

z/OS



System z Platform Test Report for z/OS and Linux Virtual Servers

Version 1 Release 10

z/OS



System z Platform Test Report for z/OS and Linux Virtual Servers

Version 1 Release 10

Note!

Before using this information and the products it supports, be sure to read the general information under "Notices" on page 351.

Ninth Edition, December 2008

This is a major revision of SA22-7997-07.

This edition applies to Parallel Sysplex environment function that includes data sharing and parallelism. Parallel Sysplex uses the z/OS (5694-A01) 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(LBCRUZ)

Internet e-mail: lbcruz@us.ibm.com

World Wide Web: www.ibm.com/systems/services/platformtest/servers/systemz.html

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.

© Copyright International Business Machines Corporation 2001, 2008.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Opening remarks

A message from our team

Welcome to *IBM System z Platform Test Report for z/OS and Linux Virtual Servers*. Our team focuses on integration testing with a platform-wide view of z/OS[®] and Linux[®] on System z[™] in the enterprise.

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, see "How to send your comments" on page xix.

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 Alexa	Lisa Dodaro	Elaine Murphy
Loraine Arnold	Eli Dow	Al Nims
Evren Ozan Baran	Bob Fantom	Phil Peters
Ryan Bartoe	Nancy Finn	Jim Rossi
Duane Beyer	Bobby Gardinor	Andrew M. Sica
Jeff Bixler	Kieron Hinds	Tom Sirc
Muriel Bixler	Alexy N. Ivanov	Karen Smolar
Jay Brenneman	Fred Lates	Wei Song
Dave Buehl	Al Lease	Paul Sonnenberg
Jon Burke	Frank LeFevre	Jeff Stokes
Jim Campbell	Tan T. Li	Jim Stutzman
Alex Caraballo	Dolores Lovallo	Mike Tebolt
Phil Chan	Scott Loveland	Lisette Toledo
Alexander Chepkasov	George Markos	Julia Tuzhikova
John Corry	Sue Marcotte	Zhao Yu Wang
Don Costello	Tammy McAllister	Tatiana Zhbannikova
Kevin Coyne	Azeem Mohammed	
Luis Cruz	Robert Muenkel	

Contents

Opening remarks	iii
Figures	xiii
Tables	xv
About this document	xvii
An overview of System z Platform Evaluation Test (zPET)	xvii
Our mission and objectives.	xvii
Our test environment	xviii
Who should read this information	xviii
How to use this information	xviii
How to find our test reports	xviii
Where to find more information	xix
How to send your comments	xix

Part 1. System z Platform Evaluation Test for z/OS 1

Chapter 1. Migrating to and using z/OS V1R10 3

z/OS V1R10 base migration experiences	3
Our high-level base migration process	3
More about our base migration activities	4
Other z/OS V1R10 migration experiences.	5
Using IBM Health Checker for zOS for migration checking	5

Chapter 2. Using the IBM System z10 Enterprise Class platform 7

Using HiperDispatch	7
Using z/OS Capacity Provisioning	8
Setting up Capacity Provisioning	9
Capacity Provisioning in action.	12
Using Parallel Sysplex InfiniBand coupling facility links	13
Configuring OSA-Express3 multi-port support for IP	13
Testing the expedited duplex completion protocol	14

Chapter 3. Migrating from SMF data set recording to log stream logging 17

Advantages of recording SMF data in log streams	17
SMF performance improvements with log stream logging	17
Management of SMF data on a per log stream basis	17
SMF data reliability with log stream logging	17
Browsing (dumping) SMF data	18
Data retention and deletion on a per log stream basis	18
Configuration considerations for log stream logging	19
Choosing CF structure log streams or DASD-only log streams for SMF data.	19
Determining SMF log stream configuration for our test environment	19
Estimating interim storage, offload, and staging data set sizes	20
CF structure and log stream definitions	22
SMFPRMxx member definition	23
Migrating to SMF log stream logging	23
Switching from SMF data set recording to SMF log stream logging	25
Using the SWITCH SMF command and the run dump program	26
Monitoring our SMF configuration	27
References for SMF log stream logging	29

	Chapter 4. RRS archive logging enhancement	31
	Chapter 5. z/OS system logger administrative data utility enhancements	33
	Chapter 6. New auxiliary and pageable storage shortage messages	37
	Changes to auxiliary storage management	37
	Changes to pageable storage management	38
	Chapter 7. Migrating to a Server Time Protocol Coordinated Timing Network	39
	Overview of STP	39
	STP terminology.	40
	STP planning considerations.	42
	Our servers and coupling facilities.	43
	Considerations for migrating from a mixed CTN to an STP-only CTN.	43
	Recovery considerations	44
	Summary planning matrix	44
	STP migration experiences	46
	Our initial Sysplex Timer (ETR-only) topology.	46
	Adding STP timing-only links	49
	CTN ID configuration and verification	52
	Stratum 1 to stratum 2 transition and verification for T75	62
	Stratum 1 to stratum 2 transition and verification for G74	69
	Reverse migration: Stratum 2 to stratum 1 transition and verification	72
	Reverse migration: Mixed CTN to ETR timing network.	75
	Migrating from a mixed CTN to an STP-only CTN	78
	Changing server roles in an STP-only CTN.	86
	Reverse migration: STP-only CTN to mixed CTN.	89
	Chapter 8. Using the IBM zIIP	97
	Prerequisites for IBM zIIP	97
	Configuring the IBM zIIP.	98
	Monitoring zIIP utilization.	100
	DB2 workloads that exercise the IBM zIIP	102
	OMEGAMON XE for z/OS 3.1.0 zIIP support	103
	IBM zIIP assisted IPsec	107
	SDM on the IBM zIIP	108
	Chapter 9. Using TPC-R V3.4 and Basic HyperSwap in our zPET environment	113
	TPC-R environment in zPET	113
	Installing and setting up TPC-R	114
	z/OS Basic HyperSwap	114
	TPC-R product documentation	115
	Chapter 10. Testing extended address volumes.	117
	Requirements for EAV	117
	z/OS software requirements for EAV	117
	Hardware requirements for EAV	117
	Setting up EAV.	117
	Our DASD (DS8000) setup for EAV	117
	Our DFSMS setup for EAV	118
	Migrating to EAV	119
	Verifying the location of data on EAV volumes	119
	zFS EAV setup and exploitation	121
	DB2 EAV 3390-A setup and exploitation	121
	DB2 non-EAV 3309-A setup and exploitation	125
	WebSphere MQ EAV setup and exploitation	128
	Chapter 11. Using High Performance FICON for System z	129
	Required hardware for zHPF	129

z/OS external interfaces for zHPF	129
IECIOSxx parmlib support for zHPF.	129
SETIOS system command support for zHPF	129
Examples of system commands for zHPF	130
 Chapter 12. Testing PPRC secondary devices in subchannel set 1	133
 Chapter 13. Using z/OS Security Server RACF	135
Reorganizing our RACF databases	135
About our RACF database reorganization	135
Approach to reorganizing our RACF databases	136
Jobs and commands used during our database reorganization	136
Results of our RACF database reorganization.	138
Password reset granularity	138
Custom fields in the RACF database	140
RACDCERT support for 4096-bit keys	142
Chapter 14. Migrating to and using ICSF HCR7750	143
Migrating to a larger PKDS.	143
Exercising the CPACF function on the System z10 EC platform.	143
Chapter 15. Using ICSF migration health checks	145
Chapter 16. Using Network Authentication Service (Kerberos)	147
Password phrase support	147
Enabling password phrase support	147
Verifying the Kerberos principal's password phrase	148
Chapter 17. Using LDAP Server	151
IBM TDS plug-in support	151
Enabling IBM TDS plug-in support	151
Verifying IBM TDS plug-in support	152
Cleaning up after our testing	152
Using TDS differentiation, currency, and certification validation	153
Implementing TDS differentiation, currency, and certification validation.	153
Verifying certificate validation.	153
Using LDAP server wait for DB2 during startup	154
Implementing LDAP server wait for DB2 during startup	154
Verifying LDAP server wait for DB2 during startup	154
Using TDS password phrase support	156
Implementing TDS password phrase support.	156
Verifying TDS password phrase support	156
Using LDAP support for RACF custom fields	158
Implementing LDAP support for RACF custom fields.	158
Verifying LDAP support for RACF custom fields	159
Using SHA and MD5 encrypted passwords	160
Implementing SHA and MD5 encrypted passwords	160
Verifying SHA and MD5 encrypted passwords	160
 Chapter 18. Using the Cryptographic Services PKI Services	163
IP version 6 support	163
UTF-8 support	163
4096-bit key support	163
Chapter 19. Using System SSL	165
Using System SSL CPACF hardware support	165
Using System SSL 4096-bit hardware support.	166
Enhancements to gskkyman	167

Chapter 20. Using z/OS UNIX System Services	169
z/OS UNIX enhancements in z/OS V1R10	169
Password phrase enhancements for rlogin, shell	169
z/OS UNIX tools: Service to display a z/OS UNIX directory	172
Examples of DIRLIST using the command invocation format	172
Validating the DIRLIST command line processor	174
z/OS UNIX health checks: USS_PARMLIB_MOUNTS and USS_CLIENT_MOUNTS	179
USS_PARMLIB_MOUNTS check	179
USS_CLIENT_MOUNTS check	182
z/OS zFS enhancements	183
Sysplex root migration from HFS to zFS	184
zFS format authorization	185
Aggregate full message from zFS	186
zFS AUDITID	186
zFS read-only mount recovery	187
Chapter 21. Migrating to CICS Transaction Server for z/OS, Version 3.2	189
Overview of migrating to CICS TS 3.2	189
Preparing to migrate to CICS TS 3.2	190
Migrating CICSplex SM	191
Migrating the CMASs	192
Migrating the MASs	192
Migrating the CICSplex SM Web User Interface	193
Experiences with migrating to CICS TS 3.2	193
Chapter 22. Migrating to CICS TG V7.0	195
Migrating the CICS TG daemon to V7	195
CICS TG daemon statistics	195
Port requirements for the statistics interface	195
Reserving the statistics interface port	196
Chapter 23. Migrating to IMS Version 10.1	197
Migration and coexistence software	197
Staging our migration	197
FPCTRL system definition macro eliminated	198
IMS Exits	198
DBRC SCI registration exit routine (DSPSCIX0)	198
OTMA routing exits (DFSYPRX0 and DFSYDRU0)	198
IMS Java migration	198
Summary of changes for Java in IMS V10	198
Service applied	199
Migration procedure	199
IMS syntax checker	199
Migrating IMS V9 RECON data sets to V10	199
RECON status prior to migration	199
RECON upgrade	200
RECON status after migration	200
IRLM support in IMS V10	201
Upgrading IMS utilities	201
Chapter 24. Migrating to IMS Transaction Manager Resource Adapter V10.2	203
Evaluating applications for potential migration	203
Updating applications that use conversational transactions	203
Chapter 25. Using JZOS	205
JZOS batch launcher	205
MVS operations and JZOS	205
JZOS installation and setup	205
JZOS Cookbook	206

Chapter 26. Using the IBM WebSphere Business Integration family of products.	207
Using WebSphere MQ shared queues and coupling facility structures	207
Our queue sharing group configuration	207
Our coupling facility structure configuration	208
Recovery behavior with queue managers using coupling facility structures.	208
Enabling WebSphere MQ Security	210
Reference material.	210
Problems encountered	211
Enabling higher availability for WebSphere MQ	212
Using WebSphere Message Broker	213
Updating the Retail_IMS workload for workload sharing and high availability	213
MQCICS — WebSphere MQ-CICS adapter/bridge workload	214
WebSphere MQ-CICS bridge monitor using clustered queues	215
WebSphere MQ-CICS adapter using shared queues.	215
Chapter 27. Using IBM WebSphere Application Server for z/OS	217
About our z/OS V1R10 test environment running WebSphere Application Server	217
Our z/OS V1R10 WebSphere test environment	217
Other changes and updates to our WebSphere test environment	221
Migrating to WebSphere Application Server for z/OS V6.1	221
Migrating to CICS Transaction Gateway V7	222
Migrating to IMS Transaction Manager Resource Adapter V10.2	222
Passing DB2 client information to the server	222
Installed TPC-R V3.4	226
Where to find more information	226
Chapter 28. Installing and configuring WebSphere Process Server for z/OS	227
WPS installation and configuration	227
WPS security	228
Chapter 29. Installing and configuring WebSphere Service Registry and Repository	
for z/OS.	229
WSRR installation and configuration	229
WSRR security	230
Chapter 30. Deploying a secure SOA solution	231
The SOA solution scenario	231
<hr/>	
Part 2. System z Platform Evaluation Test for Linux virtual servers	235
Chapter 31. About our Linux virtual server environment	237
Fundamental goals and priorities.	237
Staged implementation	238
About our environment	238
Our workloads	239
Overall configuration.	240
Production system names and usages	242
Test (MDAT) system names and usages	244
Chapter 32. Software management	245
Maintenance strategy and methodology	245
Base operating system upgrades	245
Upgrading the operating system on the Tivoli Storage Manager server	245
Upgrading WebSphere Application Server prior to an operating system upgrade.	251
z/VM 5.3 to z/VM 5.4 transition notes.	251
Middleware upgrades	251
Upgrading our application servers and deployment manager	251
Tivoli Storage Manager server upgrade.	258

Chapter 33. Systems management	259
Installing an open source VPN server	259
Altering the FSTAB and kernel to use disk-by-path by default	266
Migrating from ReiserFS v3 to the ext3 file system	267
Chapter 34. Capacity management	269
Setting up DASD groups and automatic allocation in DirMaint.	269
Planning the DASD storage groups	269
Preparing DASD for use in a storage group	270
Defining the pool in the EXTENT CONTROL file	270
Allocating minidisks with DirMaint	272
Tracking storage group utilization	273
Planning for growth	274
Chapter 35. Security management	275
Upgrading the Tivoli Access Manager policy server	275
Backing up the policy server and WebSEAL servers	275
Upgrading the policy server (LITTAM01)	276
Upgrading the WebSEAL server instances	278
Migrating IBM Tivoli Directory Server from Version 6.0 to Version 6.1	285
Configuring LDAP replication	289
Configuring WebSEAL for LDAP load balancing and failover	291
Installing WebSphere Edge Components V7.0.	293
Installing Load Balancer for IPv6	293
Configuring Load Balancer for IPv6	298
Testing load balancing and high availability	300
Creating a system init script for automated startup.	301
Creating a custom DirMaint usermod for integration with RACF	306
Chapter 36. Future Linux on System z projects	309
Appendix A. About our Parallel Sysplex environment	311
Overview of our Parallel Sysplex environment	311
Our Parallel Sysplex hardware configuration	311
Overview of our hardware configuration	311
Hardware configuration details	313
Our Parallel Sysplex software configuration	317
Overview of our software configuration	318
About our naming conventions	319
Appendix B. About our networking environment	321
Our networking configuration	321
Configuration overview	321
Our IPv6 environment configuration	322
z/OS UNIX System Services changes and additions	322
Comparing the network file systems.	324
Our VTAM configuration	324
Testing our networking environment	325
Enabling NFS recovery for system outages	325
Setting up the NFS environment for ARM and DVIPA.	325
Appendix C. About our security environment	329
Our Integrated Cryptographic Service Facility (ICSF) configuration	329
Network Authentication Service configuration	330
Our LDAP configuration	331
Integrated Security Services (ISS) LDAP exploitation	331
IBM Tivoli Directory Server (IBM TDS) exploitation	333
Appendix D. About our test workloads	335

Base system workloads	335
Application enablement workloads	336
Enterprise Identity Mapping (EIM)	336
HFS/zFS file system recursive copy/delete	336
IBM HTTP Server	336
ICSF	337
LDAP Server	337
Network Authentication Service (Kerberos)	337
z/OS UNIX Shelltest (rlogin/telnet)	338
z/OS UNIX Shelltest (TSO).	338
WebSphere Application Server for z/OS	338
WebSphere MQ for z/OS workloads	338
WebSphere Message Broker workloads	339
Networking workloads	340
Database product workloads	341
Database product OLTP workloads	341
Database product batch workloads	342
WebSphere MQ / DB2 bookstore application	343
Appendix E. Some of our RMF reports	345
RMF Monitor I Post Processor Summary	345
RMF Monitor III Online Sysplex Summary	345
RMF Workload Activity in WLM goal mode	347
Accessibility	349
Using assistive technologies	349
Keyboard navigation of the user interface	349
z/OS information	349
Notices	351
Policy for unsupported hardware.	353
Trademarks	353
Index	355

Figures

1. Example of an RMF Post Processor CPU Activity report	8
2. Capacity Provisioning components	10
3. zPET initial Sysplex Timer topology, planned mixed CTN topology, and planned STP-only CTN topology	43
4. zPET Sysplex Timer topology	47
5. System (Sysplex) Time panels, viewed from the Support Element (SE)	48
6. HCM Create Coupling Facility Link Connection dialog for defining a STP timing-only link	51
7. zPET Sysplex Timer topology with STP timing-only links	52
8. System (Sysplex) Time: STP Configuration panel	53
9. STP Configuration panel with STP ID value entered	53
10. STP configuration confirmation panel	53
11. STP CTN ID change completion panel	54
12. System (Sysplex) Time: STP Status panel, viewed from the SE on K25	55
13. System (Sysplex) Time: STP Status panel on K28, after configuring the CTN ID on K28	56
14. System (Sysplex) Time: STP Status panel on K25	57
15. System (Sysplex) Time: STP Status panel on K25, showing connectivity to the other three servers	58
16. System (Sysplex) Time: STP Status panel on K28, showing connectivity to the other three servers	59
17. System (Sysplex) Time: STP Status panel on G74, showing connectivity to the other three servers	60
18. System (Sysplex) Time: STP Status panel on T75 showing connectivity to the other three servers	61
19. zPET mixed CTN topology	62
20. System (Sysplex) Time: Timing Network panel before moving T75 to stratum 2	63
21. System (Sysplex) Time: STP Status panel before moving T75 to stratum 2	64
22. System (Sysplex) Time: ETR Configuration panel with ETR ports disabled	65
23. System (Sysplex) Time: ETR Port State Change Confirmation panel	65
24. System (Sysplex) Time: Apply ETR Configuration panel, indicating a successful configuration change	66
25. System (Sysplex) Time: STP Status panel with T75 at stratum 2	67
26. System (Sysplex) Time: Timing Network panel with T75 at stratum 2	68
27. zPET mixed CTN with T75 at stratum 2	69
28. System (Sysplex) Time: STP Status panel for G74 with G74 and T75 at stratum 2	70
29. zPET mixed CTN with two stratum 2 nodes: T75 and G74	72
30. System (Sysplex) Time: ETR Configuration panel for port enablement	73
31. ETR Configuration confirmation panel	73
32. System (Sysplex) Time: STP Status panel, showing T75 back at stratum 1	74
33. STP Configuration: STP ID removal	75
34. STP Configuration: CTN Network ID Change Confirmation panel	76
35. STP Configuration: CTN Network ID Change completion	76
36. System (Sysplex) Time: STP Status panel, showing that G74 had returned to an ETR timing network	77
37. zPET mixed CTN with FR24 and K25 removed	79
38. Initial view of the System (Sysplex) Time – Network Configuration panel on T75	80
39. System (Sysplex) Time task – Network Configuration panel with all server roles assigned	81
40. Global Timing Network ID Change Confirmation	81
41. System (Sysplex) Time: Timing Network panel on T75 – STP-only CTN	83
42. System (Sysplex) Time: ETR Configuration panel – STP-only CTN	84
43. System (Sysplex) Time: STP Status panel – STP-only CTN	85
44. zPET STP-only CTN	86
45. System (Sysplex) Time: Network Configuration panel – assigning K28 as current time server	87
46. Network Configuration Change Confirmation panel – apply CTN role change	87
47. Modify Network Configuration panel – successful CTN role change	88
48. System (Sysplex) Time: STP Status panel with K28 as current time server	88
49. zPET STP-only CTN with backup time server as current time server	89
50. System (Sysplex) Time: Network Configuration panel – K28 starting reverse migration to mixed CTN	90
51. Migration to Mixed CTN Confirmation: Confirm migration from STP-only to mixed CTN (step 1)	91
52. Migration to Mixed CTN Confirmation: Confirm migration from STP-only to mixed CTN (step 2)	91
53. Migration to Mixed CTN: STP-only to mixed CTN migration in progress	92
54. System (Sysplex) Time: Network Configuration panel – Migration to mixed CTN in progress	92
55. System (Sysplex) Time: Timing Network panel – K28 back in mixed CTN	94

56.	System (Sysplex) Time: Network Configuration panel –K28 back in mixed CTN.	94
57.	System (Sysplex) Time: ETR Configuration panel – K28 back in mixed CTN .	95
58.	System (Sysplex) Time: STP Status panel – K28 back in mixed CTN.	95
59.	Image profile for our J80 z/OS image with 2 zIIPs defined.	98
60.	SDSF display showing zIIP utilization	102
61.	OMEGAMON ZMCPU screen	104
62.	OMEGAMON System CPU Utilization 1	105
63.	OMEGAMON System CPU Utilization 2	106
64.	OMEGAMON Address Space Overview	107
65.	RMF Monitor III Processor Usage screen showing amount of zIIP eligible work by ANTAS0xx address spaces (scenario 1)	110
66.	RMF Workload Activity Report showing zIIP eligible workload by the ANTAS0xx address spaces (scenario 1)	110
67.	RMF Monitor III Processor Usage screen showing amount of work processed by zIIP processors for ANTAS0xx address spaces (scenario 2)	111
68.	RMF Workload Activity Report showing work processed by zIIP processors for the ANTAS0xx address spaces (scenario 2)	111
69.	Data Set Information panel for our zFS file system on an EAV volume	121
70.	DIRLIST display from a REXX EXEC	174
71.	Our CICS TS configuration	190
72.	Message flow in our Retail_IMS workload	214
73.	Our MQ cluster configuration for the WebSphere MQ-CICS bridge	215
74.	Our queue sharing group configuration for our WebSphere MQ-CICS adapter workload	216
75.	Our WebSphere for z/OS V6 configuration	220
76.	Components of our secure SOA scenario	232
77.	LVS PET application configuration: Logical transaction flow between application clusters	240
78.	LVS PET system configuration: z/VM LPARs hosting Linux virtual servers on two CPCs	241
79.	LVS PET system configuration: Overall system including MDAT Linux servers	242
80.	Example of the Integrated Solutions Console display	254
81.	WebSphere Application Server V6.1.0.0 Federation panel	255
82.	Migration wizard for WebSphere Application Server: Detected versions of WebSphere Application Server panel	256
83.	Migration wizard for WebSphere Application Server: Profile creation parameters panel	257
84.	IBM Tivoli Access Manager Installation dialog: Language selection panel	281
85.	IBM Tivoli Access Manager Installation dialog: Welcome panel	281
86.	IBM Tivoli Access Manager Installation dialog: License agreement acceptance panel	282
87.	IBM Tivoli Access Manager Installation dialog: Disk space required and available panel	282
88.	IBM Tivoli Access Manager Installation dialog: Review configuration options panel	283
89.	IBM Tivoli Access Manager Installation dialog: Installation completion panel	283
90.	IBM Tivoli Directory Server 6.1 installation dialog: Configuration summary panel	286
91.	WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Welcome panel	295
92.	WebSphere Edge Components: Load Balancer for IPv6 installation dialog: License agreement acceptance panel	295
93.	WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Language selection panel	296
94.	WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Setup type selection panel	296
95.	WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Feature selection panel	297
96.	WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Installation options summary panel	297
97.	WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Installation summary panel	298
98.	Our sysplex hardware configuration	312
99.	Our sysplex software configuration	318
100.	Our networking topology	321
101.	Our VTAM configuration	324
102.	NFS configuration	326
103.	Overview of our Network Authentication Service configuration.	330
104.	Integrated Security Services (ISS) LDAP environment	331
105.	IBM Tivoli Directory Server (IBM TDS) environment	333

Tables

1.	Parallel Sysplex planning library publications	xix
2.	Our high-level migration process for z/OS V1R10	3
3.	Planning steps for deploying the Server Time Protocol in our data center	44
4.	Our DASD volume configuration for EAV testing	118
5.	Applicable .jar files for various Java application environments in IMS Version 10	199
6.	DB2 Universal JDBC driver methods for passing client information to the server	223
7.	Custom priorities that we set for the JDBC datasource for our application	223
8.	Our production Linux system names, IP addresses, and usages	242
9.	Our MDAT system names, IP addresses, and usages	244
10.	Remaining panels and actions for the migration.sh GUI	257
11.	Our mainframe servers	313
12.	Our coupling facilities	315
13.	Coupling facility channel configuration on Plex 1	315
14.	Coupling facility channel configuration on Plex 2	316
15.	Other sysplex hardware configuration details	316
16.	Our production OLTP application groups	319
17.	Summary of our workloads	335

About this document

This document is a test report written from the perspective of a system programmer. The IBM® System z Platform Evaluation Test (zPET) team (also known as the z/OS 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 mainframe 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 System z Platform Evaluation Test (zPET)

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. In 2005, we expanded to add a Linux Virtual Server arm to our overall environment, which will be used to emulate leading-edge customer environments, workloads, and activities.

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 reproducible 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 on demand enterprise—just like yours.

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

- “Our Parallel Sysplex hardware configuration” on page 311
- “Our Parallel Sysplex software configuration” on page 317
- “Our networking configuration” on page 321
- “Appendix C. About our security environment” on page 329
- “Appendix D. About our test workloads” on page 335

Who should read this information

System programmers can use this information 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 a working knowledge of MVS and Parallel Sysplex concepts and terminology, and at least a basic level of experience with installing and managing the z/OS operating system, subsystems, network products, and other related software. See “Where to find more information” on page xix.

How to use this information

Use this test report as a companion to—*never instead of*—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 our test reports

We make all editions of our test reports available on our IBM Platform Test - System z Web site at:

www.ibm.com/systems/services/platformtest/servers/systemz.html

We publish our test reports twice a year, every June and December. Our December edition covers our initial test experiences with a new z/OS release, including

migration. Our June edition is the final edition for that release; it is cumulative, building upon the December edition with any new test experiences we've encountered since then. We freeze the June edition and begin anew with the next release in December. The most recent edition of our test report, as well as the final editions for previous releases of z/OS, are available on our Web site.

We also have a companion publication, *z/OS V1R8.0 System z Parallel Sysplex Recovery*, GA22-7286. In this publication, we focus on describing:

- How to be prepared for potential problems in a Parallel Sysplex
- What the indicators are to let you know there is a problem
- What actions to take to recover

The recovery scenarios we describe are based on our own experiences in our particular test environment while running z/OS V1R8, DB2® V8, IMS™ V9, WebSphere® Application Server V6.0, WebSphere MQ V6 and CICS® TS V3R1. These scenarios do not represent a comprehensive list of all possible approaches and outcomes, but do represent the approaches we have tested and that work for us.

Note: The recovery book was written in the z/OS V1R8 time frame; however, many of the recovery concepts that we discuss still apply to later releases of 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
<i>z/OS Parallel Sysplex Overview</i>	SA22-7661
<i>z/OS MVS Setting Up a Sysplex</i>	SA22-7625
<i>z/OS Parallel Sysplex Application Migration</i>	SA22-7662
<i>z/OS Planning for Installation</i>	GA22-7504

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

- See the Parallel Sysplex for OS/390 and z/OS Web site at: www.ibm.com/systems/z/advantages/pso/
- See the z/OS Managed System Infrastructure (msys) for Operations Web site at: www.ibm.com/servers/eserver/zseries/msys/msysops/
- See the IBM Education Assistant which integrates narrated presentations, Show Me Demonstrations, tutorials, and resource links to help you successfully use the IBM software products at: publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp

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:

- lbcruz@us.ibm.com, for questions about z/OS and Parallel Sysplex
- lefevre@us.ibm.com, for questions about Linux on System z

Or, you can use the **Contact now** link on our Web site at:

www.ibm.com/systems/services/platformtest/servers/systemz.html

If you are reading the PDF version of this document, you can also submit the Readers' Comments form located at the end of the 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 topic heading or page number).

Part 1. System z Platform Evaluation Test for z/OS

The System z Platform Evaluation Test (zPET) team focuses on integration testing of the z/OS and Parallel Sysplex aspects of our computing environment.

We address such topics as:

- Migration to the latest release of the z/OS operating system
- Experiences with new functionality offered in the latest z/OS release
- Experiences with various z/OS data management and transaction management products that exploit Parallel Sysplex and data sharing
- Experiences with various z/OS middleware and application enablement products

Chapter 1. Migrating to and using z/OS V1R10

This topic describes our migration to z/OS V1R10. Our migration experiences include:

- “z/OS V1R10 base migration experiences”
- “Other z/OS V1R10 migration experiences” on page 5

Here we primarily discuss our sysplex-oriented migration experiences and other related experiences. This includes the enablement of significant new functions and, if applicable, performance aspects. Detailed test experiences with major new functions beyond migration and experiences with other z/OS products appear in subsequent chapters.

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

For migration experiences with...	See this edition of our test report...
-----------------------------------	--

z/OS V1R9	<i>System z Platform Test Report for z/OS and Linux Virtual Servers, June 2008</i>
-----------	--

z/OS V1R8 and z/OS.e V1R8	<i>zSeries® Platform Test Report for z/OS and Linux Virtual Servers, June 2007</i>
---------------------------	--

z/OS V1R10 base migration experiences

This topic describes our experiences with our base migration to z/OS V1R10, without having implemented any new functions. It includes our high level migration process along with other migration activities and considerations.

Our high-level base migration process

The following is an overview of our z/OS V1R10 migration process.

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

Table 2 shows the high-level process we followed to migrate the members of our sysplex from z/OS V1R9 to z/OS V1R10.

Table 2. Our high-level migration process for z/OS V1R10

Stage	Description
Updating PARMLIB for z/OS V1R10	We created SYS1.PETZ10.PARMLIB to contain all the PARMLIB members that changed for z/OS V1R10 and we used our LOADxx member for migrating our systems one at a time. (See our December 1997 test report for an example of how we use LOADxx to migrate individual systems.)

Table 2. Our high-level migration process for z/OS V1R10 (continued)

Stage	Description
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 <i>z/OS Planning for Installation</i> and make sure you install the fixes for any APARs that relate to your configuration before you migrate.
IPLing our first z/OS V1R10 image	We brought up z/OS V1R10 on our Z2 test system and ran it there for a couple of weeks.
Updating the RACF [®] templates	To test the RACF dynamic template enhancement, we IPLed the first z/OS V1R10 image without first running the IRRMIN00 utility with PARM=UPDATE. As expected, the following message appeared: <pre> ICH579E RACF TEMPLATES ON DATABASE ARE DOWNLEVEL </pre> 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 <i>z/OS Security Server RACF System Programmer's Guide</i> , SA22-7681 for details about RACF templates.)
IPLing additional z/OS V1R10 images	We continued to bring up additional z/OS V1R10 images across our sysplex, as follows: <ol style="list-style-type: none"> 1. Migrated test system, Z1, and ran for a couple of days. 2. Migrated the rest of our test systems, Z3 and Z4, and ran for a week. 3. Migrated some of our production systems, JC0, JE0, and TPN, and ran for a couple of days. 4. Migrated three more production systems, Z0, JA0, and J80, and ran for a week. 5. At this point, we took one V1R10 production system, J80, back down to V1R9. This is part of our focus on migration testing and fallback. We ran for two full days and experienced no fallback issues. We then migrated J80 back to V1R10. 6. Migrated one more image, JF0, and ran for a week. 7. Migrated the remaining production systems, JB0 and J90.

