAP ADMINISTRATOR
   ```

   TDBM change logging works successfully.

Chapter 10. Using Kerberos (Network Authentication Service)

Setting up a Kerberos peer trust relationship between z/OS and Windows 2000

We had a request from our colleagues in z/OS LDAP development to test the z/OS LDAP client with the Microsoft Windows 2000 Active Directory service using a Kerberos authentication bind. This was to help validate a fix they were working on involving the use of Kerberos authentication to bind to IBM Directory Server.

We needed to create a Kerberos realm that included Kerberos servers on z/OS and a Windows platform. Another test team had already enabled a Windows 2000 server for Kerberos authentication and we already had a Kerberos server running on z/OS. We now needed to create a peer trust relationship between these two Kerberos servers.

Since the Windows 2000 server was set up by another test team, we don't have a lot of details to share. However, we intend to continue our testing with Kerberos peer relationships and transitive trust relationships on other platforms and from other suppliers. We'll have more to say about how to enable such scenarios at that time. For now, we will focus on what we did from the z/OS perspective to set up the peer trust relationship and how we validated our setup.

We used the following documentation to help us set up the z/OS portion of the peer trust relationship:

- *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926
- *z/OS Integrated Security Services LDAP Client Programming*, SC24-5924
- *z/OS Security Server RACF Command Language Reference*, SA22-7687

Enabling the peer trust relationship on z/OS

There are two main areas that need to be configured to enable the peer trust relationship for Kerberos authentication from the z/OS perspective: the krb5.conf configuration file and RACF. We configured these accordingly, as described below, to define a new Kerberos realm. (For more information, see the appendix containing sample Kerberos configurations in *z/OS Integrated Security Services Network Authentication Service Administration*.)

Defining the Windows 2000 realm to the Kerberos server on z/OS

We did the following to update the /etc/skrb/krb5.conf configuration file:

1. Under the [Realms] section, we defined the Windows 2000 server as follows:

```
KERBEROS.XXX.YYY.IBM.COM = {
   kdc = kerb2000.kerberos.xxx.yyy.ibm.com:88
   kpasswd_server = kerb2000.kerberos.xxx.yyy.ibm.com:464
}
```

---

2. Under the [domain_realm] section, we defined the Windows 2000 server as follows:

```
.kerberos.xxx.yyy.ibm.com = KERBEROS.XXX.YYY.IBM.COM
```

Defining the cross-realm certification in RACF

We issued the following RACF commands to define the cross-realm certification:

```
RDEFINE REALM /.../KERBEROS.XXX.YYY.IBM.COM/krbtgt/XXX.YYY.IBM.COM KERB(PASSWORD(win2k_pw))

RDEFINE REALM /.../XXX.YYY.IBM.COM/krbtgt/KERBEROS.XXX.YYY.IBM.COM  KERB(PASSWORD(zos_pw))
```

Testing the peer trust relationship

Since the intention for this scenario was to test the z/OS LDAP client, we used the z/OS LDAP client and the Windows 2000 Active Directory service to validate the newly created Kerberos realm and the peer trust relationship.

We did the following to test the peer trust relationship (all commands are issued from the z/OS UNIX shell):

1. We issued the **kinit** command to obtain Kerberos credentials. Because we were trying to obtain credentials from the Windows 2000 server, we issued the **kinit** command using a Windows 2000 Kerberos principal.

   Example: `kinit SAM@KERBEROS.XXX.YYY.IBM.COM`

   Result: `EUVF06017R Enter password:`

   If everything is set up correctly and you enter the correct password, you will simply return to the command prompt when the **kinit** command successfully completes.

   However, if there is a problem with the setup and you are unable to access the Windows 2000 server, you would see the following error message:

   ```
   EUVF06014E Unable to obtain initial credentials.
              Status 0x96c73a9a - Unable to locate security server.
   ```

   Our **kinit** command was successful and we returned to the command prompt.

---

2. We used the **klist** command to verify that we had the expected credentials.

   Example: `klist`

   Result: We received the following response, as expected:

   ```
   Ticket cache: FILE:/var/skrb/creds/krbcred_0a3ae270
   Default principal: SAM@KERBEROS.XXX.YYY.IBM.COM

   Server: krbtgt/KERBEROS.XXX.YYY.IBM.COM@KERBEROS.XXX.YYY.IBM.COM
     Valid 2004/05/13-13:21:23 to 2004/05/13-23:21:23
   ```

At this point, we were confident that we had established communication between z/OS and the Kerberos server running on Windows 2000. We could now use other applications that require binding with Kerberos authentication. In this scenario, we used the z/OS LDAP client to search the Windows 2000 Active Directory, as described in the following steps.

3. We issued the **ldapsearch** command to search the Windows 2000 Active Directory.

   Example: We entered the following command as a single line from the z/OS UNIX command prompt:

   ```
   ldapsearch -h ip_address_of_win2k_server -V 3 -S GSSAPI -s base
      -b "CN=Users,DC=kerberos,DC=xxx,DC=yyy,DC=ibm,DC=com" "objectclass=*"
   ```

   Result: We received the following response, as expected:

```
CN=Users,DC=kerberos,DC=xxx,DC=yyy,DC=ibm,DC=com
cn=Users
description=Default container for upgraded user accounts
instanceType=4
isCriticalSystemObject=TRUE
distinguishedName=CN=Users,DC=kerberos,DC=xxx,DC=yyy,DC=ibm,DC=com
objectCategory=CN=Container,CN=Schema,CN=Configuration,DC=kerberos,DC=xxx,DC=yyy,DC=ibm,DC=com
objectClass=top
objectClass=container
objectGUID=NOT Printable
name=Users
showInAdvancedViewOnly=FALSE
systemFlags=-1946157056
uSNChanged=1314
uSNCreated=1314
whenChanged=20030506135552.0Z
whenCreated=20030506135552.0Z
```

To validate the peer trust relationship between the Kerberos server on z/OS and the Kerberos server on Windows 2000, we needed to clear out our existing Kerberos credentials (that we had obtained using a Windows 2000 Kerberos principal), obtain new Kerberos credentials using a z/OS Kerberos principal, and then rerun the **ldapsearch** command.

4. We issued the following command to clear out any existing Kerberos credentials:

   ```
   kdestroy
   ```

   ---

5. We issued the **kinit** command to obtain new Kerberos credentials using a z/OS Kerberos principal.

   Example: `kinit LDAP/zOS.ibm.com`

   ---

6. We reissued the same **ldapsearch** command as before to search the Windows 2000 Active Directory.

   Example: We entered the following command as a single line from the z/OS UNIX command prompt:

   ```
   ldapsearch -h ip_address_of_win2k_server -V 3 -S GSSAPI -s base
      -b "CN=Users,DC=kerberos,DC=xxx,DC=yyy,DC=ibm,DC=com" "objectclass=*"
   ```

   Result: As expected, the response was identical to the one received before.

Network Authentication Service (NAS) enhancements in z/OS V1R6

All Integrated Security Server products are now placing their load modules in SYS1.SIEALNKE instead of maintaining their own load module data set. This is a new data set for the Integrated Security Server products.

We used *z/OS Integrated Security Services Network Authentication Service Administration* and *z/OS Migration* when migrating to this new level and using this new data set.

Accessing SYS1.SIEALNKE

Prior to starting any NAS servers, verify that the SYS1.SIEALNKE data set is link list. If it is not in link list, then you must use STEPLIB to locate the data set.

When SYS1.SIEALNKE is not in link list, the NAS server will not start and the following error is seen in the JES log.

```
|
|                        IEF403I  SKRBKDC - STARTED - TIME=20.41.45
|                        CSV003I  REQUESTED MODULE EUVFSKDC NOT FOUND
|                        CSV028I  ABEND806-04  JOBNAME=SDRBKDC  STEPNAME=SKRBKDC
|                        IEA995I  SYMPTOM DUMP OUTPUT
|                        SYSTEM COMPLETION CODE=806  REASON CODE=00000004
```

Chapter 11. Using the IBM WebSphere Business Integration family of products

The IBM WebSphere MQ (formerly MQSeries) family of products forms part of the newly re-branded WebSphere Business Integration portfolio of products. These products are designed to help an enterprise accelerate the transformation into an on demand business.

This chapter discusses the following topics:

Using WebSphere MQ shared queues and coupling facility structures

Using Websphere MQ, programs can talk to each other across a network of unlike components, including processors, operating systems, subsystems, and communication protocols, using a simple and consistent application programming interface.

We currently run WebSphere MQ for z/OS Version 5.3.1. We originally discussed our implementation of shared queues in our December 2002 edition. We continue that discussion by focusing on the usage and behavior of the coupling facility structures that support shared queues.

We used information from the following sources to set up and test our shared queues:

- *WebSphere MQ for z/OS System Administration Guide*, SC34-6053, for information about recovery from DB2, RRS, and CF failures. This document is available from the WebSphere Business Integration library at www.ibm.com/software/integration/websphere/library/.
- *WebSphere MQ in a z/OS Parallel Sysplex Environment*, SG24-6864, available from IBM Redbooks at www.ibm.com/redbooks/
- *WebSphere MQ Queue Sharing Group in a Parallel Sysplex Environment*, REDP-3636, available from IBM Redbooks at www.ibm.com/redbooks/

Our queue sharing group configuration

We currently have two queue sharing groups: one with three members and another with six members. The smaller queue sharing group is for testing new applications or configurations before migrating them to our production systems. The queue sharing groups each connect to different DB2 data sharing groups. This discussion will focus on the six-member production queue sharing group. All of the queue managers in the group run WebSphere MQ for z/OS Version 5.3.1.

Our coupling facility structure configuration

We defined our MQ coupling facility structures to use two coupling facilities (CF2 and CF3) as defined in the preflist in the structure definitions. (See "Coupling facility details" on page 7 for details about our coupling facilities.)

The following is the structure definition for our CSQ_ADMIN structure:

```
STRUCTURE NAME(MQGPCSQ_ADMIN)
          INITSIZE(15000)
          MINSIZE(15000)
          DUPLEX(ENABLED)
          SIZE(16000)
          ALLOWAUTOALT(YES)
          PREFLIST(CF3,CF2)
          REBUILDPERCENT(1)
          FULLTHRESHOLD(85)
```

We also have the following four message structures defined to support different workloads:

- MSGQ1 — for the batch stress workload
- CICS — for the CICS bridge application
- EDSW — for the IMS bridge application
- WMQI — for the WebSphere MQ Integrator retail application

The following is the structure definition for the message structure that supports the MQ-CICS bridge workload:

```
STRUCTURE NAME(MQGPCICS)
          INITSIZE(10240)
          DUPLEX(ENABLED)
          SIZE(20480)
          ALLOWAUTOALT(YES)
          PREFLIST(CF2,CF3)
          REBUILDPERCENT(1)
          FULLTHRESHOLD(85)
```

The other three message structures are defined similarly, except for the sizes. All of the structures are enabled for duplexing.

We chose to create multiple message structures in order to separate them by application. That way, if there is a problem with a structure, it will not impact the other applications. However, this is not necessarily the recommended approach from a performance perspective. See the Redbook Paper *WebSphere MQ Queue Sharing Group in a Parallel Sysplex Environment* for more information.

The CICS structure is recoverable and is backed up daily. The other structures are non-recoverable; however, we plan to make the EDSW and WMQI structures recoverable in the near future.

Testing the recovery behavior of the queue managers and coupling facility structures

We conducted the following types of test scenarios during our z/OS release testing:

- CF structure errors
- CF structure duplexing and moving structures between coupling facilities
- CF-to-CF link failures
- MQ CF structure recovery

During these tests, we monitored the behavior of the MQ queue managers as well as the behavior of applications that use shared queues.

Queue manager behavior during testing

We observed the following behavior during our test scenarios:

CF structure errors: With the MQ CICS bridge workload running, we used a local tool to inject errors into the coupling facility structures. When we injected an error into the MQ administrative structure, the structure moved to the alternate coupling facility, based on the preflist, as expected. Throughout the test, the CICS bridge workload continued to run without any errors.

CF structure rebuild on the alternate coupling facility: With system-managed CF structure duplexing active and a shared queue workload running, we issued the SETXCF STOP,REBUILD command to cause XCF to move the MQ structures to the alternate coupling facility. The queue manager produced no errors and the application continued without any interruption.

MQ structure recovery: During our normal coupling facility testing, the MQ CICS structure went into a failed state for valid reasons. This afforded us the opportunity to test MQ structure recovery. We issued the RECOVER CFSTRUCT command and the structure recovered with no errors.

We also tested recovering into an empty structure. We first issued the SETXCF FORCE command to clear the structure, followed by the RECOVER CFSTRUCT(CICS) TYPE(PURGE) command. Again, the structure recovered with no errors.

Suggested MQ maintenance

During the course of our testing, we applied the following WebSphere MQ APARs:

- PQ72242
- PQ72755
- PQ74895
- PQ75276
- PQ76590
- PQ77039
- PQ77396
- PQ77558
- PQ78586
- PQ82073
- PQ82506

Additional experiences and observations

MQ abends during coupling facility failures: Although coupling facility failures are extremely rare under normal operations, we induce many failures in our environment in the course of our testing. When coupling facility failures occur which have an impact on WebSphere MQ, such problems generally manifest themselves as MQ dumps with abend reason codes that start with 00C51*nnn*. Many of these are actually coupling facility problems or conditions that result in MQ having a problem and are not necessarily MQ problems in their own right. When such abends occur, we suggest that you analyze the system log for any IXC or IXL messages that might indicate a problem with a coupling facility.

Intra-group queuing: We have all members of the queue sharing group set up for intra-group queueing. This was done by altering the queue manager to enable intra-group queuing. SDSF makes use of the SYSTEM.QSG.TRANSMIT shared queue for transmitting data between SDSF servers instead of the cluster queues. It continues to use the cluster queues and channels for members not in the queue sharing group. Currently all systems in our sysplex have the SDSF MQ function enabled so job output for one system can be viewed from any other system in the sysplex.

Effects of DB2 and RRS failures on MQ: We also tested how MQ reacts when DB2 or RRS become unavailable. The following are some of our observations:

- APAR PQ77558 fixes a problem with MQ V5.3.1 when RRS is cancelled while the queue manager is running.
- When DB2 or RRS become unavailable, the queue manager issues an error message to report its loss of connectivity with DB2 and which subsystem is down. An example of such a message is:

  ```
  CSQ5003A !MQJA0 CSQ5CONN Connection to DB2 using DB1G pending, no active DB2
  ```

  When DB2 becomes available again, MQ issues a message to report that it is again connected to DB2. For example:

  ```
  CSQ5001I !MQJA0 CSQ5CONN Connected to DB2 DBD1
  ```
- MQ abend reason codes that indicate a DB2 failure start with 00F5*nnnn*.

Notes about MQ coupling facility structure sizes:

- All of our MQ coupling facility structures are defined to allow automatic alter (by specifying ALLOWAUTOALT(YES) in the structure definitions in the CFRM policy), whereby XCF can dynamically change the size of a structure, as necessary. This is beneficial because it allows XCF to automatically increase the size of a message structure as needed to hold more messages.
- When we first defined the CSQ_ADMIN structure, we made it 10000K bytes in size. Our original sizing was based on the guidelines in *WebSphere MQ for z/OS Concepts and Planning Guide*, GC34-6051. However, we have since migrated to a higher CFCC level and increased the number of queue managers in the queue sharing group, which increases the size requirement for the CSQ_ADMIN structure. As a result, the queue manager recently failed to start because the CSQ_ADMIN structure was too small and issued the following message:

  ```
  CSQE022E !MQJA0 Structure CSQ_ADMIN unusable, size is too small
  ```

  We used the SETXCF START,ALTER command to increase the size of the structure. The following is an example of the command we issued:

  ```
  SETXCF START,ALTER,STRNAME=MQGPCSQ_ADMIN,SIZE=16000
  ```

  Accordingly, we also increased the value of INITSIZE() and MINSIZE() for CSQ_ADMIN in the CFRM policy from 10000 to 15000 to accommodate the increase in usage.

Implementing WebSphere MQ shared channels in a distributed-queuing management environment

We implemented shared channels within the larger of our two queue sharing groups to bolster our distributed-queuing management (DQM) environment. Previously, we have had a DQM workload that exercised distributed messaging using MQ channels that provided an environment to test channel functionality such as SSL, as well as more general testing such as load stress. For z/OS V1R5, we modified the underlying DQM environment to utilize both shared inbound and shared outbound channels without having to change the workload application. We are now able to handle higher amounts of inbound messages from remote MQ clients and, at the same time. provide transparent failover redundancy for those inbound messages.

Our MQ ″clients″ are in fact full MQ servers on distributed platforms such as Linux and Windows 2000.

Our shared channel configuration

The following sections describe the configuration of our shared inbound and outbound channels. We used information in *WebSphere MQ Intercommunication*, SC34-6059, to plan our configuration.

Shared inbound channels

We decided to implement the shared channel environment on our sysplex using TCP/IP services because our distributed DQM clients are mainly TCP/IP clients. All queue managers in the queue sharing group were configured to start group listeners on the same TCP port (1415), as described in the MQ intercommunication guide.

Example: The following is an example of the command to start group listeners on TCP port 1415:

```
START LISTENER INDISP(GROUP) PORT(1415)
```

The MQ intercommunication guide describes how the group listener port maps to a generic interface that allows the queue sharing group to be seen as a single network entity. For our DQM environment, we configure the Sysplex Distributor service of z/OS Communications Server to serve as the TCP/IP generic interface. This is a slight departure from the intercommunication guide, which utilizes DNS/WLM to provide the TCP/IP generic interface. VTAM generic resources is another available service that can provide the generic interface for channels defined using LU6.2 connections.

Example: The following is an example of our Sysplex Distributor definition for TCP port 1415:

```
VIPADYNAMIC
VIPADEFINE MOVEABLE IMMED 255.255.255.0  192.168.32.30
VIPADISTRIBUTE DEFINE 192.168.32.30 PORT 1415
DESTIP 192.168.49.31 192.168.49.32 192.168.49.33 192.168.49.34 92.168.49.36 192.168.49.38
ENDVIPADYNAMIC
```

We added this definition to the TCP/IP profile of one of our queue sharing groups (in this case 192.168.49.32), but it can be added to any TCP/IP host within the sysplex in which the queue sharing group resides. The IP addresses listed for DESTIP are the XCF addresses of the queue managers in our queue sharing group. The remote client can then specify 192.168.32.30 (or, correspondingly, the host name MQGP, which maps to that IP address in our DNS server for our 192.168.*xx.xx* LAN) on its sender channel, which then causes the receiver channel start to be load-balanced using the WLM mechanisms of Sysplex Distributor.

Example: The following is an example of our definitions for the remote sender channel and the local receiver channel:

```
DEFINE  CHANNEL(DQMLNXP.TO.DQMMQGP) +
  REPLACE +
  CHLTYPE(SDR) +
  XMITQ(DQMMQGP.XMIT.QUEUE) +
  TRPTYPE(TCP) +
  DISCINT(15) +
  CONNAME('MQGP(1415)')

DEFINE  CHANNEL(DQMLNXP.TO.DQMMQGP) +
  REPLACE +
  CHLTYPE(RCVR) +
  QSGDISP(GROUP) +
  TRPTYPE(TCP)
```

Note that QSGDISP(GROUP) specifies that a copy of this channel is defined on each queue manager in the queue sharing group. This allows the inbound channel start request to be serviced by any queue manager in the queue sharing group. At this point, messages can be placed on application queues that are either shared or local to the queue manager (as long as they are defined on each queue manager in the queue sharing group, specifying QSGDISP(GROUP) in the definitions).

Shared outbound channels

The MQ intercommunication guide states that an outbound channel is a shared channel if it moves messages from a shared transmission queue. Thus, we defined a shared transmission queue for our outbound channels, along with an outbound sender channel with a QSGDISP of GROUP. This enables the queue managers in the queue sharing group to perform load-balanced start requests for this channel.

Example: The following is our definition for the shared transmission queue:

```
DEFINE  QLOCAL(DQMLNXP.XMIT.QUEUE) +
  REPLACE +
  QSGDISP(SHARED) +
  CFSTRUCT(MSGQ1)
  TRIGGER +
  TRIGDATA(DQMMQGP.TO.DQMLNXP) +
  INITQ(SYSTEM.CHANNEL.INITQ) +
  USAGE(XMITQ)  +
  STGCLASS(DQMSTG)
```

Example: The following are our definitions for the local sender channel and the remote receiver channel:

```
DEFINE CHANNEL(DQMMQGP.TO.DQMLNXP) +
  REPLACE +
  CHLTYPE(SDR) +
  XMITQ(DQMLNXP.XMIT.QUEUE) +
  QSGDISP(GROUP) +
  TRPTYPE(TCP) +
  DISCINT(15) +
  CONNAME(remote_client_host_name)

DEFINE CHANNEL(DQMMQGP.TO.DQMLNXP) +
  REPLACE +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)
```

Testing shared channel recovery

Based on the information in the MQ intercommunication guide, as well as information in the IBM Redbook, *WebSphere MQ in a z/OS Parallel Sysplex Environment*, SG24-6864, we tested several scenarios for shared channel recovery. For each scenario, we varied the DISCINT parameter of the channels in order to strike a balance between manual channel status observation and load-balanced channel starts. For our particular workload and environment, we set it to 60 seconds.

By observing the WLM goals for the Sysplex Distributor (using the NETSTAT VDPT command), we were able to ascertain the queue manager on which the inbound channel likely would start. In all of our tests, our sysplex workload mix caused queue manager CSQC on system JC0 to be favored as the destination of the Sysplex Distributor.

The following are the shared channel recovery scenarios that we tested, along with our experiences and observations:

Testing channel initiator failure

Action: Cancel the CHINIT address space.

Expected Results: The channel initiator fails, but the associated queue manager remains active. The queue manager monitors the failure and initiates recovery processing.

Actual Results: From a NETSTAT VDPT display, we observed that system JC0 had a WLM goal of 13 and JB0 had a goal of 12. All other queue manager systems had lower WLM goals. Thus, we expected the channel to start on JC0 (queue manager CSQC) and recover to JB0 (queue manager CSQB) when CSQCCHIN was cancelled.

With our DQM workload running over a shared inbound channel from our remote Linux host (queue manager LNXP) to the CSQC member of our MQGP queue sharing group, we canceled CSQCCHIN. The application continued to run successfully after the channel restarted on CSQB (as it had the highest WLM goal in the queue sharing group). After we restarted CSQCCHIN, when the channel timed out on CSQB, the next set of messages caused the channel to restart on CSQC.

The following messages appeared on the queue manager where CSQCCHIN was canceled:

```
CSQ3201E !MQJC0 ABNORMAL EOT IN PROGRESS FOR USER=
        CONNECTION-ID=CSQCCHIN THREAD-XREF=
CSQM052I !MQJC0 CSQMPCRT Shared channel recovery
        completed for CSQC, 1 channels found, 0 FIXSHARED, 1 recovered
```

Testing queue manager failure

Action: Cancel the MSTR address space.

Expected Results: The queue manager fails (failing the associated channel initiator). Other queue managers in the queue sharing group monitor the event and initiate peer recovery.

Actual Results: With our DQM workload running from our remote Linux host (queue manager LNXP) to the CSQC member of our MQGP queue sharing group, we canceled CSQCMSTR. The channel restarted on queue manager CSQB and the application continued to run successfully. After CSQCMSTR had completely restarted and channel DQMLNXP.TO.DQMMQGP timed out to CSQB, the next set of messages caused the channel to restart on CSQC (CSQCCHIN was restarted when CSQCMSTR restarted).

Testing DB2 failure

Action: Cancel the DB2 subsystem.

Expected Results: Channel state information is stored in DB2, so a loss of connectivity to DB2 becomes a failure when a channel state change occurs. Running channels can continue running without access to these resources. On a failed access to DB2, the channel enters the retry state.

Actual Results: During normal operations, we lost the connection to the DB2 subsystem from queue manager CSQA (which is also a member of our MQGP queue sharing group). Subsequent attempts to display, start, or stop shared channels failed with the following error message:

```
CSQM294I – CSQA CSQMDRTS CANNOT GET INFORMATION FROM DB2
```

We then had to wait until the connection to DB2 was re-established in order to change the state of any shared channels. This corresponds to results discussed in *WebSphere MQ in a z/OS Parallel Sysplex Environment*.

Using WebSphere Business Integration Message Broker

WebSphere Business Integration Message Broker is the latest version of the product formerly known as WebSphere MQ Integrator. This section continues the discussion of our experiences with WebSphere MQ Integrator V2.1 from our December 2003 edition and includes our experiences migrating to WebSphere Business Integration Message Broker V5.0.

Note: To simplify the discussion, we'll refer to WebSphere Business Integration Message Broker as WBIMB, and WebSphere MQ Integrator as WMQI. However, these abbreviations are not officially sanctioned by IBM, so you should not use them to try to locate information or for product ordering, for instance.

Testing WMQI V2.1 on DB2 V8

We brought up one of our WMQI V2.1 brokers (CSQ1BRK) on DB2 V8 just to see if this would work. To do this, we first deleted the broker and it's broker database on DB2 V7, and then recreated the broker using DB2 V8 for the broker and application databases. We also tested the scenario where the broker database is on DB2 V8 and the application databases is on DB2 V7 and had no problems. We tested using a variant of the Retail WMQI application that we described in our December 2003 edition.

Setting the _BPXK_MDUMP environment variable to write broker core dumps to MVS data sets

By default, broker core dumps are written to the home directory of the owner of the broker's started task in z/OS UNIX, with each dump in a separate file with a unique identifier. For example, in our sysplex, the started task owner for all brokers is MQSTEST, so the dumps are written to MQSTEST's home directory in the z/OS UNIX file system (/u/mqstest).

The problem is that these dumps are often quite large and can quickly fill up the HFS if it is not carefully monitored. Also, since we have four brokers running on different systems in the sysplex, with each broker using MQSTEST as the started task owner, it can often be difficult to determine which broker created which core dump.

We solved these problems by customizing our brokers to write core dumps to MVS data sets instead of to the started task owner's home directory. We specified a different data set for each broker's core dumps. To do this, we added the environment variable _BPXK_MDUMP to the ENVFILE file and also to the mqsicompcif file (both of which reside in the broker's component directory in the

z/OS UNIX file system) so that if the broker is re-customized in the future, this change will not be lost. It is important to remember that the broker may have to be restarted to pick up changes to the ENVFILE.

We did the following to define the _BPXK_MDUMP environment variable and prepare the target MVS data sets to receive the broker core dumps:

1. We defined the _BPXK_MDUMP environment variable in the ENVFILE file to specify the name of the MVS data set to receive broker core dumps.

   Example: We added the following line to the end of the ENVFILE file on system Z2 to cause broker core dumps on that system to be written to the MVS data set WMQI.COREDUMP.Z2:

   ```
   _BPXK_MDUMP=WMQI.COREDUMP.Z2
   ```

---

2. Similarly, we also defined the _BPXK_MDUMP environment variable in the mqsicompcif file, between the (ENVIRONMENTBEGIN) and (ENVIRONMENTEND) tags.

   Example: We added the following to the mqsicompcif file on system Z2:

   ```
   (ENVIRONMENTBEGIN)
   _BPXK_MDUMP=WMQI.COREDUMP.Z2
   (ENVIRONMENTEND)
   ```

---

3. We allocated an empty MVS sequential data set for each data set name that we specified by a _BPXK_MDUMP environment variable. The data sets must already exist in order for core dumps to be written to them.

   Example: The following is an example of the attributes we specified to allocate the MVS data sets:

```
Data Set Name . . . . : WMQI.COREDUMP.Z2

General Data                          Current Allocation
 Management class . . : NOMIG          Allocated cylinders : 750
 Storage class  . . . : STANDARD       Allocated extents . : 1
  Volume serial . . . : PPRD0B
  Device type . . . . : 3390
 Data class . . . . . : **None**      Current Utilization
  Organization  . . . : PS             Used cylinders  . . : 0
  Record format . . . : FBS            Used extents  . . . : 0
  Record length . . . : 4160
  Block size  . . . . : 4160
  1st extent cylinders: 750
  Secondary cylinders : 250
  Data set name type  :                SMS Compressible  . : NO