More about our base migration activities

This topic highlights additional details about some of the base migration activities that we perform with each new release, including running with mixed product levels, using concatenated PARMLIB, and recompiling automation EXECs.

Running with mixed product levels

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

- z/OS V1R9 and z/OS V1R10
- z/OS V1R9 JES2 and z/OS V1R10 JES2
- z/OS V1R9 JES3 and z/OS V1R10 JES3

Using concatenated PARMLIB

We continue to use concatenated PARMLIB support to add or update PARMLIB members for z/OS V1R10. 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 V1R10 in one place and makes it easier to migrate multiple systems. Rather than change many PARMLIB members each time we migrate another system to V1R10, we just add the PARMLIB statements at the appropriate places in SYS0.IPLPARAM(LOADxx) to allow that system to use SYS1.PETZ10.PARMLIB.

Recompiling REXX EXECs for automation

We recompiled our IBM Tivoli® System Automation for z/OS REXX™ EXECs when we migrated to z/OS V1R10. We discuss the need to recompile these REXX EXECs in our December 1997 test report.

Other z/OS V1R10 migration experiences

This topic highlights additional details about some of our migration experiences that are specific to z/OS V1R10.

Using IBM Health Checker for zOS for migration checking

In previous z/OS releases, we used the IBM Migration Checker for z/OS tool to check the applicability of various migration actions on our currently running systems. With z/OS V1R10, these migration checks are now built into the IBM Health Checker for z/OS.

For our migration checking, we followed the information about using health checks for migration purposes in *z/OS Migration*, under the topic “Using IBM Health Checker for zOS for migration checking.”

Additional information about IBM Health Checker for z/OS can be found in *IBM Health Checker for z/OS: User's Guide*.

Chapter 2. Using the IBM System z10 Enterprise Class platform

This topic describes our deployment and use of the new IBM System z10 Enterprise Class (z10 EC) platform, which was announced and made available in the first quarter of 2008. We discuss the following aspects of our experiences:

- “Using HiperDispatch”
- “Using z/OS Capacity Provisioning” on page 8
- “Using Parallel Sysplex InfiniBand coupling facility links” on page 13
- “Configuring OSA-Express3 multi-port support for IP” on page 13
- “Testing the expedited duplex completion protocol” on page 14

Using HiperDispatch

We have implemented HiperDispatch on all of the z/OS images on our System z10 EC platform.

As you begin to plan to implement HiperDispatch in your environment, we recommend that you refer to the following two technical documents that are available at www.ibm.com/support/techdocs/:

- TD104518 — z/OS Positioning Software for the z10 EC Server
- WP101229 — z/OS: Planning Considerations for HiperDispatch Mode

After we installed the prerequisite software identified in the TD104518 technical document, we enabled HiperDispatch by including the new value `HIPERDISPATCH=YES` in an `IEAOPTxx` member and used the `SET OPT=xx` command to begin using HiperDispatch. This results in the following message:

```
IRA8601 HIPERDISPATCH MODE IS NOW ACTIVE
```

However, after HiperDispatch is started, we found that there is no obvious means to determine whether HiperDispatch mode is still active hours or days later without reviewing the syslogs. Over time, we found the following ways to tell:

- **RMF™ Monitor III Data Portal for z/OS:** We looked at a CPC Report for a z/OS image. There are three new columns that show the number of logical processors with high, medium, and low affinity to the LPAR. If these are marked N/A, then you know that HiperDispatch is not currently active for the LPAR. If numeric values appear, then HiperDispatch is currently active.
- **RMF Post Processor Report:** There is a field, named `HIPERDISPATCH`, at the top of the CPU Activity report that indicates whether HiperDispatch was active during the interval.

Figure 1 on page 8 shows an example of the RMF Post Processor CPU Activity report for one of our systems where HiperDispatch was active.

C P U A C T I V I T Y										
z/OS V1R9				SYSTEM ID Z2			START 04/16/2008-15.00.00		PAGE 1	
				RPT VERSION V1R9 RMF			END 04/16/2008-15.30.00		INTERVAL 000.30.00	
				E56 SEQUENCE CODE 00000000000699FF			H I P E R D I S P A T C H = Y E S		CYCLE 0.100 SECONDS	
0---CPU---				----- TIME %			LOG PROC		--I/O INTERRUPTS--	
NUM	TYPE	ONLINE	LPAR BUSY	MVS BUSY	PARKED	SHARE %	RATE	% VIA	TPI	
0	CP	100.00	17.59	17.52	0.00	100.0	414.1	4.73		
1	CP	100.00	50.72	50.69	0.00	100.0	1407	6.42		
2	CP	100.00	15.18	15.12	0.00	100.0	275.4	4.77		
3	CP	100.00	18.59	18.53	0.00	100.0	347.7	4.67		
4	CP	100.00	26.36	26.29	0.00	100.0	604.8	6.84		
5	CP	100.00	47.69	47.86	0.00	52.6	982.8	8.27		
6	CP	100.00	0.37	94.77	99.45	0.0	0.00	0.00		
7	CP	100.00	0.08	100.0	99.77	0.0	0.00	0.00		
8	CP	100.00	0.07	100.0	99.78	0.0	0.00	0.00		
9	CP	100.00	0.07	100.0	99.78	0.0	0.00	0.00		
TOTAL/AVERAGE			17.67	29.47		552.6	4032	6.50		

Figure 1. Example of an RMF Post Processor CPU Activity report

- **IBM Tivoli OMEGAMON® XE on z/OS:** OMEGAMON XE on z/OS Version 4.1.0, with PTFs and a related workstation Interim Fix applied (PTFs UA39283 and UA39284, APARs OA23220 and OA23223) provides support for HiperDispatch through the workstation-based Tivoli Enterprise Portal (TEP) interface and the OMEGAMON 3270-based interface.

Essentially, both the 3270-based and the TEP-based interfaces provide the same HiperDispatch status and statistics. These include the following information:

- The LPAR's current HiperDispatch status, as On, Off, or n/a. The n/a value indicates that the required level of operating system and hardware support is not available on the current system.
- The name of the LPAR, LPAR cluster, and LPAR group, if available
- The LPAR current, minimum, and maximum weights (IRD) for standard CPs, zAAPs, and zIIPs.
- For each logical processor, grouped by standard CP, zAAP, and zIIP:
 - Logical CPU ID
 - HiperDispatch priority, as High, Medium, or Low
 - Physical processor share guaranteed to the logical processor, as a percentage
 - Physical processor dispatch utilization, as a percentage
 - Physical processor LPAR overhead, as a percentage
 - HiperDispatch status, as Online, Offline, Parked, Park Pending, or Reserved

Using z/OS Capacity Provisioning

The System z10 EC platform introduces just-in-time deployment of additional computing capacity, known as Capacity on Demand (CoD). The new functions are designed to provide more flexibility and to make it easier to dynamically change capacity when business requirements dictate. For example, additional capacity can be dynamically activated using granular activation controls directly from the management console of the z10 EC, without the need to interact with IBM Support.

The Capacity on Demand architecture implemented in the System z10 EC provides more flexibility, granularity, and responsiveness than previous implementations for both the customer and for IBM. In addition, this architecture provides an enhanced set of Capacity on Demand Application Programming Interfaces (APIs) for use by systems management and automation software.

z/OS Capacity Provisioning is delivered as part of the z/OS MVS Base Control Program (BCP) component and includes the following components:

- Capacity Provisioning Control Center (CPCC)—the workstation code
The CPCC, installed on a workstation, is the graphical user interface to the Capacity Provisioning Manager. Through this interface, administrators work with provisioning policies and domain configurations and can transfer these to the Capacity Provisioning Manager.
- Capacity Provisioning Manager (CPM)—the z/OS server program
The CPM helps you manage the general and special purpose processor capacity (CP, zAAP, and zIIP) of the System z10 EC platform running one or more instances of the z/OS operating system. The CPM uses the enhanced set of Capacity on Demand Application Programming Interfaces (APIs).
The z/OS Capacity Provision Manager can be configured to manually provision capacity, via operator commands, and to autonomically provision capacity based on real time feedback from IBM Workload Manager (WLM).

The following topics discuss our implementation and deployment experiences with the z/OS Capacity Provisioning solution. For complete product details, see *z/OS MVS Capacity Provisioning User's Guide*, SC33-8299. Also, the IBM Redbook, *IBM System z10 Enterprise Class Capacity on Demand*, SG24-7504, is available at www.redbooks.ibm.com/abstracts/sg247504.html

Setting up Capacity Provisioning

We relied on *z/OS MVS Capacity Provisioning User's Guide*, SC33-8299 to guide us through our initial installation and setup. We found that the *User's Guide* clearly and adequately articulates each installation task, so we will not reiterate them here. However, we will use this opportunity to present an additional customization step that we performed in order to provide a higher level of availability for the RMF Distributed Data Server (DDS) which, in turn, provided higher availability for our Capacity Provisioning deployment.

Specifically, since the RMF DDS is a single data collection point for all of the RMF data gatherers within a sysplex, we needed a way to ensure that the RMF data gatherers could dynamically find the RMF DDS so that the DDS could be started (or restarted) anywhere in the sysplex, either manually, by a systems automation package, or by ARM.

Figure 2 on page 10 provides a visual representation of the Capacity Provisioning components and their respective relationship to one another. The TCP/IP and OMPROUTE components are intentionally omitted for brevity.

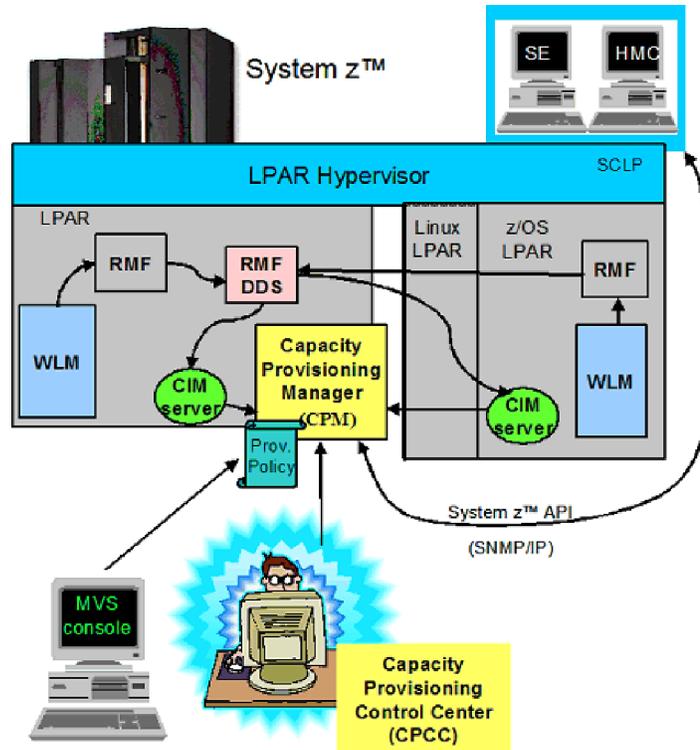


Figure 2. Capacity Provisioning components

To provide a higher level of availability for the RMF DDS, we performed the following steps:

1. Set up the necessary RACF authorizations for the RMF DDS started procedure to issue the SIOCSVIPA IOCTL command via the MODDVIPA utility.
2. Configure TCP/IP and OMROUTE profiles across the sysplex to support a new application activated dynamic VIPA (DVIPA).
3. Configure the CIMSERVICES to specify the DVIPA that the RMF DDS would be activating.
4. Modify the RMF DDS procedure to dynamically create the dynamic VIPA upon startup, as well as delete it upon an orderly shutdown.

The following topics describe these steps in more detail and illustrate the respective configuration statements.

RACF authorizations

Authorize the GPMSSERVE started procedure so that it can call the MODDVIPA utility.

1. Issue the following RDEFINE command for each system image (*sysname*) in the sysplex:

```
RDEFINE SERVAUTH (EZB.MODDVIPA.sysname.tcpname) UACC(NONE)
```

For example:

```
RDEFINE SERVAUTH (EZB.MODDVIPA.TPN.TCPIP) UACC(NONE)
RDEFINE SERVAUTH (EZB.MODDVIPA.JC0.TCPIP) UACC(NONE)
RDEFINE SERVAUTH (EZB.MODDVIPA.JB0.TCPIP) UACC(NONE)
:
RDEFINE SERVAUTH (EZB.MODDVIPA.Z0.TCPIP) UACC(NONE)
```

2. Issue the following PERMIT command for each system image (*sysname*) in the sysplex:

```
PERMIT EZB.MODDVIPA.sysname.tcpname ACCESS(READ) CLASS(SERVAUTH) ID(user1)
```

For example:

```
PERMIT EZB.MODDVIPA.TPN.TCPIP ACCESS(READ) CLASS(SERVAUTH) ID(GPMSERVE)
PERMIT EZB.MODDVIPA.JC0.TCPIP ACCESS(READ) CLASS(SERVAUTH) ID(GPMSERVE)
PERMIT EZB.MODDVIPA.JB0.TCPIP ACCESS(READ) CLASS(SERVAUTH) ID(GPMSERVE)
PERMIT EZB.MODDVIPA.Z0.TCPIP ACCESS(READ) CLASS(SERVAUTH) ID(GPMSERVE)
:
:
PERMIT EZB.MODDVIPA.J80.TCPIP ACCESS(READ) CLASS(SERVAUTH) ID(GPMSERVE)
```

3. Issue the following SETROPTS command:

```
SETROPTS RACLIST(SERVAUTH) REFRESH
```

TCP/IP profile definitions

Reserve the dynamic VIPA address in each TCP/IP profile so that the GPMSERVE started procedure can activate it when the started procedure is started in any z/OS image. Note that the VIPARANGE statement must be contained within the TCP/IP profile's VIPADYNAMIC block.

```
VIPADYNAMIC
;
;----- RMF DDS (GPMSERVE) HA DVIPA
VIPARANGE DEFINE 255.255.255.255 172.31.1.1
;
ENDVIPADYNAMIC
```

Dynamic routing—OMPROUTE profile definition

Dynamic routing is required in order for this deployment to provide the necessary high availability connectivity. We use OSPF. Therefore we defined the following OSPF interface statement in each z/OS image's OMPROUTE profile across our sysplex:

```
;
; Dynamic VIPA Interface added for GPMSERVE DVIPA support
;
OSPF_Interface
  IP_Address           = 172.31.1.1
  NAME                 = IGNORED
  Subnet_Mask          = 255.255.0.0
  MTU                  = 65535
  Advertise_VIPA_Routes = HOST_ONLY
  Cost                 = 1
  Subnet               = NO
  Attaches_To_Area    = 1.1.1.1;
```

CIMServer envar file

The CIMServer also needs to maintain communication to the GPMSERVER and, therefore, needs to be configured to specify the same application activated DVIPA.

```
RMF_CIM_HOST=172.31.1.1
```

GPMSERVE proc modification

The final step involves modifying the GPMSERVE started procedure to invoke the MODDVIPA utility so that it will create the dynamic VIPA.

```
//GPMSERVE PROC MEMBER=HS
//*      PARM='TRAP(ON),ENVAR(ICLUI_TRACETO=STDERR)&MEMBER'
//*
//*****
//*
//* Cleanup: *
//* this step will delete the application activated DVIPA prior *
//* to creating it for the case where the GPMSERVE ASID *
//* had not previously ended normally/cleanly. *
//* RC=8 is expected if the DVIPA was not in use *
//*
```

```

//DELDVP EXEC PGM=MODDVIPA,REGION=0K,TIME=1440,
//          PARM='POSIX(ON) ALL31(ON)/-p TCPIP -d 172.31.1.1'
//*
/**----- create the DVIPA -----*
/**
//TCPDVP EXEC PGM=MODDVIPA,REGION=0K,TIME=1440,
//          PARM='POSIX(ON) ALL31(ON)/-p TCPIP -c 172.31.1.1'
//*
/**-----*
/**
//STEP1 EXEC PGM=GPMDDSRV,REGION=0M,TIME=1440,
//          PARM='TRAP(ON)/&MEMBER'
//GPMINI DD DISP=SHR,DSN=SYS1.SERBPWSV(GPMINI)
//GPMHTC DD DISP=SHR,DSN=SYS1.SERBPWSV(GPMHTC)
//CEEDUMP DD DUMMY
//SYSPRINT DD DUMMY
//SYSOUT DD DUMMY
/**-----*
/**          delete the DVIPA upon exit
/**
//TCPDVP EXEC PGM=MODDVIPA,REGION=0K,TIME=1440,
//          PARM='POSIX(ON) ALL31(ON)/-p TCPIP -d 172.31.1.1'
/**-----*
/**
//          PEND

```

Capacity Provisioning in action

After our Capacity Provisioning policy was installed and activated, the Capacity Provisioning Manager began monitoring our z10 and provisioned additional capacity.

The following message capture illustrates how CPM added and removed up to three zAAPs, as well as performed several model conversions, taking our z10 from a model 719 up to a model 723:

CPM added 1 zAAP:

```

14.49.30 S0086030 BPXM023I (CPOSRV) 341
341 CP04108I Activation of resources for CPC H91 successfully initiated:
341 model 719 (0/0) with 1 zAAPs and 0 zIIPs
14.50.27 S0086030 BPXM023I (CPOSRV) CP03030I Command completed successfully for CPC H91

```

CPM removed 1 zAAP:

```

19.12.50 S0086030 BPXM023I (CPOSRV) 086
086 CP04109I Deactivation of resources for CPC H91 successfully initiated:
086 model 719 (0/0) with 0 zAAPs and 0 zIIPs
19.13.49 S0086030 BPXM023I (CPOSRV) CP03032I Command completed successfully for CPC H91

```

CPM added 3 zAAPs:

```

23.12.50 S0086030 BPXM023I (CPOSRV) 524
524 CP04108I Activation of resources for CPC H91 successfully initiated:
524 model 719 (0/0) with 1 zAAPs and 0 zIIPs
23.13.46 S0086030 BPXM023I (CPOSRV) CP03030I Command completed successfully for CPC H91
23.20.46 S0086030 BPXM023I (CPOSRV) 369
369 CP04108I Activation of resources for CPC H91 successfully initiated:
369 model 719 (0/0) with 2 zAAPs and 0 zIIPs
23.21.43 S0086030 BPXM023I (CPOSRV) CP03030I Command completed successfully for CPC H91
23.28.43 S0086030 BPXM023I (CPOSRV) 903
903 CP04108I Activation of resources for CPC H91 successfully initiated:
903 model 719 (0/0) with 3 zAAPs and 0 zIIPs
23.29.41 S0086030 BPXM023I (CPOSRV) CP03030I Command completed successfully for CPC H91

```

CPM added 1 CP: The z10 model is now a 720:

```

23.36.41 S0086030 BPXM023I (CPOSRV) 967
967 CP04108I Activation of resources for CPC H91 successfully initiated:
967 model 720 (1/0) with 3 zAAPs and 0 zIIPs
23.37.38 S0086030 BPXM023I (CPOSRV) CP03030I Command completed successfully for CPC H91

```

```

CPM added 1 CP: The z10 model is now a 721:
23.44.38 S0086030 BPXM023I (CPOSRV) 059
059 CP04108I Activation of resources for CPC H91 successfully initiated:
059 model 721 (2/0) with 3 zAAPs and 0 zIIPs
23.45.33 S0086030 BPXM023I (CPOSRV) CP03030I Command completed successfully for CPC H91

CPM added 1 CP: The z10 model is now a 722:
23.52.33 S0086030 BPXM023I (CPOSRV) 314
314 CP04108I Activation of resources for CPC H91 successfully initiated:
314 model 722 (3/0) with 3 zAAPs and 0 zIIPs
23.53.30 S0086030 BPXM023I (CPOSRV) CP03030I Command completed successfully for CPC H91

CPM added 1 CP: The z10 model is now a 723:
03.02.50 S0086030 BPXM023I (CPOSRV) 840
840 CP04108I Activation of resources for CPC H91 successfully initiated:
840 model 723 (4/0) with 3 zAAPs and 0 zIIPs
03.03.51 S0086030 BPXM023I (CPOSRV) CP03030I Command completed successfully for CPC H91

```

Using Parallel Sysplex InfiniBand coupling facility links

Parallel Sysplex InfiniBand (PSIFB) links are the latest coupling facility links. They were announced on 26 February 2008, in US hardware announcement 108-154, *IBM System z10 Enterprise Class – The forward-thinking mainframe for the twenty-first century*, as well as in the 6 May 2008 US hardware announcement 108-269, *IBM System z10 Enterprise Class: Helping to meet global 24x7 demands for information services with improvements for Internet access and coupling*.

For a brief description of the capabilities and advantages of PSIFB links, see the PSIFB Web page at www.ibm.com/systems/z/advantages/pso/ifb.html.

We have implemented two PSIFB coupling facility links between our z10 EC CPC and each of our z10 BC CPCs, and we have also connected the two z10 BC CPCs with two PSIFB links.

With PSIFB, each physical fiber connection can carry multiple coupling facility connections, known as CIB channels. We are exploiting this capability by defining multiple CIB channels on each PSIFB fiber.

We have also defined multiple CIB channels for both our production sysplex and our test sysplex over these same fibers.

Sound confusing? There is a great deal of information available on the PSIFB Web page to help you decide why and where you can use PSIFB links and how to implement them.

Configuring OSA-Express3 multi-port support for IP

We used the following documentation to configure the IBM Open Systems Adapter-Express3 (OSA-Express3) multi-port support for IP:

- *z/OS Communications Server: IP Configuration Guide*/OS Communications Server: IP Configuration Guide
- *z/OS Communications Server: IP Configuration Reference*/OS Communications Server: IP Configuration Reference

With OSA-Express2 hardware, only one port was available per CHPID. The new OSA-Express3 Gigabit Ethernet (GbE) cards now provide up to 4 ports per card. The OSA-Express3 hardware is a dual density card that supports two CHPIDs,

with each CHPID supporting two ports. We will be describing the support that was added to z/OS Communications Server to enable use of the second port.

The existing support for sharing OSA ports between images has not changed. On our CPCs, we share each OSA between all images as a matter of test coverage. z/OS V1R10 added a new PORTNUM keyword that is used in the TRLE definition. If the PORTNUM keyword is not present in the TRLE definition, it will default to using port 0. To use a port other than 0, the PORTNUM keyword must be specified.

In our existing configuration, we had our OSA-Express3 feature configured to support the default port 0, as:

```
SUBNET71 TRLE READ=(0920),WRITE=(0921),LNCTL=MPC,DATAPATH=(0922),
          MPCLEVEL=QDIO,PORTNAME=THUMP920,PORTNUM=0
```

We created a new TRLE, specifying a new, unique PORTNAME and PORTNUM=1. The PORTNAME specifies which port will be used by the OSA, as in this example:

```
SUBNT712 TRLE READ=(0926),WRITE=(0927),LNCTL=MPC,DATAPATH=(0928),
          MPCLEVEL=QDIO,PORTNAME=THUMP921,PORTNUM=1
```

The DISPLAY NET command now also displays the PORTNUM value in the command response:

```
IST097I DISPLAY ACCEPTED
IST075I NAME = SUBNT712, TYPE = TRLE
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST087I TYPE = LEASED           , CONTROL = MPC , HPDT = YES
IST1954I TRL MAJOR NODE = PET71NT2
IST1715I MPCLEVEL = QDIO      MPCUSAGE = SHARE
IST2263I PORTNAME = THUMP921  PORTNUM = 1  OSA CODE LEVEL = 070
IST1577I HEADER SIZE = 4096 DATA SIZE = 0 STORAGE = ***NA***
IST1221I WRITE DEV = 0927 STATUS = ACTIVE   STATE = ONLINE
IST1577I HEADER SIZE = 4092 DATA SIZE = 0 STORAGE = ***NA***
IST1221I READ  DEV = 0926 STATUS = ACTIVE   STATE = ONLINE
IST1221I DATA DEV = 0928 STATUS = ACTIVE   STATE = N/A
```

In the TCP/IP profile, we added the following DEVICE and LINK statements to support the second port:

```
DEVICE THUMP921 MPCIPA  NONROUTER  AUTORESTART
LINK OSA9201  IPAQNET  THUMP921  VMAC
```

Testing the expedited duplex completion protocol

CFLEVEL 16 was installed in our hardware environment as part of the EC driver upgrade to Driver 76D on our z10 EC server, H91. Our z10 BC server, M31, also came with Driver 76D and, thus, its coupling facility LPARs also operate at CFLEVEL 16.

One major change in CFLEVEL 16 is the *expedited duplex completion protocol* which is intended to provide system-managed duplexing performance enhancements by allowing one of architected synchronous duplexing signals that are exchanged between the primary and secondary copies of a system-managed duplexed structure to complete asynchronously. This is expected to provide improvements in service times for synchronous coupling facility structure duplexing requests.

The expedited duplex completion protocol support is integrated in z/OS V1R10 (HBB7750). APAR OA25130 for z/OS V1R6 through z/OS V1R9 (HBB7709 through

HBB7740) rolls down support for CFLEVEL 16 exploitation of the expedited duplex completion protocol. In addition, APAR OA25130 provides critical fixes for this support for z/OS V1R10 (HBB7750).

This support also introduces a switch to control the use of the expedited duplex completion protocol. The protocol is disabled by default and can be explicitly enabled or disabled by using the SETXCF FUNCTIONS command, as in the following examples:

```
SETXCF FUNCTIONS,ENABLE=DUPLEXCF16
IXC373I XCF / XES OPTIONAL FUNCTIONS ENABLED:
      DUPLEXCF16
```

```
SETXCF FUNCTIONS,DISABLE=DUPLEXCF16
IXC373I XCF / XES OPTIONAL FUNCTIONS DISABLED:
      DUPLEXCF16
```

In order to persist across IPLs, you can also enable or disable the expedited duplex completion protocol in the COUPLExx parmlib member, as in the following examples:

```
FUNCTIONS ENABLE(DUPLEXCF16)
```

```
FUNCTIONS DISABLE(DUPLEXCF16)
```

The SETXCF FUNCTIONS command has a system scope and should be issued to every exploiting system in a Parallel Sysplex. Downlevel systems that do not support the expedited duplex completion protocol (such as z/OS V1R6 through z/OS V1R9 systems without APAR OA25130) can coexist in the same Parallel Sysplex as those systems that do support the new protocol. However, only CF requests originating from the enabled systems will be able to take advantage of the new protocol. Therefore, it is recommended that all systems be enabled to take advantage of the expedited duplex completion protocol.

The D XCF,COUPLE command shows whether the function has been enabled. If enabled, the OPTIONAL FUNCTION STATUS field shows ENABLED, as in this example:

```
D XCF,COUPLE
IXC357I 11.09.16 DISPLAY XCF
:
OPTIONAL FUNCTION STATUS:
      FUNCTION NAME          STATUS      DEFAULT
      DUPLEXCF16            ENABLED    DISABLED
```

The system-managed duplexing performance enhancements in CFLEVEL 16 can only be realized if both copies of a duplexed structure reside on CFs that have CFLEVEL 16 installed. The first connector that supports the expedited duplex completion protocol that connects to a structure instance duplexed between two CFLEVEL 16 CFs will set flags in both copies of the structure to indicate that the structure is enabled for the new protocol. To ensure that all connectors can exploit the new protocol, we suggest to stop and restart duplexing for each structure duplexed between two CFLEVEL 16 CFs after the DUPLEXCF16 function is dynamically enabled via the SETXCF FUNCTIONS command.

Once the protocol is enabled in structures that are duplexed between two CFLEVEL 16 CFs, the D XCF,STRUCTURE command will then show:

```
D XCF,STR,STRNM=MQGTCICS
IXC360I 11.37.07 DISPLAY XCF
:
CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
```

```
-----  
CSQEMQGTCSQ101 01 00010419 Z1 CSQ1MSTR 0244 ACTIVE NEW,OLD  
CSQEMQGTCSQ303 03 00030181 Z3 CSQ3MSTR 0151 ACTIVE NEW,OLD
```

ENABLED FOR EXPEDITED DUPLEX COMPLETION PROTOCOL

We ran several stress tests in the various supported CFCC environments in our configuration and compared the coupling facility performance data of structures duplexed between two CFLEVEL 16 CFs against the baseline we had previously taken to make sure we saw no degradation in performance.

In the final configuration, when we had CFLEVEL 16 on both CF1 and CF4, we observed noticeable improvements in our CF service times between the two CFs.

One of the things we tested was the migration of some of our CFs to the new CFLEVEL 16 and a mixture of different supported CFLEVELs. Table 12 on page 315 shows the different configurations we tested. In addition, we paid particular attention to whether the new CFLEVEL requires any structure size increases. Since CFLEVEL 16 does provide a structure increment size change from 512K to 1M, as part of our migration to the new CFLEVEL, we downloaded and ran the recommended CFSizer program, available at www.ibm.com/systems/support/z/cfsizer/.

The program ran against all structures in our CFRM policy. We strongly recommend that you visit this Web site if you are migrating to a higher level of CFCC. We paid particular attention to the structures running on CFCC level 16 coupling facilities. We found that for some of our structures, we did have to increase structure sizes by up to three storage increment sizes (3M bytes) based on the CFSIZER recommendations for our final configuration.

Chapter 3. Migrating from SMF data set recording to log stream logging

Beginning with z/OS V1R9, you can configure z/OS MVS Systems Management Facility (SMF) to use the system logger to write records to log streams. This topic gives an overview of the benefits of migrating SMF to log streams, reviews configuration considerations, and takes you through our migration process.

Advantages of recording SMF data in log streams

The SMF exploitation of system logger services provides many benefits, such as:

- SMF performance improvements with log stream logging
- Management of SMF data on a per log stream basis
- SMF data reliability with log stream logging
- Browsing (dumping) SMF data
- Data retention and deletion on a per log stream basis

SMF performance improvements with log stream logging

With SMF support for log streams, data is captured faster than if using MANx data sets. In addition, since system logger manages data flow and available storage, there is no concern over buffer overrun due to MANx data set switch processing.

This support also allows for more efficient dumping, as dump processing can be run against a given log stream which might hold just a subset of your SMF data (as described in “Management of SMF data on a per log stream basis”).

Management of SMF data on a per log stream basis

SMF allows you to determine which SMF records to send to a given log stream on a per system basis. This allows more customization of how SMF records are managed and grouped. You can also choose to merge SMF data from multiple systems into a single log stream to give a sysplex view, isolate certain SMF record types to a particular log stream, or group certain record types together as best suits your environment.

This ability to filter SMF data on a log stream basis makes dump processing more efficient as well. Dump programs can be run against the log stream that is holding the SMF record types in which you are interested; it is not necessary to crawl through unrelated data.

SMF data reliability with log stream logging

System logger protects an exploiter’s data against a single point of failure. By using a log stream to store real-time data, SMF takes advantage of the data reliability provided – further, this mechanism is managed by Logger processing, and is of no functional impact to the exploiting application.

To explain this further, we must understand the flow of data once it is written to a log stream. Here is a brief overview:

Data written to a log stream is kept in interim storage until it is offloaded to DASD log data sets. The actual storage mediums used depend on the type of log stream:

- Coupling facility (CF) structure log streams store log data in a coupling facility list structure.
- DASD-only log streams store log data in data space local buffers.

While data is in interim storage, Logger manages a duplex copy of the data. For DASD-only log streams, log stream data is duplexed to staging data sets. CF structure based log stream data may be duplexed to a variety of storage mediums (staging data sets, local buffers, or another XES structure via System Managed duplexing). This duplex copy of data serves as a backup should the primary storage copy be lost.

Once an offload is triggered (caused by a high data threshold being reached, for example), data is written out to more permanent DASD log data sets and scratched from interim storage.

It is important to note that the actual location of log data is of no concern to an application attempting to read it – Logger browse (read) processing manages this overhead, and it is abstract to the exploiter.

This management helps ensure data is recoverable should a system failure or disaster occur. We'll touch on this topic again briefly when the test environment is discussed in "Determining Log Stream configuration for the Integration Test environment".

For detailed information, see chapter 9 in *z/OS MVS Setting Up a Sysplex*.

Browsing (dumping) SMF data

When it is necessary to dump SMF data, you use the IFASMF DL dump job. Simply specify the appropriate log stream name or names along with the dates and times in which you are interested. The IFASMF DL program dumps the log stream data to sequential data sets which you can then use to produce reports.

Data retention and deletion on a per log stream basis

System logger manages data retention on an individual log stream basis. This allows you to determine how long to keep record types in a particular log stream. For SMF log stream data, you control how long data is to be kept by specifying the REDPD and AUTODELETE parameters on the log stream definition, as follows:

RETPD(*days*)

Specifies the number of days that SMF data should be retained in the log stream. After this period expires, data is eligible for deletion. For example, specifying RETPD=365 will cause data to be retained for one year before it can be deleted.

AUTODELETE(YES|NO)

When AUTODELETE(YES) is specified, system logger automatically deletes log data for which the retention period has expired. If AUTODELETE(NO) is specified, SMF data will not be deleted automatically when it becomes eligible for deletion (that is, when its retention period has expired).

Configuration considerations for log stream logging

We took several considerations into account when planning our configuration for log stream logging. These included:

- Choosing CF structure log streams or DASD-only log streams for SMF data
- Determining SMF log stream configuration for the test environment
- Estimating interim storage, offload, and staging data set sizes
- CF structure and log stream definitions
- SMFPRMxx member definition

Choosing CF structure log streams or DASD-only log streams for SMF data

There are many factors to consider when deciding whether to use a CF structure log stream or DASD-only log stream. In “SMF data reliability with log stream logging” on page 17, we discussed differences in log stream type interim storage and data flow. Another important consideration in planning is the scope (single system versus multi-system) of the SMF data to record in a given log stream:

- CF structure log streams can be connected and written to from multiple systems concurrently; so, to write SMF data from multiple systems into a single log stream you must use a CF log stream.
- If each log stream is going to be written to by a single system, then you can choose to use CF or DASD-only log streams.

Note: DASD-only log streams can *only* be connected to from one system at a time.

For our testing, we chose to use both CF structure and DASD-only log streams. However, there are many other factors to consider when choosing between the two types of log streams and your decision should be based on your environment. If you are not familiar with system logger and log streams, see *z/OS MVS Setting Up a Sysplex* for more information.

Determining SMF log stream configuration for our test environment

For our test environment, we are using both CF structure and DASD-only log streams. Thus, we grouped the SMF record types for each type of log stream, as follows:

- DASD-only log streams (one per system)
 - SMF type 0-29 records in one log stream
 - SMF type 70-79 records in a second log stream
 - SMF type 30 records only in a third log stream (because these records are cut at a high rate in our environment)
 - All other record types will go to a default log stream
- CF structure log stream (written to by all systems)
 - SMF type 88 records from all systems in one log stream

For example, on system Z4, we created the following log streams:

```
IFASMF.SMF0T029.Z4
IFASMF.SMF70T79.Z4
IFASMF.SMF30.Z4
```

IFASMF.SMFDFLT.Z4
IFASMF.SMF88.PLEX2

We also created an IFASMF_SMF88 CF structure for the IFASMF.SMF88.PLEX2 log stream.

The topic of planning log stream configuration is discussed in detail in various publications, such as *z/OS MVS Setting Up a Sysplex* under the topic, “Determine Which Log Streams Map to Which Coupling Facility Structures,” and the IBM Redbook, *Systems Programmer’s Guide to: z/OS System Logger*.

Estimating interim storage, offload, and staging data set sizes

The following topics describe our estimations of interim storage, offload, and staging data set sizes.

Interim storage for DASD-only log streams

The IBM Redbook, *Systems Programmer’s Guide to: z/OS System Logger*, describes this best, as:

For DASD-only log streams, system logger uses local buffers in system logger’s data space for interim storage. It then duplexes the data simultaneously to staging data sets. Unlike CF structure-based log streams, you have no control over this processing; system logger always uses this configuration for DASD-only log streams.

Interim storage for CF structure log streams

For IFASMF.SMF88.PLEX2, our CF structure-based log stream, our interim storage is a CF structure. To calculate the appropriate structure sizes for most system logger exploiters, use the CFSizer tool available at www.ibm.com/systems/support/z/cfsizer/.

At the time of this article, SMF was not yet an exploiter of the CFSizer tool; therefore, we had to come up with structure sizes for our installation on our own.

We did not want our SMF type 88 records to be sitting in the CF for long. Therefore, we wanted a small CF structure. To determine a good structure size, we looked at how much SMF type 88 data we were writing per day and then estimated what size structure to create.

For example, on this test system, we were writing six cylinders worth of SMF type 88 data per day. Since this is just estimation, we assumed that we write the same amount of SMF type 88 data on the rest of the images in this sysplex.

In this test sysplex we have four systems. We are going to write SMF type 88 data to the IFASMF.SMF88.PLEX2 CF structure log stream from each of these four systems. Therefore, we multiplied the number of cylinders SMF used for type 88 data (six cylinders) by four to estimate the amount of space that the IFASMF_SMF88 structure needs per day:

Structure	Size (cyls)	Size (M bytes)
IFASMF_SMF88	24 (6 × 4)	17

This means that a 17 M byte IFASMF_SMF88 structure should hold approximately a day’s worth of SMF type 88 data from all four systems. Since we do not want our SMF data sitting in the CF all day before being offloaded to DASD, we decided to use a smaller structure size than that.

Also, remember that, when sizing CF structures, not all of the space will be available for the log stream data. Some of it is overhead used by coupling facility control code (CFCC); system logger also uses space to store control information related to a given log stream. In our test environment, we observed overhead size to be roughly 8M bytes.

Because all of this data is going to be offloaded to DASD, there is no advantage to having a large structure size. We simply want the structure big enough so that the system logger is not offloading constantly or encountering frequent full conditions. We also wanted to account for spikes in IXGWRITE activity which potentially could also trigger a full condition. Based on this, along with our earlier observations, we decided to size our IFASMF_SMF88 structure to be 15M bytes.

For configurations where multiple SMF log streams are defined to the same structure (or you are collecting all SMF data in a single log stream), you would likely want to use a larger structure size. Again, there is no advantage to making the structure very large—because the SMF log streams are being used in a funnel-like manner, all data will be offloaded eventually. The structure size should generally be large enough to accommodate the peak level of write activity you are likely to encounter including short term spikes without encountering a structure or entry full condition—that is, an offload should be triggered by the HIGHOFFLOAD threshold value.

As we mentioned earlier, there is currently no CFsizer tool support for SMF. However, you may want to look at other system logger exploiter recommendations which have similar usage characteristics (that is, using log streams as a funnel for information) as an example, such as IMS. The IBM Redbook, *Systems Programmer's Guide to: z/OS System Logger*, talks about other system logger exploiters in detail.

Sizing for staging data sets

To ensure maximum recoverability, we decided to define IFASMF.SMF88.PLEX2, our CF structure type log stream, to always use staging data sets as a duplexing medium. This ensures that a hardened, failure-independent copy of data in interim storage exists on persistent media, protecting data against multiple failures.

We did not specify the staging data set size (STG_SIZE) for IFASMF.SMF88.PLEX2 so that system logger would use the default value, which is the amount specified in the SIZE parameter of the structure definition in the CFRM policy. Basically, we allocate a staging data set that is as large as the CF structure, 15M bytes.

Sizing for offload data sets

The main goal when setting up the offload data set is to ensure that the size is large enough to avoid frequent data set switches during offload processing. A rule of thumb to follow is to set the size large enough to handle multiple offloads and not cause a switch during the day.

The offload data sets were sized at 1000 cylinders each to begin with. Use the LS_SIZE parameter, specified in 4K blocks, to tell the system logger what size offload data sets to use for a log stream. To convert from cylinders to 4K blocks:

1 cylinder = 180 4K blocks

So, we defined LS_SIZE as 180000 (1000 × 180) blocks.

SMF type 88 records showed that 1000 cylinder offload data sets were large enough to handle a few days worth of SMF data stored in the IFASMF.SMF88.PLEX2 log stream.

CF structure and log stream definitions

Here are sample structure and log stream definitions we made in our CFRM and system logger policies.

Example of our SMF structure definition in the CFRM policy:

```
STRUCTURE NAME(IFASMF_SMF88)
SIZE(15360)
DUPLEX(ALLOWED)
PREFLIST(CFAA,CFAB)
```

Example of our SMF structure definition in the system logger policy:

```
DEFINE STRUCTURE NAME(IFASMF_SMF88)
LOGSNUM(1) MAXBUFSIZE(65276)
```

We chose 65276 as our MAXBUFSIZE value. The system logger documentation suggests that you use that size unless you need it to be bigger or you know what size you really need. SMF publications recommend a value between 33024 and 65532, so we will use the system logger recommendation.

This MAXBUFSIZE value results in an ELEMENT size of 256, rather than 512. SMF type 88 records showed that this size, along with the other configuration decisions we made, did not cause any structure or staging data set full events resulting in frequent offloads.

Example of the CF structure type log stream in the system logger policy:

```
DEFINE LOGSTREAM
NAME(IFASMF.SMF88.PLEX2) LS_SIZE(180000)
STRUCTNAME(IFASMF_SMF88)
HIGHOFFLOAD(60) LOWOFFLOAD(35)
AUTODELETE(YES) RETPD(2)
LOGGERDUPLEX(UNCOND)
STG_DUPLEX(YES)
DUPLEXMODE(UNCOND)
OFFLOADRECALL(NO)
```

We decided to use the default HIGHOFFLOAD(60) and LOWOFFLOAD(35) values. These log streams will be used primarily to write data and occasionally to retrieve (dump) it. This means that the log stream offload will begin at 60 percent full and offload data to the 35 percent full point. The capacity between the HIGHOFFLOAD point and the 100 percent full mark acts as a buffer to allow system logger to keep accepting new write requests while an offload is in progress. Depending on usage characteristics, you can use different values or increase the structure space available. It is important to look at performance related data (as we discuss in "Monitoring our SMF configuration" on page 27) and attempt to avoid structure full type conditions. This is important because if the structure runs out of available space (100 percent full), system logger will stop accepting new writes from applications until space can be made available via offload.

As mentioned previously, we decided to always duplex SMF data to staging data sets to ensure maximum recoverability of log data, so we set DUPLEXMODE(UNCOND) and STG_DUPLEX(YES).

For other SMF data types, we are using DASD-only log streams. Here is an example of the DASD-based type log stream:

```
DATA TYPE(LOGR)
DEFINE LOGSTREAM NAME(IFASMF.SMF30.Z4)
DASDONLY(YES)
```

```
STG_SIZE(12800)
LS_SIZE(180000)
AUTODELETE(YES)
RETPD(2)
HIGHOFFLOAD(60)
LOWOFFLOAD(35)
```

You might notice that we chose a staging data set size of 50M bytes (12800 4K blocks). Similar to our CF structure size, this was based on analysis of the amount of data we were writing and our write characteristics. Based on our requirements for this particular log stream, we decided SMF data should be retained for two days. Thus, we defined our log streams to use AUTODELETE(YES) and RETPD(2).

SMFPRMxx member definition

To activate SMF upon system IPL, we made the following changes to our SMFPRMxx parmlib member:

```
RECORDING(LOGSTREAM),
DEFAULTLSNAME(IFASMF.SMFDFLT.&SYSNAME),
LSNAME(IFASMF.SMF0T029.&SYSNAME,TYPE(0:29)),
LSNAME(IFASMF.SMF30.&SYSNAME,TYPE(30)),
LSNAME(IFASMF.SMF70T79.&SYSNAME,TYPE(70:79)),
LSNAME(IFASMF.SMF88.PLEX2,TYPE(88)),
PROMPT(LIST),
```

```
/* Prompt parameter allows you to dynamically switch */
/* between logging and data set recording via SETSMF */
/* command */
```

We also left the DSNNAME statement for our MANx data sets in the parmlib member. By doing this, the MANx data sets are available if we ever need to dynamically switch back to SMF data set recording.

```
DSNNAME(SYS1.SMF.&SYSNAME..MANS,
        SYS1.SMF.&SYSNAME..MANT,
        SYS1.SMF.&SYSNAME..MANU,
        SYS1.SMF.&SYSNAME..MANV),
```

Migrating to SMF log stream logging

When we decided to exploit SMF log stream logging, we wanted to migrate systems one or a few at a time, in a manner that would not disturb the users of our SMF data. We started with a single system and, once we were satisfied, we switched the rest of our systems to use log streams, as well.

Prior to SMF log stream logging, whenever one of our MANx data sets filled up, our SMF data would be dumped into a new generation data group (GDG) data set. These data sets were named similar to:

```
SMFDATA.SMFZ4.G1503V00
```

where Z4 is the system name and G1503V00 represents the generation and version numbers.

Our goal was that, during the migration, the location of the SMF data and the data set names would not change. This would allow our end users to run their jobs to post process the SMF data without changing them or with very minor changes.

We considered the following two ways to accomplish this:

1. Dump SMF data when needed

This was probably the easiest method and the one we would suggest but it would have required our end users to run an additional job and possibly to make some minor changes to their jobs. Basically, whenever the end users wanted to look at SMF data, they would have to dump the data they are looking for from the log streams into a data set using the same name convention that their jobs currently expect as input. (See "Using the SWITCH SMF command and the run dump program" on page 26 for an example of the IFASMF DL dump program.)

2. Dump SMF data once a day

We can schedule a job to run once a day and dump the SMF data from the log streams into a data set. When naming this data set, we can use the same naming convention that the end users' jobs take as input. Thus, our end users will not have to change their post processing jobs.

For instance, we can use IBM Tivoli NetView® for z/OS automation facilities to submit a job everyday at 1:00 AM on each system. This job runs a REXX program (below) to figure out the previous day's date and creates the control cards for the SMF dump program. Then it executes the IFASMF DL program and dumps the SMF data out to a GDG data set.

Below is the sample REXX program, which we stored in OZ2.REXX(GETDATE):

```

/*** REXX ***/
SysID = MVSVAR('SYSNAME')

jday = DATE('Base')
jdayyest = jday - 1
jyear = LEFT(DATE('Standard',jdayyest,'Base'),4)
jdate = jyear || RIGHT(DATE('Days',jdayyest,'Base'),3,'0')

Queue "      LSNAME(IFASMF.SMF70T79." || SysID || ",OPTIONS(DUMP))"
Queue "      LSNAME(IFASMF.SMF0T029." || SysID || ",OPTIONS(DUMP))"
Queue "      LSNAME(IFASMF.SMF30." || SysID || ",OPTIONS(DUMP))"
Queue "      LSNAME(IFASMF.SMFDFLT." || SysID || ",OPTIONS(DUMP))"
Queue "      OUTDD(DUMPOUT,TYPE(0:255))"
Queue "      ABEND(NORETRY)"
Queue "      DATE(" || jdate || ", " || jdate || ")"
Queue "      START("0000")"
Queue "      END("2400")"

"PIPE STACK | PAD 80 | CHOP 80 | > DDNAME=SMFCNTL"

```

Below is the sample JCL to run the REXX program. The REXX program resides in member GETDATE in library OZ2.REXX.

```

//OZST JOB 'OZAN',MSGCLASS=A,CLASS=A
//*****
//* BUILD THE CONTROL CARDS FOR THE SMF DUMP PGM
//*****
//GETDATE EXEC PGM=IKJEFT01,
//      DYNAMNBR=50,
//      PARM='%GETDATE'
//SYSPROC DD DISP=SHR,DSN=OZ2.REXX
//SMFCNTL DD DISP=(NEW,PASS,DELETE),DSN=&&SMFCNTL,
//      SPACE=(TRK,(1,1)),UNIT=SYSDA,VOL=SER=,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=9040)
//SYSIN DD DISP=(NEW,PASS,DELETE),
//      SPACE=(TRK,(1,1)),UNIT=SYSDA,VOL=SER=,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=9040)
//SYSTSIN DD DUMMY
//DUMPOUT DD DUMMY
//SYSTSPRT DD SYSOUT=*
//*****
//* ALLOCATE THE NEXT GDG ENTRY
//*****
//ALLOC1 EXEC PGM=IEFBR14,COND=(4,LT)

```

```

//DUMPOUT DD DSN=OZ2.TEMP(+1),
//        DISP=(NEW,CATLG,DELETE),
//        DCB=(SMFDATA.MODEL.DSCB),
//        UNIT=LOGS,
//        SPACE=(CYL,(750,750))
//*****
//* DUMP THE SMF DATA
//*****
//DUMP1 EXEC PGM=IFASMF DL,COND=(4,LT)
//DUMPOUT DD DSN=OZ2.TEMP(+1),DISP=OLD,
//        SPACE=(CYL,(750,750),RLSE),
//        DCB=(SMFDATA.MODEL.DSCB)
//SYSIN DD DISP=(OLD,DELETE),DSN=&&SMFCTL
//SYSPRINT DD SYSOUT=*
//*****
//* NOTE THE NAME OF THE NEWEST GDG FOR FUTURE REFERENCE
//*****
//GDGLIST EXEC PGM=SMFGDG,COND=(4,LT)
//STEPLIB DD DSN=USER.LINKLIB,DISP=SHR,
//        VOL=SER=CMNSTC,UNIT=3390
//SYSUDUMP DD SYSOUT=*
//SMFGDG DD DSN=OZ2.TEMP(+1),DISP=SHR
//LOG DD DSN=OZ2.GDG.LIST,DISP=SHR

```

Switching from SMF data set recording to SMF log stream logging

Once the SMFPRM xx parmlib member is ready, there are a few different ways to switch:

1. IPL with the SMF parmlib member, such as the one described in “SMFPRM xx member definition” on page 23.
2. Run the SET SMF= xx command and specify the SMFPRM xx parmlib member to switch dynamically.
3. Run the SETSMF RECORDING(LOGSTREAM) command to switch dynamically.

All of these methods will generate an outstanding reply message. We replied with U to keep the options in the specified parmlib member.

The first time through, we dynamically switched using the second option: We issued the MVS system command SET SMF=Z4 and received the following:

```

IEE967I 07.12.02 SMF PARAMETERS 849
MEMBER = SMFPRMZ4
MULCFUNC -- DEFAULT
LISTDSN -- DEFAULT
STATUS(010000) -- DEFAULT
MAXDORM(3000) -- DEFAULT
DDCONS(YES) -- DEFAULT
LASTDS(MSG) -- DEFAULT
NOBUFFS(MSG) -- DEFAULT
INTVAL(30) -- DEFAULT
DUMPABND(RETRY) -- DEFAULT
REC(PERM) -- DEFAULT
ACTIVE -- DEFAULT
BUFSIZMAX(0256M) -- PARMLIB
BUFUSEWARN(80) -- PARMLIB
SYNCVAL(00) -- PARMLIB
SYS(EXITS(IEFUSI)) -- PARMLIB
SYS(EXITS(IEFUJV)) -- PARMLIB
SYS(EXITS(IEFU85)) -- PARMLIB
SYS(EXITS(IEFU84)) -- PARMLIB
SYS(EXITS(IEFU83)) -- PARMLIB
SYS(EXITS(IEFU29)) -- PARMLIB

```

```

SYS(EXITS(IEFUJI)) -- PARMLIB
SYS(EXITS(IEFACTRT)) -- PARMLIB
SYS(INTERVAL(SMF,SYNC)) -- PARMLIB
SYS(DETAIL) -- PARMLIB
SYS(TYPE(0,2,3,6:10,14,15,22:24,26,30,32,33,41,42,
    47:48,59,61:69,70:79,80:83,85,88,89,90:91,94,98,
    100:103,108,110, 115:117,120,130,134,148:151,161,
    200,244,245)) -- PARMLIB
SID(Z4) -- DEFAULT
JWT(2400) -- PARMLIB
MEMLIMIT(00512M) -- PARMLIB
DSNAME(SYS1.SMF.Z4.MANV) -- PARMLIB
DSNAME(SYS1.SMF.Z4.MANU) -- PARMLIB
DSNAME(SYS1.SMF.Z4.MANT) -- PARMLIB
DSNAME(SYS1.SMF.Z4.MANS) -- PARMLIB
PROMPT(LIST) -- PARMLIB
LSNAME(IFASMF.SMF70T79.Z4,TYPE(70:79)) -- PARMLIB
LSNAME(IFASMF.SMF30.Z4,TYPE(30)) -- PARMLIB
LSNAME(IFASMF.SMF0T029.Z4,TYPE(0:29)) -- PARMLIB
LSNAME(IFASMF.SMF88.PLEX2,TYPE(88)) -- PARMLIB
DEFAULTLSNAME(IFASMF.SMFDFLT.Z4) -- PARMLIB
RECORDING(LOGSTREAM) -- PARMLIB

```

*7187 IEE357A REPLY WITH SMF VALUES OR U

We replied U to the IEE357A message.

Next, we ran the D SMF command to verify that SMF is indeed using the log streams. The following is an example of the command response:

```

IFA714I 10.53.42 SMF STATUS 604
LOGSTREAM NAME          BUFFERS      STATUS
A-IFASMF.SMFDFLT.Z4    15069      CONNECTED
A-IFASMF.SMF0T029.Z4   7076      CONNECTED
A-IFASMF.SMF30.Z4      9935      CONNECTED
A-IFASMF.SMF70T79.Z4  56084     CONNECTED
A-IFASMF.SMF88.PLEX2   0         CONNECTED

```

Using the SWITCH SMF command and the run dump program

SMF also provides a new dump program for use with log streams, IFASMFDL. It can take multiple log streams as input and write its output to multiple data sets. For details on the IFASMFDL program, see *z/OS MVS System Management Facilities (SMF)*.

Here is an example of the JCL to execute the program for collecting SMF data:

```

//IFASMFDL JOB MSGLEVEL=(1,1),MSGCLASS=H,CLASS=A,REGION=0M,
// NOTIFY=&SYSUID
//DUMP1 EXEC PGM=IFASMFDL
//OUT1 DD DSN=0Z.SMF88.Z4,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(100,100),RLSE),UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        LSNAME(IFASMF.SMF30.Z4)
        LSNAME(IFASMF.SMF70T79.Z4)
        LSNAME(IFASMF.SMF0T029.Z4)
        LSNAME(IFASMF.SMFDFLT.Z4)
        OUTDD(OUT1,TYPE(0:255)),START(0000),END(2400)
//*

```

Overall, the routine for looking at SMF data is still the same:

1. Run the "SWITCH SMF" command to transfer the log stream data from the buffers into the appropriate log streams.
2. Run the IFASMFDL dump program to dump the SMF data.

Since we are now using the system logger to manage the offloading and archiving of our SMF data, we are not using the IEFU29L exit. If you want to, you can still use the combination of the IEFU29L exit and the SWITCH SMF command to handle the archiving of your SMF log stream data.

Monitoring our SMF configuration

As shown in “Switching from SMF data set recording to SMF log stream logging” on page 25, the D SMF command shows the log streams that SMF is using, the buffer sizes, and whether or not SMF is connected to the log streams. The D SMF,O command shows the options that SMF is currently using, just like the SET SMF=xx output.

We used the D LOGGER command to see the structures to which the log streams are connected, their status, and the number of connections:

D LOGGER,L,LSN=IFASMF.SMF30.Z4

```
IXG601I  10.55.46  LOGGER DISPLAY 631
INVENTORY INFORMATION BY LOGSTREAM
LOGSTREAM          STRUCTURE          #CONN  STATUS
-----          -
IFASMF.SMF30.Z4    *DASDONLY*          000001  IN USE
  SYSNAME: Z4
  DUPLEXING: STAGING DATA SET
  GROUP: PRODUCTION
```

NUMBER OF LOGSTREAMS: 000001

D LOGGER,L,LSN=IFASMF.SMF88.PLEX2

```
IXG601I  10.56.19  LOGGER DISPLAY 676
INVENTORY INFORMATION BY LOGSTREAM

LOGSTREAM          STRUCTURE          #CONN  STATUS
-----          -
IFASMF.SMF88.PLEX2  IFASMF_SMF88        000001  IN USE
  SYSNAME: Z4
  DUPLEXING: STAGING DATA SET
  GROUP: PRODUCTION
```

NUMBER OF LOGSTREAMS: 000001

We also used the D LOGGER command to display the staging data set that a log stream is using, its location, size, and other information:

D LOGGER,C,LSN=IFASMF.SMF30.Z4,D

```
IXG601I  10.56.57  LOGGER DISPLAY 685
CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM Z4
LOGSTREAM          STRUCTURE          #CONN  STATUS
-----          -
IFASMF.SMF30.Z4    *DASDONLY*          000001  IN USE
  DUPLEXING: STAGING DATA SET
  STGDSN: IXGLOGR.IFASMF.SMF30.Z4.PETPLEX2
  VOLUME=P2LG06  SIZE=012960 (IN 4K)  % IN-USE=003
  GROUP: PRODUCTION
  JOBNAME: SMF          ASID: 001B
  R/W CONN: 000000 / 000001
  RES MGR./CONNECTED: *NONE* / NO
  IMPORT CONNECT: NO
```

NUMBER OF LOGSTREAMS: 000001

D LOGGER,C,LSN=IFASMF.SMF88.PLEX2,D

```
IXG601I  10.57.17  LOGGER DISPLAY 697
```

```

CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM Z4
LOGSTREAM          STRUCTURE          #CONN  STATUS
-----
IFASMF.SMF88.PLEX2  IFASMF_SMF88          000001  IN USE
  DUPLEXING: STAGING DATA SET
  STGDSN: IXGLOGR.IFASMF.SMF88.PLEX2.Z4
  VOLUME=P2LG03  SIZE=003960 (IN 4K)  % IN-USE=001
  GROUP: PRODUCTION
  JOBNAME: SMF          ASID: 001B
  R/W CONN: 000000 / 000001
  RES MGR./CONNECTED: *NONE* / NO
  IMPORT CONNECT: NO

NUMBER OF LOGSTREAMS: 000001

```

For CF type log streams, we used the D LOGGER and D XCF commands to collect structure-related information:

D LOGGER,STR,STRN=IFASMF_SMF88

```

IXG601I 10.57.43  LOGGER DISPLAY 704
INVENTORY INFORMATION BY STRUCTURE
STRUCTURE          GROUP          CONNECTED
-----
IFASMF_SMF88      PRODUCTION
  IFASMF.SMF88.PLEX2          YES

```

NUMBER OF STRUCTURES: 000001

and

D XCF,STR,STRNM=IFASMF_SMF88

```

IXC360I 10.58.05  DISPLAY XCF 719
STRNAME: IFASMF_SMF88
STATUS: ALLOCATED
EVENT MANAGEMENT: POLICY-BASED
TYPE: LIST
POLICY INFORMATION:
  POLICY SIZE      : 15360 K
  POLICY INITSIZE : N/A
  POLICY MINSIZE  : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: N/A
  DUPLEX         : ALLOWED
  ALLOWREALLOCATE: YES
  PREFERENCE LIST: CF21      CF22
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
ACTIVE STRUCTURE
-----
  ALLOCATION TIME: 04/17/2008 10:53:08
  CFNAME        : CFAB
  COUPLING FACILITY: XXXXXXXX.IBM.02.0000000699FF
  PARTITION: 13  CPCID: 00
  ACTUAL SIZE   : 15360 K
  STORAGE INCREMENT SIZE: 512 K
  USAGE INFO    TOTAL    CHANGED  %
  ENTRIES:      305      5      1
  ELEMENTS:     25038    32      0
  PHYSICAL VERSION: C2423136 D57CB61F
  LOGICAL  VERSION: C2423136 D57CB61F
  SYSTEM-MANAGED PROCESS LEVEL: 8

```

```
DISPOSITION      : DELETE
ACCESS TIME      : 0
MAX CONNECTIONS  : 32
# CONNECTIONS    : 1
```

Another way to monitor your configuration is by post processing SMF type 88 records. Once you do SMF logging for a while (say, 24 hours), you can dump your data and create a System Logger Activity Report using the IXGRPT1 macro. For more information about the IXGRPT1 macro and the report that it generates, see *z/OS MVS System Management Facilities (SMF)*. Information about how to react to this data can be found in the IBM Redbook, *Systems Programmer's Guide to: z/OS System Logger*.

Since we estimated our structure sizes, we wanted to pay special attention to our sizing decisions. To monitor our structures, we looked at the following data in the IXGRPT1 report:

- **# Type 2 and 3 writes:** At least the Type 3 column should be zero. If not, then the structure size might need to be adjusted. Type 3 writes indicate a write request that was processed after a structure full condition was encountered.
- **# offload events:** If too frequent, the high and low offload thresholds or the structure size might need to be adjusted. Note, however, that offloads are not an indication of a problem. More interesting is the reason that the offload is triggered, so consideration should be given to other SMF values, such as the number of structure full events.
- **# structure full events:** This should be a rare occurrence, as well. If the value in this field is frequently greater than zero, consider adjusting the structure size and checking log stream performance data.
- **# DASD shifts:** These occur every time the system creates an offload data set. This should be a small percentage of the offload events. Otherwise, the offload data sets might be too small.
- **# staging threshold was reached:** If too frequent, the staging data set might be too small.
- **# staging data set full:** As with the staging threshold, check the size of the log stream's staging data set, as it might be too small. If you are unsure what it should be, size it similar to the CF structure to which the log stream is connected.

References for SMF log stream logging

- IBM Redbook, *Systems Programmer's Guide to: z/OS System Logger*
- *z/OS MVS Setting Up a Sysplex*
- *z/OS MVS System Management Facilities (SMF)*
- *z/OS MVS System Commands*
- *SMF Recording with MVS Logger*, by Riaz Ahmad and Jeff McDonough

Chapter 4. RRS archive logging enhancement

RRS makes use of a several system logger log streams. One of the optional log streams is the RRS archive log stream. RRS never uses the data that is stored in this log stream. It is there for the installation to use in case of a major outage. When the RRS archive log stream is enabled, RRS writes to it for each completed UR so there can be a considerable performance impact. This is why it is optional and not required.

Prior to z/OS V1R10, there was no easy way to turn archive logging on and off. You had to delete the log stream prior to starting RRS if you did not want to log. Starting with z/OS V1R10 you can use the SETRRS ARCHIVELOGGING system command to turn archive logging on and off when RRS is operational, without having to delete the log stream.

In our zPET environment, we always run with ARCHIVELOGGING turned on to exercise more code paths during our testing. We exploited this new functionality by dynamically stopping and restarting RRS archive logging using the new SETRRS commands. The results were as expected.

For a complete description of the SETRRS ARCHIVELOGGING command, see *z/OS MVS System Commands*.

Chapter 5. z/OS system logger administrative data utility enhancements

z/OS V1R10 introduces multiple enhancements to the z/OS system logger administrative data utility, IXCMIAPU. We exploited these enhancements and found them useful, so we are sharing them with you here:

- One of the shortcomings of the tool was that it would stop processing after running into an error in a single request. A new request verb, CONTINUE, has been added for TYPE LOGR requests to allow processing to continue past errors in subsequent requests. The placement of the CONTINUE request is significant. If errors occur in requests before the CONTINUE request is specified, the system logger performs syntax checking on the remaining requests, as usual, but does not execute them. If an error occurs after CONTINUE is specified, the system logger records the error and attempts to execute the remaining requests in the job step.
- DASD-only log streams are always duplexed to staging data sets. This is not optional, so the IXCMIAPU utility would fail when creating a DASD-only log stream with the duplexing options STG_DUPLEX, DUPLEXMODE, and LOGGERDUPLEX specified. Not being able to specify at least the default values, even though you cannot change them, was confusing. Starting in z/OS V1R10, the utility will not fail when these default values are specified. For instance:

```
STG_DUPLEX(YES)
DUPLEXMODE(UNCOND)
LOGGERDUPLEX(UNCOND)
```

Inappropriate duplexing configuration requests for DASD-only log streams are not allowed.