  Creation date . . . : yyyy/mm/dd     Referenced date . . : ***None***
  Expiration date . . : ***None***
```

Note: We found that each time a broker writes a core dump, it simply appends to the end of the dump data set; therefore, it is important to monitor and clear out these data sets on a regular basis.

Resolving a EC6–FF01 abend in the broker

We ran into a problem where, immediately upon starting the broker, it would fail with an EC6–FF01 abend. We traced the cause of this problem back to the HFS for the broker component directory being completely full. Once we allocated more space for the HFS, the broker was able to successfully start.

Migrating WebSphere MQ Integrator V2.1 to WebSphere Business Integration Message Broker V5.0

We used the WBIMB V5.0 documentation in the online WebSphere Business Integration Information Center (publib.boulder.ibm.com/infocenter/wbihelp/index.jsp) and in the IBM Redbook, *Migration to WebSphere Business Integration Message Broker V5*, SG24-6995, to perform our migration.

Migration activities on the Windows platform

We performed the following activities to migrate to WBIMB V5.0 on the Windows platform:

1. We installed DB2 V8 on our Windows XP system, which replaced DB2 V7 that was already installed.

---

2. We had to migrate the existing broker and configuration manager databases so that they could be used with DB2 V8.

   Example: We issued the following commands from the DB2 command line processor to migrate the broker databases from DB2 V7 to DB2 V8:

   ```
   migrate database MQSIBKDB
   migrate database MQSICMDB
   migrate database MQSIMRDB
   ```

---

3. We performed some testing to verify that the WMQI V2.1 broker, configuration manager, and Control Center were still working following the database migration.

---

4. We installed the WBIMB V5.0 tool kit on our Windows XP system.

   To do this, we followed the instructions in the online Information Center and the Redbook cited earlier. We strongly recommend that you read the relevant sections in the Redbook before starting your migration. As the instructions state, it is important to first uninstall the WMQI V2.1 Control Center *excluding data,* so that the data remains on the Windows system.

Migration activities on the z/OS platform

We have not yet tested the documented migration path for taking existing WMQI V2.1 brokers to WBIMB V5.0. Since the broker is a runtime component, we felt that simply deleting the V2.1 broker, recreating it as a V5.0 broker, and deploying the appropriate flows would be a satisfactory migration path. However, we do intend to try migrating one of our brokers by using the jobs and documentation that WBIMB V5.0 provides and we will report on any relevant experiences in a future test report.

For now, we deleted our WMQI V2.1 brokers and recreated them as WBIMB V5.0 brokers with Fix Pack 01. We noticed that the installation directory in WBIMB V5.0 is different than it was in WMQI V2.1 (for us, it was /wbimb50/mqsi/V5R0M0 in V5.0, and /wbimb50 in V2.1). Later on, after applying Fix Pack 02 and Fix Pack 03

(we did one right after the other), we noticed that the installation directory had again changed (now it was /wbimb50/V5R0M1), so we again had to update our jobs with the correct directory path.

As the documentation points out, it is important to note that Java version 1.4.0 is the minimum level that is required for WBIMB V5.1 on z/OS. See the section, "Checking the level of Java (z/OS)," in the online Information Center.

Applying WBIMB V5.0 Fix Pack 02 and Fix Pack 03

It is important to realize that FixPack 02 is actually a new release of WBIMB—namely, WBIMB 5.0.1. One difference that we found is the change in the installation directory path, as we mentioned earlier. This means that the JCL in the jobs generated by the mqsicustomize step needs to be updated to point to the new directory path. We found that the easiest way to do this was by recreating the broker and customizing it with a mqsicompcif file that uses the correct installation library.

After applying the Fix Packs, we also ran into a problem where we could run the customization verification program (job name BIP$JCVP) with no errors, but when we started the broker, it would come down after a minute or two with a 4039 user abend. We found that we had not listed all of the necessary directories in the LIBPATH in the mqsicompcif and ENVFILE files (both of these files reside in the broker's component directory in the z/OS UNIX file system). We corrected the problem and the broker started and ran successfully.

Some useful WBIMB Web sites

We found the following Web sites useful when working with WebSphere Business Integration Message Broker:

- README files for WebSphere MQ family products: www.ibm.com/software/integration/mqfamily/support/readme/
- Fix Packs for WebSphere Business Integration Brokers: www.ibm.com/software/integration/mqfamily/support/summary/wbib.html
- SupportPacs for WebSphere MQ family products: www.ibm.com/software/integration/support/supportpacs/product.html

  Note: We found SupportPac IP13 for WebSphere Business Integration Brokers to be particularly useful.
- IBM Redbooks: www.ibm.com/redbooks/

Chapter 12. Using IBM WebSphere Application Server for z/OS

This chapter describes our experiences using IBM WebSphere Application Server for z/OS and related products. There are two environments currently being supported:

- The first is WebSphere Application Server for z/OS V4.0.1 running on z/OS V1R5. See "About our z/OS V1R5 test environment running WebSphere Application Server."
- The second is WebSphere Application Server for z/OS V5.*x* running on z/OS V1R6. See "Migrating to WebSphere for z/OS V5.*X*" on page 177

Note: References to WebSphere Application Server for z/OS V5.*x* appear in the text as ″WebSphere for z/OS V5.*x*″ or simply ″V5.*x*.″ References to WebSphere Application Server for z/OS V4.*x* appear in the text as ″WebSphere for z/OS V4.*x*″ or simply ″V4.*x*.″ References to earlier releases, such as WebSphere Application Server for OS/390 V3.5 appear as ″Standard Edition V3.5″ or simply ″SE V3.5.″

About our z/OS V1R5 test environment running WebSphere Application Server

Over the past few months, we have made a number of changes and updates to our WebSphere Application Server for z/OS test environment. In this chapter, we provide a level-set view of our current test environment and provide details about the changes we've made and our experiences along the way.

Our z/OS V1R5 WebSphere test environment

This section provides an overview of our z/OS V1R5 WebSphere test environment, including the set of software products and release levels that we run, the Web application configurations that we support, and the workloads that we use to drive them.

Current software products and release levels

The following information describes the software products and release levels that we use on the z/OS platform and on the workstation platform.

Software products on the z/OS platform: In addition to the elements and features that are included in z/OS V1R5 (migrated up from z/OS V1R4), our WebSphere test environment includes the following products:

- WebSphere for z/OS V4.0.1, service level W401502 (PTF UQ75293)
- WebSphere Application Server for OS/390 SE V3.5, PTF 41 (PTF UQ74996)
- IBM SDK for z/OS, Java 2 Technology Edition V1.3.1
- WebSphere Studio Workload Simulator V1.0
- DB2 V7.1
  - JDBC (PQ61146)
- CICS TS V2.2
  - CICS Transaction Gateway (CTG) V5.0
- IMS V7.1
  - IMS Connector V1.2

Software products on the workstation platform: On our workstations, we use the following tools to develop and test our Web applications:

- VisualAge for Java Enterprise Edition V4.0
- WebSphere Studio V4.0.1
- WebSphere Studio Application Developer, EE V4.0.1
- WebSphere Studio Workload Simulator V1.0

Current Web application configurations and workloads

Figure 24 provides a high-level illustration of a typical WebSphere test environment on one of our z/OS images.

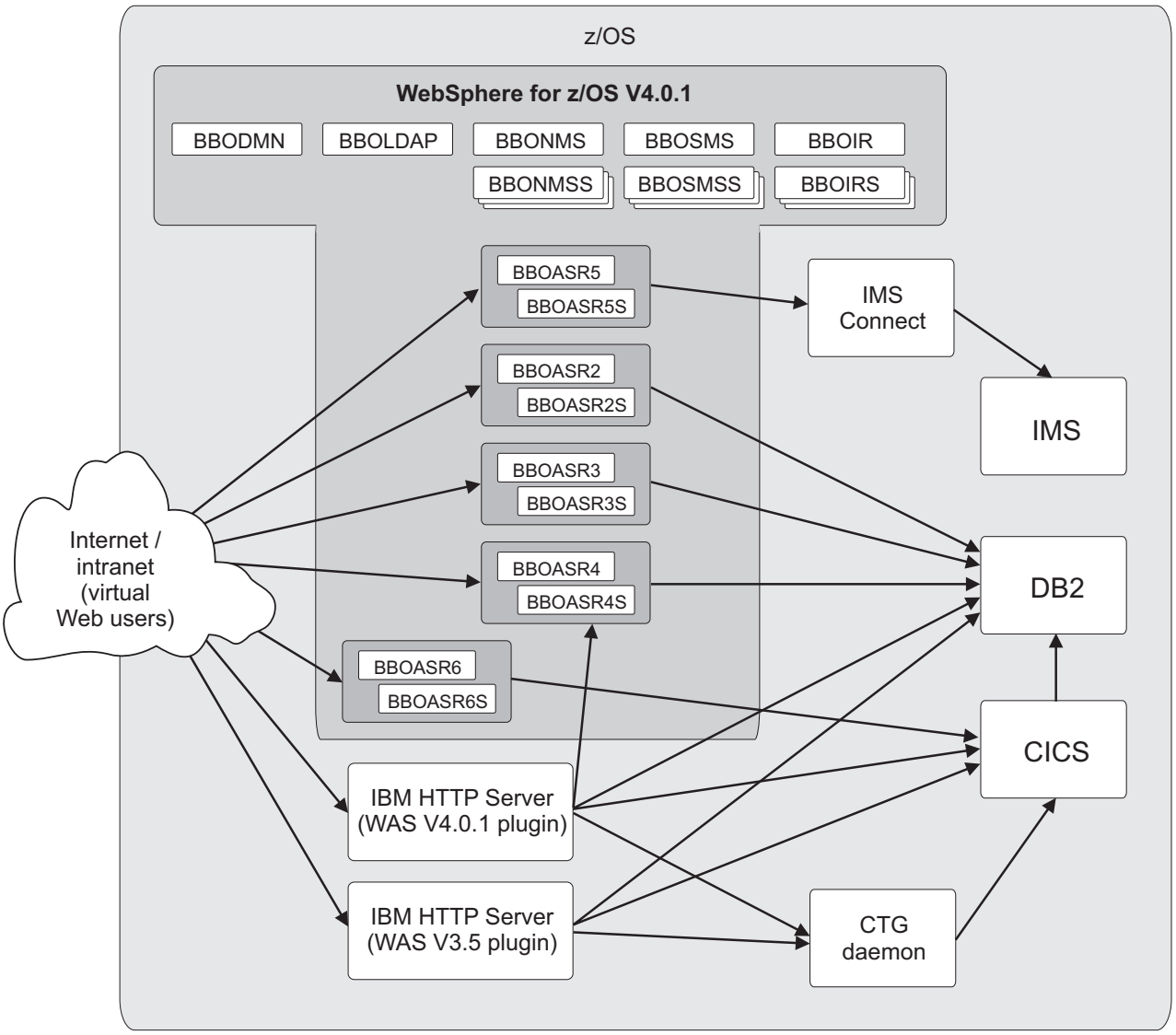


Figure 24. Typical WebSphere test environment on a single z/OS image

The following information describes the Web application configurations and workloads that we run in our test environment.

Web application configurations: We run a variety of configurations of WebSphere Application Server and Web applications in our environment. The major variations differ in terms of the level of WebSphere Application Server and how and where the Web applications are deployed.

These variations currently include the following:

- WebSphere for z/OS V4.0.1 J2EE servers, with direct access via HTTP Transport Handlers
- WebSphere for z/OS V4.0.1 plugin to IBM HTTP Server, with:
  - Web applications configured in the plugin, with:
    - No access to a J2EE server
    - Java Servlets and JavaServer Pages (JSP) files accessing Enterprise JavaBeans (EJB) components (or enterprise beans) in a J2EE server on the same system
    - Java Servlets and JSP files accessing enterprise beans in a J2EE server on a different system
  - Pass-through to Web applications configured in J2EE servers (nothing deployed in the plugin) that are:
    - On the same system
    - On a different system
- WebSphere SE V3.5 plugin to IBM HTTP Server

Web application workloads: We run a number of different Web application workloads in our test environment on z/OS. Generally, each workload drives HTTP requests to Web applications that consist of any combination of static content (such as HTML documents and images files), Java Servlets, JSP pages, and EJB beans. These Web applications use various connectors to access data in our DB2, CICS, or IMS subsystems.

Our Web application workloads currently include the following:

- J2EE applications (including persistent (CMP and BMP) and stateless session EJB beans) that:
  - Access DB2 using JDBC
  - Access CICS using the CICS Common Client Interface (CCI)
  - Access IMS using the IMS Connector for Java CCI
- Non-J2EE applications (only static resources, Servlets, and JSP pages) that:
  - Access DB2 using JDBC
  - Access CICS using CICS CTG
  - Access IMS using IMS Connect
- Other variations of the above applications, including those that:
  - Access secure HTTPS connections using SSL
  - Perform basic mode authentication
  - Use HTTP session data
  - Use connection pooling

Recent changes and updates to our WebSphere test environment

The following sections describe the recent changes and updates we have made to our WebSphere test environment along with our own particular experiences with implementing them.

Note: Over the past year, while we have migrated our Web applications to WebSphere for z/OS V4.0.1, we have also continued to test with and run our SE V3.5 environments and applications. However, be aware that service support for SE V3.5 ends in August, 2003. WebSphere for z/OS V5.0 is now generally available and applications currently running on SE V3.5 should be aggressively migrated to either V4.0.1 or V5.0.

WebSphere for z/OS V4.0.1 service updates

Currently, we are at the W401408 service level for WebSphere for z/OS V4.0.1.

WebSphere for z/OS V4.0.1 has been releasing service updates at a rapid pace and, as a result, we have progressed through numerous service level updates. Although it is sometimes difficult to keep up with the pace, in addition to providing problem fixes, the service updates continually provide new functionality, so we feel it's well worth the effort.

A listing of service levels for WebSphere for z/OS V4.0 and V4.0.1 is available in the WebSphere for z/OS support Web pages, available at www.ibm.com/software/webservers/appserv/zos_os390/support/.

The WebSphere publications are also periodically updated with new or changed information resulting from service updates. The most current publications are available in the WebSphere for z/OS library Web pages, available at www.ibm.com/software/webservers/appserv/zos_os390/library/.

Migrating to service level W401400: Of all the service updates we've performed, migrating to service level W401400 was a big step for us. This service level rolls up all of the changes from previous service levels, allowing you to update from any prior level of V4.0.1. However, this service level update can introduce a significant amount of change all at once, so it's important to carefully read all of the HOLDDATA, as some of these changes might require you to make other adjustments to your environment. For example, the default class loader mode for the J2EE servers changed from compatibility mode to application mode. For some of our applications, we needed to add a property to the server's jvm.properties in order to set the loader mode back to compatibility mode so that they could continue to run.