- The system logger now manages the STG_DUPLEX, DUPLEXMODE and LOGGERDUPLEX pending updates for CF type log streams so that the log stream will not have to be disconnected from each system around the sysplex in order for the updates to take effect; a CF user-managed structure rebuild will be enough.
- The IXCMIAPU utility, when run with the DETAIL parameter set to YES now displays the following additional information for each log stream that it lists:
 - The GMT date and time that each log stream was defined
 - Each data set in the LOG STREAM DATA SET INFO section now shows:
 - The youngest (highest) block ID for the data set
 - The highest relative byte address (RBA) for each offload data set, which indicates the total number of bytes written to the data set
 - The system name of the system that last modified the data set (which could be when the data set was newly allocated, closed, or marked for deletion). This is not the name of the system that last wrote to the data set.

The following example will give you a better idea of what these fields look like:

```
LOGSTREAM NAME(ATR.UTCPXJ8.ARCHIVE) STRUCTNAME(RRSLOG_ARCHIVE)
LS_DATACLAS(LS_MGMTCLAS() LS_STORCLAS()) HLQ(RRS) MODEL(NO) LS_SIZE(76800)
STG_MGMTCLAS() STG_STORCLAS() STG_DATACLAS() STG_SIZE(76800) LOWOFFLOAD(0)
HIGHOFFLOAD(80) STG_DUPLEX(YES) DUPLEXMODE(UNCOND) RMNAME() DESCRIPTION()
RETPD(5) AUTODELETE(YES) OFFLOADRECALL(NO) DASDONLY(NO) DIAG(NO)
LOGGERDUPLEX(UNCOND) EHLQ(NO_EHLQ) GROUP(PRODUCTION)
```


POSSIBLE ORPHANED LOG STREAM DATA SETS:

NUMBER OF POSSIBLE ORPHANED LOG STREAM DATA SETS: 0

LOGR Inventory Record Summary:

LOGR COUPLE DATA SET FORMAT LEVEL: HBB7705

/* Functional Items: */
/* SMDUPLEX(1) */

Type	Formatted	In-use
-----	-----	-----
LSR (Log Stream)	4,000	851
LSTRR (Structure)	80	80
DSEXTENT (Data Set Extent)	30	4

Chapter 6. New auxiliary and pageable storage shortage messages

Storage shortages are one of the most crucial and highly critical to resolve because real storage is a finite resource. An unresolved real storage shortage can result in a system outage. One of the biggest unresolved problems is in the handling of address spaces which allocate and fix too much storage or use too much auxiliary storage.

With z/OS V1R10, changes were made so that the system now monitors address spaces with the highest storage increases. This information is then used to build the list of address spaces that are the highest users of storage. These changes have led to the creation of additional messages and an enhanced ENF 55 signal to provide more information earlier about a storage shortage, such as auxiliary storage shortages and pageable storage shortages.

One example of the new storage handling is that during an auxiliary storage shortage, the system identifies a non-swappable, non-system address space with the largest slot increase and now marks it non-dispatchable, so that the address space cannot further increase the amount of slots. The system issues new message IRA210E to identify any address space that has been marked as non-dispatchable, as in the following examples:

```
IRA210E NETSA31 ASID 0028 SET NON DISPATCHABLE
IRA210E OMPROUTE ASID 0120 SET NON DISPATCHABLE
IRA210E GPMSERVE ASID 0116 SET NON DISPATCHABLE
```

The system then issues message IRA211I when the storage shortage is relieved and the address space is set back to being non-swappable, as in the following examples:

```
IRA211I NETSA31 ASID 0028 SET DISPATCHABLE
IRA211I GPMSERVE ASID 0116 SET DISPATCHABLE
IRA211I OMPROUTE ASID 0120 SET DISPATCHABLE
```

Changes to auxiliary storage management

When 50% of the auxiliary storage is allocated, the system issues message IRA205I and issues an ENF 55 signal with an additional qualifier that has been added to inform listening applications of this.

When 70% of the auxiliary storage is allocated, the system issues messages IRA200E and IRA206I with an ENF 55 signal which will contain a list of the top 20 address spaces contributing to the auxiliary storage shortage. If any of the address spaces with the highest increase are swappable, then the system will logically swap the address space, issue message IRA210E, and no longer dispatch the address space. For more information, see the STORAGENSWDWP keyword in *z/OS MVS Initialization and Tuning Reference*.

When 85% of auxiliary storage is allocated, the system issues message IRA210E. Then, if there are any swappable address spaces with a high amount of fixed storage, the system will logically swap the address space and no longer dispatch it. Refer again to the STORAGENSWDWP keyword in *z/OS MVS Initialization and Tuning Reference*.

Also at the 85% allocation mark, the system will now issue message IRA220I and WTOR IRA221D. By default, the system will issue IRA220I with the top five auxiliary storage users, as in the following example:

```
IRA220I CRITICAL AUXILIARY SHORTAGE 173
IRA220I ! ## ! USER      ! ASID ! PAGES      ! SLOTS      !
IRA220I +----+-----+-----+-----+-----+
IRA220I ! 01 ! DUMPSRV ! 0005S! 0000350342 ! 0000606258 !
IRA220I ! 02 ! WQWSRW1S ! 0128 ! 0000073178 ! 0000064510 !
IRA220I ! 03 ! CSQ1BRK ! 0272 ! 0000024828 ! 0000042030 !
IRA220I ! 04 ! CSQ1BRK ! 017B ! 0000023449 ! 0000042020 !
IRA220I ! 05 ! CSQ1BRK ! 0264 ! 0000024614 ! 0000041253 !
IRA221D REPLY M FOR MORE, E TO END, ## TO CANCEL A USER
```

For more information, see the STORAGEWTOR keyword in *z/OS MVS Initialization and Tuning Reference*.

Changes to pageable storage management

The system issues new message IRA405I when 50% of total real storage is fixed. You can control this value by using the IRA405I keyword in the IEAOPTxx parmlib member. The ENF 55 signal is also issued with a qualifier added to inform any listening applications about this. For more information, see the IRA405I keyword in *z/OS MVS Initialization and Tuning Reference*.

When 80% of real storage is fixed, the system issues messages IRA400E and IRA404I and issues an ENF 55 signal with a list of the top 20 contributors of shortage. The system will then logically swap any of the top 20 contributors, if they are swappable. If the address space is non-swappable, the system will then issue IRA410E and set the address space as non-dispatchable. For more information, see the STORAGENSWDP keyword in *z/OS MVS Initialization and Tuning Reference*.

When 90% of real storage is allocated, the system continues with tasks described at the 80% mark. If the pageable storage shortage lasts longer than 15 seconds, the system issues the new messages IRA420I and IRA421D, with the top five contenders listed by default. See message IRA220I in *z/OS MVS System Messages, Vol 9 (IGF-IWM)*. Also see the STORAGEWTOR keyword in the IEAOPTxx parmlib member in *z/OS MVS Initialization and Tuning Reference* for information about how to make message IRA420I display up to 20 address spaces at once.

Chapter 7. Migrating to a Server Time Protocol Coordinated Timing Network

This topic discusses our experiences with migrating to a Server Time Protocol (STP) Coordinated Timing Network (CTN) in the Poughkeepsie Development Lab. We begin with a brief overview of STP and related terminology, as well as a high-level overview of the timing topology in our zPET environment. We then discuss both the planning considerations and our actual migration steps to deploy STP in our data center.

We relied on the IBM Redbook, *Server Time Protocol Planning Guide*, SG24-7280, to provide the necessary information and technical details to help guide us through the migration process. The latest edition is available on the IBM Redbooks Web site at www.ibm.com/redbooks/.

Note that, while many of the steps we document might also apply to other data center migration efforts, the migration steps and the order of those steps as we present them are unique to our data center and, thus, you should not consider them to be universal.

Overview of STP

The Server Time Protocol (STP) feature is designed to provide the capability for multiple servers and coupling facilities (CFs) to maintain time synchronization with each other without requiring a Sysplex Timer[®] external time reference (ETR). The following servers and coupling facilities were able to support STP when it was first introduced:

- IBM System z9[®] Enterprise Class (z9[™] EC)
- IBM System z9 Business Class (z9 BC)
- IBM eServer[™] zSeries 990 (z990)
- IBM eServer zSeries 890 (z890)

The recently available IBM System z10 Enterprise Class (z10 EC) also supports STP.

Server Time Protocol is a server-wide facility that is implemented in the Licensed Internal Code (LIC) of z10 EC, z9 EC, z9 BC, z990, and z890 CPCs and presents a single view to Processor Resource/Systems Manager[™] (PR/SM[™]). STP uses a message-based protocol in which timekeeping information is passed over externally defined coupling link. STP supports the following coupling links:

- InterSystem Channel-3 (ISC-3) links configured in peer mode
- Integrated Cluster Bus-3 (ICB-3) links
- Integrated Cluster Bus-4 (ICB-4) links
- Parallel Sysplex over InfiniBand (PSIFB) coupling links

These can be the same links that are already being used in a Parallel Sysplex for CF message communication.

By using the same links to exchange timekeeping information and coupling facility messages in a Parallel Sysplex, STP can scale with distance. Servers exchanging messages over short distance links, such as ICB-3 and ICB-4, are designed to meet more stringent synchronization requirements than servers exchanging messages

over long distance links, such as ISC-3 (distances up to 100 kilometers), where the synchronization requirements are less stringent. This is an enhancement over the current Sysplex Timer implementation, which does not scale with distance.

STP supports the following activities:

- Allow clock synchronization for supported IBM System z servers and CFs without requiring a Sysplex Timer ETR.
- Support a multi-site timing network of up to 100 kilometers (62 miles) over fiber optic cabling, allowing a Parallel Sysplex to span these distances.
- Potentially reduce the cross-site connectivity required for a multi-site Parallel Sysplex.
- Coexist with an ETR network.
- Allow use of dial-out time services to set the time to an international time standard, such as Coordinated Universal Time (UTC), as well as adjust to the time standard on a periodic basis.
- Allow setting of local time parameters, such as time zone and Daylight Saving Time (DST).
- Allow automatic updates of Daylight Saving Time.

While STP does not require a Sysplex Timer, STP does support concurrently migrating from a timing network entirely synchronized to the IBM Sysplex Timer ETR (ETR network) to a timing network consisting of both Sysplex Timer ETRs and STP-enabled z9 and zSeries servers (mixed CTN), as well as migrating to a timing network consisting entirely of STP-enabled z9 and zSeries servers without any Sysplex Timer ETRs (STP-only CTN).

As shown in Figure 3 on page 43, our Parallel Sysplex is currently synchronized to the Sysplex Timer ETR. This is referred to as an ETR network. We discuss the planning and migration steps that we took to migrate from an ETR network to a mixed Coordinated Timing Network (CTN) then to an STP-only CTN. We also include explicit timing network configurations, migration scenarios, message captures, and panel captures as we moved our Parallel Sysplex environment from time synchronization using Sysplex Timer ETRs to using both Sysplex Timer ETRs and STP, then to using only STP and no Sysplex Timer ETRs.

STP terminology

Along with the new STP technology is new terminology. Within the scope of this information, the following terms and definitions apply:

ETR timing mode

A server is considered to be in *ETR timing mode* when the its time of day (TOD) clock has been initialized to and is being advanced by stepping signals received from a Sysplex Timer ETR. To be in ETR timing mode, the server must be part of an ETR network.

STP timing mode

A server is considered to be in *STP timing mode* when the its time of day (TOD) clock has been both initialized to Coordinated Server Time (CST) and is being advanced at the rate of the local hardware oscillator. In STP timing mode, the server's TOD clock is adjusted (steered) as needed in order to either maintain or attain time synchronization with the timing network's Coordinated Server Time. To be in STP timing mode, the server must be part of an STP network. Coordinated Server Time is defined below.

STP-capable server

An *STP-capable server* is any z10 EC, z9 EC, z9 BC, zSeries 990, or zSeries 890 server or CF that has all of the required STP LIC installed.

STP-enabled server

An *STP-enabled server* is an STP-capable server or CF that has the STP function enabled. Even after the LIC to support STP is installed on a server, the STP function cannot be used until it is enabled.

STP-configured server

An *STP-configured server* is a server that has been configured with a Coordinated Timing Network ID (CTN ID) so that it can participate in a Coordinated Timing Network (CTN). When the STP network ID portion of the CTN ID is not specified, the server is not configured to be part of a CTN and, therefore, is not an STP-configured server.

stratum

STP distributes time messages in layers, or *stratums*. The top layer, (stratum 1) distributes time messages to the layer immediately below it (stratum 2). Stratum 2, in turn, distributes time messages to stratum 3.

Coordinated Server Time (CST)

The *Coordinated Server Time* represents the time value to which all servers and coupling facilities in a Coordinated Timing Network (CTN) are synchronized.

Coordinated Timing Network (CTN)

A *Coordinated Timing Network* is a collection of servers that are all time synchronized to a common time value called Coordinated Server Time (CST). The servers that make up a CTN must all be configured with a common identifier, referred to as a Coordinated Timing Network ID (CTN ID). All servers in a CTN maintain an identical set of time-control parameters that are used to coordinate the time of day (TOD) clocks.

A CTN can be either of the following:

mixed CTN

A *mixed CTN* is a Coordinated Timing Network where the Sysplex Timer provides the timekeeping information to a heterogeneous mix of both Sysplex Timer synchronized servers and servers that are synchronized with Coordinated Server Time (CST).

STP-only CTN

An *STP-only CTN* is a timing network that does not require a Sysplex Timer ETR.

The following definitions are necessary to understand the roles that need to be assigned for certain servers in an STP-only CTN:

preferred time server

Using the STP panels provided at the HMC, a server must be assigned that has preference to be the stratum-1 server of an STP-only CTN. This is the *preferred time server*. This server should have connectivity to all servers that are destined to be the stratum 2 servers of an STP-only CTN. The connectivity can be either ISC-3 links in peer mode, ICB-3 links, or ICB-4 links.

backup time server

Optionally, it is highly recommended to also assign a *backup time server* whose role is to take over as the

stratum-1 server. The backup time server is a stratum-2 server that has connectivity to the preferred time server, as well as to all other stratum-2 servers that are connected to the preferred time server.

current time server

The *current time server* is the active stratum-1 server in an STP-only CTN. At the HMC, the current time server must be assigned to either the preferred or the backup time server. In most cases, the current time server is assigned to the preferred time server when the configuration is initialized. Subsequently, if there is a need to reassign the roles, the current time server can be concurrently assigned to the backup time server. This action may be part of a planned reconfiguration of the preferred time server.

arbiter

Optionally, at the HMC a server may be assigned to be the *arbiter* server. The arbiter server provides additional means for the backup time server to determine whether it should take over as the current time server in the event of unplanned exception conditions.

Coordinated Timing Network ID (CTN ID)

The *CTN ID* is an identifier that is used to indicate whether the server has been configured to be part of a Coordinated Timing Network (CTN) and, if so, it identifies the Coordinated Timing Network (CTN). The CTN ID is comprised of the following two fields:

1. One field that defines the STP network ID
2. One field that defines the ETR network ID

For more information about STP concepts and definitions, see the IBM Redbook, *Server Time Protocol Planning Guide*, SG24-7280.

STP planning considerations

Figure 3 on page 43 provides a before-and-after illustration of both the initial Sysplex Timer topology and the planned STP timing topology.

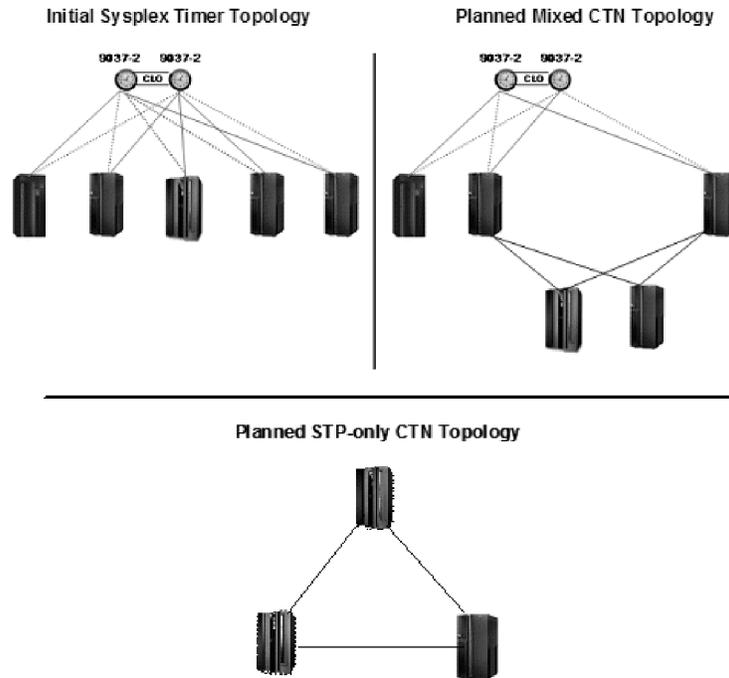


Figure 3. zPET initial Sysplex Timer topology, planned mixed CTN topology, and planned STP-only CTN topology

Server Time Protocol Planning Guide presents the detailed considerations and steps that are required to plan any STP migration. Here we highlight only the steps that we took that are unique to our environment.

Our servers and coupling facilities

At the time of our migration, the five existing servers in our data center fell into the following categories:

- **Non-STP-capable servers**

Our data center included one non-STP-capable server, a 2064-212 (z900) CPC, named FR24, which can coexist with STP-capable servers in a mixed CTN but cannot coexist with STP-capable servers in an STP-only CTN.

- **STP-capable servers**

Our data center included the following STP-capable servers:

- Two 2096-S07 (z9 BC) CPCs, named K25 and K28
- One 2094-S38 (z9 EC) CPC, named T75
- One 2084-327 (z990) CPC, named G74

Considerations for migrating from a mixed CTN to an STP-only CTN

All servers in an STP-only CTN must be STP-capable. Therefore, before configuring an STP-only CTN, all non-STP-capable servers must be removed from the Parallel Sysplex configuration. We could not migrate to an STP-only CTN configuration until we removed the z900 server from our sysplex. Later, when the IBM System z10 EC server was introduced, we were able to replace our non-STP-capable z900 server with a z10 EC (2097-E56) server, named H91, thereby allowing us to configure an STP-only CTN.

Recovery considerations

Our Sysplex Timer topology is such that all of the servers in our data center maintain fully redundant Sysplex Timer connectivity. We wanted to ensure that each migration step had to maintain at least this same level of resiliency for time synchronization. We recognized that K25 did not have peer link connectivity to every other STP-capable server in the data center and, therefore, would not have fully redundant timing connectivity in certain CTN configurations. To eliminate this vulnerability, we configured redundant STP timing-only links between K25 and G74.

Summary planning matrix

Table 3 provides an overview of the planning steps that we took to deploy STP in our environment, based on the information in *Server Time Protocol Planning Guide*.

Table 3. Planning steps for deploying the Server Time Protocol in our data center

Migration step	Migration action	Description	Notes and comments applicable to our data center
1	Recognize the hardware platforms and their respective supported timing modes	ETR only, mixed CTN, and STP-only CTN	Two 2096-S07 (z9 BC) One 2094-S38 (z9 EC) One 2084-327 (z990)
		ETR only and mixed CTN coexistence	zSeries 800 and zSeries 900 One 2064-212 (z900) with an ICF zone. MTOF is satisfied since this server is already connected to both Sysplex Timer ETRs.
2	Message Time Order Facility (MTOF)	MTOF is an STP pre-requisite.	All servers satisfy the MTOF requirement. (The 2064-212 is connected to the ETRs.)
3	Upgrade the HMC to V2.9.1.	Requires HMC application V2.9.1	Needed to upgrade the HMC
4	Install EC levels and MCLs.	z9, z890, and z990 servers must be made STP-capable by concurrently installing STP Licensed Internal Code (LIC)	The three z9 servers will need the latest level of Driver 63J.
			The one z990 will need the latest level of Driver 55K. This step makes each server STP-capable but not STP-enabled. See step 6 for STP Enablement.
5	Install Sysplex Timer LIC.	The Sysplex Timer ETRs require a LIC upgrade.	This is a non-disruptive concurrent apply only if the ETRs are in a Sysplex Timer Expanded Availability (EA) configuration.
6	Enable the STP facility by installing Feature Code (FC) 1021.	The STP facility must be <i>enabled</i> on each STP-capable server	Must be applied to each of the servers that were made STP-capable in step 4.
			Concurrently install on one server then verify via step 7. Then repeat steps 6-7 for each remaining STP-capable server.

Table 3. Planning steps for deploying the Server Time Protocol in our data center (continued)

Migration step	Migration action	Description	Notes and comments applicable to our data center
7	Verify STP facility enablement.	Verification step	<p>From the upgraded HMC, select the Sysplex Timer task for the server where the STP facility was enabled in step 6.</p> <p>Look for new panels. See Figure 5 on page 48 for an example.</p>
8	Verify z/OS supported levels and latest service.	The STP feature is supported on z/OS V1R7 and higher.	All z/OS images were at z/OS V1R8 with the latest maintenance applied weekly.
9	Install STP timing-only link support.	Install the necessary IOCP, HCD, and HCM maintenance for STP timing-only link support.	<p>This support is needed in order to achieve stated migration objective.</p> <p>Schedule this maintenance installation to coincide with the weekly service window.</p>
10	Install z/OS STP enablement SPE.	The z/OS STP enablement support was delivered as a ++APAR at the time of this writing.	<p>The z/OS V1R8 version of the enablement APAR is the only one that is needed.</p> <p>Schedule this maintenance installation to coincide with the weekly service window. Once STP is generally available, the STP enablement APAR will be available as part of the required STP software maintenance in step 8.</p>
11	Install z/OS coexistence support.	z/OS toleration support is required for z/OS V1R4, V1R5, and V1R6 systems if they are in a Parallel Sysplex and are running on servers that are in a mixed CTN.	Not applicable as all z/OS images were at z/OS V1R8.
12	Update SYS1.PARMLIB(CLOCKxx).	Optional	Leave existing CLOCKxx member as is and allowed all the new parameters to use their default values.
13	IPL z/OS.	An IPL is required after the z/OS STP enablement APAR is installed.	<ol style="list-style-type: none"> Schedule this step to coincide with the weekly service window IPLs. Ensure that 100 percent of the STP support was already installed on all of the hardware to avoid an additional IPL.
14	Define STP timing-only links.	IODF definition step	This step can be performed anytime prior to actually transitioning G74 to a stratum-2 server.
15	Install peer link fiber between the two servers for STP timing-only link usage.	Physical ISC peer link fiber installation	<p>This step can be performed anytime prior to performing the IODF activation in step 16.</p> <p>Run two peer links, one ICB and one ISC-3, between K25 and G74.</p>

Table 3. Planning steps for deploying the Server Time Protocol in our data center (continued)

Migration step	Migration action	Description	Notes and comments applicable to our data center
16	Activate the IODF.	Required for timing-only links defined in step 14	Need to activate the IODF on the two servers where the two STP timing-only links were defined.
17	Configure STP timing-only links on the servers at both ends.	Verification step	Verify that the timing-only links come online. <ul style="list-style-type: none"> • Minimum: Physical configure of PCHID • Optional: Logical configure of z/OS CHPID

STP migration experiences

This topic documents our migration experiences. We begin by briefly describing our initial data center topology and then proceed through the STP migration.

Our initial Sysplex Timer (ETR-only) topology

Our existing data center topology consisted of five IBM mainframe servers, all of which were connected to a pair of Sysplex Timer ETRs, as illustrated in Figure 4 on page 47.

The two ETRs were inter-connected via Control Link Oscillator (CLO) connections in support of the recommended Sysplex Timer Expanded Availability (EA) configuration. This EA configuration provides ETR network recovery in the event of a link failure, an ETR failure, or a power outage since both ETRs are simultaneously transmitting the same time-synchronized data to all of the attached servers.

zPET Sysplex Timer Topology

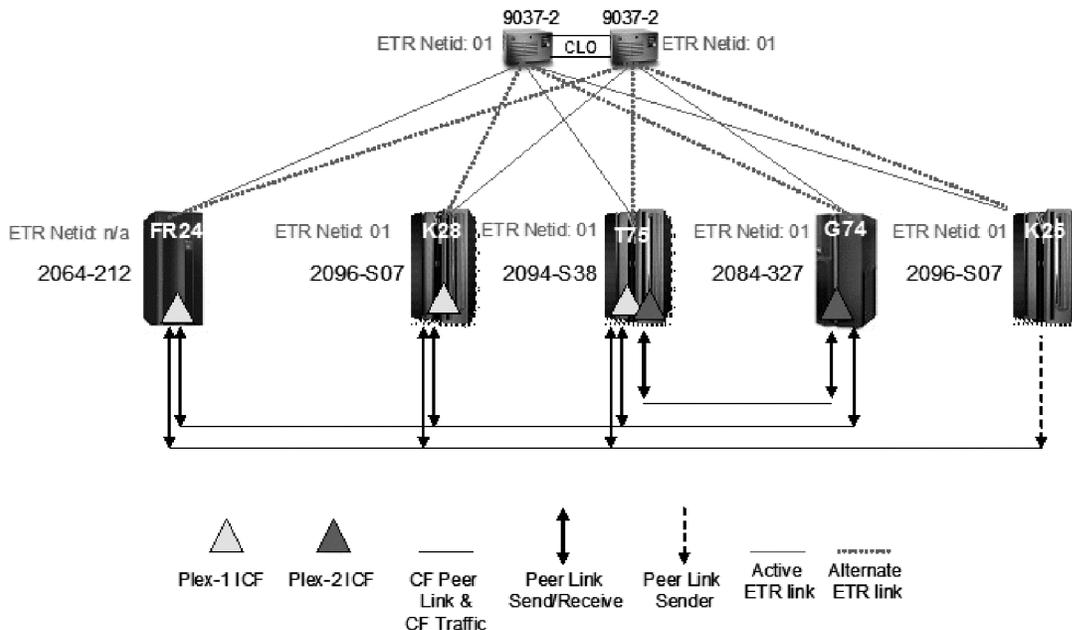


Figure 4. zPET Sysplex Timer topology

The System (Sysplex) Time task on the HMC or SE is used to access the various facilities for display and management of the CTN. There are now six tabs that you can display on the System (Sysplex) Time panel, as shown in Figure 5 on page 48. Note that, for a mixed CTN, time management tasks, such as time zone and leap second offsets, are still executed from the Sysplex Timer Console application. In an STP-only CTN, all time management tasks are executed from the System (Sysplex) Time task.

All six tabs are displayed only if the server has at least one ETR card installed and the STP feature is installed. The following list summarizes the conditions under which each of the tabs will be visible:

- The **ETR Configuration** and **ETR Status** tabs are only shown if the server has ETR cards installed.
- If at least one ETR card is installed but the STP facility is not enabled, the only tabs that will be available are the **ETR Configuration** and **ETR Status** tabs.
- The **Timing Network**, **Network Configuration**, **STP Configuration**, and **STP Status** panels are only present if the server is STP-enabled.

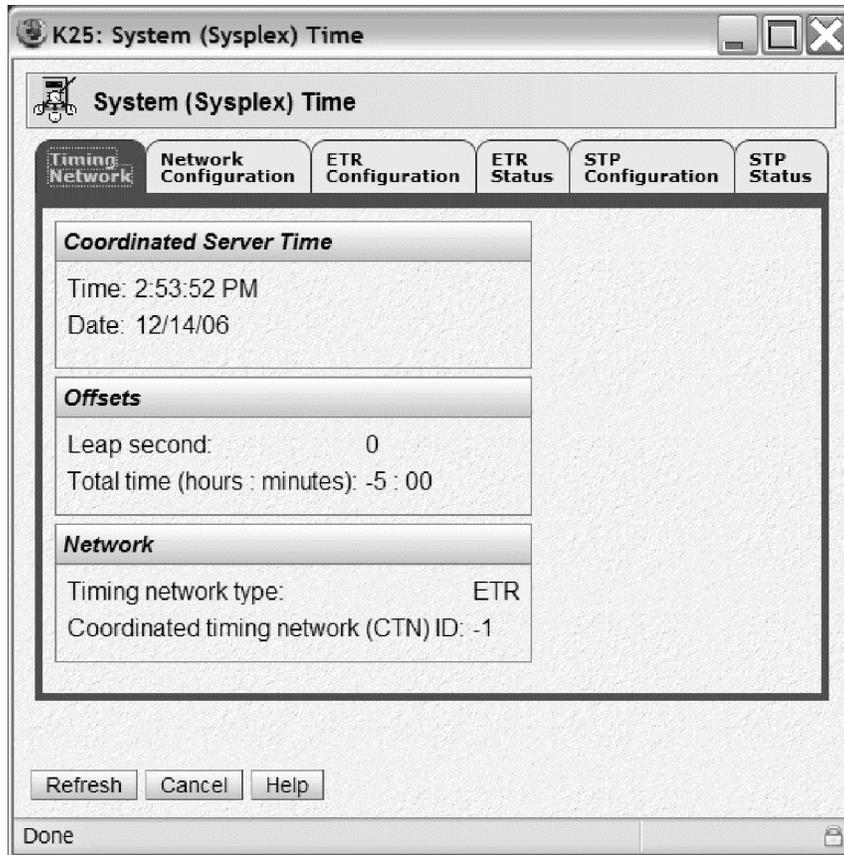


Figure 5. System (Sysplex) Time panels, viewed from the Support Element (SE)

Issuing the DISPLAY ETR command on all of the LPARs in the Parallel Sysplex that are using this timing network confirms that all of the servers are in ETR timing mode—that is, their TOD clocks are being advanced by stepping signals received from a Sysplex Timer ETR.

```
IEA282I 06.26.55 TIMING STATUS
SYNCHRONIZATION MODE = ETR
CPC PORT 0 <== ACTIVE      CPC PORT 1
OPERATIONAL                 OPERATIONAL
ENABLED                     ENABLED
ETR NET ID=01              ETR NET ID=01
ETR PORT=04                ETR PORT=04
ETR ID=00                  ETR ID=01
```

In addition to the DISPLAY ETR command, routing the DISPLAY XCF,SYSPLEX,ALL command to any z/OS image in the sysplex also reinforces this topology, as shown in the following sample output:

```
IXC335I 09.27.53 DISPLAY XCF
SYSTEM  TYPE SERIAL LPAR STATUS TIME          SYSTEM STATUS
JC0     2084 B52A   0C 09/28/2006 09:27:51 ACTIVE          TM=ETR
JB0     2084 B52A   01 09/28/2006 09:27:51 ACTIVE          TM=ETR
TPN     2064 1526   09 09/28/2006 09:27:50 ACTIVE          TM=ETR
Z0      2064 1526   01 09/28/2006 09:27:49 ACTIVE          TM=ETR
J80     2094 299E   07 09/28/2006 09:27:53 ACTIVE          TM=ETR
JF0     2094 299E   06 09/28/2006 09:27:50 ACTIVE          TM=ETR
JA0     2084 B52A   2A 09/28/2006 09:27:52 ACTIVE          TM=ETR
J90     2064 1526   05 09/28/2006 09:27:50 ACTIVE          TM=ETR
JH0     2096 FE2D   01 09/28/2006 09:27:49 ACTIVE          TM=ETR
JE0     2084 B52A   22 09/28/2006 09:27:50 ACTIVE          TM=ETR
```

Note that, starting with z/OS V1R7, the SYSTEM STATUS field includes a timing mode (TM) portion that indicates whether an LPAR resides on a server that is in ETR timing mode or in STP timing mode. For definitions of the ETR and STP timing modes, see “STP terminology” on page 40.

Adding STP timing-only links

As mentioned in “Recovery considerations” on page 44, we needed to define STP timing-only links between K25 and G74 in order to maintain fully redundant timing synchronization between all servers in our data center.

STP timing-only links are coupling links that allow two servers to be synchronized using STP messages when a CF does not exist at either end of the coupling link. Both HCD and HCM have been enhanced to allow you to define timing-only links with the new STP control unit.

We used HCD to define one ISC-3 timing-only link (CHPID type CFP) and HCM to define one ICB-3 timing-only link (CHPID type CBP). In our environment, the ISC-3 link that we planned to use connected CHPID K25.0.16 (ISC-3 PCHID K25.191) to G74.0.96 (ISC-3 PCHID G74.100). The ICB-3 link that we planned to use connected CHPID K25.0.02 (ICB-3 PCHID K25.2A0) to G74.0.97 (ICB-3 PCHID G74.680).

The following HCD dialog lists the coupling peer links (not all CHPIDs shown) as they were defined in our IODF before defining the timing-only links:

```

CF Channel Path Connectivity List

Select one or more channel paths, then press Enter.

Source processor ID . . . . . : K25
Source channel subsystem ID . . : 0
Source partition name . . . . . : *

-----Source-----      -----Destination-----      -CU-
/ CHPID  Type Mode Occ   Proc.CSSID      CHPID  Type Mode Type
- 00     CBP  SHR  N     T75.0          60     CBP  SPAN CFP
- 02     CBP  SHR  N
- 14     CFP  SHR  N     K28.0          A9     CFP  SHR  CFP
- 16     CFP  SHR  N
- 18     CFP  SHR  N     T75.0          71     CFP  SPAN CFP

```

You can see that there are no coupling peer links defined between K25 and G74. However, we have already defined the CHPIDs for each end of the timing links.

The Connect to CF Channel Path dialog in HCD is used to define the timing-only link, as follows:

```

Connect to CF Channel Path

Specify the following values.

Source processor ID . . . . . : K25
Source channel subsystem ID . . : 0
Source channel path ID . . . . . : 16
Source channel path type . . . . : CFP

Destination processor ID . . . . . G74      +
Destination channel subsystem ID . . 0      +
Destination channel path ID . . . . . 96    +

Timing-only link . . . . . YES

```

Notice the new **Timing-only link** parameter, which we set to YES in this case. It is important to note that if a CF image is in the access list of the CHPID on either end of this intended link, HCD will reject the creation of the timing-only link. In fact, if any coupling peer links are already defined between the two servers, HCD will also reject the creation of the timing-only link.

Pressing Enter displays the Add CF Control Unit and Devices dialog, as follows:

```

Add CF Control Unit and Devices

Confirm or revise the CF control unit number and device numbers
for the CF control unit and devices to be defined.

Processor ID . . . . . : K25
Channel subsystem ID . . . : 0
Channel path ID . . . . . : 16          Operation mode . . : SHR
Channel path type . . . . . : CFP

Control unit number . . . . FFF1 +

Device number . . . . . : _____
Number of devices . . . . . : 0
  
```

Notice that the control unit number is generated by HCD so we can accept it as it is. More importantly, notice that the number of devices generated is 0. This means that an STP timing-only control unit has no devices associated with it. This is a key difference between a coupling peer link and a timing-only link. Since there are no devices defined for timing-only links, z/OS cannot use it to send coupling messages. However, the STP facility can use either type of link to send STP messages.

After confirming this dialog as well as the Add CF Control Unit and Devices dialog for the G74 side of this peer link definition, the CF Channel Path Connectivity List dialog appears, as follows:

```

                CF Channel Path Connectivity List

Select one or more channel paths, then press Enter.

Source processor ID . . . . . : K25
Source channel subsystem ID . . : 0
Source partition name . . . . . : *

-----Source-----      -----Destination-----      -CU-
/ CHPID  Type Mode Occ   Proc.CSSID      CHPID  Type  Mode Type
- 00     CBP  SHR  N     T75.0          60    CBP   SPAN CFP
- 02     CBP  SHR  N
- 14     CFP  SHR  N     K28.0          A9    CFP   SHR  CFP
- 16     CFP  SHR  N     G74.0          96    CFP   SPAN STP
- 18     CFP  SHR  N     T75.0          71    CFP   SPAN CFP
  
```

The new timing-only link is distinguished by the CU Type of STP.

The HCM dialogs are similar, providing a new check box to specify a peer link as an STP timing-only link, as shown in Figure 6 on page 51.

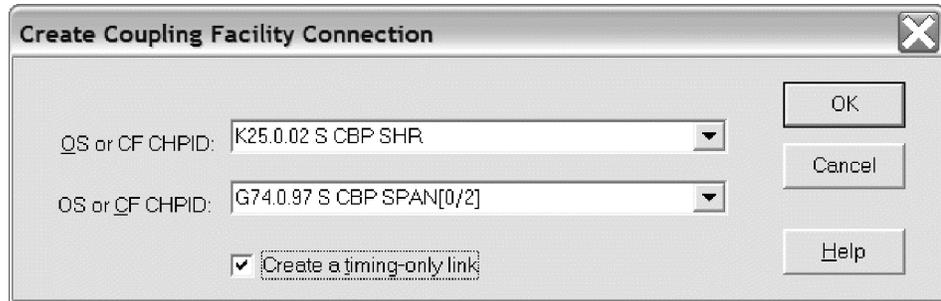


Figure 6. HCM Create Coupling Facility Link Connection dialog for defining a STP timing-only link

After the physical cables were connected between the two servers, we did a dynamic IODF ACTIVATE across all of the images on both of the servers. We were then able to configure the CHPIDs online to the z/OS images on both K25 and G74. To verify the link status, we issued a DISPLAY M=CHP(xx) command for each timing-only CHPID. The following is a sample of the command response from one of those images:

```
RO JH0,D M=CHP(16)
IEE174I 14.00.21 DISPLAY M
CHPID 16: TYPE=22, DESC=COUPLING FACILITY PEER, ONLINE
```

Issuing this same command for a CHPID used for coupling facility traffic results in the following response, which includes information related to CF connectivity and devices:

```
RO JB0,D M=CHP(03)
IEE174I 13.54.04 DISPLAY M
CHPID 03: TYPE=22, DESC=COUPLING FACILITY PEER, ONLINE
COUPLING FACILITY 002064.IBM.02.000000051526
                    PARTITION: 04 CPCID: 00
                    CONTROL UNIT ID: FFEA

SENDER PATH      PHYSICAL      LOGICAL      CHANNEL TYPE
03 / 0519        ONLINE        ONLINE        CFP

COUPLING FACILITY SUBCHANNEL STATUS
TOTAL: 56 IN USE: 2 NOT USING: 54 NOT USABLE: 0
DEVICESUBCHANNEL STATUS
FEBB3280 OPERATIONAL
FEBC3281 OPERATIONAL
```

⋮

Figure 7 on page 52 shows our existing Sysplex Timer topology after being updated with the STP timing-only links.

zPET Sysplex Timer Topology with STP Timing only Links

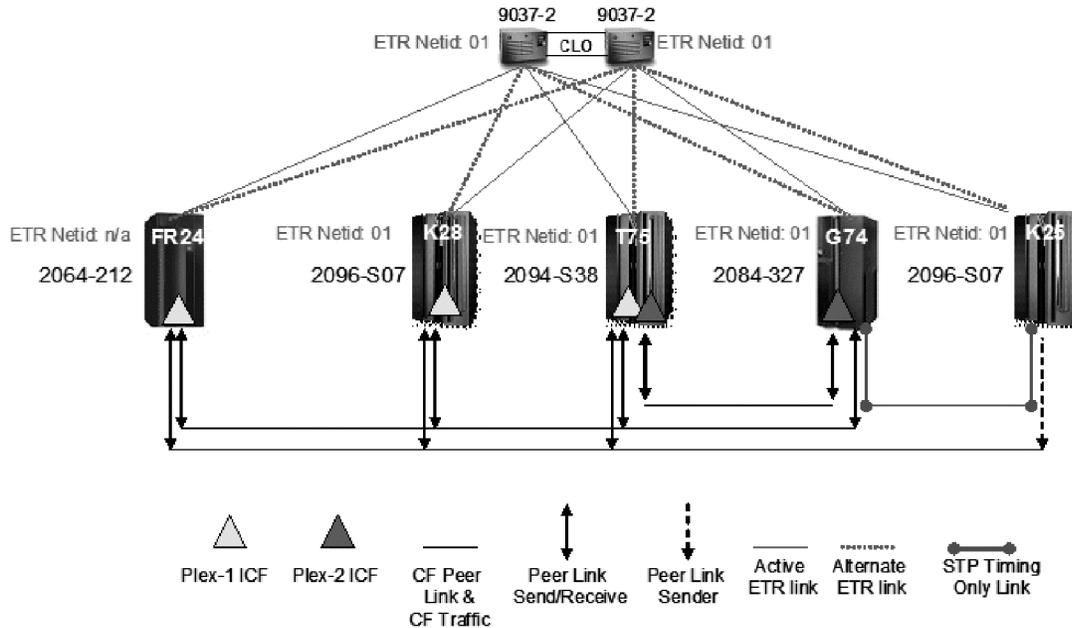


Figure 7. zPET Sysplex Timer topology with STP timing-only links

CTN ID configuration and verification

Our next step involved configuring a matching CTN ID on each STP-enabled server in our data center. The CTN ID is comprised of two fields in the form of *STP ID - ETR network ID*.

The STP Configuration tab of the System (Sysplex) Time task is used to configure the CTN ID on each STP-enabled server. Figure 8 on page 53 shows how the initial CTN ID contained a blank STP ID field, while the ETR network ID field had already been filled in.

We entered an STP ID of PETCTN on this panel on each STP-enabled server in our data center to initialize each STP facility. Note that the ETR network ID portion of the CTN ID is actually 01 (zero-one); the leading zero is not displayed.

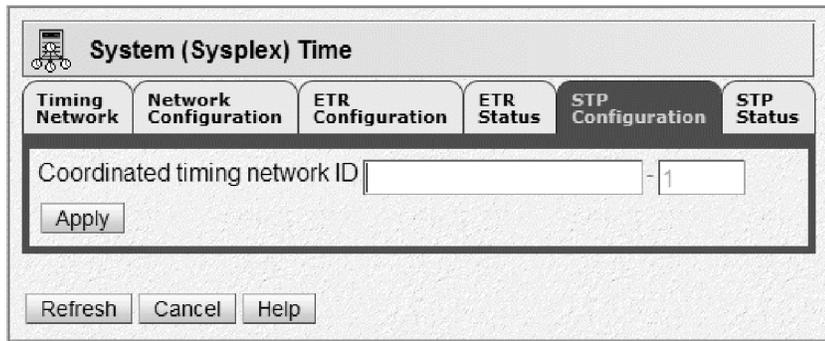


Figure 8. System (Sysplex) Time: STP Configuration panel

Once we entered the STP ID field, we clicked **Apply** to configure a CTN ID of PETCTN-01 on the server. Figure 9, Figure 10, and Figure 11 on page 54 show the sequence of STP configuration panels involved with configuring the CTN ID on one STP-enabled server.

Figure 9 shows where the STP ID portion of the CTN ID was entered.

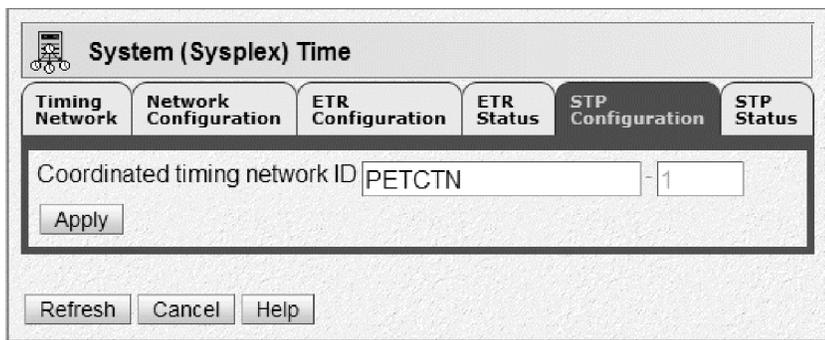


Figure 9. STP Configuration panel with STP ID value entered

Figure 10 shows the confirmation panel that appears when a CTN ID change is detected.

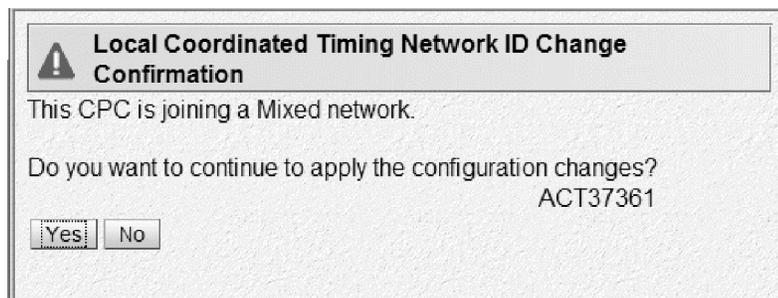


Figure 10. STP configuration confirmation panel

Figure 11 on page 54 shows the panel that acknowledges that the CTN ID was successfully changed.

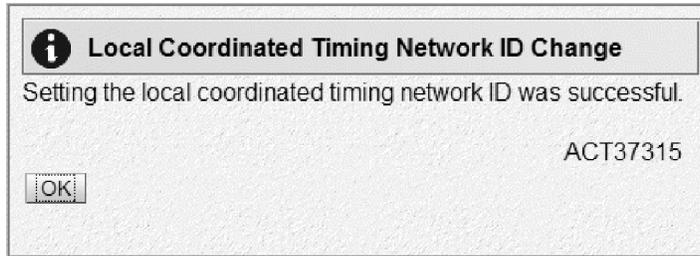


Figure 11. STP CTN ID change completion panel

After a CTN ID has been configured on an STP-enabled server, each z/OS image running on that server will post an unsolicited message indicating that it has dynamically and non-disruptively detected that the Coordinated Timing Network ID has been changed, provided that the STPMODE parameter was either explicitly set to Y or allowed to default to Y in the CLOCKxx member that was used during IPL.

Example: The following message indicates that z/OS system JH0 has detected the CTN ID change. Notice how the CTN ID is the concatenation of the PETCTN STP ID and the 01 ETR network ID, resulting in a CTN ID of PETCTN-01.

```
IXC438I COORDINATED TIMING INFORMATION HAS BEEN UPDATED
        FOR SYSTEM: JH0
        PREVIOUS ETR NETID: 01
        CURRENT CTN ID:  PETCTN -01
```

It is important to note that this server is still synchronized to the Sysplex Timer ETR and is still considered to be in ETR timing mode. Issuing the z/OS DISPLAY ETR command from a z/OS image on this STP-configured server shows the following:

```
IEA282I 16.01.22 TIMING STATUS
SYNCHRONIZATION MODE = ETR
CPC PORT 0          ACTIVE ==>  CPC PORT 1
OPERATIONAL          OPERATIONAL
ENABLED              ENABLED
ETR NET ID=01       ETR NET ID=01
ETR PORT=02        ETR PORT=12
ETR ID=00           ETR ID=01
THIS SERVER IS PART OF TIMING NETWORK PETCTN -01
```

Here we see that the D ETR command returns additional information showing the CTN ID of the mixed CTN in the last line of the display. The SYNCHRONIZATION MODE = ETR means that the server is still connected to the Sysplex Timer ETRs and, therefore, remains in ETR timing mode.

Figure 12 on page 55 shows STP Status tab of the System (Sysplex) Time task, which displays the timing configuration from the server's perspective.

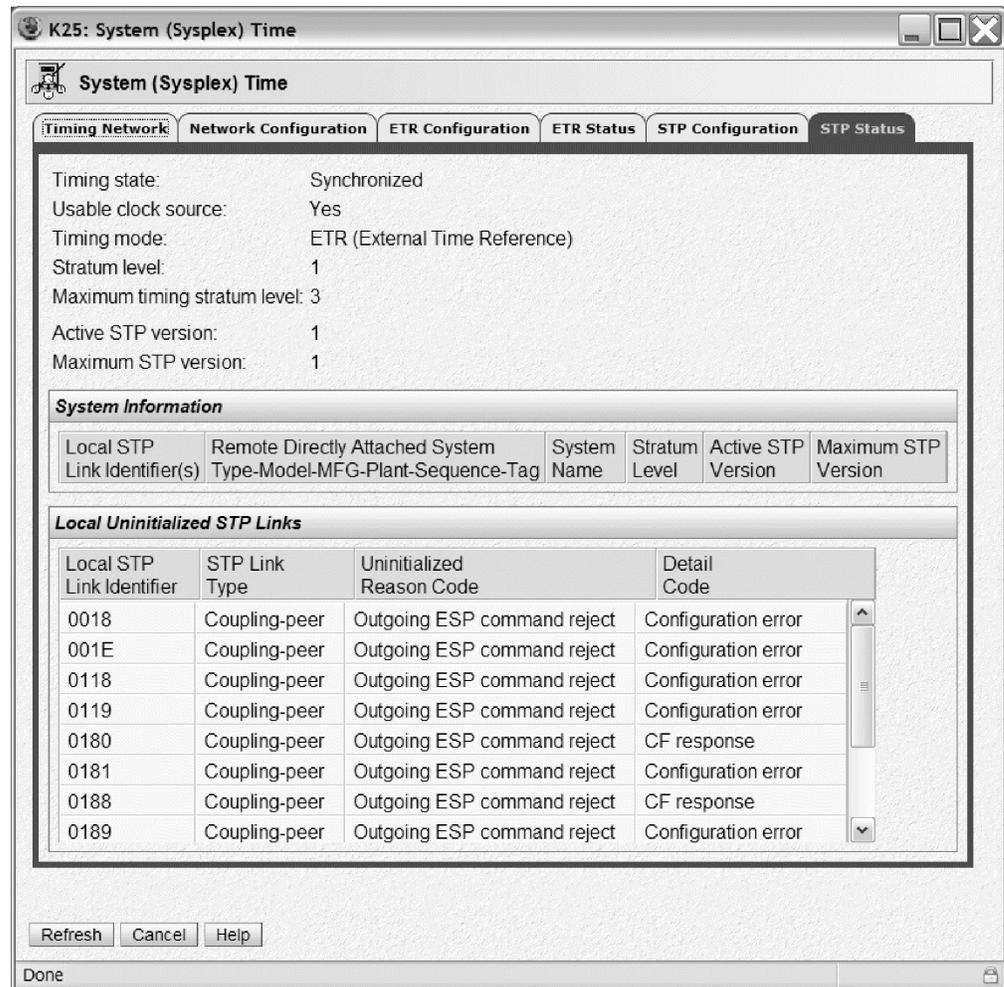


Figure 12. System (Sysplex) Time: STP Status panel, viewed from the SE on K25

You can see that, even with STP configured on this server, this server is still synchronized to an ETR, as reflected by the **Timing State** and **Timing Mode** fields.

Figure 12 also demonstrates that none of the other servers connected to this one have been configured with a matching CTN ID as there are no directly attached systems listed in the **System Information** section of the panel.

Once a matching CTN ID had been configured on each of the STP-enabled servers in our data center, we used the STP Status panel to verify that the STP facility was exchanging STP timing signals over the peer links connecting the STP-configured servers.

Next, we configured a second server, K28, with the same CTN ID (PETCTN-01) and verified the configuration using the System (Sysplex) Time STP Status panel on K28. Figure 13 on page 56 contains the results of this configuration action.

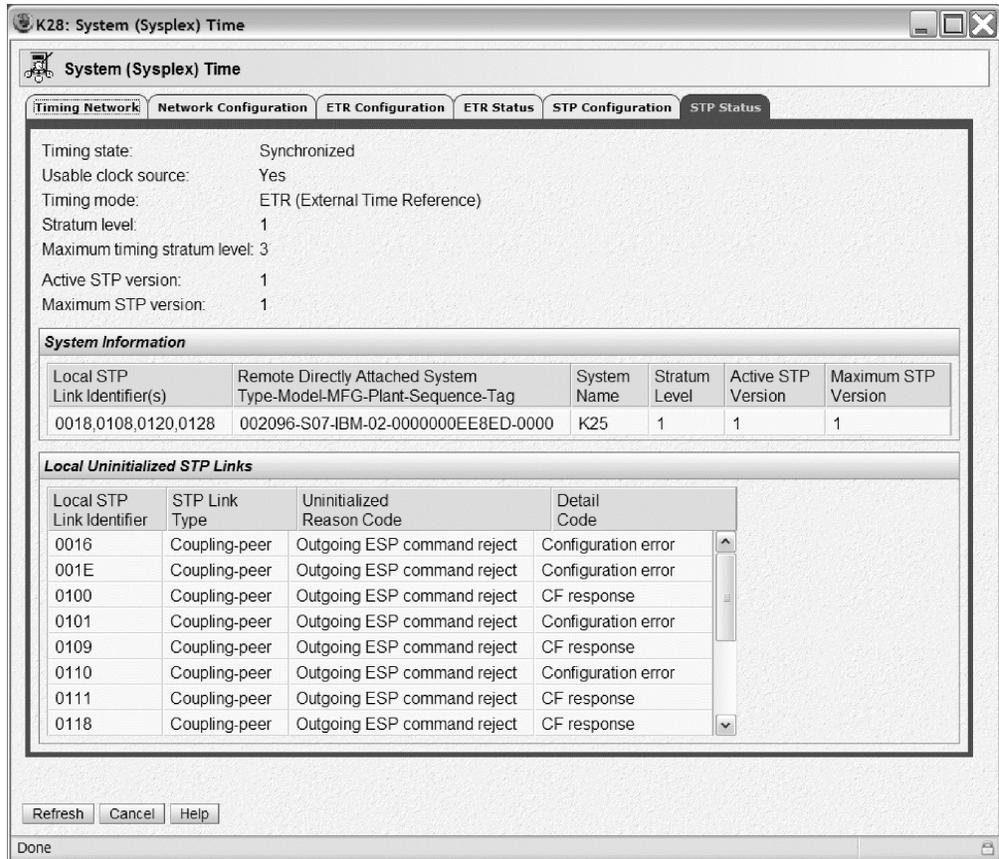


Figure 13. System (Sysplex) Time: STP Status panel on K28, after configuring the CTN ID on K28

Immediately upon configuring the STP ID on K28, an entry for K25 appears in the **System Information** section of the STP Status panel.

- The value K25 in the **System Name** field indicates that a matching CTN ID has been detected between this server (K28) and K25.
- The **Local STP Link Identifiers** field lists all of the peer links where matching CTN IDs have been exchanged between these two servers. Specifically, K28 is using PCHIDs 0018, 0108, 0120 and 0128 to send and receive STP signals to and from K25.

Also note that, because both servers are still synchronized to the ETRs, they are each at the stratum 1 level in the timing hierarchy, as indicated by the **Stratum Level** fields.

Figure 14 on page 57 provides another verification view, this time from the K25 side of the configuration. The **System Information** section of the STP Status tab shows that the four peer links being used to exchange STP signals to K28 are PCHIDs 001E, 0118, 0181, and 0190.

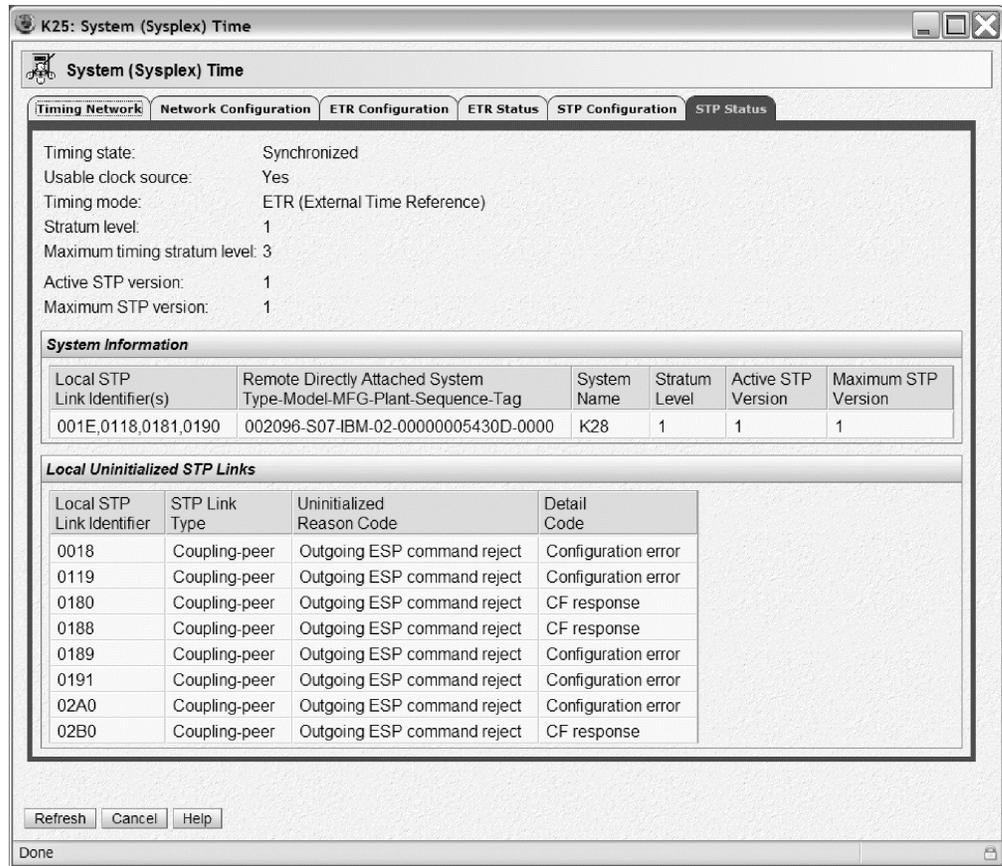


Figure 14. System (Sysplex) Time: STP Status panel on K25

K28 was a stand alone coupling facility in that it was activated as a single LPAR configured with Internal Coupling Facility (ICF) processors. Therefore, when a matching CTN ID was configured on K28, we do not see any indication of a CTN ID change from the CF operating system, as we experienced on a server with active z/OS images.

Therefore, from a z/OS perspective, an STP configuration verification for this ICF was accomplished by issuing the z/OS DISPLAY CF command from any z/OS image residing on K25, then comparing the response received with the list of STP links listed on K25's System (Sysplex) Time STP Status panel for K28 in the System Information section.

Example: Comparing the following response from the DISPLAY CF command on z/OS image JH0 with the information shown in Figure 14 verifies that the STP facility has initialized the same peer links that the JH0 image is using for connectivity to the CF1 coupling facility:

```
RO JH0,D CF,CFNM=CF1

IXL150I 11.16.50 DISPLAY CF
COUPLING FACILITY 002096.IBM.02.00000005430D
PARTITION: 01 CPCID: 00
CONTROL UNIT ID: FFFC

NAMED CF1

CF REQUEST TIME ORDERING: REQUIRED AND ENABLED

SENDER PATH          PHYSICAL          LOGICAL          CHANNEL TYPE
```

```

01 / 001E      ONLINE      ONLINE      CBP
13 / 0190      ONLINE      ONLINE      CFP
14 / 0118      ONLINE      ONLINE      CFP
15 / 0181      ONLINE      ONLINE      CFP

```

At this point in our STP migration effort, two of the four STP-enabled servers have now been made STP-configured. We then configured matching CTN IDs on the remaining two STP-enabled servers and verified that the four STP-configured servers were correctly exchanging STP signals over all of the expected peer links.

Figure 15 through Figure 18 on page 61 show the STP Status panels for K25, K28, G74, and T75, respectively, demonstrating that all four of the STP-enabled servers were properly configured and communicating with each other.