Allowing multiple service levels and rolling service updates: We made changes to our WebSphere for z/OS V4.0.1 setups to allow us to have multiple service levels within the sysplex and to enable us to perform rolling service updates to the various systems. To accomplish this, we made the following two significant changes:

- Added a qualifier to our V4.0.1 product data sets for each service level (for instance, WAS401.**PTF400**.SBB*)
- Implemented an alternate HFS structure for our V4.0.1 product service level HFSs

The additional data set qualifier, while not necessary, allows us to catalog all of the data sets for the various levels simultaneously and helps us to easily identify each. The alternate HFS structure is necessary to allow rolling service. *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, available on the WebSphere for z/OS library Web page, discusses these topics in several places:

- In the chapter on setting up your sysplex for a rolling upgrade
- In the chapter on installing new releases and maintenance levels
- In the appendix on using an alternate HFS structure for product upgrades

Using this setup whenever possible, we have successfully performed rolling service updates of several maintenance levels seamlessly across multiple systems. Our workloads that run in conjunction with Sysplex Distributor continued to run uninterrupted, with other systems handling the work while one system was being updated. (See "Using Sysplex Distributor with WebSphere for z/OS V4.0.1" on page 176 for additional information about how we run with Sysplex Distributor.)

Adding a z/OS.e image to our WebSphere test environment

Early in 2002, we added another system instance into our V4.0.1 setup, bringing the number of system instances in our WebSphere host cluster to four. (See the section on enabling WebSphere for z/OS V4.0 on a sysplex in our December 2002 edition for details on how we initially expanded to a multi-system setup on our sysplex.) The new system that we added (JH0) runs z/OS.e while our other three systems (J80, JB0, and JF0) continue to run z/OS. We are pleased to report that we have experienced no problems running our WebSphere setups in a mixed environment with z/OS and z/OS.e.

The following restrictions apply to a mixed host cluster containing both z/OS and z/OS.e systems:

- There is currently a restriction that the CICS CCI connectors for WebSphere can only connect in local mode. This means that both the application server and the target CICS subsystem must reside on the same z/OS image. As a result, we cannot run our V4.0.1 workloads that use CICS CCI connectors on our z/OS.e system. (For more information, see the section on using the CICS ECI connector in a J2EE server in WebSphere for z/OS V4.0.1 in our December 2002 edition.
- Although CICS and IMS subsystems themselves cannot run on z/OS.e systems, we can still run our SE V3.5 Web applications which use CTG and IMS Connect to access CICS and IMS subsystems on other z/OS systems in our sysplex.

Adding HTTP Transport Handlers to all J2EE servers

Many of our Web applications are now deployed entirely in J2EE servers. We have enabled the HTTP and HTTPS Transport Handlers for each of our V4.0.1 J2EE servers to allow a Web browser to directly access the components deployed in a J2EE server's Web container (such as Servlets, JSP pages, and static HTML documents). Most of our workloads now run by directly connecting to the J2EE servers.

We do continue to run some workloads that initially target an HTTP server configured with the V4.0.1 plugin. The application server plugin running in this HTTP server does the following:

- Runs Web applications (consisting of Servlets, JSP pages, and static HTML documents) that run entirely in the HTTP server's address space and do not access enterprise beans. These are applications that we initially migrated from an HTTP server using the SE V3.5 plugin to the V4.0 plugin.
- Runs Web applications configured in the plugin that access enterprise beans in J2EE servers. These are applications that we initially migrated to use enterprise beans.
- Redirects HTTP requests to Web containers in J2EE servers. These applications are fully migrated to our J2EE servers—that is, the Servlets, JSP pages, static HTML documents, and enterprise beans are deployed in the J2EE server.

Figure 25 on page 174 summarizes all of these modes of HTTP access to our Web applications:

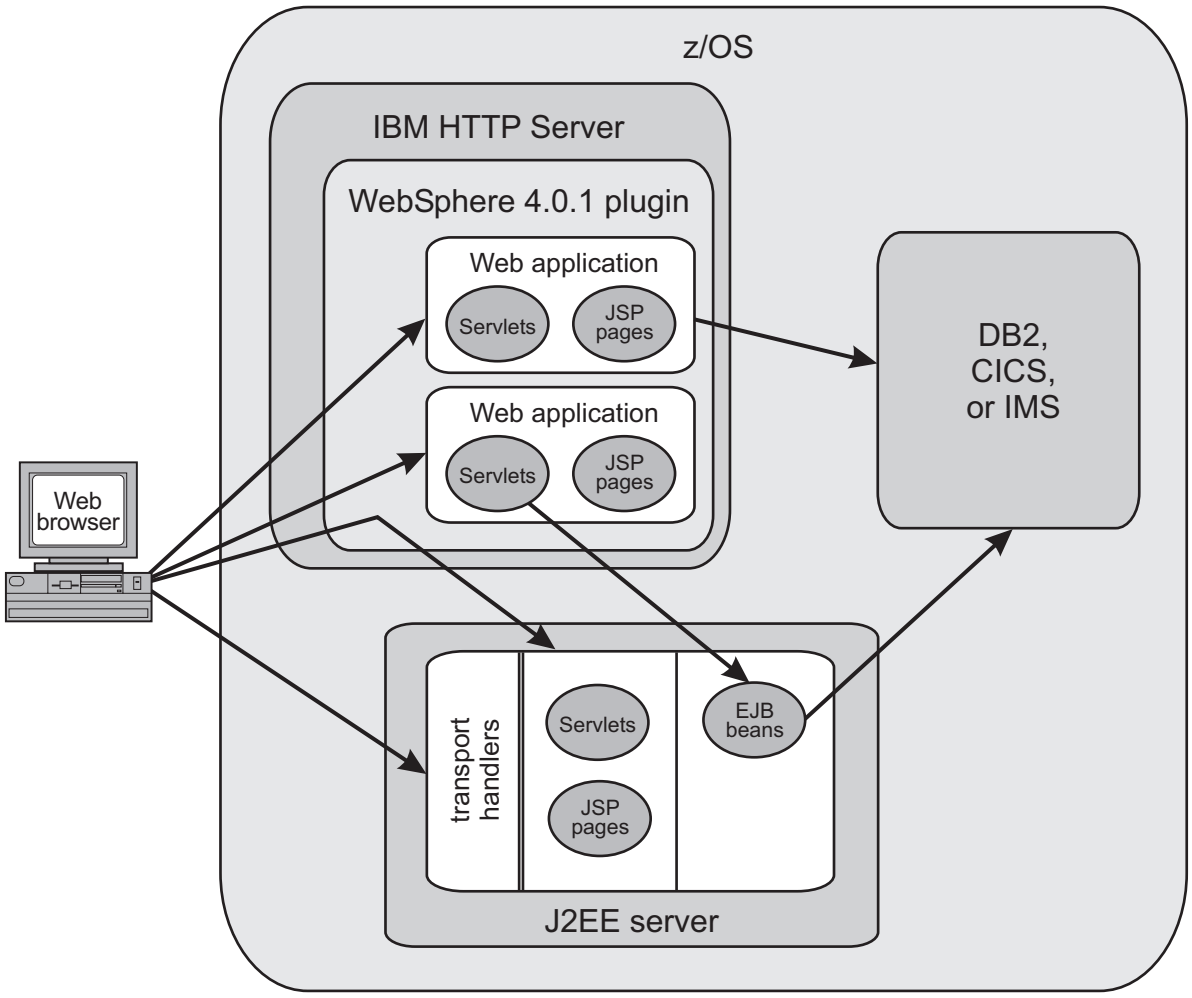


Figure 25. Summary of the modes of HTTP access to our Web applications

For complete information on enabling the HTTP and HTTPS Transport Handlers, see the section on the WebSphere environment for Web applications in the chapter on the J2EE server in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

For some excellent information on how to configure Web applications in a WebSphere for z/OS V4.0 or V4.0.1 environment, see the white paper, *WebSphere Application Server V4.0 and V4.0.1 for z/OS and OS/390: Configuring Web Applications*, WP100238, published by the Washington Systems Center and available on the IBM Techdocs Web site at www.ibm.com/support/techdocs/.

Using PROGxx members to dynamically load modules into the LPA

We now use PROG*xx* members in our MVS system parameter library (parmlib) that contain SETPROG LPA commands to dynamically load WebSphere V4.0.1 modules into the link pack area (LPA). The COMMND*xx* member invokes these PROG*xx* members to automatically load our V4.0.1 modules when we IPL the system. To conserve LPA storage, these PROG*xx* members are only invoked on the systems that are in our WebSphere host cluster.

Likewise, we use PROG*xx* members to do the following:

- APF authorize the SBBOLOAD, SBBOLD2, and SBBOLPA data sets

- Add the SBBOLOAD data set to the linklist (LNKLST*xx*)

We use two sets of PROG*xx* members: one for use during system initialization and the other for use during rolling service updates (with no system IPL) to dynamically add and delete modules and data sets for LPA, linklist, and APF authorizations.

Periodic recycling of J2EE servers

By default, the J2EE servers in WebSphere for z/OS V4.0.1 will recycle after processing 50,000 transactions. During this recycling, the address space for the J2EE server region is stopped and a new one is started. This allows for periodic release of system resources (such as memory and DB2 threads) and, for example, can help clean up after applications that exhibit problems during the initial development and test phases.

We found that, while running on J2EE servers that are configured to run with only a single server region per system, our Web applications were unavailable for up to one minute while a new server region was initializing.

To minimize the impact of these outages, you can use the V4.0.1 System Management EUI (SM EUI) to increase the number of transactions between server recycles. However, to eliminate these outages altogether, we found that either of the following methods work:

- Disable the server recycling on the J2EE server. We experienced no problems while running our J2EE servers in a single address space for extended periods of days at a time.
- Enable multiple server regions per system (via the workload management (WLM) policy). The MIN_SRS and MAX_SRS environment variables define the minimum and maximum number of address spaces, respectively.

On servers where our applications are stable, we prefer to disable the server recycling. We also prefer to use multiple server regions per system only for workload balancing purposes once an application has stabilized, not to maintain availability during server recycles.

BBOO_WORKLOAD_PROFILE environment variable

The V4.0.1 J2EE servers use an environment variable (BBOO_WORKLOAD_PROFILE) to determine workload-related processing decisions, such as the number of threads to use in a server region. The default value of this variable is NORMAL. We found that with the default setting, the throughput for each server region was lower than we desired, requiring us to run multiple server regions per system to handle heavier workload levels.

There are three other possible values for this variable: IOBOUND, CPUBOUND, or LONGWAIT. For IOBOUND and CPUBOUND, the number of available CPUs determines the number of server region threads. The server's job log displays the thread pool size that the control and server regions are actually using. In our case, we found desirable results by using the CPUBOUND setting for our J2EE servers. Each of the four systems in our V4.0.1 host cluster have different hardware platforms and varying amounts of resources available (for instance, system J80 runs on a G6 server while system JF0 runs on a z900 server). The thread pool size varies for server regions on each system.

You must use care when changing the setting of this variable in conjunction with running multiple server regions per system, as both actions can increase the demand for system resources and, when used together, may require that you make other adjustments to your system. For example, changing one of our J2EE servers

to use CPUBOUND caused both an increase in the number of worker threads used by each server region and an increase in the number of DB2 threads used by the server. When we later changed this server to use multiple address spaces per system, the total number of DB2 threads used by all of the server region address spaces increased proportionally.

Using Sysplex Distributor with WebSphere for z/OS V4.0.1

We have added the use of the Sysplex Distributor function into our V4.0.1 setups. As of the writing of this report, we have chosen to limit the use of this function to our non-session-based workloads. This is due to a restriction that requires a supported Web server and WebSphere plugin on a distributed (non-z/OS) platform to act as a front end for session-based HTTP requests. APAR PQ68250 lifts this restriction and provides an updated WebSphere plugin for IBM HTTP Server for z/OS.

When using the Sysplex Distributor, we experienced a problem when starting a server instance. In V4.0.1 J2EE servers, the control region for the server actually handles the incoming HTTP requests to the J2EE server and the server region handles the work. As soon as initialization of the control region was complete, it began accepting requests from the Sysplex Distributor. However, these requests were not actually processed until the associated server region completed initialization and was available for work. While the time between these two occurrences generally would be only a minute or so, it was long enough for the HTTP requests to time out. This effectively resulted in an outage of our Web site which, ironically, was exactly what we were trying to prevent by incorporating the Sysplex Distributor.

APAR PQ63711 provides a fix for this problem, along with some new environment variables to control when to start accepting HTTP requests. A new variable, BBOO_ACCEPT_HTTP_WORK_AFTER_MIN_SRS defines the number of server regions that must be ready for work before the server accepts HTTP requests. Once the specified number of server regions are ready, the HTTP/HTTPS Transport Handlers start accepting work. Additionally, the BBOO_ACCEPT_HTTP_WORK_AFTER_MIN_SECS variable defines the amount of time, in seconds, to wait for the desired number of server regions as defined by BBOO_ACCEPT_HTTP_WORK_AFTER_MIN_SRS before accepting work over the HTTP/HTTPS Transport Handlers. The default is 300 seconds and the minimum value is 10 seconds.

For more information on using and configuring the Sysplex Distributor with WebSphere, see the white paper, *Use of WebSphere for z/OS with Sysplex Distributor*, WP100312, published by the Washington Systems Center and available on the IBM Techdocs Web site.

Changing the default checking of the J2EE servers by the V4.0.1 plugin

The V4.0.1 plugin for the IBM HTTP Server for z/OS attempts to periodically communicate with the J2EE servers. In our test environment, all of our J2EE servers might not be up or available all of the time. When this happens, we receive BBOU516E error messages. Also, we generally configure our J2EE servers to handle requests only from certain HTTP servers with the V4.0.1 plugin (using the virtual host settings in the J2EE server's webcontainer.conf file).

To limit the set of J2EE servers with which the V4.0.1 plugin attempts to communicate, we added the following property to the configuration file for the plugin:

```
appserver.java.extraparm=-Dcom.ibm.ws390.wc.includedWebContainers=server1,server2,...
```

We also added the following properties to change the default frequency, in minutes, with which the V4.0.1 plugin checks for new J2EE servers and new Web applications on the J2EE servers:

```
appserver.java.extraparm=-Dcom.ibm.ws390.wc.serverCheckInterval=interval
appserver.java.extraparm=-Dcom.ibm.ws390.wc.webappupdateInterval=interval
```

For complete information about the use of these properties, see the appendix on using the alternate configuration option in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*.

Resolving SEC6 abends after migrating to z/OS V1R5

After migrating to z/OS V1R5, our HTTP servers configured with the V4.0.1 plugin began to experience numerous SEC6 abends. We discovered that the reason was that we did not have the following environmental variable set for the HTTP servers:

```
JAVA_PROPAGATE=NO
```

For details about the proper configuration of the HTTP server when using the alternate configuration option, see the appendix about the alternate configuration option in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

Where to find more information

During our testing, we used documentation from several sources, listed below. They contain all of the documents that we have cited throughout the course of this chapter.

- IBM CICS Transaction Gateway documentation, available at http://www.ibm.com/software/ts/cics/library/
- IBM HTTP Server for OS/390 documentation, available at http://www.ibm.com/software/websphere/httpservers/library.html
- IBM TechDocs (flashes, white papers, etc.), available at www.ibm.com/support/techdocs/
- IBM WebSphere Application Server for z/OS and OS/390 documentation, available at http://www.ibm.com/software/webservers/appserv/zos_os390/library/
- IBM WebSphere Studio documentation, available at http://www.ibm.com/software/websphere/studio/library.html
- IBM WebSphere Studio Workload Simulator documentation, available at www.ibm.com/software/awdtools/studioworkloadsimulator/library/
- *Java Servlet Specification, v2.2*, available at http://java.sun.com/products/servlet/
- *Java 2 Platform Enterprise Edition Specification, v1.2*, available at http://java.sun.com/products/j2ee/
- *JavaServer Pages Specification, Version 1.1*, available at http://java.sun.com/products/jsp/
- *J2EE Connector Architecture*, available at http://java.sun.com/j2ee/connector/

Migrating to WebSphere for z/OS V5.*X*

We recently completed the migration of our WebSphere workloads from V4.0.1 to V5.0. We still consider our production WebSphere environment to be V4.0.1, while we have the V5.0 product coexisting on the same systems.

Once our WebSphere for z/OS V5.0.2 migration was completed, we decided to proceed with the migration to WebSphere for z/OS V5.1.

Note: Please be aware that WebSphere for z/OS V4.0.1 is not supported on z/OS V1R6. You will have to migrate all of your V4.0.1 applications and servers to WebSphere for z/OS V5.*X* prior to migrating to V1R6.

About our migration to WebSphere for z/OS V5.X

Our overall experience so far

WebSphere for z/OS V5.0 is another major enhancement in the history of the product. As is often typical of such a change, the initial migration presents a fair amount of “pain” and complexity. Among the biggest changes in V5.0 are updates to the support levels of J2EE from 1.2 to 1.3, new terminology, a new topology, and a new administration interface.

Much of our migration effort has been spent on planning, educating ourselves, and getting familiar with the new version. The good news is that there is an abundance of information available. The bad news is that there is an abundance of information available—and you need to read it!

Since there is extensive documentation elsewhere, we will keep things brief here and point you to some of the more helpful documents. We list some useful Web sites at the end of this section. We have them bookmarked in our browsers and visit them frequently!

About our migration to WebSphere for z/OS V5.X

Our overall experience so far: We are pleased to report our migration to WebSphere Application Server V5.1 and JDK1.4.1 was successful. Overall, our migration from WebSphere V5.0.2 to V5.1 was very smooth and uneventful, perhaps our quickest migration so far. The two versions of WebSphere for z/OS version 5 remain very similar. Applications do not require any modifications and can remain unchanged between the two levels. However, there are significant performance gains using the newer level.

Current software products and release levels

The following information describes the software products and release levels that we use on the z/OS platform and on the workstation platform.

Note: WebSphere for z/OS V4.0.1 is not supported on z/OS V1R6. When migrating your base z/OS environment to V1R6, you must upgrade to WebSphere for z/OS V5.X. In addition, WebSphere for z/OS V5.1 requires the JDK level be at JDK1.4.1 to run properly.

Software products on the z/OS platform: In addition to the elements and features that are included in z/OS V1R6, our WebSphere test environment includes the following products:

- WebSphere for z/OS V5.1
- JDK1.4.1
- WebSphere for z/OS V5.0.2, service level W502002 (PTF UQ85128)
- WebSphere for z/OS V4.0.1, service level W401607 (PTF UQ83165)
- IBM SDK for z/OS, Java 2 Technology Edition V1.3.1 (February 2004 maintenance rollup, PTF UQ84703)
- WebSphere Studio Workload Simulator V1.0

- WebSphere MQ for z/OS V5.3.1
- DB2 V7.1
  - JDBC (PQ69861)
- CICS TS V2.2
  - CICS Transaction Gateway (CTG) V5.0.1
- IMS V7.1
  - IMS Connector V2.1 (with APAR PQ82862)

Software products on the workstation platform: On our workstations, we use the following tools to develop and test our Web applications:

- WebSphere Studio Application Developer, IE V5.0.1
- WebSphere Studio Workload Simulator V1.0

Specifics from our migration to WebSphere for z/OS V5.1

The migration from V5 to V5.1 proved to be smooth. The major difference between the releases is the JDK support. WebSphere for z/OS V5.0.2 requires JDK 1.3.1 to run properly, while WebSphere for z/OS V5.1 requires JDK 1.4.1 to run properly.

When migrating, we recommend using WebSphere for z/OS V5.1 service level W510004, which contains required fixes for migrating configurations from V5.0.2 to V5.1.

The technical experts at Washington Systems Center (WSC) have created a document with details on migrating from WebSphere for z/OS V5.0 to WebSphere for z/OS V5.1. This document is entitled **Migrating a Configuration from V5.0 to V5.1 — Overview, Explanation, Checklist, and Example** and is available at: *http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100441*.

RTLS warning messages WebSphere for z.OS V5.1 on z/OS V1R6: When running WebSphere for z/OS V5.1 on z/OS V1R6, you can expect to see the following messages in various customization jobs and server job logs:

```
CEE3608I The following messages pertain to the invocation command run-time
options.

CEE3611I The run-time option RTLS was an invalid run-time option or is not
supported in this release of Language Environment.
```

WebSphere for z/OS V5.1 specifies RTLS off, due to conflicts with RTLS and and XPLINK in z/OS V1R2, V1R3, V1R4, and V1R5. RTLS support was removed in z/OS V1R6 and the resulting error messages are due to the WebSphere for z/OS RTLS settings. These messages will not cause and failures and can be ignored.

zAAP projection tool

JDK 1.4.1 running with V1R6 on a z890 or z990 will provide support for zAAP processor exploitation. See Chapter 4, “Using zSeries Application Assist Processors (zAAPs),” on page 59 for detailed information on the benefits of zAAP.

Before you move to a zAAP supported environment you will want to do some capacity planning to determine how many zAAPs you will need. A zAAP Projection Tool, a modified Java SDK 1.3, is available that offers some of the same functionality which is incorporated into Java SDK 1.4 and higher. This tool gathers usage information about how much CPU time is spent executing Java code which could potentially run on zAAPs.

The following Washington System Center (WSC) Techdocs provide information on obtaining, installing, and using the zAAP Projection Tool:

- **Installation of the zAAP projection Tool Instrumented SDK in WebSphere for z/OS V5**

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100431
- **z/OS Performance: Capacity Planning Considerations for zAAP Processors**

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100417

Our current configuration and workloads

In our environment, WebSphere for z/OS V5.X coexists on systems with WebSphere for z/OS V4.0. In the V5.X setups on those systems, we use STEPLIBs in the JCL to point to the V5.0 libraries.

Our test and production configurations: Our current V5.0 setup contains two cells: T1 for our test system and P1 for our production systems. Both cells are configured as network deployment cells.

Our T1 cell is configured as follows:

- Resides entirely on one of our test systems (Z1)
- Contains six different J2EE servers, each running different applications (as described below)

Our P1 cell is the V5.0 equivalent to our V4.0.1 setup and is configured as follows:

- Spans four production systems in our sysplex (J80, JB0, JF0, and JH0)
- Contains five different clusters, each of which spans all four systems. Each cluster contains four J2EE servers—one J2EE server per system.
- Each cluster corresponds to one of the single J2EE servers in our T1 cell. Initially, we configure and deploy applications on a test J2EE server in the T1 cell and then deploy them to the corresponding server cluster in the P1 cell.

In addition, we set up a new test cell using WebSphere for z/OS V5.1 that is a replication of our production setups. This setup allowed for enhanced testing of a would be production environment.

Our Web application workloads: The following applications run in the J2EE servers on both our T1 and P1 cells:

- J2EE server 1 runs the the WebSphere V5.0 IVP and samples gallery applications. This server only resides on one system in our P1 cell; therefore, it is not contained within a cluster.
- J2EE server 2 runs our bookstore application, accessing DB2 and WebSphere MQ
- J2EE server 3 runs our trade3 application, accessing DB2 and WebSphere MQ
- J2EE server 4 runs our PETRTWDB2 application, accessing DB2
- J2EE server 5 runs our PETDSWIMS application, accessing IMS
- J2EE server 6 runs our PETNSTCICS application, accessing CICS

Figure 26 on page 181 shows the server address spaces in our P1 cell.

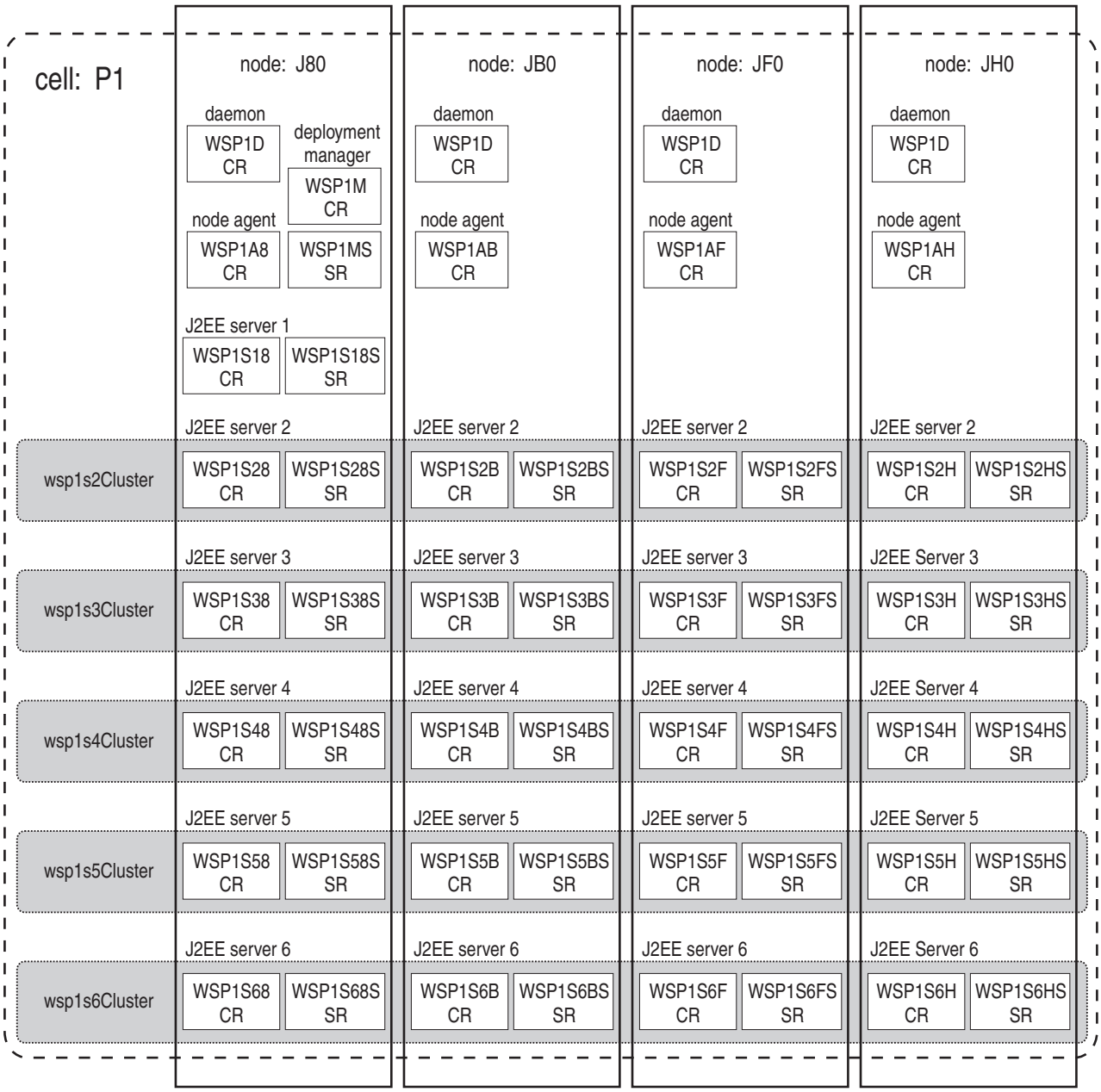


Figure 26. Our WebSphere for z/OS V5.0 configuration

About our naming conventions: After some experimentation, we settled upon a naming convention for our WebSphere setups. Our address space names are of the following format:

`WSccs[n]y[S]`

where:

WS The first two characters are always “WS” to identify a WebSphere resource.

cc Cell identifier:
- **T1** Test cell 1
- **P1** Production cell 1

s[*n*] Server type. For J2EE server control regions and server regions, *n* is the instance number of the server within the node:

A Node agent
D Daemon
M Deployment manager
S*n* J2EE server control region, instance *n*

y System identifier:
1 Z1 (test)
8 J80 (production)
B JB0 (production)
F JF0 (production)
H JH0 (production)

[S] Servant flag. This is appended to the name of a J2EE server control region to form the name of the associated servant region(s).

Example: The name WSP1S18S indicates a **W**eb**S**phere **p**roduction cell **1** J2EE server **s**erver region **1** on system J**8**0.

Server short names are specified in upper case. Server long names are the same as the short names, but are specified in lower case.

Specific documentation we used

Documentation to assist you with the usage of your product is available in many places. We have found that the Washington Systems Center documentation is very good and very often this same information is also in the information center. While we offer a set of generic links to documentation, see "Where to find more information" on page 184 for more information, we also wanted to take this opportunity to highlight the specific documentation we used and found especially useful.

For our current WebSphere for z/OS V5.1 configuration, we found the following technical documents were especially good at getting us up and running quickly:

- **WebSphere Variables to control WAS operator message routing**

  This article describes how to manage operator message routing in WebSphere for z/OS V5 and can be found at:
  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101116
- **DB2 UDB / JCC Connectors**

  This article describes how to enable WebSphere for z/OS V5.0.2 to use the DB2 Universal JDBC Driver and can be found at:
  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101663
- **WAS 5.0 Plugin for HTTP Server / Virtual Hosts / Sysplex Distributor**

  This article describes how to configure and troubleshoot the WebSphere for z/OS V5 HTTP Server and can be found at:
  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS829

Running WebSphere for z/OS V5.0 and V5.1 on the same system

When running WebSphere for z/OS V5.0 and V5.1 on the same system, only one level of the run-time modules can be loaded into LPA. The other level must have STEPLIBs to the proper version of modules. However, WebSphere for z/OS specifies the `BBORTSS5` module must be loaded into LPA regardless of whether the other modules are loaded into LPA or specified in STEPLIBs.

If you are looking to run with both levels on the same system, you should load the V5.1 `BBORTSS5` into LPA. The level is compatible with both version of the product.

Changes and updates to our WebSphere environment for V5.X

The following are some of the changes and updates that we've needed to make as we continue our migration to WebSphere for z/OS V5.0.

Topology and configuration changes for V5.X

Some of the biggest changes for V5.0 are the application server topology and configuration. They're also among the first things that need to be addressed in the planning and setup of the V5.0 environment. Be prepared to repeat the setup activity more than once as you get the hang of it and iron out the details. After doing it once or twice, it starts to make more sense and becomes much easier.

WebSphere for z/OS V5.0 has two major modes: base application server node and network deployment (ND) node. While your particular setup may not require the network deployment, we strongly suggest that you plan for it. The first time around, it was a daunting task for us just to get the first base application server node set up and running. However, when we tried to set up the deployment manger node and then "federate" the base server into it, we found holes in the naming convention that we had been using.

We highly recommend the white paper *Washington Systems Center Sample WebSphere for z/OS ND Configuration*, WP100367, which provides an excellent description of this process as well as a step-by-step sample to help get you through. This white paper is available on the IBM Techdocs Web site at www.ibm.com/support/techdocs/.

Administration console

The WebSphere for z/OS System Management End User Interface (SMEUI) has been replaced by a Web application (accessed via a Web browser) that closely resembles the version available on the distributed platform. While it took some getting used to, we've found that, overall, we like using it. Like anything new, it does take some time to find your way around.

One of the biggest changes to get used to is the scoping level. Many of the entries can be set for the cell, node, or server level of scoping. Be aware of the level that is currently set for the display on the admin console. Also, items that are assigned at higher levels can be overridden at lower levels (cell is highest, server is lowest). While this is very handy, try to set the scoping at the highest level you can as definitions at the server level can be easily overlooked.

WebSphere for z/OS V5.0 service updates

Currently, we are at the W502002 service level for WebSphere for z/OS. We strongly suggest that you obtain and maintain the latest service levels of the product. Many of the problems that we encountered were already resolved by service updates by the time we hit them.

We also have structured our setups to allow for rolling service and multiple levels of the application server, Java, and MQ products within the sysplex. To accomplish this, we employed techniques similar to our V4.0.1 setups. See our December 2003 edition for further details. For V5.0, this is well documented in *WebSphere Application Server for z/OS V5.0: Getting Started*, GA22-7957.

The white paper *WebSphere Application Server for z/OS V5 - Planning for Test, Production and Maintenance*, WP100396, provides some very good pointers, considerations, and sample configurations that allow for various service strategies. This white paper is available on the IBM Techdocs Web site at www.ibm.com/support/techdocs/.

Updating our WebSphere applications

We did not use the migration tools provided in V5.0 to migrate our applications. Instead, we decided to re-work our applications as we were migrating. This allowed us to exploit some of the new functionality provided by V5.0's full support of the J2EE 1.3 specification levels.

To do this, we employed WebSphere Studio Application Developer, Integration Edition (WSADIE), Version 5.0.1. All of our applications were previously migrated to WSADIE 4.0.1. Thus, we were able to export these applications from WSADIE 4.0.1 and then import them into WSADIE 5.0.1. We then performed some minor updates using WSADIE's migration tools. This allowed us to quickly update our applications.

We're very happy to report that WSADIE V5.0 and WebSphere for z/OS V5.0 have become tightly integrated. We were able to fully test all applications within the WebSphere Application Server Test Environment in WSADIE, directly accessing our resources (such as DB2, CICS, IMS, and MQ) on our z/OS test systems. Once tested, we deployed the applications directly to z/OS.

JDBC Connector setup for DB2

In order to use DB2 V7.1 with WebSphere for z/OS V5.0, you need to apply APAR PQ69861 for DB2. This provides a new mechanism to allow the JDBC driver to locate and load the DSNJDBC_JDBCProfile.ser file.

See *Using DB2 for z/OS in WebSphere for z/OS Version 5*, TD101072, for a step-by-step guide to assist you in the proper setup for using DB2 with WebSphere for z/OS V5.0. This white paper is available on the IBM Techdocs Web site at www.ibm.com/support/techdocs/.

CICS Transaction Gateway Connector V5.0.1 in local mode

In order to run your CICS Transaction Gateway (CTG) J2C connectors in local mode (that is, where the application server and CICS server run on the same z/OS system), you need to extract the libCTGJNI.so library from the CICS CTG 5.0.1 install directory.

See *Connecting to CICS Transaction Server from WebSphere for z/OS Version 5*, WP100375, for a step-by-step guide to assist you in this process. This white paper is available on the IBM Techdocs Web site at www.ibm.com/support/techdocs/.

IMS Connector V2.1 in local mode

As with our CICS Transaction Gateway Connector, we ran into problems when we tried to run our IMS Connector in local mode. APAR PQ82862 for IMS Connect V2.1 resolves the problem.

Where to find more information

During our testing, we used documentation from several sources, listed below. They contain all of the documents that we have cited throughout the course of this chapter.

- IBM WebSphere Application Server for z/OS and OS/390 documentation, available at http://www.ibm.com/software/webservers/appserv/zos_os390/library/
- IBM WebSphere Application Server Version 5.*X* Information Center, available at publib.boulder.ibm.com/infocenter/wasinfo/index.jsp
- IBM Techdocs (flashes, white papers, etc.), available at www.ibm.com/support/techdocs/
- *Java 2 Platform Enterprise Edition Specification, v1.2*, available at http://java.sun.com/products/j2ee/

Chapter 13. Using EIM authentication

Enterprise Identity Mapping (EIM) is an IBM @server infrastructure architecture that defines a set of services and extensions to LDAP to transform the user identity associated with a work request as it moves between systems having different user administration schemes as part of a multi-tiered application in a heterogeneous environment. EIM offers a new approach to easily manage multiple user registries and user identities in an enterprise by providing an architecture for describing the relationships between entities (such as individual users and system resources) in the enterprise and the many identities that represent them.

In z/OS V1R5, EIM has added support for the following bind types:

- Client authentication using a digital certificate over an SSL connection
- Kerberos authentication for clients and servers using a trusted third party protocol
- CRAM-MD5 password protection using a hashed password

The following sections describe our experiences deploying each bind type.

Client authentication using digital certificates

We used information from the following sources to help us set up and test EIM with client authentication using digital certificates:

- *z/OS Integrated Security Services EIM Guide and Reference*, SA22-7875
- *z/OS Integrated Security Services LDAP Server Administration and Use*, SC24-5923
- *z/OS Integrated Security Services LDAP Client Programming*, SC24-5924
- *z/OS HTTP Server Planning, Installing, and Using*, SC34-4826
- *z/OS Crytographic Services System Secure Sockets Layer Programming*, SC24-5901

We used the System Secure Sockets Layer (SSL) gskkyman utility to generate our key databases and certificates. We know from past experience that *z/OS HTTP Server Planning, Installing, and Using* does an excellent job of explaining how to generate certificates. We used that document to assist in the generation of the certificates for EIM/LDAP, even though the HTTP server was not involved. (Basically, a certificate is a certificate, regardless of the application with which it will be used.) We then used the EIM and LDAP documentation to assist in the implementation and use of the certificates.

Resolving problems during our testing

EIM domain name missing the country attribute: The EIM domain that we had established in our previous testing did not have a country attribute in its domain name. The EIM domain name is the full distinguished name (DN) of the EIM domain. The gskkyman utility requires a domain name to contain, at a minimum, common name (cn), organization (o), and country (c). We created a new EIM domain name containing the required attributes so that our client certificates would have associated entries in the EIM domain controller. It is critical not only to insure that the domain name contains the minimum required attributes, but also that the domain name in the EIM domain controller matches the domain name in the client certificate.

Using the documented example for creating LDAP suffix and user objects: To set up a new suffix, we used the "Example for creating LDAP suffix and user objects" in *z/OS Integrated Security Services EIM Guide and Reference*. If you use this example, it is important to make sure that a blank line exists between the object entries in the sample ldif file. The blank line is what signals the end of one entry and the beginning of the next. When we created the ldif file, the blank lines were missing and, thus, the **ldapadd** command failed.

We also did not include the entry that defines the country object (c=us). This is because our slapd.conf file only defines a suffix for both the organization and country (o=ibm,c=us). If we had included the entry for the country object as in the example, we would have had to add that suffix to our slapd.conf file. This illustrates the importance of understanding the structure of the data in your directory and how it is processed.

Testing the client authentication using digital certificates

Enabling the EIM domain controller for SSL processing: We followed the instructions in the LDAP Server documentation to enable our EIM domain controller for SSL processing. We also used the HTTP Server documentation to generate the key database and digital certificates.

Testing the client certificate: We issued the following **ldapsearch** command to verify that SSL processing was working properly:

```
ldapsearch  -h ip_address  -S EXTERNAL  -Z  -l 689  -K client_key_database
     -P client_key_database_password  -V 3  -p 689  -s base  -b "" "objectclass=*"
```

This worked successfully.

We then issued the following eimadmin command using a client certificate:

```
eimadmin  -lD  -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'  -h ldaps://ip_address:689
     -K client_key_database  -P client_key_database_password  -S EXTERNAL
```

This also worked successfully.

Again, it is important to understand the structure of your data. Also, because the command syntax is long and complex, it is easy to make a mistake and it's not always clear from the error messages what is wrong.

Kerberos authentication

We used information from the following sources to help us set up and test Kerberos authentication:

- *z/OS Integrated Security Services EIM Guide and Reference*, SA22-7875
- *z/OS Integrated Security Services LDAP Server Administration and Use*, SC24-5923
- *z/OS Integrated Security Services LDAP Client Programming*, SC24-5924
- *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926

We heavily relied on the LDAP documentation to set up the environment to allow EIM to bind using Kerberos. There are many different ways to set up this environment—what we describe here is but one way.

Clearing up a documentation inaccuracy

In the chapter on EIM APIs in *z/OS Integrated Security Services EIM Guide and Reference*, it states in three different places that, “To connect to an EIM domain using Kerberos information, you need to do so from a non-z/OS platform.” This statement occurs under the descriptions of the eimListUserAccess, eimQueryAccess, and eimRemoveAccess APIs.

The functionality to connect to an EIM domain from a z/OS platform using Kerberos information is available in z/OS V1R5. The above statement is left over from the previous release and, unfortunately, was not removed from the documentation in time for the general availability (GA) of z/OS V1R5. However, the statement will be removed from the next release of the documentation.

Testing the Kerberos authentication

Enabling the EIM domain controller for Kerberos processing: We followed the instructions in the LDAP Server documentation to enable our EIM domain controller for Kerberos processing. In addition, refer to our December 2002 edition for more information on enabling LDAP Server for use with Kerberos authentication.

Testing the EIM bind with Kerberos authentication: We used the **eimadmin** command for our testing. The EIM documentation describes the command syntax and how to bind using Kerberos authentication.

Before we could issue the **eimadmin** command, we first had to obtain a Kerberos ticket by issuing the **kinit** command, as follows:

```
kinit kerberos_principal
```

Once we had the Kerberos ticket, we issued the following **eimadmin** command specifying that we wanted to bind with Kerberos authentication:

```
eimadmin  -lR  -d 'ibm-eimDomainName=Domain,o=org,c=us'  -r 'RACF SSL'
     -h ldap://ip_address  -S GSSAPI
```

Note that the **eimadmin** command does not have any bind credentials. The `-S GSSAPI` specifies to use the Kerberos ticket obtained from the **kinit** command. However, unless something is done, the Kerberos principal identified in the ticket does not have authorization to issue EIM commands. The EIM documentation does not specify how to do this. However, there are several ways to identify or associate the Kerberos principal to IDs (or, LDAP DNs) that are already authorized to issue EIM commands. We’ll describe one method that we used.

Using TDBM mapping to associate a Kerberos principal to an EIM-authorized ID: We chose to use TDBM mapping to associate the Kerberos principal to an EIM-authorized DN, as described in *z/OS Integrated Security Services LDAP Server Administration and Use* under the “Identity mapping” section of the chapter on “Kerberos authentication”. To do this, we created an ldif file, kerberos.ldif, containing the following:

```
dn: cn=eim ssl administrator,o=eimssl,c=us
changetype:modify
add:x
objectclass: ibm-securityIdentities
altSecurityIdentities: KERBEROS:principal@REALM
```

We then issued the following ldapmodify command to update the LDAP DN:

```
ldapmodify  -h ip_address  -D "cn=LDAP Administrator"  -w password
     -f /etc/ldap/kerberos.ldif
```

We were then able to issue **eimadmin** commands binding with Kerberos authentication. However, note that the Kerberos principal will only be able to issue those EIM commands for which its associated LDAP DN is authorized.

CRAM-MD5 password protection

We used information from *z/OS Integrated Security Services EIM Guide and Reference*, SA22-7875 to help us test CRAM-MD5 password protection:

No work is needed to set up for using CRAM-MD5 binds. All we needed to do was issue the **eimadmin** command and specify the CRAM-MD5 bind option. We did not encounter any problems using the CRAM-MD5 bind.

The following are a couple of examples of using the **eimadmin** command with the CRAM-MD5 bind option:

Example: We used the following command to list a domain:

```
eimadmin  -lD  -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'  -h ldap://ip_address:389
     -b 'cn=eim ssl administrator,o=eimssl,c=us'  -w password  -S CRAM-MD5
```

Example: We used the following command to list a registry:

```
eimadmin  -lR  -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'  -r 'RACF SSL'  -h ldap://ip_address:389
     -b 'cn=eim ssl administrator,o=eimssl,c=us'  -w password  -S CRAM-MD5
```

EIM enhancements in z/OS V1R6

In z/OS V1R6 the following EIM enhancements were tested:

- "x.509 certificate registries"

We used the following documentation to help us plan and implement these enhancements:

- *z/OS Integrated Security Services EIM Guide and Reference*
- *z/OS Crytographic Services System Secure Sockets Layer Programming*

x.509 certificate registries

First, we created an x.509 certificate registry. We used the **eimadmin** command for all of our testing.

Example: We used the following command to create the x.509 registry:

```
eimadmin -aR -h ldap://<ip address>:389 -b 'cn=EIM Administrator'
-w <password> -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'
-r 'Cert Maps' -y X509 -n 'Registry for Certificates'
```

Example: We used the following command to list the registry for verification:

```
eimadmin -lR -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'
-r 'Cert Maps' -h ldap://<ip address>:389 -b 'cn=EIM Administrator'
-w <password>
```

Result:

```
source registry: Cert Maps
  registry kind: SYSTEM
  registry type: X509
    description: Registry for Certificates
        lookups: ENABLED
       policies: DISABLED
```

Testing associations

We used the following procedures to create an association using the name stored within a certificate:

Obtain the certificate in a file:

We used the System Secure Sockets Layer (SSL) gskkyman utility to retrieve an existing client certificate from an existing kdb database.

1. Issued the gskkyman command
2. Select the kdb
3. Select option 1 for Manage keys and certificates
4. Select option 'n' for the certificate to export
5. Select option 6 for Export certificate to a file
6. Select option 2 for Base64 ASN.1 DER

Example: After the certificate is in a file, we entered the EIM command to add an association from a certificate:

```
eimadmin -aA -h ldap://<ip address>:389 -b 'cn=EIM Administrator' -w <password>
-d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us' -r 'Cert Maps'
-i "eim ssl administrator" -E <certificate file name> -t admin
```

Result: This failed with the following error message because we did not create the identifier:

```
ITY4030 Service eimAddAssociation() returned error 248
ITY0025 EIM identifier not found or insufficient access to EIM data.
```

Example: Create the identifier before the association. We used the following command to create the identifier:

```
eimadmin -aI -i 'eim ssl administrator' -n 'Identifier for eim ssl admin'
-d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'  -h ldap://<ip address>:389
-b 'cn=EIM Administrator' -w <password>
```

Next, we verified by listing the identifier:

```
eimadmin -lI -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'
-i 'eim ssl administrator'  -h ldap://<ip address>:389
-b 'cn=EIM Administrator' -w <password>
```

Result:

```
            unique identifier: eim ssl administrator
             other identifier: eim ssl administrator
                  description: Identifier for eim ssl admin
```

This worked successfully. We reissued the association command. This also worked successfully.

Note: If an identifier is used that already exists, you would not see the error on the first issuance of the command and the addition of the identifier would not be required.

Example: We listed the association for validation:

```
eimadmin -lA -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'
-i 'eim ssl administrator' -h ldap://<ip address>:389
-b 'cn=EIM Administrator' -w <password>
```

Result: That successfully created an association from a certificate located in a file.