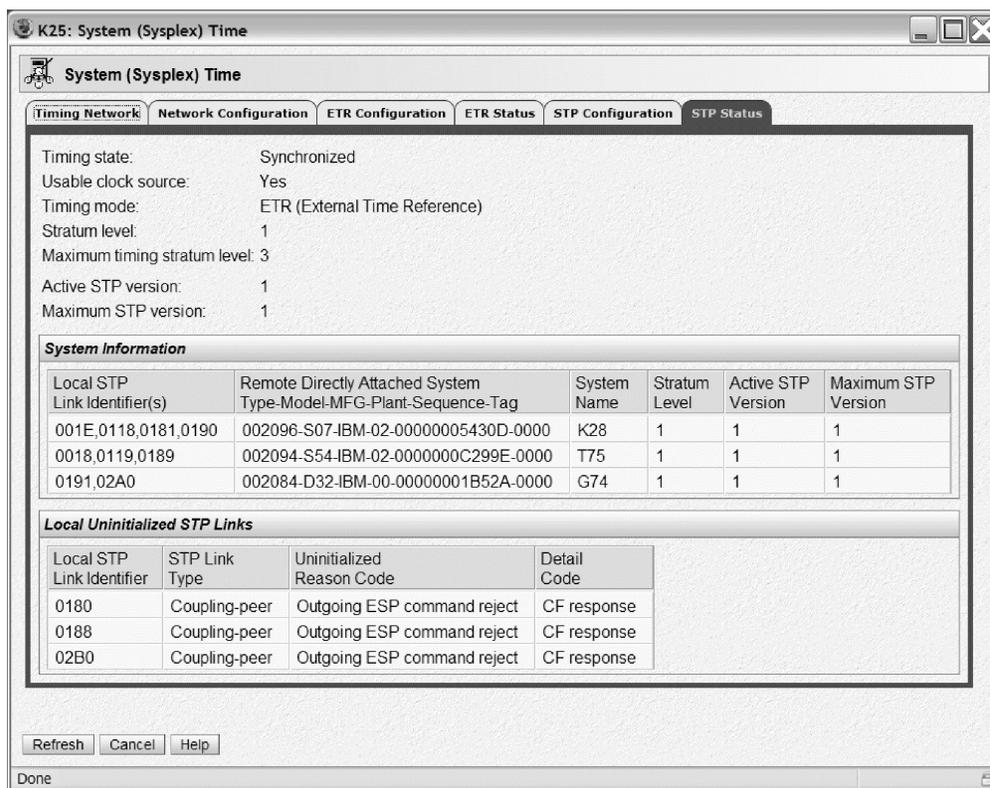


Figure 15. System (Sysplex) Time: STP Status panel on K25, showing connectivity to the other three servers

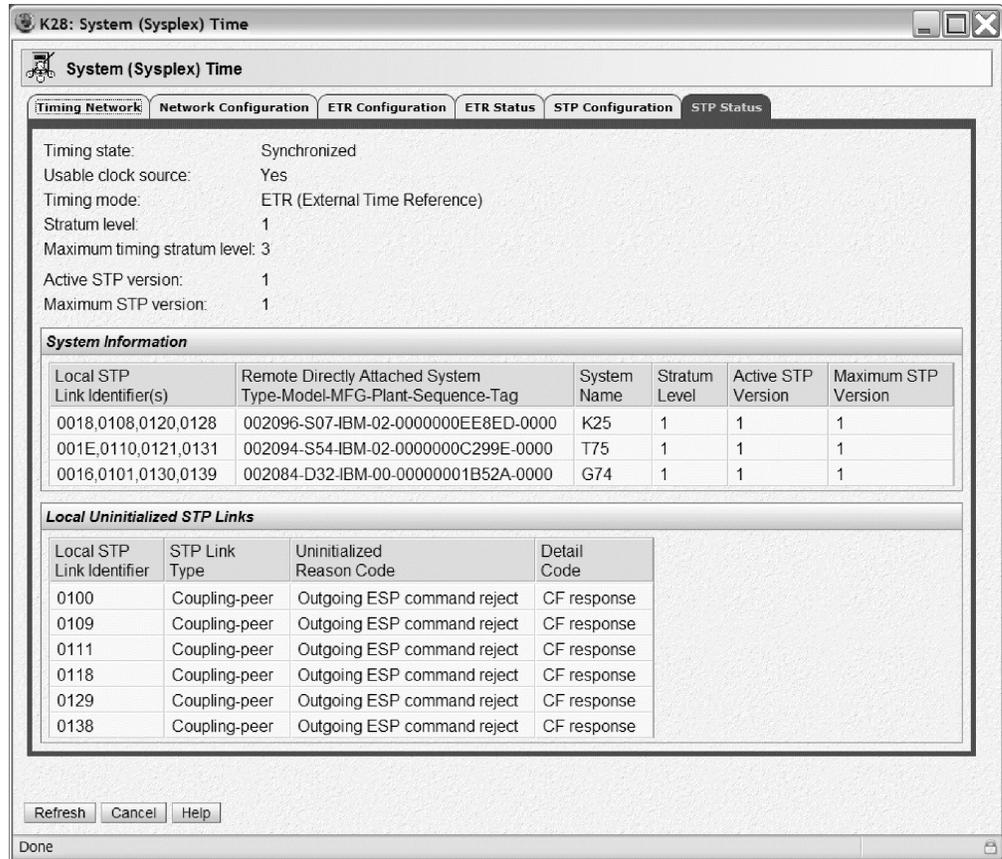


Figure 16. System (Sysplex) Time: STP Status panel on K28, showing connectivity to the other three servers

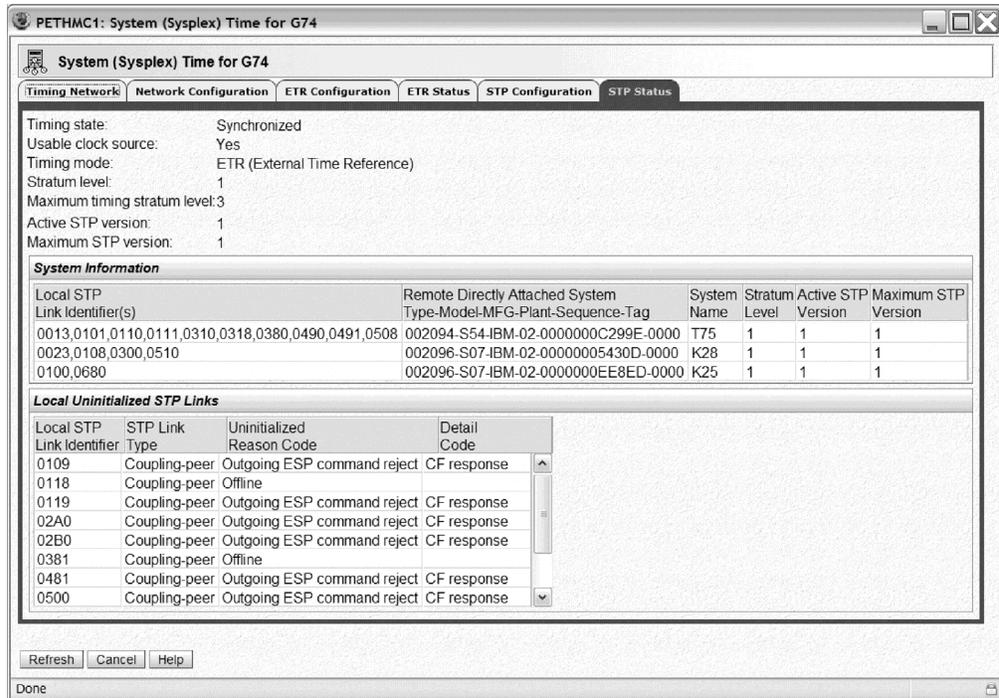


Figure 17. System (Sysplex) Time: STP Status panel on G74, showing connectivity to the other three servers

Notice that some of the coupling peer links in the Local Uninitialized STP Links section of this status panel reflect a value of Outgoing ESP command reject in the **Uninitialized Reason Code** field. This is expected for all peer links connected between our STP-configured servers and the CF partition on our non-STP-capable z900 server (2064-212). Issuing the z/OS DISPLAY CF command from a z/OS LPAR on G74 confirms this, as shown in the following example command response:

```
D CF,CFNM=CF3

IXL150I 14.00.38 DISPLAY CF
COUPLING FACILITY 002064.IBM.02.000000051526
                                PARTITION: 04 CPCID: 00
                                CONTROL UNIT ID: FFEA

NAMED CF3

CF REQUEST TIME ORDERING: REQUIRED AND ENABLED

SENDER PATH  PHYSICAL      LOGICAL      CHANNEL TYPE
02 / 0109    ONLINE          ONLINE      CFP
03 / 0519    ONLINE          ONLINE      CFP
07 / 0500    ONLINE          ONLINE      CFP
0A / 0119    ONLINE          ONLINE      CFP
1A / 0481    ONLINE          ONLINE      CBP
1B / 02A0    ONLINE          ONLINE      CBP
1C / 02B0    ONLINE          ONLINE      CBP
1D / 0690    ONLINE          ONLINE      CBP
```

Also note that Figure 17 indicates that PCHIDs 0118 and 0381 are offline. The reason for this is that these are unused CFP CHPIDs in the configuration and, therefore, they will remain in the **Local Uninitialized STP Links** section of the status panel. Finally, notice that G74 is able to exchange STP timing signals with K25 over the previously defined STP timing-only links.

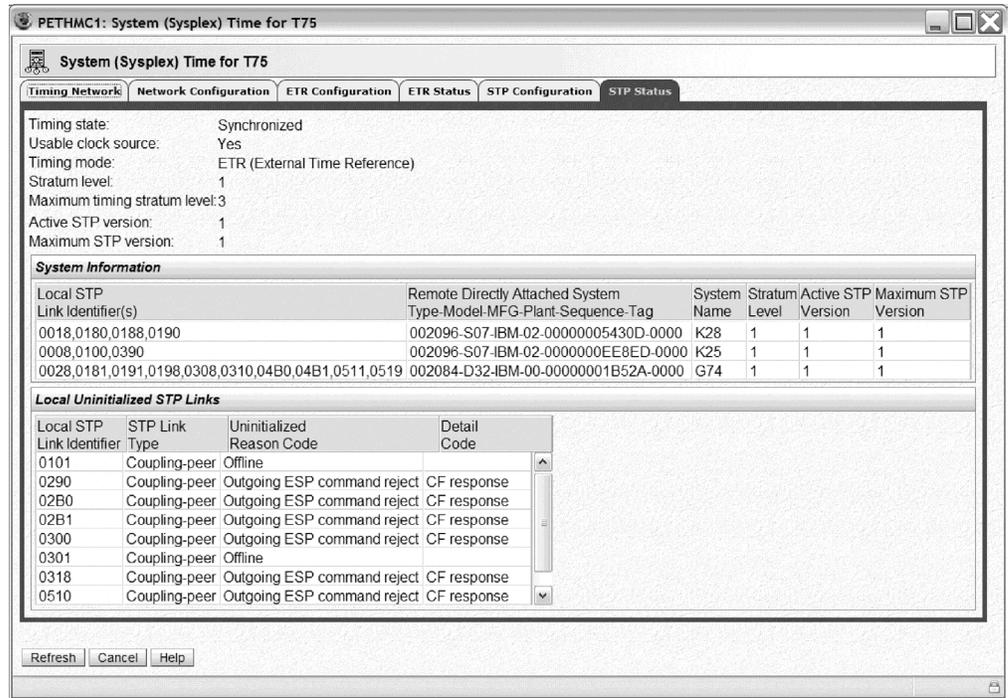


Figure 18. System (Sysplex) Time: STP Status panel on T75 showing connectivity to the other three servers

Figure 19 on page 62 illustrates the timing topology within our data center up to this point in the migration effort.

zPET Mixed CTN Topology

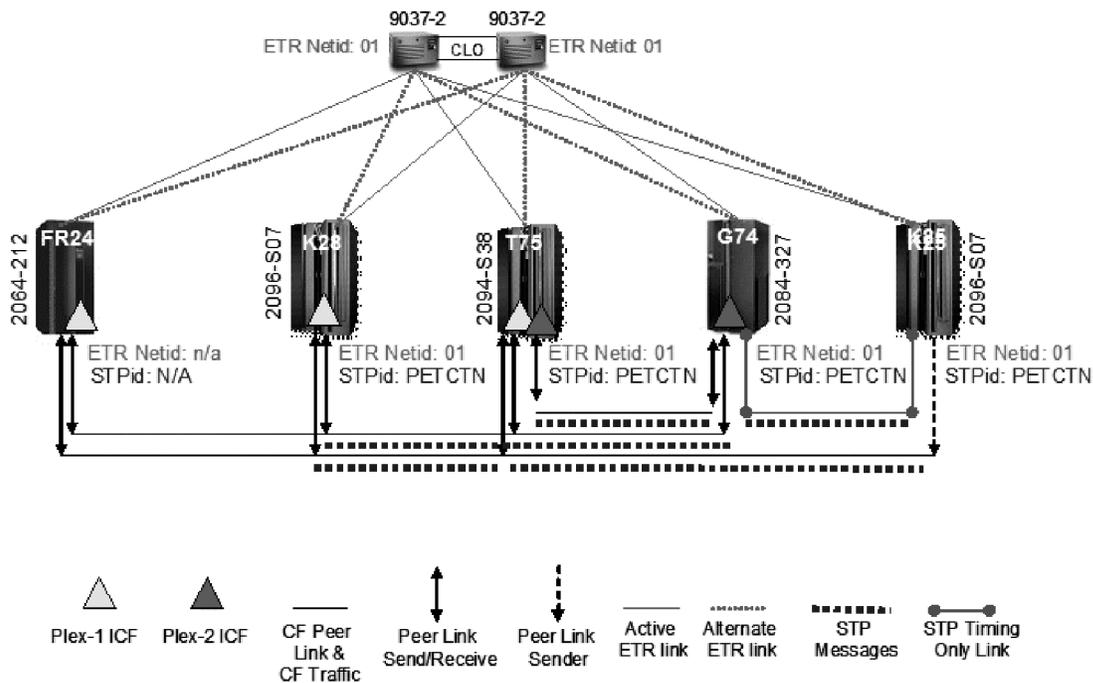


Figure 19. zPET mixed CTN topology

Note the following points about Figure 19:

- All five servers maintained their fully redundant connectivity to the two ETRs at all times through these steps and have also maintained their original ETR network ID of 01.
- Because all of the z/OS images had been IPLed after all of the STP hardware support was installed, all of the z/OS images remained up and running in their respective sysplex without experiencing any impact or without the need for an additional IPL.
- The four STP-configured servers have a CTN ID of PETCTN-01 configured on them. The 2064-212 does not support STP and, therefore, does not have a matching CTN ID but still remains synchronized to the ETRs.
- STP signals are now being exchanged over existing CF peer links, as well as over new STP timing-only links.
- All STP-configured servers now have fully redundant peer link connectivity to every other STP-configured server in the data center.

Stratum 1 to stratum 2 transition and verification for T75

Transitioning an STP-configured server from a stratum 1 position to a stratum 2 position within the timing hierarchy involves intentionally disabling all of the ETR ports on that server.

Once the ETR ports have been disabled, the server will rely on the STP facility to keep it synchronized to the Sysplex Timer ETR. It accomplishes this by using the STP timing signals received from one or more of its STP-configured peers, which still remain directly connected to the ETRs.

Note: It is important to point out that, in a mixed CTN, which has been configured up to this point in this migration effort, the ETRs will continue to remain as the CTN time source. In the simplest terms, this means that an STP-configured server can maintain synchronized timing in a mixed CTN without being directly connected to an ETR, provided that the server is both connected to and is receiving STP messages from at least one STP-configured server that is still connected to at least one ETR.

In this step, we will disable the ETR ports on T75 (the z9 EC CPC) so that it transitions to a stratum 2 position in the timing hierarchy. This step again requires the use of the System (Sysplex) Time task on the SE. Figure 20 shows the initial panel that appears when the System (Sysplex) Time task is selected on T75.

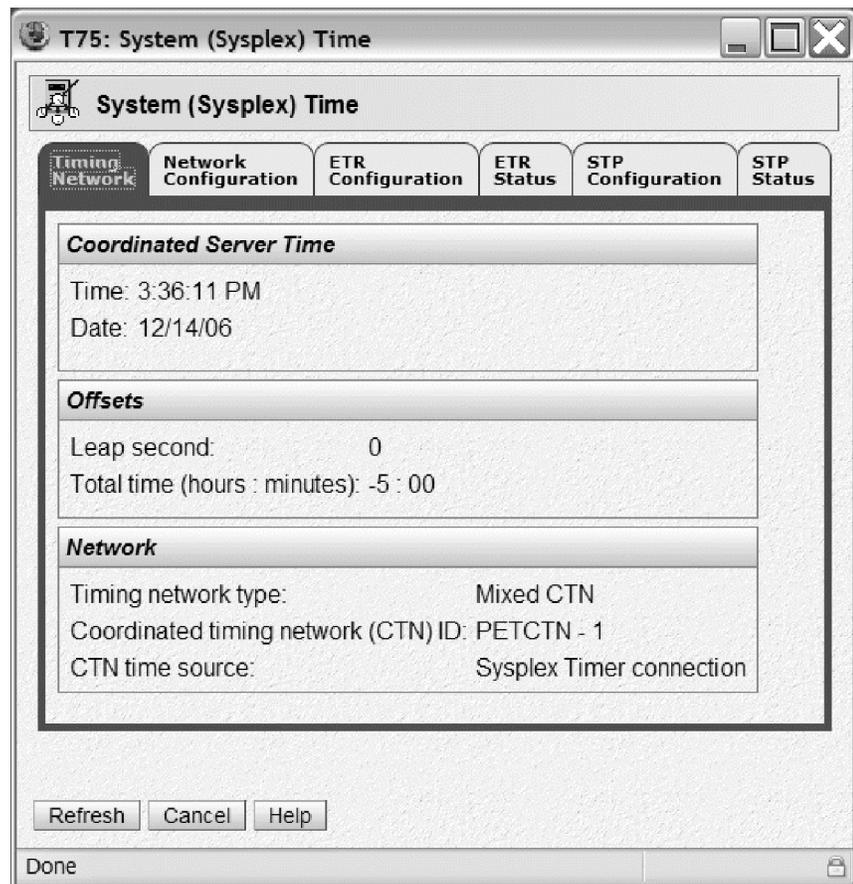


Figure 20. System (Sysplex) Time: Timing Network panel before moving T75 to stratum 2

To ensure that T75 would maintain equal or better timing resiliency during this migration step, we selected the **STP Status** tab as a preliminary verification step, as shown in Figure 21 on page 64.

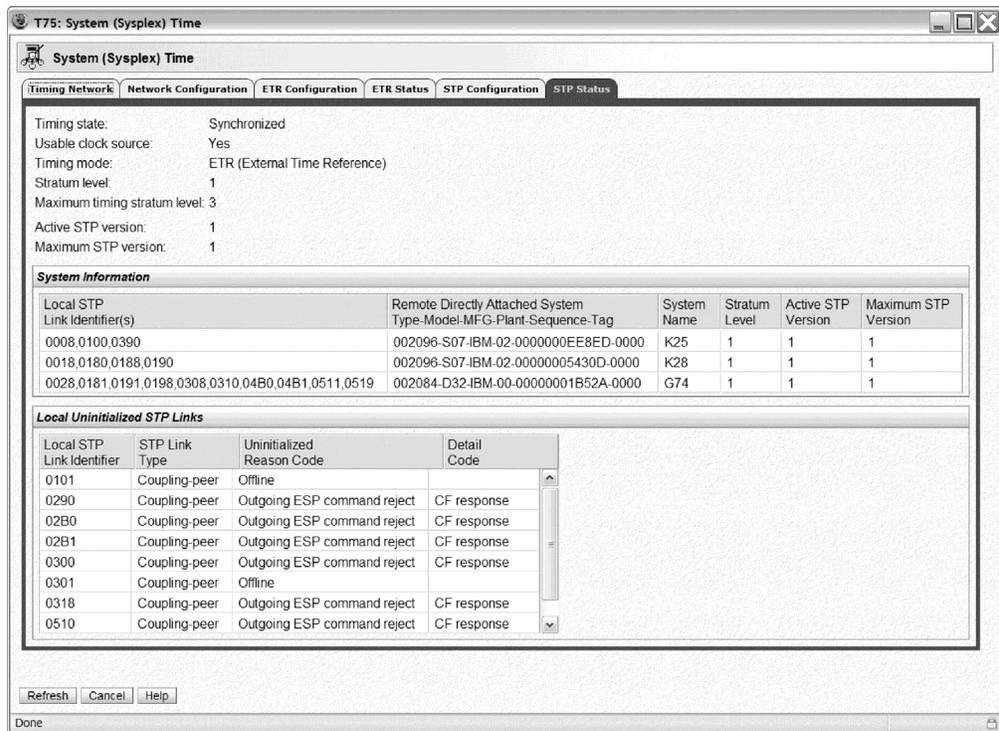


Figure 21. System (Sysplex) Time: STP Status panel before moving T75 to stratum 2

This status panel shows that T75 has the following links that can be used to exchange STP timing signals:

- Three peer links to K25 (0008, 0100, 0390)
- Four peer links to K28 (0018, 0180, 0188, 0190)
- Ten peer links to G74 (0028, 0181, 0191, 0198, 0308, 0310, 04B0, 04B1, 0511, 0519)

Thus, when T75 is moved to the stratum 2 position, the server will be receiving STP timing signal from each of the other three STP-configured servers and there are a total of 17 initialized peer links over which to receive those STP timing signals.

We proceeded to disable the ETR ports on T75, as all of our migration criteria were satisfied.

We selected the **ETR Configuration** tab so that we could disable the ETR ports. Figure 22 on page 65 shows how we selected the **Disabled** buttons for both ETR ports.

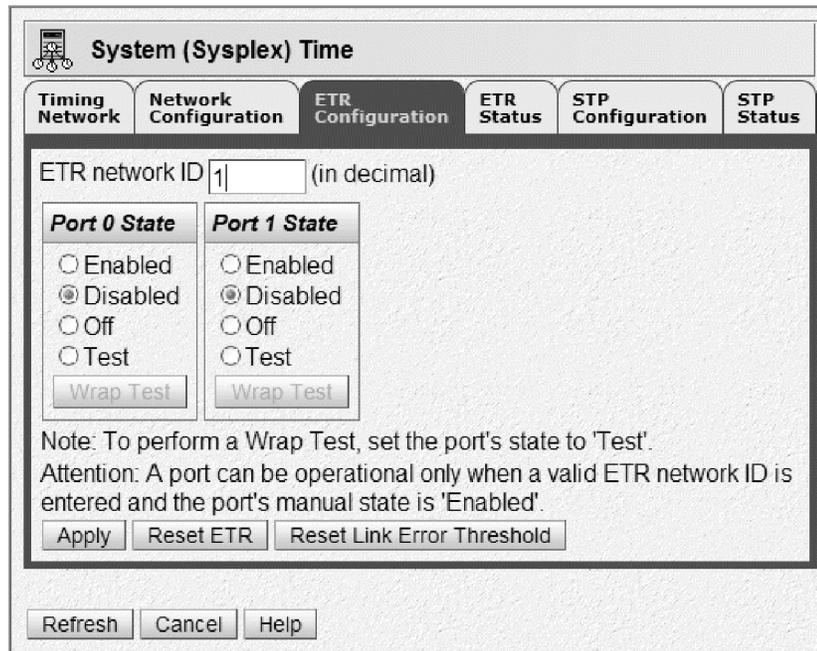


Figure 22. System (Sysplex) Time: ETR Configuration panel with ETR ports disabled

After we clicked the **Apply** button, another panel confirms that the operator truly understands that this is a potentially disruptive action, as shown in Figure 23.

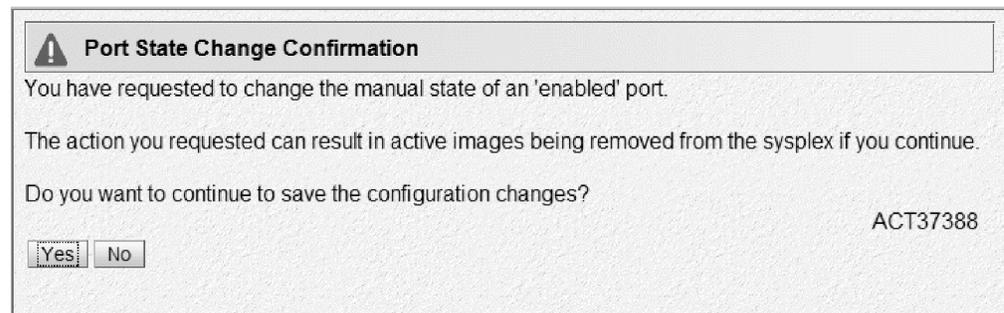


Figure 23. System (Sysplex) Time: ETR Port State Change Confirmation panel

Because we were comfortable with the configuration change, we proceeded by clicking the **Yes** button. Figure 24 on page 66 shows the results of this ETR configuration change.

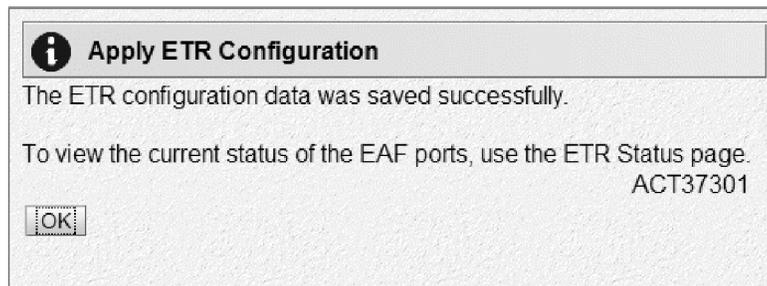


Figure 24. System (Sysplex) Time: Apply ETR Configuration panel, indicating a successful configuration change

In addition to the System (Sysplex) Time verification panel, the z/OS images residing on the server where the ETR ports were disabled will post messages IEA393I and IEA380I, as shown below. Since we simultaneously disabled both ETR ports in the same operation, z/OS posted one message per port disablement.

```
15:08:49.64 *IEA393I ETR PORT 0 IS NOT OPERATIONAL. THIS MAY BE A CTN
              CONFIGURATION CHANGE.
15:08:49.64 *IEA393I ETR PORT 1 IS NOT OPERATIONAL. THIS MAY BE A CTN
              CONFIGURATION CHANGE.
15:08:49.64 IEA380I THIS SYSTEM IS NOW OPERATING IN STP TIMING MODE.
```

Note: It is important to point out that if only one of the two ETR ports had been disabled, the server would have remained directly connected to one of the two ETRs and, thus, would have remained at the stratum 1 position. In that situation, another ETR port disablement step would be needed in order to move the server to the stratum 2 position in the mixed CTN timing hierarchy.

We again used the STP Status panel of the System (Sysplex) Time task to verify that we experienced a successful transition, as shown in Figure 25 on page 67.

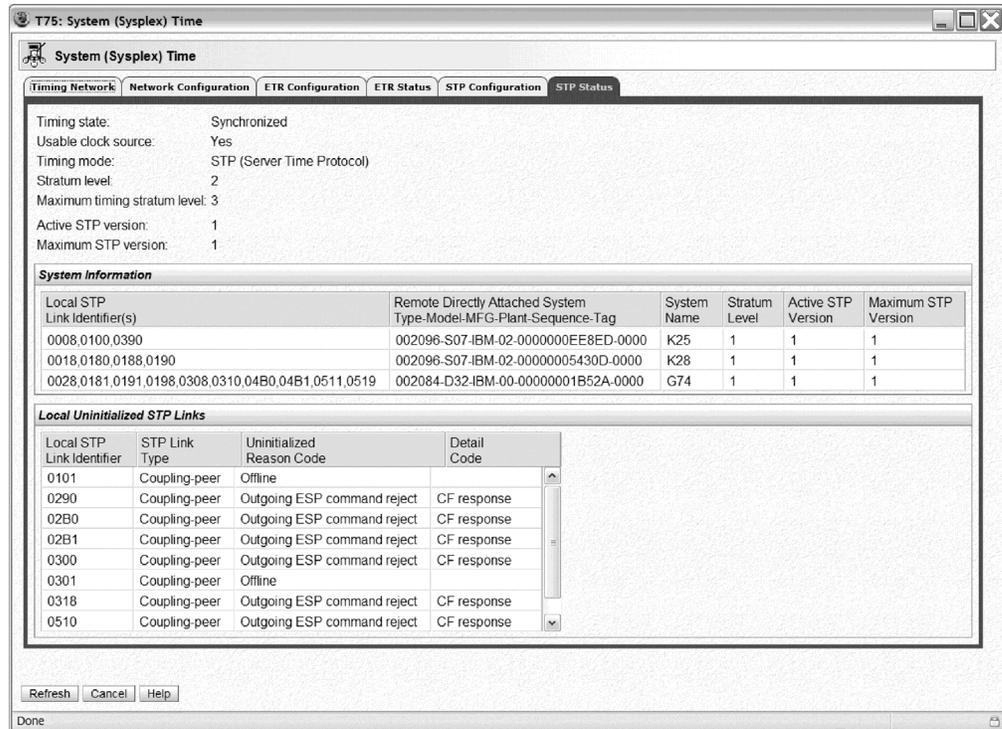


Figure 25. System (Sysplex) Time: STP Status panel with T75 at stratum 2

Note the following about Figure 25:

- The **Timing state** remains synchronized.
- **Note:** The mixed CTN is still synchronized to the Sysplex Timer ETRs.
- The **Timing mode** on T75 is now being reported as STP (Server Time Protocol).
- T75 has, in fact, transitioned to stratum 2, as reflected by the value 2 in the **Stratum level** field.
- Each of the other three STP-configured servers (K28, K25, and G74) all remained at stratum level 1.
- All of the original peer links shown under **STP Link Identifiers** remained active.

In addition to the changed stratum level, the timing mode has now changed to reflect that T75 is now in STP timing mode. Because T75 is in a mixed CTN, the CTN time source remains the Sysplex Timer ETR, as shown in Figure 26 on page 68.

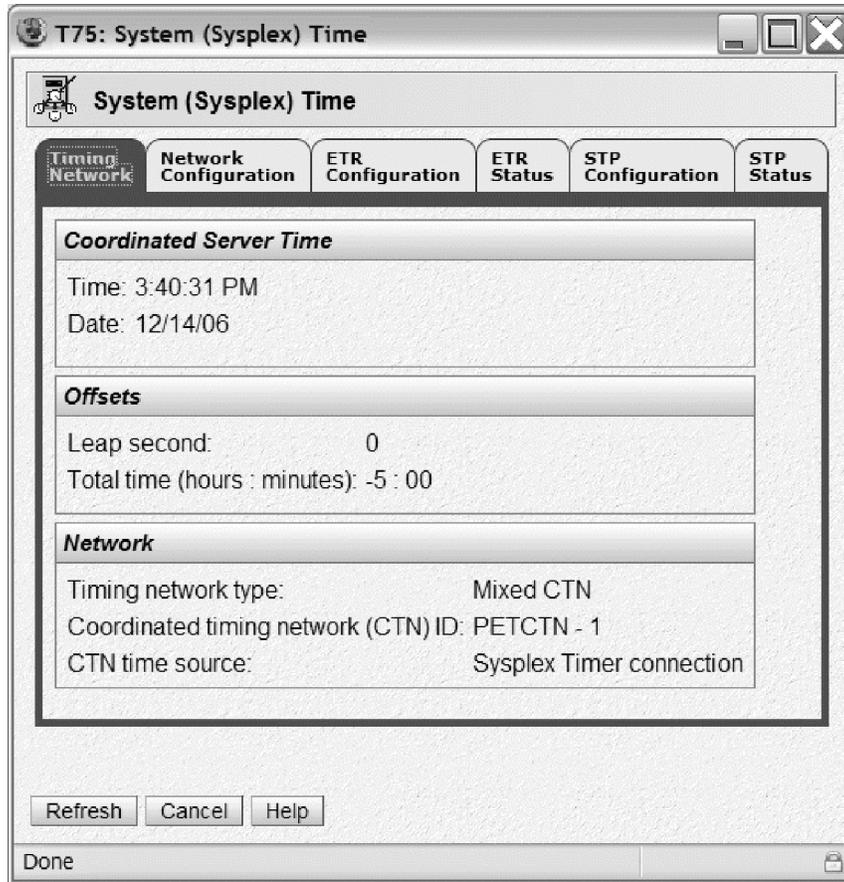


Figure 26. System (Sysplex) Time: Timing Network panel with T75 at stratum 2

We used the z/OS commands DISPLAY ETR and DISPLAY XCF for further verification.

We issued the DISPLAY ETR command to all of the z/OS images residing on T75 to confirm that they are all operating on a stratum 2 server, as shown by the following sample response:

```
IEA386I 15.38.10 TIMING STATUS
SYNCHRONIZATION MODE = STP
THIS SERVER IS A STRATUM 2
CTN ID = PETCTN -01
NUMBER OF USABLE TIMING LINKS = 17
```

Note: The message ID returned by the DISPLAY ETR command has changed from IEA282I to IEA386I when STP is configured on a server. Also, the synchronization mode now indicates that the server is synchronized to the STP facility. The rest of message IEA386I describes timing network information, such as the stratum, the CTN ID, and the number of usable peer links over which the server can receive timing signals.

Issuing the DISPLAY XCF,SYSPLEX,ALL command on any z/OS image in the sysplex now shows that all of the z/OS images running on T75 have a timing mode of STP (TM=STP), as shown in the following sample response:

```
IXC335I 10.45.43 DISPLAY XCF
SYSTEM TYPE SERIAL LPAR STATUS TIME SYSTEM STATUS
TPN 2064 1526 09 08/26/2006 10:45:38 ACTIVE TM=ETR
J80 2094 299E 07 08/26/2006 10:45:43 ACTIVE TM=STP
```

JC0	2084	B52A	0C	08/26/2006	10:45:38	ACTIVE	TM=ETR
Z0	2064	1526	01	08/26/2006	10:45:40	ACTIVE	TM=ETR
JA0	2084	B52A	2A	08/26/2006	10:45:40	ACTIVE	TM=ETR
JB0	2084	B52A	01	08/26/2006	10:45:38	ACTIVE	TM=ETR
J90	2064	1526	05	08/26/2006	10:45:38	ACTIVE	TM=ETR
JH0	2096	FE2D	01	08/26/2006	10:45:40	ACTIVE	TM=ETR
JF0	2094	299E	06	08/26/2006	10:45:40	ACTIVE	TM=STP
JE0	2084	B52A	22	08/26/2006	10:45:40	ACTIVE	TM=ETR

In this case, z/OS images J80 and JF0 reside on T75. All of the other z/OS images are on servers that are in ETR timing mode. The STP Status panel for the other three servers in the mixed CTN will confirm that they remain at stratum 1, while T75 is now a stratum 2 server, as seen in the **System Information** section on the STP Status panel.