```
            unique identifier: eim ssl administrator
                  association: ADMIN
              source registry: Cert Maps
                registry type: X509
                registry user: <SDN>CN=EIMSSLADMINISTRATOR,O=EIMSSL,C=US</SDN>
                               <IDN>CN=AE TEAM CA FOR JA0,OU=INTEGRATION TEST,
                               O=AE TEAM,L=POK,ST=NY,C=US</IDN> <HASH_VAL>CF752E
                               7699A95818799E8AC70CD6A9F8BCA0B35D</HASH_VAL>
```

Removing an association using the name stored within a certificate:
Example: We removed the association previously created, by issuing the following command:

```
eimadmin -pA -h ldap://<ip address>:389 -b 'cn=EIM Administrator'
-w <password> -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'
-r 'Cert Maps' -i 'eim ssl administrator' -E <certificate file name>-t admin
```

Example: Next, we issued the list command to verify that the association does not exist:

```
eimadmin -lA -d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us'
-i 'eim ssl administrator' -h ldap://<ip address>:389 -b
'cn=EIM Administrator' -w <password>
```

Result: The association was removed.

Removing an association using the -u flag: **Example:** We issued the following command to remove an association using the -u flag:

```
eimadmin -pA -h ldap://<ip address>:389 -b 'cn=EIM Administrator' -w <password>
-d 'ibm-eimDomainName=SSL Domain,o=eimssl,c=us' -r 'Cert Maps' -i 'eim ssl
administrator' -u '<SDN>CN=EIM SSL ADMINISTRATOR,O=EIMSSL,C=US</SDN>
<IDN>CN=AE TEAM CA FOR JA0, OU=INTEGRATION TEST,O=AE TEAM,L=POK,ST=NY,C=US</IDN>
<HASH_VAL>CF752E7699A95818799E8AC70CD6A9F8BCA0B35D</HASH_VAL>'  -t admin
```

Result: The removal was successful.

Testing Filtering

Next, we tested the x.509 certificate filtering.

Example: We issued the following command to create a filter policy:

```
eimadmin -aY -h ldap://<ip address>:389 -b 'cn=EIM Administrator'
-w <password> -d 'ibm-eimDomainName=Small Domain,o=EIM'
-r 'Cert Maps' -J 'O=EIMSSL,C=US' -F 'OU=INTEGRATION TEST,
O=AE TEAM,L=POK,ST=NY,C=US' -T 'RACF Insco' -u C00002 -t filter
```

Example: We issued the list filter command to validate the add filter command:

```
eimadmin -pY -h ldap:// ://<ip address>:389 -b 'cn=EIM Administrator' -w <password>
-d 'ibm-eimDomainName=Small Domain,o=EIM' -r 'Cert Maps' -J 'O=EIMSSL,C=US'
-F 'OU=INTEGRATION TEST,O=AE TEAM,L=POK,ST=NY,C=US' -t filter
```

Result: The following is returned:

```
              source registry: Cert Maps
                  policy type: FILTER
                       filter: <SDN>O=EIMSSL,C=US</SDN><IDN>OU=INTEGRATION
```

```
            TEST,O=AE TEAM,L=POK,ST=NY,C=US</IDN>
               target registry: RACF Insco
          target registry user: C00002
               domain policies: DISABLED
       source registry lookups: ENABLED
       target registry lookups: ENABLED
      target registry policies: DISABLED
```

Example: We issued the following command to delete the filter policy:

```
eimadmin -pY -h ldap://<ip address>:389 -b 'cn=EIM Administrator'
-w <password> -d 'ibm-eimDomainName=Small Domain,o=EIM' -r 'Cert Maps'
-J 'O=EIMSSL,C=US' -F 'OU=INTEGRATION TEST,O=AE TEAM,L=POK,ST=NY,C=US'
-T 'RACF Insco' -u C00002 -t filter
```

Result: The filter policy was successfully removed.

Create an x.509 certificate filter policy using a certificate

Example: We issued the following command to create a certificate filter policy:

```
eimadmin -aY -h ldap://<ip address>:389 -b 'cn=EIM Administrator'
-w <password> -d "ibm-eimDomainName=Small Domain,o=EIM" -r 'Cert Maps'
-G <certificate file name> -J O= -F OU=  -T 'RACF Insco' -u C00002 -t filter
```

Example: Next, we issued the list filter command to verify using the certificate as the list criteria:

```
eimadmin -lY -h ldap://<ip address>:389 -b "cn=EIM Administrator"
-w <password> -d 'ibm-eimDomainName=Small Domain,o=EIM'
-r 'Cert Maps' -G <certificate file name> -J O= -F OU= -t filter
```

Result: Here is what returned:

```
source registry: Cert Maps
                    policy type: FILTER
                         filter:
<SDN>=EIMSSL,C=US</SDN><IDN>OU=INTEGRATION TEST,
O=AE TEAM,L=POK,ST=NY,C=US</IDN>
                target registry: RACF Insco
           target registry user: C00002
                domain policies: DISABLED
       source registry lookups: ENABLED
       target registry lookups: ENABLED
      target registry policies: DISABLED
```

Example: To delete the filter policy using the certificate as the delete criteria, we issued the following command:

```
eimadmin -pY -h ldap://<ip address>:389 -b "cn=EIM Administrator"
-w <password> -d "ibm-eimDomainName=Small Domain,o=EIM" -r 'Cert Maps'
-T 'RACF Insco' -u C00002 -G <certificate file name> -J O= -F OU= -t filter
```

Result: The list filter was issued and nothing was returned.

Appendix A. Some of our parmlib members

This section describes how we have set up some of our parmlib members for z/OS. Table 13 summarizes our new and changed parmlib members for z/OS V1R5 and z/OS.e V1R5. Samples of some of our parmlib members are available on the Samples page of our Web site.

Table 13. Summary of our parmlib changes for z/OS V1R5 and z/OS.e V1R5

Member name	z/OS release	Change summary	Related to
IFAPRD*xx*	z/OS.e V1R5	No changes from z/OS.e V1R3. (See our December 2002 edition.)	z/OS.e, dynamic enablement
IEASYMPT	z/OS.e V1R5	No changes from z/OS.e V1R3. (See our December 2002 edition.)	z/OS.e, dynamic enablement
IEASYS*xx*	z/OS.e V1R5	No changes from z/OS.e V1R3. (See our December 2002 edition.)	z/OS.e
LOAD*xx* (Actually in SYS0.IPLPARM. See note below.)	z/OS V1R5	Changed the PARMLIB statement to use SYS1.PETR15.PARMLIB	concatenated parmlib
	z/OS.e V1R5	No changes from z/OS.e V1R3. (See our December 2002 edition.)	z/OS.e
LPALST*xx*	z/OS V1R5	Added: SYS1.SIATLPA	JES3
PROG*xx* (APF additions)	z/OS V1R5	Added: SYS1.SHASLINK SYS1.SHASMIG	JES2
PROG*yy* (LNKLST)	z/OS V1R5	Added to LNKLST*xx*: SYS1.SHASLINK SYS1.SHASMIG	JES2
		Added to LNKLST*xx*: SYS1.SIATLIB SYS1.SIATLINK SYS1.SIATMIG	JES3

Note: As of our OS/390 R6 testing, we changed LOAD*xx* to use a generic name, IEASYMPT, for our IEASYM*xx* member. We have successfully used the name IEASYMPT for our migrations through all subsequent releases of OS/390 and z/OS. Only the entries in SYS0.IPLPARM changed.

Appendix B. Some of our RMF reports

In this appendix we include some of our RMF reports, as indicated in “z/OS performance” on page 46.

RMF Monitor I post processor summary report

The following figure contains information from our *RMF Monitor I Post Processor Summary Report*. Some of the information we focus on in this report includes CP (CPU) busy percentages and I/O (DASD) rates. This report contains information for the same date and time interval as the report in Figure 29 on page 197.

```
                          R M F   S U M M A R Y   R E P O R T

OS/390                      SYSTEM ID JA0             START 01/18/2001-10.00.00  INTERVAL 00.30.00
  REL. 02.10.00               RPT VERSION 02.10.00      END   01/18/2001-10.30.00  CYCLE 0.100 SECONDS

    INT   CPU   DASD  DASD  TAPE   JOB   JOB   TSO   TSO   STC   STC  ASCH  ASCH  OMVS  OMVS SWAP DEMAND
   MM.SS  BUSY  RESP  RATE  RATE   MAX   AVE   MAX   AVE   MAX   AVE   MAX   AVE   MAX   AVE RATE PAGING
        OS/390                    SYSTEM ID JA0
   30.00  52.0    16  78.6   0.0     0     0     0     0   283   282     0     0     3     3 0.00   0.00
        OS/390                    SYSTEM ID JB0
   30.00  55.9    14  89.9   0.0   113   113     0     0   170   169     0     0     2     2 0.00   0.01
        OS/390                    SYSTEM ID J80
   30.00  36.5    16 907.7   0.0     1     1     6     6   300   292     0     0     3     3 0.00   0.01
        OS390                     SYSTEM ID JC0
   29.59  25.3    11 151.0   0.0     0     0     0     0   280   279     0     0     1     1 0.00   0.22
        OS/390                    SYSTEM ID JE0
   29.59  26.0    13 142.1   0.0     0     0     0     0   280   278     0     0     3     3 0.00   0.12
        z/OS  V1R1                SYSTEM ID Z2
   30.00  13.3    22  33.1         0     0     0     0   264   263     0     0     4     4 0.00   0.00
        z/OS  V1R1                SYSTEM ID Z3
   30.00  13.8    21  33.5         0     0     0     0   273   271     1     0     7     5 0.00   0.02
        z/OS  V1R1                SYSTEM ID TPN
   29.59  67.8     5  74.5   0.0     0     0     1     1   277   275     0     0     2     2 0.00   0.00
        z/OS  V1R1                SYSTEM ID JG0
   30.00  23.8    11 387.2   0.0   113   113     3     3   175   173     1     0     8     4 0.00   0.00
        z/OS  V1R1                SYSTEM ID JH0
   29.59  30.7    17 610.1   0.0     0     0     2     2   282   280     0     0     2     2 0.00   0.06
        z/OS  V1R1                SYSTEM ID J90
   30.00  25.1     9 641.4   0.0   115   113     6     4   181   176     1     0    13     6 0.00   0.42
        z/OS  V1R1                SYSTEM ID JF0
   29.59  35.9    12  1022   0.0     0     0     0     0   289   287     0     0     2     2 0.00   0.25
        z/OS  V1R1                SYSTEM ID Z0
   30.00  31.5     9  1347   0.0   114   113    10     9   325   312     1     0    42    35 0.00   0.10
```

Figure 27. Example RMF Monitor I post processor summary report

RMF Monitor III online sysplex summary report

The following figure contains information from the RMF *Monitor III Online Sysplex Summary Report*. This is a real-time report available if you are running WLM in goal mode. We highlighted some of our goals and actuals for various service classes and workloads. At the time this report was captured we were running 1130 CICS transactions/second. Note that this report is a snapshot, as opposed to Figure 29 on page 197, which is based on a 1/2-hour interval.

```
         HARDCOPY  RMF 2.10.0 Sysplex Summary - UTCPLXJ8       Line

WLM Samples: 479     Systems: 14 Date: 01/18/01 Time: 10.05.00 Range: 1

                      >>>>>>>>XXXXXXXXXXXXXXXXXX<<<<<<<<

Service Definition: WLMDEF01              Installed at: 12/07/00, 09.35
     Active Policy: WLMPOL01              Activated at: 12/07/00, 09.50
               ------- Goals versus Actuals --------  Trans --Avg. Resp
               Exec Vel  --- Response Time ---  Perf  Ended  WAIT EXECU
Name     T  I  Goal Act  ---Goal--- --Actual--  Indx  Rate   Time   Tim
BATCH    W          83                              0.025 1.076  59.8
BATCHHI  S  2    50  83                      0.60  0.000
DISCR    S  D        88                              0.025 1.076  59.8
CICS     W          N/A                               1130  0.000  0.08
CICS     S  2       N/A   0.600 80%          99%   0.50  961.9 0.000  0.08
CICSCONV S  3       N/A  10.00 50%          51%  1.00  5.883 0.000  13.4
CICSCPSM S  1       N/A  0.500 90%         100%  0.50  1.325 0.000  0.00
CICSDEFA S  3       N/A  1.000 90%         100%  0.50  1.308 0.000  0.26
CICSMISC S  3       N/A  1.000 90%         100%  0.50  159.8 0.000  0.01
CICSRGN  S  2    60  54                      1.11  0.000
IMS      W          N/A                               44.15 0.000  0.14
IMSTMHI  S  2       N/A  0.500 90%          96%  0.70  42.17 0.000  0.14
IMSTMLOW S  4       N/A  1.000 90%         100%  0.50  1.983 0.000  0.12
STC      W          65                              0.642 0.002  43.1
DB2HIGH  S  2    50  56                      0.90  0.000 0.000  0.00
IMS      S  2    50  57                      0.88  0.000
IMSHIGH  S  2    60  50                      1.20  0.000
OMVS     S          65                              0.333 0.002  1.28
         1  2    30  75                      0.40  0.000 0.000  0.00
         2  3    20  50                      0.40  0.125 0.002  8.72
         3  5    10  69                      0.14  0.208 0.002  1.96
OMVSKERN S  1    40  72                      0.55  0.308 0.002  6.79
TPNS     S  2    70  78                      0.89  0.000 0.000  0.00
SYSTEM   W          46                              0.042 1.201  32.6
SYSOTHER S      N/A  63    N/A                       0.000 0.000  0.00
SYSSTC   S      N/A  39    N/A                       0.042
SYSTEM   S      N/A  59    N/A                       0.000 0.000  0.00
TSO      W          80                              1.750 0.025  2.53
TSO      S  2       80  2.000 AVG  2.555  AVG  1.28  1.750 0.025  2.53
⋮
```

Figure 28. Example RMF Monitor III online sysplex summary report

RMF workload activity report in WLM goal mode

The following figure illustrates a couple of sections from our RMF *Workload Activity Report* in goal mode. This report is based on a 1/2-hour interval. Highlighted on the report you see 99.2% of our CICS transactions are completing in 0.5 seconds, and our CICS workload is processing 1130.54 transactions per second.

```
1                                                 W O R K L O A D   A C T I V I T Y
                                                                                                                          PAGE   8
         OS/390                SYSPLEX UTCPLXJ8          DATE 01/18/2001           INTERVAL 30.00.307   MODE = GOAL
         REL. 02.10.00         RPT VERSION 02.10.00      TIME 10.00.00

                                                  POLICY ACTIVATION DATE/TIME 12/07/2000 09.50.19

   REPORT BY: POLICY=WLMPOL01   WORKLOAD=CICS        SERVICE CLASS=CICS       RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=2
                                                     CRITICAL     =NONE

   TRANSACTIONS    TRANS.-TIME  HHH.MM.SS.TTT
   AVG      0.00   ACTUAL                  54
   MPL      0.00   EXECUTION               81
   ENDED 1733071   QUEUED                   0
   END/S  962.79   R/S AFFINITY             0
   #SWAPS      0   INELIGIBLE               0
   EXCTD 1350758   CONVERSION               0
   AVG ENC  0.00   STD DEV                255
   REM ENC  0.00
   MS ENC   0.00

              ------------------------------RESPONSE TIME BREAKDOWN IN PERCENTAGE------------------------------  ------STATE------
   SUB    P   TOTAL  ACTIVE  READY   IDLE  -----------------------------WAITING FOR-----------------------------  SWITCHED TIME (%)
   TYPE                                    CONV  I/O PROD MISC LOCK                                               LOCAL SYSPL REMOT
   CICS  BTE    123     2.0    0.1    0.0   120  0.7  0.0  0.0  0.0                                                79.8  40.5   0.0
   CICS  EXE   63.1     4.0    2.0   10.6   0.0  4.7 27.4 14.3  0.0                                                 0.9   0.4   0.0
   DB2   EXE   12.0     1.3    0.0    0.0   0.0  4.4  0.0  5.7  0.5                                                 0.0   0.0   0.0
   SMS   EXE   37.7     3.5    0.0    0.0   0.0 33.9  0.0  0.0  0.3                                                 0.0   0.0   0.0

   VELOCITY MIGRATION:   I/O MGMT   N/A     INIT MGMT  N/A

           ---RESPONSE TIME---  EX   PERF
           HH.MM.SS.TTT         VEL  INDX   Comman
   GOAL    00.00.00.600  80.0%
   ACTUALS
   *ALL                  99.2%   N/A  0.5
   JA0                   99.1% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   JB0                   99.2% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   JC0                   99.2% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   JE0                   99.2% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   JF0                   99.3% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   JG0                   99.3% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   JH0                   97.8% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   J80                   99.2% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   J90                   99.3% N/A  0.5    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
   Z0                     0.0% N/A  N/A    0.0   0.0   0.0    0.0   0.0  0.0  0.0                                 0.0  0.0  0.0
                                                                                                                          PAGE   9
         OS/390                SYSPLEX UTCPLXJ8          DATE 01/18/2001           INTERVAL 30.00.307   MODE = GOAL
         REL. 02.10.00         RPT VERSION 02.10.00      TIME 10.00.00

                                                  POLICY ACTIVATION DATE/TIME 12/07/2000 09.50.19

                                                 ----------RESPONSE TIME DISTRIBUTION----------
      ----TIME----    --NUMBER OF TRANSACTIONS--    -------PERCENT-------  0    10   20   30   40   50   60   70   80   90   100
      HH.MM.SS.TTT    CUM TOTAL       IN BUCKET     CUM TOTAL   IN BUCKET  |....|....|....|....|....|....|....|....|....|....|
   <  00.00.00.300         1701K          1701K         98.1        98.1   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
   <= 00.00.00.360         1708K           7534         98.6         0.4   >
   <= 00.00.00.420         1713K           4658         98.8         0.3   >
   <= 00.00.00.480         1716K           3061         99.0         0.2   >
   <= 00.00.00.540         1718K           2011         99.1         0.1   >
   <= 00.00.00.600         1719K           1556         99.2         0.1   >
   <= 00.00.00.660         1720K           1083         99.3         0.1   >
   <= 00.00.00.720         1721K            827         99.3         0.0   >
   <= 00.00.00.780         1722K            616         99.4         0.0   >
   <= 00.00.00.840         1722K            518         99.4         0.0   >
   <= 00.00.00.900         1722K              0         99.4         0.0   >
   <= 00.00.01.200         1724K           1771         99.5         0.1   >
   <= 00.00.02.400         1729K           4959          100         0.3   >
   >  00.00.02.400         1733K           3935          100         0.2   >


   REPORT BY: POLICY=WLMPOL01   WORKLOAD=CICS
                                          cics workload

   TRANSACTIONS    TRANS.-TIME  HHH.MM.SS.TTT   --DASD I/O--   ---SERVICE----   --SERVICE RATES--   PAGE-IN RATES    ----STORAGE----
   AVG     66.98   ACTUAL                 123   SSCHRT  2464   IOC      57004K   ABSRPTN     27842   SINGLE     0.0   AVG      11478.3
   MPL     66.98   EXECUTION               81   RESP    14.4   CPU     541183K   TRX SERV    27842   BLOCK      0.0   TOTAL     768820
   ENDED 2035027   QUEUED                   0   CONN     2.8   MSO       2513M   TCB        8793.1   SHARED     0.0   CENTRAL  635117
   END/S 1130.54     R/S AFFINITY             0   DISC     5.2   SRB     245916K   SRB         4638.8   HSP        0.0   EXPAND   133702
   #SWAPS      0   INELIGIBLE               0   Q+PEND   6.4   TOT       3357M   RCT            0.0   HSP MISS   0.0
   EXCTD 1351007   CONVERSION               0   IOSQ     0.0   /SEC      1865K   IIT            4.8   EXP SNGL   0.1   SHARED     0.00
   AVG ENC  0.00   STD DEV              1.717                                    HST            0.0   EXP BLK    0.1
   REM ENC  0.00                                                                 APPL %       746.4   EXP SHR    0.0
   MS ENC   0.00
                                                  W O R K L O A D   A C T I V I T Y
 .
 .
 .
```


Figure 29. Example RMF workload activity report in WLM goal mode

Appendix C. Availability of our test reports

The following information describes the variety of ways in which you can obtain our test reports.

Our publication schedule is changing

Starting in 2003, our publication schedule changed somewhat from our traditional quarterly cycle as a result of the planned change in the development cycle for annual z/OS releases. Keep an eye on our Web site for announcements about the availability of new editions of our test report.

Availability on the Internet: You can view, download, and print the most current edition of our test report from our z/OS Integration Test Web site at:

www.ibm.com/servers/eserver/zseries/zos/integtst/

Our Web site also provides all of our previous year-end editions, each of which contains all of the information from that year's interim editions.

You can also find our test reports on the z/OS Internet Library Web site at:

www.ibm.com/servers/eserver/zseries/zos/bkserv/

Each edition is available in the following formats:

- IBM BookManager BOOK format

  On the Web, BookManager documents are served as HTML via IBM BookServer. You can use your Web browser (no plug-in or other applications are needed) to view, search, and print selected topics. You can also download individual BOOK files and access them locally using the IBM Softcopy Reader or IBM Library Reader. You can get the Softcopy Reader or Library reader free of charge from the IBM Softcopy Web site at www.ibm.com/servers/eserver/zseries/softcopy/.
- Adobe Portable Document Format (PDF)

  PDF documents require the Adobe Acrobat Reader to view and print. Your Web browser can invoke the Acrobat Reader to work with PDF files online. You can also download PDF files and access them locally using the Acrobat Reader. You can get the Acrobat Reader free of charge from www.adobe.com/products/acrobat/readstep.html.

Softcopy availability: BookMaster BOOK and Adobe PDF versions of our test reports are included in the OS/390 and z/OS softcopy collections on CD-ROM and DVD. For more information about softcopy deliverables and tools, visit the IBM Softcopy Web site (see above for the Web site address).

A note about the currency of our softcopy editions

Because we produce our test reports toward the end of the product development cycle, just before each new software release becomes generally available (GA), we cannot meet the production deadline for the softcopy collections that coincide with the product's GA release. Therefore, there is normally a one-edition lag between the release of our latest test report edition and the softcopy collection in which it is included. That is, the test report that appears in any given softcopy collection is normally one edition behind the most current edition available on the Web.

Hardcopy availability: Our December 2001 edition was the last edition to be published in hardcopy. As of 2002, we no longer produce a hardcopy edition of our year-end test reports. You can still order a printed copy of a previous year-end edition (using the order numbers shown in Table 14 below) through your normal ordering process for IBM publications.

Available year-end editions: The following year-end editions of our test report are available:

Table 14. Available year-end editions of our test report

		Covers our test experiences for...		
Title	**Order number**	**This year...**	**And these releases...**	**Softcopy collection kits**
z/OS Parallel Sysplex Test Report	SA22-7663-09	2003	z/OS V1R4	SK3T-4269-11 SK3T-4271-11
z/OS Parallel Sysplex Test Report	SA22-7663-07	2002	z/OS V1R3 and V1R4	SK3T-4269-07 SK3T-4271-07
z/OS Parallel Sysplex Test Report	SA22-7663-03	2001	z/OS V1R1 and V1R2	SK3T-4269-03 SK3T-4270-04 SK3T-4271-03
OS/390 Parallel Sysplex Test Report	GC28-1963-19	2000	OS/390 V2R9 and V2R10	SK2T-6700-24 SK2T-6718-14
OS/390 Parallel Sysplex Test Report	GC28-1963-15	1999	OS/390 V2R7 and V2R8	SK2T-6700-17
OS/390 Parallel Sysplex Test Report	GC28-1963-11	1998	OS/390 V2R5 and V2R6	SK2T-6700-15 and -17
OS/390 Parallel Sysplex Test Report	GC28-1963-07	1997	OS/390 V1R3 and V2R4	SK2T-6700-11 and -13
OS/390 Parallel Sysplex Test Report	GC28-1963-03	1996	OS/390 V1R1 and V1R2	SK2T-6700-07
S/390 MVS Parallel Sysplex Test Report	GC28-1236-02	1995	MVS/ESA SP V5	none

Other related publications: From our Web site, you can also access other related publications, including our companion publication, *OS/390 Parallel Sysplex Recovery*, GA22-7286, as well as previous editions of *OS/390 e-Business Integration Test (eBIT) Report*.

Appendix D. Useful Web sites

We have cited the IBM books we used to do our testing as we refer to them in each topic in this test report. This chapter contains listings of some of the Web sites that we reference in this edition or previous editions of our test report.

IBM Web sites

Table 15 lists some of the IBM Web sites that we reference in this edition or previous editions of our test report:

Table 15. Some IBM Web sites that we reference

Web site name or topic	Web site address
IBM Terminology (includes the *Glossary of Computing Terms*)	www.ibm.com/ibm/terminology/
IBM CICS Transaction Gateway	www.ibm.com/software/ts/cics/library/
IBM HTTP Server library	www.ibm.com/software/websphere/httpservers/library.html
IBMLink	www.ibm.com/ibmlink/
IBM mainframe servers Internet library	www.ibm.com/servers/eserver/zseries/library/
IBM Redbooks	www.ibm.com/redbooks/
IBM Systems Center Publications (IBM TechDocs — flashes, white papers, etc.)	www.ibm.com/support/techdocs/
Linux at IBM	www.ibm.com/linux/
Net.Data Library	www.ibm.com/software/data/net.data/library.html
OS/390 Internet library	www.ibm.com/servers/s390/os390/bkserv/
Parallel Sysplex	www.ibm.com/servers/eserver/zseries/pso/
Parallel Sysplex Customization Wizard	www.ibm.com/servers/eserver/zseries/zos/wizards/parallel/
System Automation for OS/390	www.ibm.com/servers/eserver/zseries/software/sa/
WebSphere Application Server	www.ibm.com/software/webservers/appserv/
WebSphere Application Server library	www.ibm.com/software/webservers/appserv/zos_os390/library/
WebSphere Studio library	www.ibm.com/software/websphere/studio/library.html
WebSphere Studio Workload Simulator	www.ibm.com/software/awdtools/studioworkloadsimulator/library/
z/OS Consolidated Service Test	www.ibm.com/servers/eserver/zseries/zos/servicetst
z/OS downloads	www.ibm.com/servers/eserver/zseries/zos/downloads/
z/OS Integration Test (includes information from OS/390 Integration Test and e-business Integration Test (ebIT))	www.ibm.com/servers/eserver/zseries/zos/integtst/
z/OS Internet library	www.ibm.com/servers/eserver/zseries/zos/bkserv/
z/OS UNIX System Services	www.ibm.com/servers/eserver/zseries/zos/unix/
z/OS.e home page	www.ibm.com/servers/eserver/zseries/zose/

Table 15. Some IBM Web sites that we reference (continued)

Web site name or topic	Web site address
z/OS.e Internet library	www.ibm.com/servers/eserver/zseries/zose/bkserv/

Other Web sites

Table 16 lists some other non-IBM Web sites that we reference in this edition or previous editions of our test report:

Table 16. Other Web sites that we reference

Web site name or topic	Web site address
Cisco Systems	www.cisco.com/
Java Servlet Technology	java.sun.com/products/servlet/
Java 2 Platform, Enterprise Edition (J2EE)	java.sun.com/products/j2ee/
JavaServer Pages (JSP)	java.sun.com/products/jsp/
J2EE Connector Architecture	java.sun.com/j2ee/connector/
SuSE Linux	www.suse.com/

Appendix E. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`www.ibm.com/servers/eserver/zseries/zos/bkserv/`

One exception is command syntax that is published in railroad track format; screen-readable copies of z/OS books with that syntax information are separately available in HTML zipped file form upon request to mhvrcfs@us.ibm.com.

Notices

This information was developed for products and services offered in the USA.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

COPYRIGHT LICENSE:

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

these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

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

AIX
BatchPipes
BookManager
BookMaster
CICS
CICSPlex
DB2
DB2 Connect
DFS
DFSMS/MVS
DFSMShsm
DFSMSrmm
Enterprise Storage Server
ESCON
@server
FICON
IBM
ibm.com
IBMLink
IMS
Infoprint
Language Environment
Library Reader
MQSeries
MVS
MVS/ESA
Net.Data
NetView
Notes
OS/2
OS/390
Parallel Sysplex
PR/SM
Processor Resource/Systems Manager
QMF
RACF
RAMAC
Redbooks
RMF
RS/6000
S/390
SP
SupportPac
Sysplex Timer
Tivoli
TotalStorage
VisualAge
VSE/ESA
VTAM
WebSphere
z/OS
z/OS.e
z/VM
zSeries

The following terms are trademarks of other companies:

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

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

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

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

Index

M

N

O

P

R

S

T

U

V

W

Z

Readers' Comments — We'd Like to Hear from You

z/OS
zSeries Platform Test Report
Version 1 Release 6

Publication No. SA22-7997-00

We appreciate your comments about this publication. Feel free to comment on specific errors or omissions, accuracy, organisation, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

For general questions, please call ″Hello IBM″ (phone number 01803/313233).

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

Comments:

Thank you for your support.

To submit your comments:

- Send your comments to the address on the reverse side of this form.

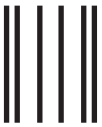
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
SA22-7997-00

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department B6ZH, Mail Station P350
2455 South Road
Poughkeepsie, NY
12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in USA

SA22-7997-00