Figure 27 illustrates the timing topology in our data center up to this point of the migration.

zPET Mixed CTN Topology with one Stratum 2 Server

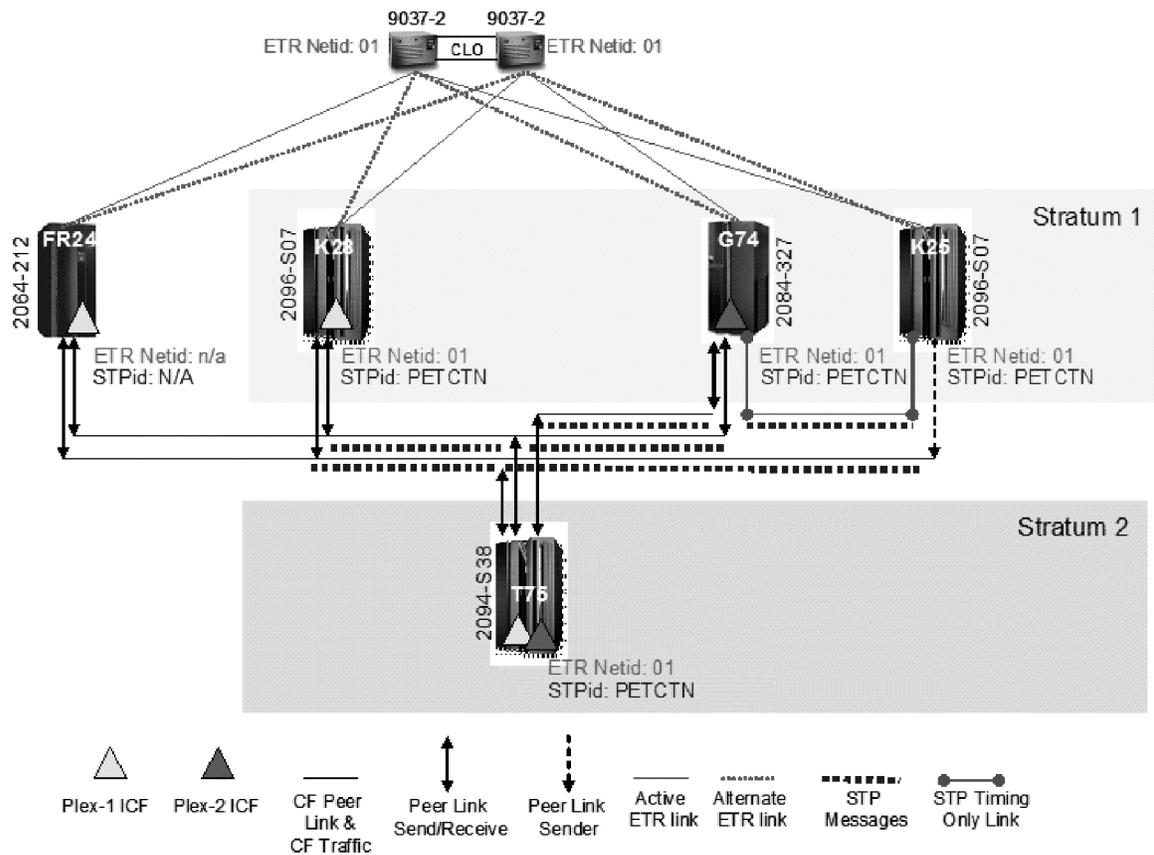


Figure 27. zPET mixed CTN with T75 at stratum 2

Stratum 1 to stratum 2 transition and verification for G74

The next step in the STP migration plan was to move G74 to stratum 2 by disabling both of its ETR ports. Because the procedure for accomplishing this is identical to that used to transition T75 to stratum 2 (as described in “Stratum 1 to

stratum 2 transition and verification for T75” on page 62) and in the interest of brevity, this topic only includes the displays and screen captures that we used to verify the G74 transition.

After we confirmed the ETR port disablement on G74, we verified the server's new position in the timing topology by issuing the z/OS commands DISPLAY ETR and DISPLAY XCF,SYSPLEX,ALL and by examining the server's STP Status panel from within the System (Sysplex) Time task on the SE.

As shown in Figure 28, when we disabled the ETR ports on G74, the STP Status panel verified that G74 remained synchronized, that it was now at the stratum 2 level, and that it maintained the following peer link connectivity:

- Six timing links to stratum1 servers
 - Four of these are CF peer links that are connected to K28
 - Two of these are STP timing-only links which are connected to K25
- Ten CF peer links that are connected to the stratum 2 server, T75

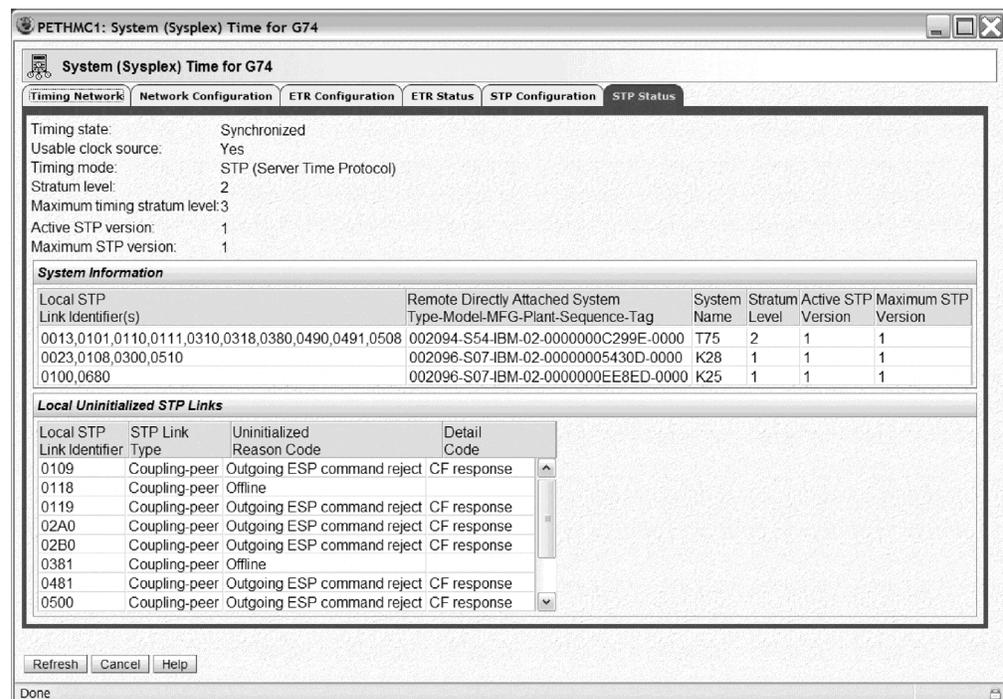


Figure 28. System (Sysplex) Time: STP Status panel for G74 with G74 and T75 at stratum 2

We also performed z/OS verifications by issuing the DISPLAY ETR and DISPLAY XCF,SYSPLEX,ALL commands.

The DISPLAY ETR command routed to a z/OS image residing on G74 resulted in the following response:

```
IEA386I 21.02.56 TIMING STATUS
SYNCHRONIZATION MODE = STP
THIS SERVER IS A STRATUM 2
CTN ID = PETCTN -01
NUMBER OF USABLE TIMING LINKS = 16
```

This verifies several important aspects of the migration:

- The z/OS image resides on a server that is currently synchronized using STP.

- This server is at the stratum 2 level.
- This server belongs to the PETCTN-01 mixed CTN.
- There are a total of 16 timing links.

The DISPLAY XCF,SYSPLEX,ALL command issued on any z/OS image in the sysplex returned the following information:

```
IXC335I 21.02.32 DISPLAY XCF
SYSTEM  TYPE SERIAL LPAR STATUS TIME          SYSTEM STATUS
JC0     2084 B52A 0C   10/01/2006 21:02:29 ACTIVE          TM=STP
JB0     2084 B52A 01   10/01/2006 21:02:30 ACTIVE          TM=STP
TPN     2064 1526 09   10/01/2006 21:02:31 ACTIVE          TM=ETR
Z0      2064 1526 01   10/01/2006 21:02:30 ACTIVE          TM=ETR
J80     2094 299E 07   10/01/2006 21:02:32 ACTIVE          TM=STP
JF0     2094 299E 06   10/01/2006 21:02:30 ACTIVE          TM=STP
JA0     2084 B52A 2A   10/01/2006 21:02:29 ACTIVE          TM=STP
J90     2064 1526 05   10/01/2006 21:02:29 ACTIVE          TM=ETR
JH0     2096 FE2D 01   10/01/2006 21:02:31 ACTIVE          TM=ETR
JE0     2084 B52A 22   10/01/2006 21:02:30 ACTIVE          TM=STP
```

This shows that four more z/OS images in the sysplex (JC0, JB0, JA0, and JE0) have transitioned to STP time synchronization on G74.

Figure 29 on page 72 illustrates the new timing topology up to this point. It shows how each server has maintained redundant timing synchronization, either by Sysplex Timer links, CF peer links, or STP timing-only peer links.

zPET Mixed CTN Topology with two Stratum 2 Servers

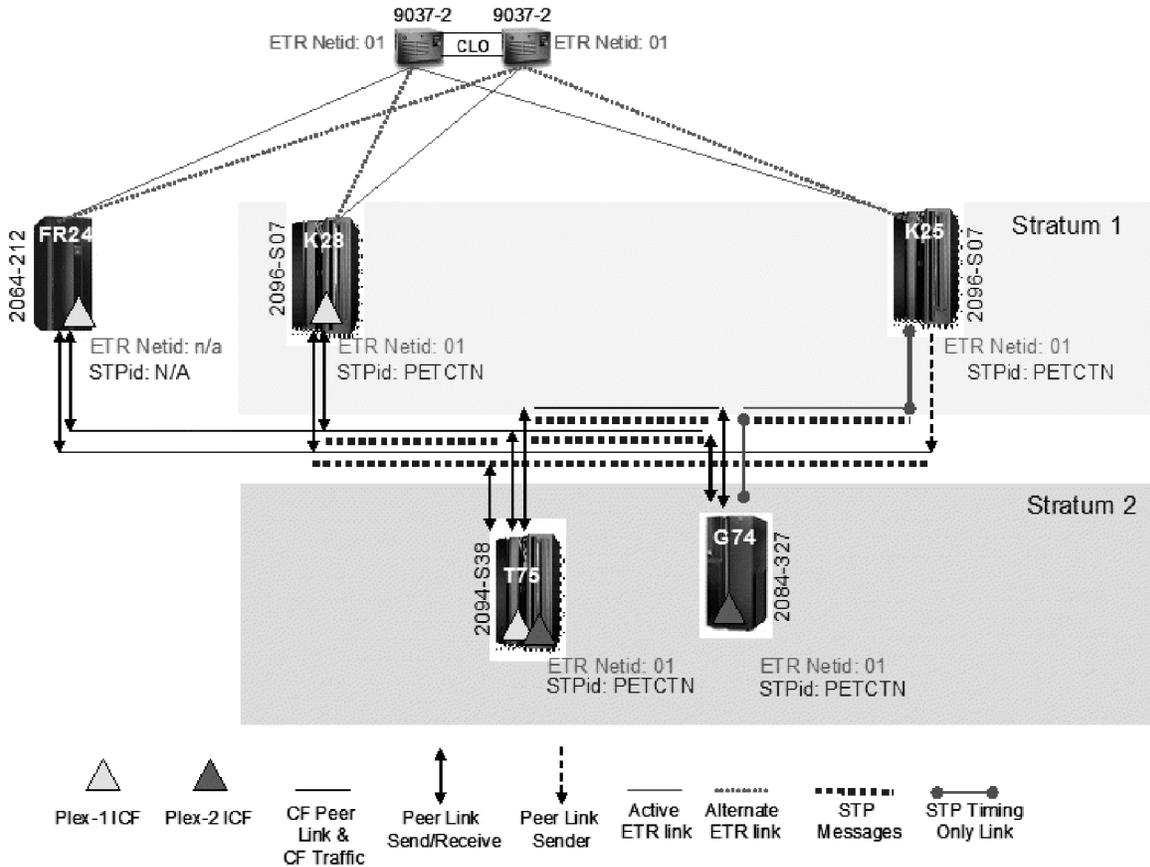


Figure 29. zPET mixed CTN with two stratum 2 nodes: T75 and G74

Reverse migration: Stratum 2 to stratum 1 transition and verification

An important step in any concurrent migration strategy is the ability to concurrently roll back any changes to a stable starting point. STP provides this concurrent reverse migration support.

In this topic, we show the steps that we took to reverse our timing network topology from the one shown in Figure 29 to the original topology shown in Figure 4 on page 47.

The first step is to move all stratum 2 servers to the stratum 1 level by enabling their respective ETR ports. Figure 30 on page 73, Figure 31 on page 73, and Figure 32 on page 74 show the panels that we used to enable the ETR ports and to verify that the task succeeded.

We used the ETR Configuration panel to enable the ETR ports, as shown in Figure 30 on page 73. However, before enabling either ETR port, the ETR Status panel should be used to verify that the ETR card status shows Light detected for each port and that the ETR status word state shows Semi-operational for each port. This will ensure that the ETR ports are in a state that can be used to receive

Sysplex Timer signals before enabling them. Enabling each port involves selecting the **Enabled** button for each, then clicking the **Apply** button.

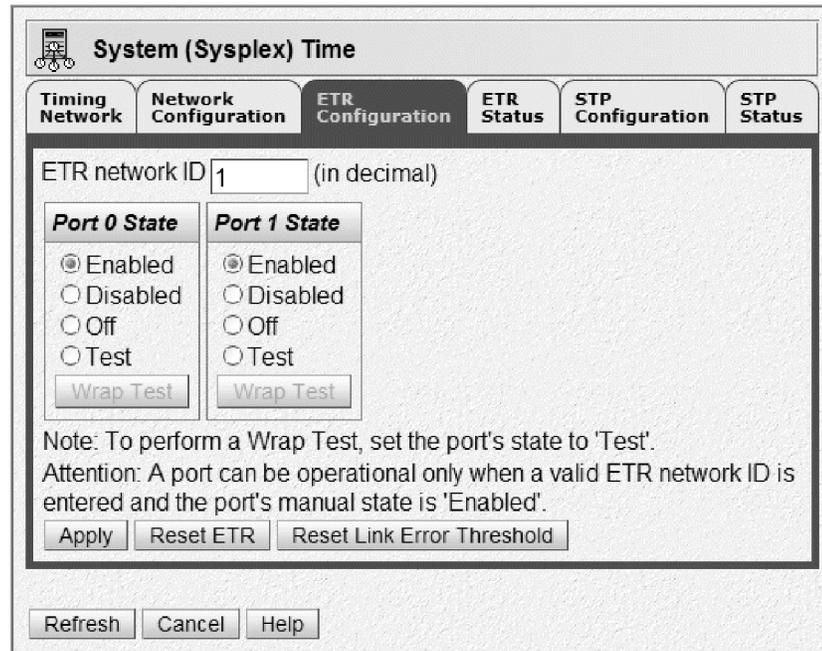


Figure 30. System (Sysplex) Time: ETR Configuration panel for port enablement

A confirmation panel appears to indicate that the ETR ports were successfully enabled, as shown in Figure 31.

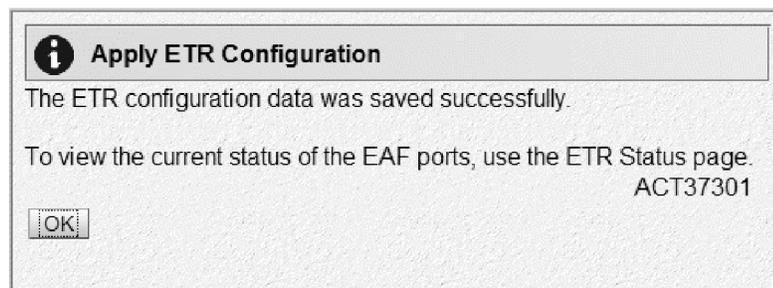


Figure 31. ETR Configuration confirmation panel

The z/OS images residing on that server will post a message for each ETR port that has been enabled. The following z/OS message capture illustrates how z/OS recognized the availability of the ETR ports when they have been re-enabled and how z/OS has dynamically adjusted the system's time of day (TOD) clock to maintain synchronized timing with the newly connected ETRs.

The following z/OS messages accompany the reverse transition:

```
J80      06275 00:08:59.37 IEA267I ETR PORT 0 IS NOW AVAILABLE.
J80      06275 00:08:59.37 IEA267I ETR PORT 1 IS NOW AVAILABLE.
J80      06275 00:09:12.75 IEA260I THE CPC IS NOW OPERATING IN ETR MODE.
J80      06275 00:09:20.08 IEA273I TOD CLOCKS DYNAMICALLY ADVANCED TO MAINTAIN
                                ETR SYNCHRONISM.
```

We again used the STP Status panel from within the System (Sysplex) Time task on the HMC to verify that the server has properly transitioned within the timing

network. Figure 32 indicates that T75 has now transitioned back to a stratum 1 position and is once again back in ETR timing mode.

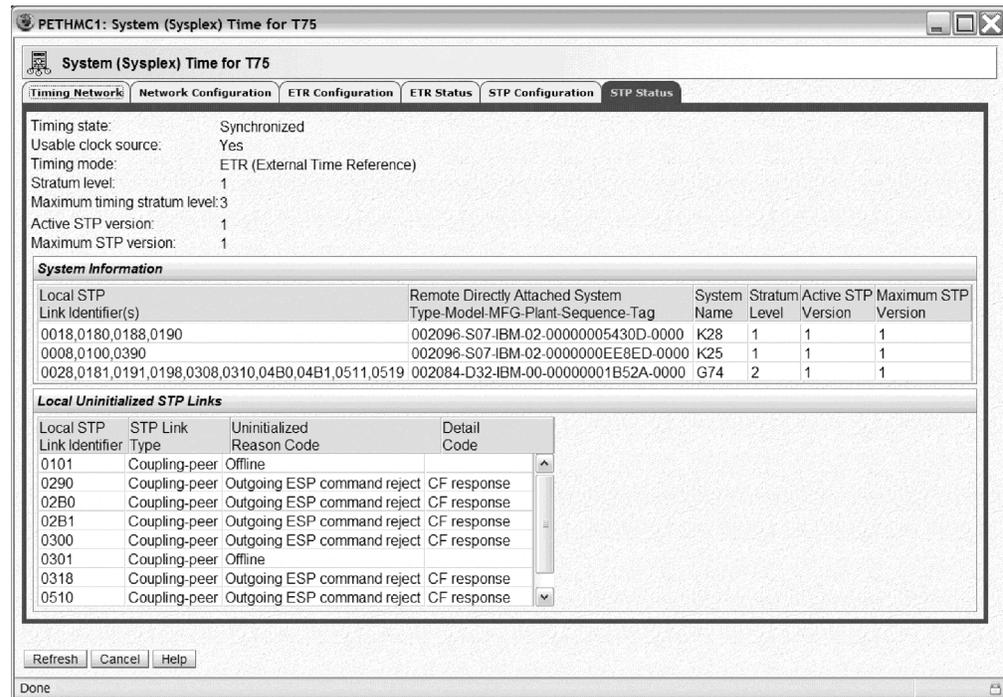


Figure 32. System (Sysplex) Time: STP Status panel, showing T75 back at stratum 1

Additional verification of the server's timing status involved issuing the z/OS DISPLAY ETR and DISPLAY XCF,SUSPLEX,ALL commands. For example, the following message capture shows that the J80 z/OS image residing on the T75 server recognizes that both of the ETR ports are now operational:

```
IEA282I 00.32.38 TIMING STATUS
SYNCHRONIZATION MODE = ETR
CPC PORT 0 <== ACTIVE CPC PORT 1
OPERATIONAL OPERATIONAL
ENABLED ENABLED
ETR NET ID=01 ETR NET ID=01
ETR PORT=09 ETR PORT=09
ETR ID=01 ETR ID=00
THIS SERVER IS PART OF TIMING NETWORK PETCTN -01
```

The next message capture resulting from issuing the DISPLAY XCF,SYSPLEX,ALL command on any of the z/OS images shows that both of the z/OS images (J80 and JF0) residing on the T75 server now report a timing mode of ETR (TM=ETR):

```
IXC335I 00.33.24 DISPLAY XCF
SYSTEM TYPE SERIAL LPAR STATUS TIME SYSTEM STATUS
JC0 2084 B52A 0C 10/02/2006 00:33:20 ACTIVE TM=STP
JB0 2084 B52A 01 10/02/2006 00:33:19 ACTIVE TM=STP
TPN 2064 1526 09 10/02/2006 00:33:18 ACTIVE TM=ETR
Z0 2064 1526 01 10/02/2006 00:33:19 ACTIVE TM=ETR
J80 2094 299E 07 10/02/2006 00:33:24 ACTIVE TM=ETR
JF0 2094 299E 06 10/02/2006 00:33:20 ACTIVE TM=ETR
JA0 2084 B52A 2A 10/02/2006 00:33:20 ACTIVE TM=STP
J90 2064 1526 05 10/02/2006 00:33:19 ACTIVE TM=ETR
JH0 2096 FE2D 01 10/02/2006 00:33:20 ACTIVE TM=ETR
JE0 2084 B52A 22 10/02/2006 00:33:21 ACTIVE TM=STP
```

Next, we enabled the ETR ports on G74 to move it back to the stratum 1 level. (Note that the respective screen captures have been omitted for brevity.) At this point, all of the z/OS images residing on STP-configured servers in our sysplex indicated that they were back in ETR timing mode and that they were also still part of the PETCTN-01 mixed CTN. The following sample IEA282I and IXC335I message captures illustrate these points:

```
IEA282I 11.59.14 TIMING STATUS
SYNCHRONIZATION MODE = ETR
CPC PORT 0 <== ACTIVE CPC PORT 1
OPERATIONAL OPERATIONAL
ENABLED ENABLED
ETR NET ID=01 ETR NET ID=01
ETR PORT=04 ETR PORT=04
ETR ID=00 ETR ID=01
THIS SERVER IS PART OF TIMING NETWORK PETCTN -01

IXC335I 12.00.59 DISPLAY XCF
SYSTEM TYPE SERIAL LPAR STATUS TIME SYSTEM STATUS
JC0 2084 B52A 0C 10/02/2006 12:00:55 ACTIVE TM=ETR
JB0 2084 B52A 01 10/02/2006 12:00:57 ACTIVE TM=ETR
TPN 2064 1526 09 10/02/2006 12:00:55 ACTIVE TM=ETR
Z0 2064 1526 01 10/02/2006 12:00:57 ACTIVE TM=ETR
J80 2094 299E 07 10/02/2006 12:00:59 ACTIVE TM=ETR
JF0 2094 299E 06 10/02/2006 12:00:56 ACTIVE TM=ETR
JA0 2084 B52A 2A 10/02/2006 12:00:57 ACTIVE TM=ETR
J90 2064 1526 05 10/02/2006 12:00:54 ACTIVE TM=ETR
JH0 2096 FE2D 01 10/02/2006 12:00:55 ACTIVE TM=ETR
JE0 2084 B52A 22 10/02/2006 12:00:57 ACTIVE TM=ETR
```

Reverse migration: Mixed CTN to ETR timing network

The final step for completely backing out of a mixed CTN and returning to the original ETR only timing network is to remove the STP ID portion of the CTN ID for each STP-configured server. To perform this step, we used the STP Configuration panel from within the System (Sysplex) Timer task on the HMC to remove the STP ID portion of the CTN ID.

Figure 33 through Figure 35 on page 76 show the sequence of panels associated with removing the STP ID from the CTN ID. Specifically, Figure 33 shows the removal of the STP ID.

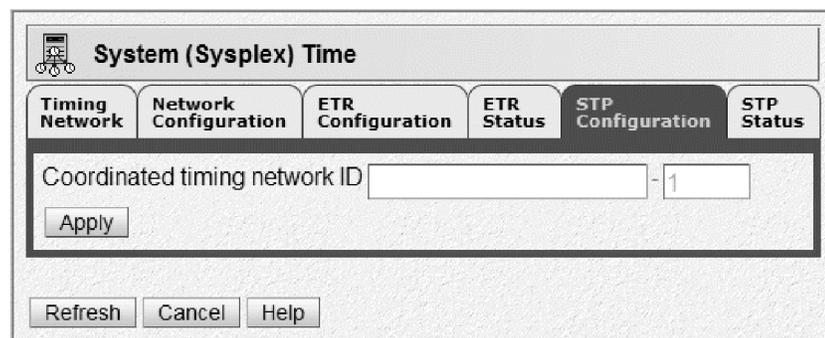


Figure 33. STP Configuration: STP ID removal

After clicking the **Apply** button on the STP Configuration panel, a confirmation panel appeared, as shown in Figure 34 on page 76.

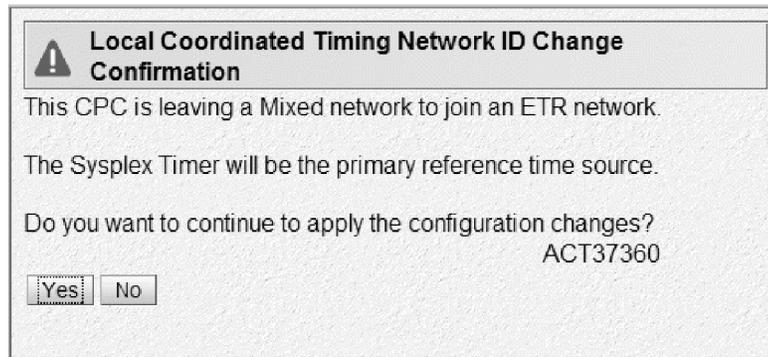


Figure 34. STP Configuration: CTN Network ID Change Confirmation panel

After clicking the **Yes** button on the CTN Network ID Change Confirmation panel, a final confirmation panel appeared indicating that the configuration change was successfully completed, as shown in Figure 35.

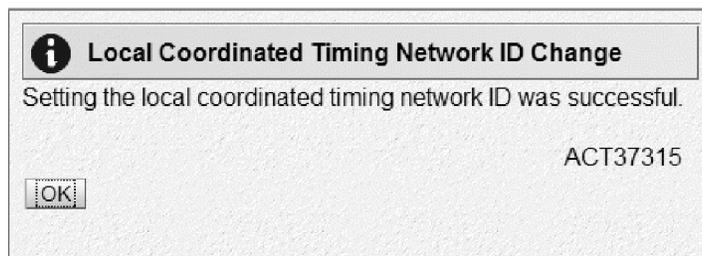


Figure 35. STP Configuration: CTN Network ID Change completion

The z/OS images residing on the server where the STP ID had just been removed posted a message indicating that the CTN ID had changed, as well. The following is an example of the message indicating that z/OS recognized that the CTN ID changed:

```
IXC438I COORDINATED TIMING INFORMATION HAS BEEN UPDATED
        FOR SYSTEM: JC0
        PREVIOUS CTN ID:  PETCTN  -01
        CURRENT  ETR NETID:  01
```

We again used the STP Status panel to confirm that G74 was no longer in the mixed CTN. By examining the STP Status panel shown in Figure 36 on page 77, we were able to confirm that G74 no longer had any servers listed in the System Information section.

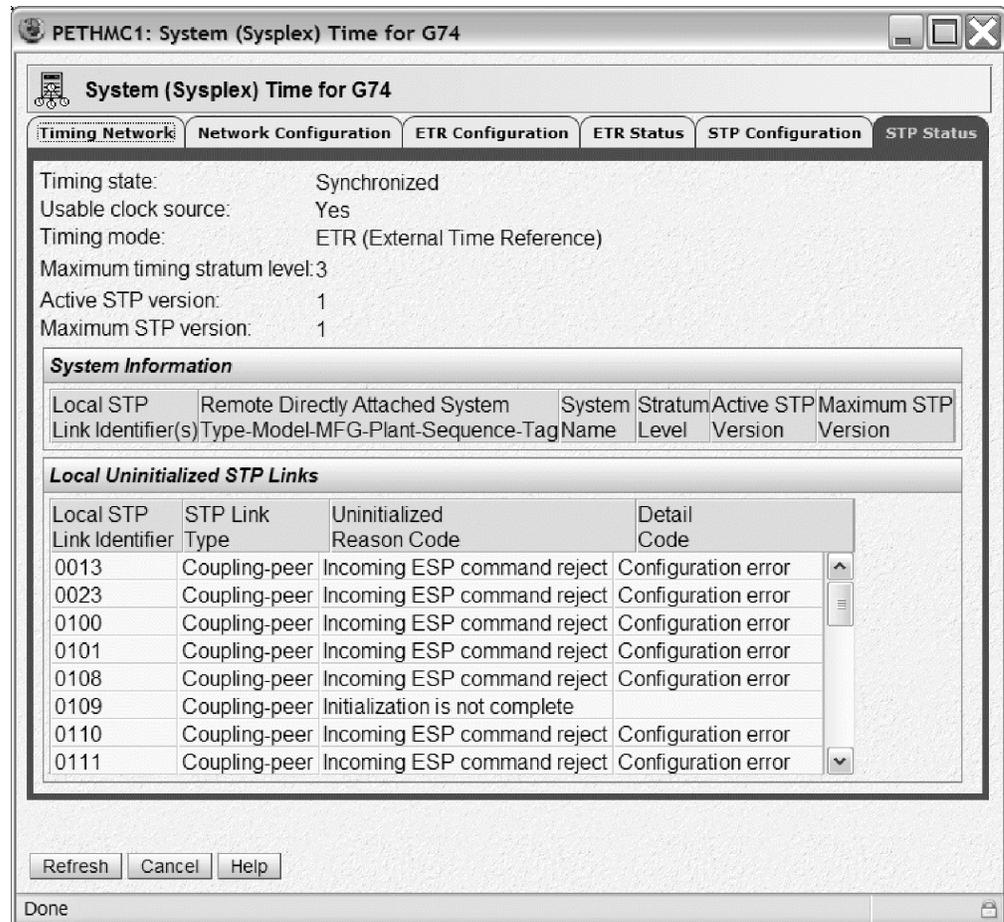


Figure 36. System (Sysplex) Time: STP Status panel, showing that G74 had returned to an ETR timing network

Issuing the z/OS command DISPLAY ETR on any z/OS image residing on G74 provided the final confirmation that G74 had non-disruptively been returned to an ETR timing network. The following is an example of the command response indicating that the server returned to an ETR timing network, as the message no longer indicates that the server is part of the PETCTN-01 mixed CTN:

```
IEA282I 12.15.52 TIMING STATUS
SYNCHRONIZATION MODE = ETR
CPC PORT 0 <== ACTIVE      CPC PORT 1
OPERATIONAL                OPERATIONAL
ENABLED                    ENABLED
ETR NET ID=01              ETR NET ID=01
ETR PORT=04                ETR PORT=04
ETR ID=00                  ETR ID=01
```

Although they were rarely encountered during our migration testing, we also observed some system logger abends during migration from a mixed CTN to an ETR-only network. Operations log facility (OPERLOG) experienced the following SVC dumps on images on the CPC for which the ETR ports were enabled:

```
IEA267I ETR PORT 0 IS NOW AVAILABLE.
IEA267I ETR PORT 1 IS NOW AVAILABLE.
IEA260I THE CPC IS NOW OPERATING IN ETR MODE.
IEA273I TOD CLOCKS DYNAMICALLY ADVANCED TO MAINTAIN ETR SYNCHRONISM.
IXG063I LOGGER ABENDED AND REQUESTED AN
SVC DUMP WHILE PROCESSING LOGSTREAM: SYSPLEX.OPERLOG
STRUCTURE: LOGGER_OPERLOG
```

```
MODULE=IXGF2WRT,ABEND=S01C5,  
REASON=00040003  
IEA794I SVC DUMP HAS CAPTURED:  
DUMPID=011 REQUESTED BY JOB (CONSOLE )  
TITLE=COMPON=LOGGER,COMPID=5752SCLOG,  
ISSUER=IXGR1REC,MODULE=IXGF2WRT,ABEND=S01C5,REASON=00040003
```

This is working as designed because, according to *z/OS MVS Planning: Operations* and *z/OS MVS Setting Up a Sysplex*, the system logger uses the system clock GMT value as an authorization key when writing to the coupling facility on behalf of the log stream. If you change the GMT, specifically turning the clock back, system logger will not be able to write to the log stream until the new GMT is greater than the old GMT. Thus, depending on how long the stratum 2 server resides in STP-timing mode, the TOD clock may drift enough for OPERLOG to detect that a system on an ETR-timing CPC may be less advanced than one on the STP-timing CPC. The S01C5 abend is taken to indicate this condition but OPERLOG continues operating without any other problems. This may be encountered on any images on the STP-timing server if they are actively writing log blocks to the OPERLOG log stream. A LOGREC entry might be recorded in addition to the dump. Refer to *z/OS MVS System Codes* for the appropriate system action and system programmer response for this condition, especially if it persists.

We then repeated the same steps to delete the STP ID on each of the remaining STP-configured servers so that all servers would return to their original ETR timing network configuration and original Sysplex Timer timing synchronization, as shown in Figure 4 on page 47.

Migrating from a mixed CTN to an STP-only CTN

In order to concurrently migrate from an ETR-only timing network to an STP-only CTN, you must first configure a mixed CTN as an intermediate step. This is because of the way a CTN ID is concurrently defined. The CTN ID in a STP-only CTN is comprised of an STP ID portion concatenated with a null ETR ID. In order to maintain time synchronization when migrating from an ETR-only timing network to an STP-only CTN, you must first define the STP ID. By definition, this step creates a mixed CTN.

Next, using the Network Configuration panel in the System (Sysplex) Time task, the STP facility will remove the ETR ID to create the *STP ID – null ETR ID* pair for an STP-only CTN ID. We will discuss this second step in more detail later but, for now, our first step in configuring an STP-only CTN in our data center was to return our STP-capable servers to the mixed CTN topology shown in Figure 19 on page 62.

As we discussed in the planning considerations in “Considerations for migrating from a mixed CTN to an STP-only CTN” on page 43, we needed to remove our z900 (2064-212) server, FR24, from our Parallel Sysplex before migrating to an STP-only CTN. Since we run a significant portion of our workload on this server during z/OS integration testing, our STP-only migration could not proceed until such time that we no longer required the z900 server images in our Parallel Sysplex and we could remove this non-STP-capable server from our configuration. This change window occurred as we began to prepare for the IBM System z10 EC server, as described in “Appendix A. About our Parallel Sysplex environment” on page 311, since a z900 is not supported in the same Parallel Sysplex as a System z10 server. In addition, as we migrated our z/OS images to z/OS V1R9, we no

longer needed to run z/OS.e in our Parallel Sysplex since z/OS.e is supported only up to z/OS.e V1R8, so we decided to remove our second System z9 BC (2096-S07) server, K25.

After removing the z900 and z9 BC servers, our Parallel Sysplex environment then looked as shown in Figure 37.

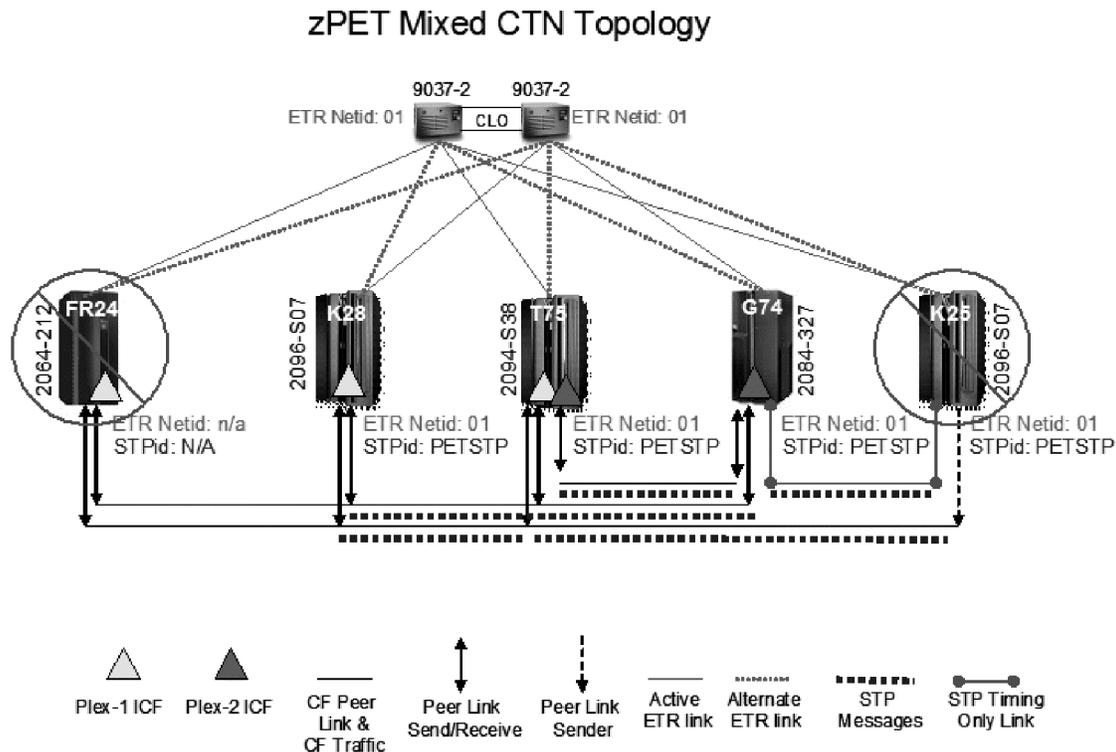


Figure 37. zPET mixed CTN with FR24 and K25 removed

At this point, we were able to proceed with the migration to an STP-only CTN. This involved assigning the role of preferred time server, optionally assigning the role of backup time server, and optionally assigning the role of arbiter (if a backup time server role is also assigned) all on the intended current time server using the **Network Configuration** tab of the System (Sysplex) Time task on the HMC.

First, we had to decide which of our three remaining servers would provide sufficient connectivity to serve as the preferred time server and backup time server. In our current configuration, both K28 and T75 house internal coupling facility LPARs, each of which has at least one z/OS image on every other server that requires connectivity to it. Therefore, since K28 and T75 are both connected to every other server via coupling facility peer links and can provide timing synchronization for every other server, they were our best candidates for preferred and backup time servers. Between these two servers, we chose T75 to be the preferred time server because it has z/OS images residing on it. K28 is a CF-only server and, thus, lacks the ability to produce solicited and unsolicited z/OS messages that are produced during CTN configuration changes, which are useful for immediate verification, problem diagnosis, and change history.

We chose to assign the arbiter role to G74.

Figure 38 shows the **Network Configuration** tab of the System (Sysplex) Time task on the HMC for T75 before assigning any roles.

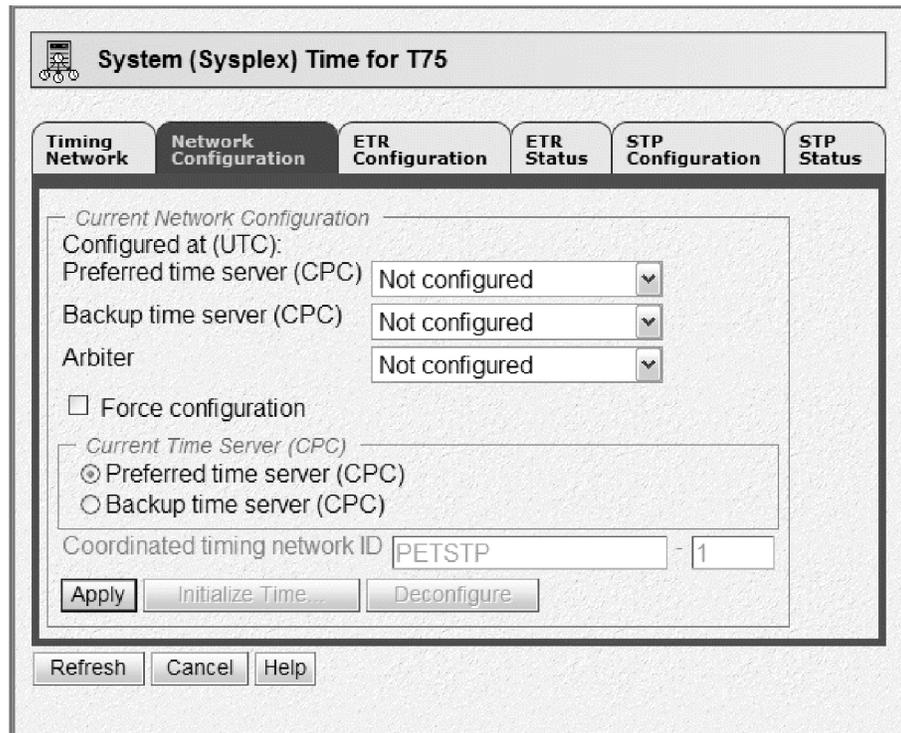


Figure 38. Initial view of the System (Sysplex) Time – Network Configuration panel on T75

Next, we used the drop-down lists for **Preferred time server (CPC)**, **Backup time server (CPC)**, and **Arbiter** in the Current Network Configuration section of the **Network Configuration** tab to assign the preferred time Server role to T75, backup time server role to K28, and arbiter role to G74. Note that this assignment must be done from the intended current time server, which will become the stratum 1 server (that is, the time source) in the CTN. As described in “STP terminology” on page 40, the current time server role must be assigned to either the preferred or backup time server. The Current Time Server (CPC) section of the **Network Configuration** tab allows the operator to select whether the current time server role is to be assigned to the preferred or backup time server. In order to assign the current time server to the preferred time server, you must use the **Network Configuration** tab on the System (Sysplex) Time task for the preferred time server CPC. Alternatively, in order to assign the current time server to the backup time server, you must use the **Network Configuration** tab on the System (Sysplex) Time task for the backup time server CPC.

Figure 39 on page 81 shows this configuration in our environment.

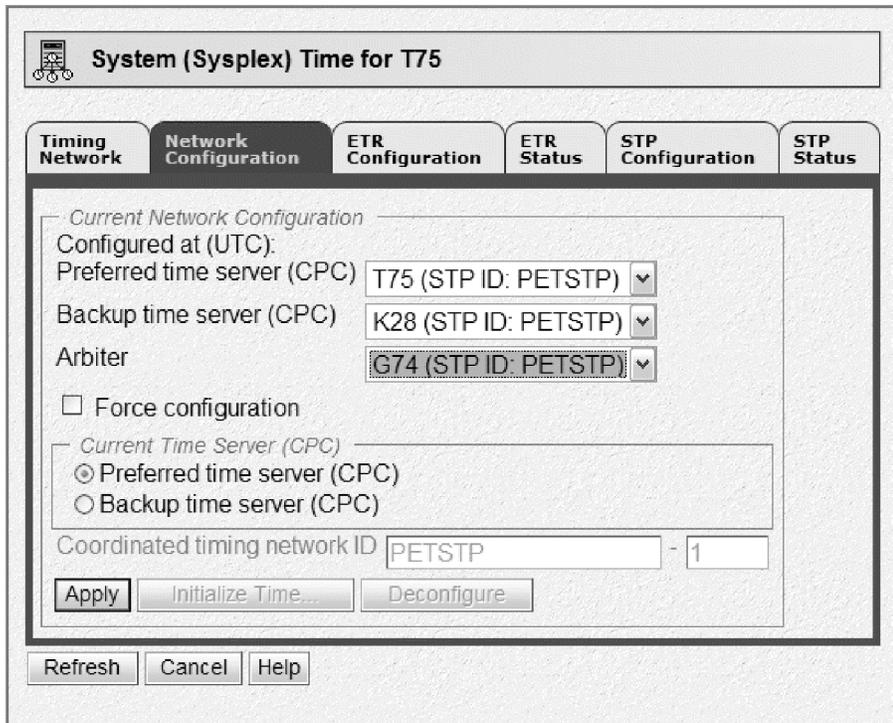


Figure 39. System (Sysplex) Time task – Network Configuration panel with all server roles assigned

Clicking the **Apply** button on the **Network Configuration** tab causes the mixed CTN to STP-only CTN migration to begin. Figure 40 shows the Global Timing Network ID Change Confirmation dialog that appears.

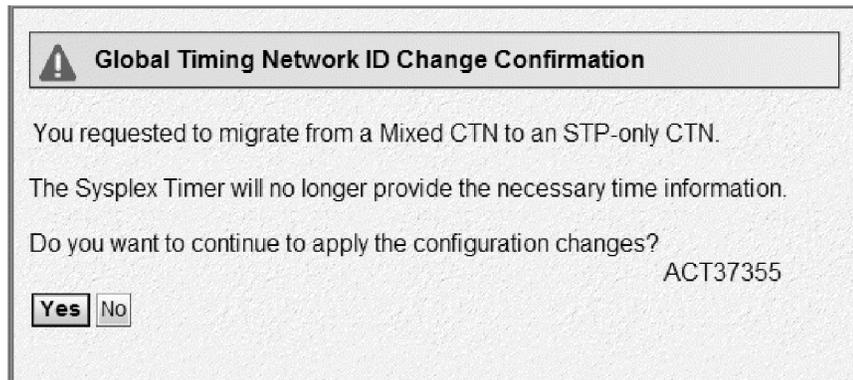


Figure 40. Global Timing Network ID Change Confirmation

Clicking **Yes** in Figure 40 begins the transition to an STP-only CTN. As this transition begins, z/OS images residing on the servers in the CTN will provide various messages that indicate the progress of the transition.

First, any images residing on servers that were previously directly attached to the ETRs (and, therefore, were stratum 1 servers) in the mixed CTN will report a loss of Sysplex Timer connectivity (message ID IEA393I) on each active ETR port, as in this example:

```
*IEA393I ETR PORT 0 IS NOT OPERATIONAL. THIS MAY BE A CTN CONFIGURATION CHANGE.
*IEA393I ETR PORT 1 IS NOT OPERATIONAL. THIS MAY BE A CTN CONFIGURATION CHANGE.
```

In addition, any images residing on servers that were previously directly attached to the ETRs will recognize the transition to STP timing mode and issue message IEA380I:

```
IEA380I THIS SYSTEM IS NOW OPERATING IN STP TIMING MODE.
```

Note that images residing on servers that were previously operating at stratum 2 in the mixed CTN were already in STP timing mode (that is, they were using STP to remain synchronized to the stratum 1 servers in the mixed CTN, which in turn were synchronized to the ETRs) and, therefore, will not present message IEA380I.

Each z/OS image should recognize the CTN ID change from PETSTP-01 to PETSTP and should report this via message IXC438I, as in this example:

```
IXC438I COORDINATED TIMING INFORMATION HAS BEEN UPDATED
        FOR SYSTEM: J80
        PREVIOUS CTNID:  PETSTP  -01
        CURRENT  CTNID:  PETSTP
```

This CTN ID change does not occur simultaneously across a Parallel Sysplex and several messages may be seen that indicate that there is a temporary mismatch of CTN IDs between images, including CFs, in the Parallel Sysplex. First, IXC439E is issued by z/OS images that recognize other images with mismatched CTN IDs, which might indicate that those images with mismatched CTN IDs might be synchronized to a different time reference. During the transition window, this message is expected, as in this example:

```
*IXC439E ALL SYSTEMS IN SYSPLEX UTCPLXJ8 ARE NOT SYNCHRONIZED
        TO THE SAME TIME REFERENCE.
        SYSTEM: JF0  IS USING CTNID: PETSTP  -01
        SYSTEM: J80  IS USING CTNID: PETSTP
        SYSTEM: JA0  IS USING CTNID: PETSTP  -01
        SYSTEM: JB0  IS USING CTNID: PETSTP  -01
        SYSTEM: J90  IS USING CTNID: PETSTP  -01
        SYSTEM: JC0  IS USING CTNID: PETSTP  -01
        SYSTEM: JE0  IS USING CTNID: PETSTP  -01
```

As more images transition to the STP-only CTN, IXC439E will show the updated CTN IDs, as in this example:

```
*IXC439E ALL SYSTEMS IN SYSPLEX UTCPLXJ8 ARE NOT SYNCHRONIZED
        TO THE SAME TIME REFERENCE.
        SYSTEM: JF0  IS USING CTNID: PETSTP
        SYSTEM: J80  IS USING CTNID: PETSTP
        SYSTEM: JA0  IS USING CTNID: PETSTP  -01
        SYSTEM: JB0  IS USING CTNID: PETSTP
        SYSTEM: J90  IS USING CTNID: PETSTP  -01
        SYSTEM: JC0  IS USING CTNID: PETSTP
        SYSTEM: JE0  IS USING CTNID: PETSTP
```

At the same time, z/OS images in the Parallel Sysplex might also recognize that CFs also have mismatched CTN IDs, which might indicate that those images with mismatched CTN IDs might be synchronized to a different time reference. This condition is reported via message IXL162E with reason CTNID MISMATCH:

```
*IXL162E CF REQUEST TIME ORDERING: REQUIRED AND WILL NOT BE ENABLED
        COUPLING FACILITY 002094.IBM.02.0000000C299E
        PARTITION: 23  CPCID: 00
        REASON: CTNID MISMATCH. CF CTNID: PETSTP
```

This particular message is reported by a z/OS image that has not yet made the transition to the PETSTP CTN, whereas this CF has already done so.

Finally, when all z/OS images have completed the transition to the STP-only CTN, message IXC435I will report that all images have matching CTN IDs and are now synchronized to the same time reference:

```
IXC435I ALL SYSTEMS IN SYSPLEX UTCPLXJ8 ARE NOW SYNCHRONIZED
      TO THE SAME TIME REFERENCE.
      SYSTEM: JF0 IS USING CTNID: PETSTP
      SYSTEM: J80 IS USING CTNID: PETSTP
      SYSTEM: JA0 IS USING CTNID: PETSTP
      SYSTEM: JB0 IS USING CTNID: PETSTP
      SYSTEM: J90 IS USING CTNID: PETSTP
      SYSTEM: JC0 IS USING CTNID: PETSTP
      SYSTEM: JE0 IS USING CTNID: PETSTP
```

When any z/OS images with mismatched CTN IDs have completed the transition to the STP-only CTN, message IXL161I will report that CF request time ordering is once again enabled, as the CTN ID mismatches with any CFs will also have been resolved:

```
IXL161I CF REQUEST TIME ORDERING: REQUIRED AND ENABLED
      COUPLING FACILITY 002094.IBM.02.0000000C299E
      PARTITION: 23 CPCID: 00
```

At this point, our servers were configured in an STP-only CTN and we verified this by using the System (Sysplex) Timer task as well as by issuing z/OS display commands.

First, we used the **Timing Network** tab on the System (Sysplex) Timer task for T75 at the HMC to verify the timing network type, the CTN ID, and the CTN time source, as shown in Figure 41.

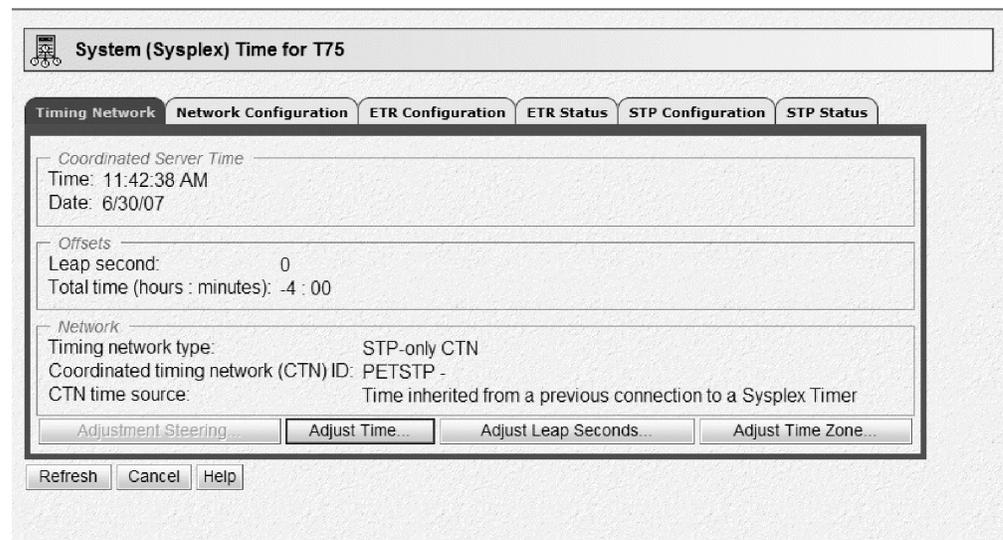


Figure 41. System (Sysplex) Time: Timing Network panel on T75 – STP-only CTN

Note that the timing network type is STP-only CTN and the CTN ID is PETSTP. Also note that the CTN time source says Time inherited from a previous connection to a Sysplex Timer. This means that this STP-only CTN was concurrently migrated from an ETR timing network via a mixed CTN.

We also used the **ETR Configuration** tab for T75 to verify that the ETR ports were disabled as reported by message IEA393I. This is shown in Figure 42 on page 84.

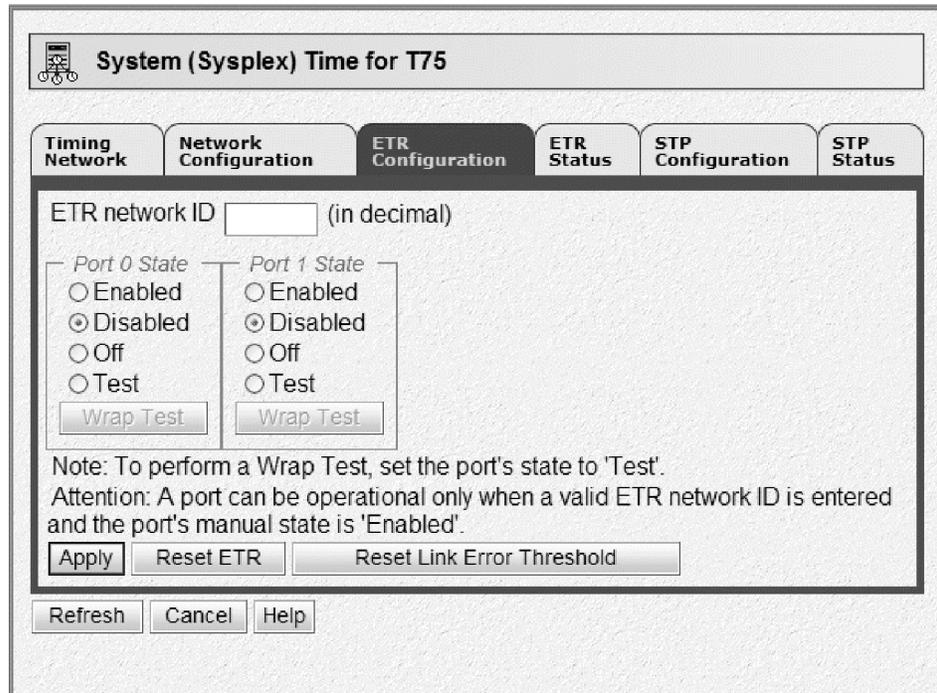


Figure 42. System (Sysplex) Time: ETR Configuration panel – STP-only CTN

Finally, the **STP Status** tab for T75 shows that it is the stratum 1 server, synchronized in STP timing mode, and all other servers to which it is directly connected are at stratum 2. This implies that T75 is the current time server in an STP-only CTN. Figure 43 on page 85 illustrates these points.

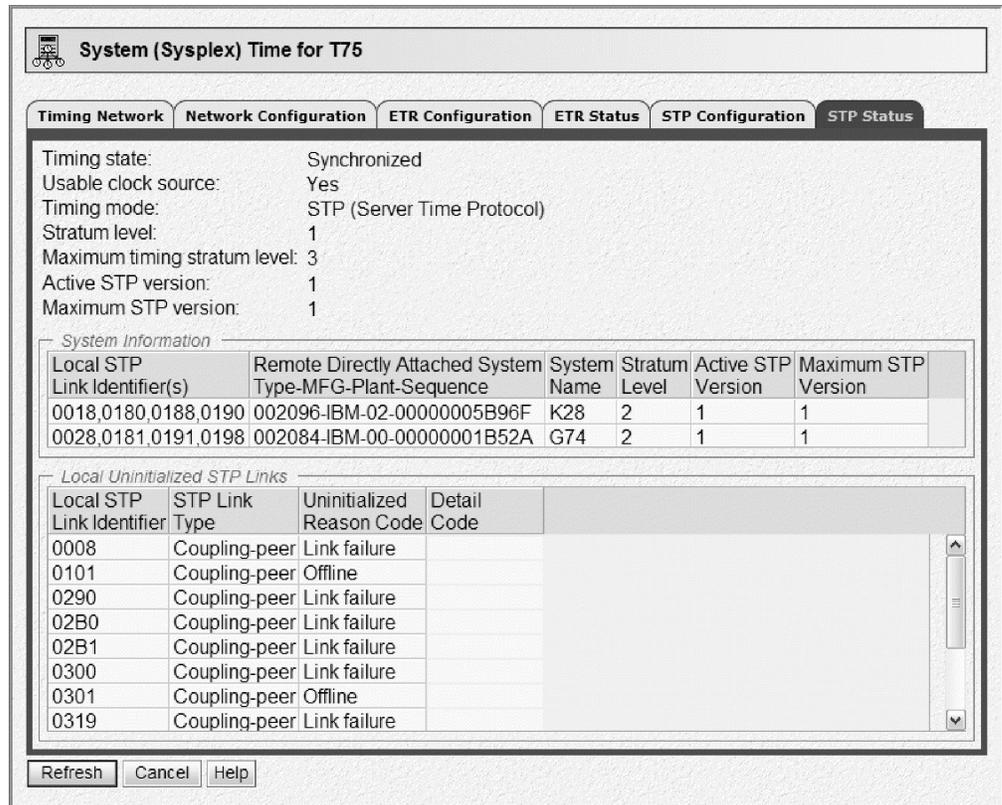


Figure 43. System (Sysplex) Time: STP Status panel – STP-only CTN

We also used the z/OS commands DISPLAY XCF,SYSPLEX,ALL and DISPLAY ETR to verify the STP-only CTN configuration.

When we issued the DISPLAY XCF,SYSPLEX,ALL command on any image in the sysplex, the following response was displayed:

```
IXC335I 11.48.43 DISPLAY XCF
SYSTEM  TYPE SERIAL LPAR STATUS TIME          SYSTEM STATUS
J80     2094 299E 07 06/30/2007 11:48:43 ACTIVE          TM=STP
JC0     2084 B52A 0C 06/30/2007 11:48:38 ACTIVE          TM=STP
JA0     2084 B52A 2A 06/30/2007 11:48:40 ACTIVE          TM=STP
JB0     2084 B52A 01 06/30/2007 11:48:38 ACTIVE          TM=STP
J90     2094 299E 05 06/30/2007 11:48:38 ACTIVE          TM=STP
JF0     2094 299E 06 06/30/2007 11:48:40 ACTIVE          TM=STP
JE0     2084 B52A 22 06/30/2007 11:48:40 ACTIVE          TM=STP
```

This shows that all images in the sysplex were in STP timing mode (TM=STP).

When we issued the DISPLAY ETR command on a z/OS image running on the current time server (T75), the following response was displayed:

```
IEA386I 11.48.49 TIMING STATUS
SYNCHRONIZATION MODE = STP
THIS SERVER IS A STRATUM 1
CTN ID = PETSTP
THE STRATUM 1 NODE ID = 002094.S54.IBM.02.0000000C299E
THIS IS THE PREFERRED TIME SERVER
```

This shows that the image is running in STP timing mode on the stratum 1 server (the current time server) that is identified by node ID 002094.S54.IBM.02.0000000C299E, which is also the preferred time server.

Because K28 has only one coupling facility LPAR, we could not use the DISPLAY ETR command to verify the role of K28 as the backup time server. However, we issued a DISPLAY ETR command to a z/OS image running on the arbiter server (G74):

```
IEA386I 11.51.38 TIMING STATUS
SYNCHRONIZATION MODE = STP
THIS SERVER IS A STRATUM 2
CTN ID = PETSTP
THE STRATUM 1 NODE ID = 002094.S54.IBM.02.0000000C299E
THIS IS THE ARBITER SERVER
NUMBER OF USABLE TIMING LINKS = 8
```

This response shows that the image is running in STP timing mode on a stratum 2 server that is the Arbiter and that this server has eight usable links over which it can receive STP timing signals from the stratum 1 server identified by node ID 002094.S54.IBM.02.0000000C299E.

Figure 44 illustrates the STP-only CTN that we have configured up to this point. Note the color-coded stratum levels that help illustrate the timing hierarchy in our CTN.

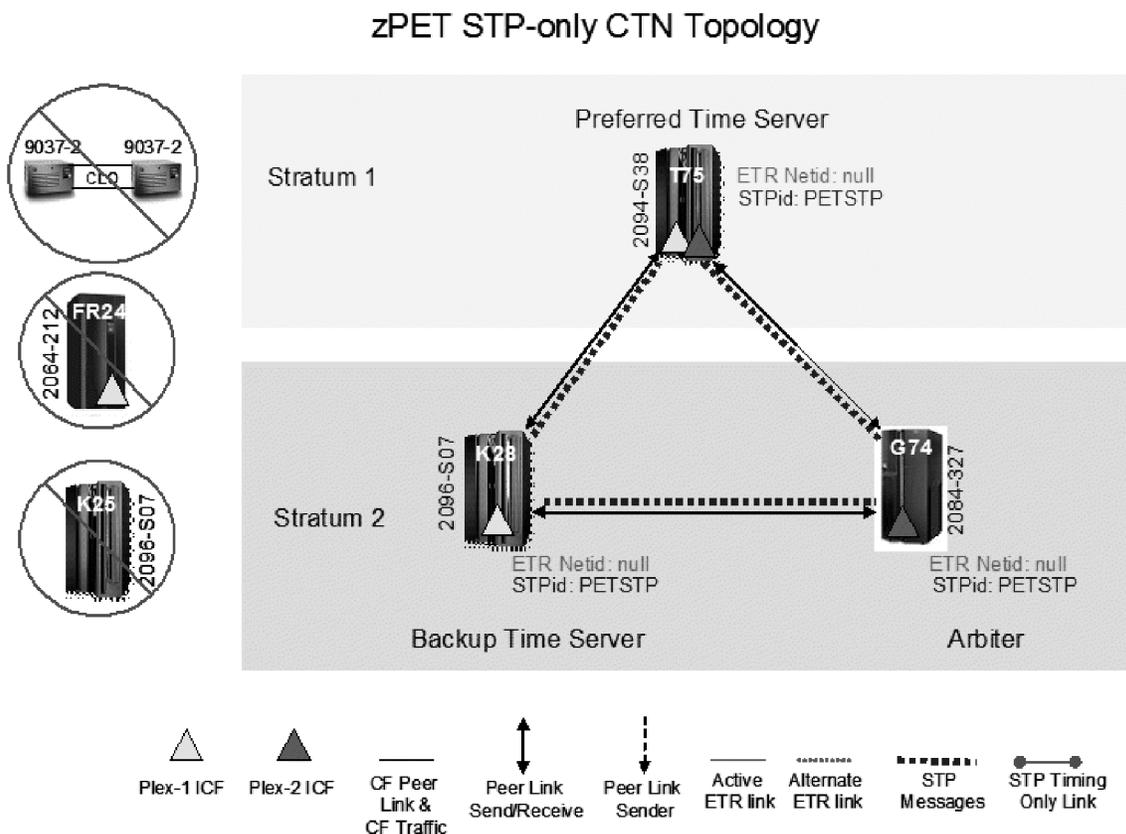


Figure 44. zPET STP-only CTN

Changing server roles in an STP-only CTN

STP provides the ability to dynamically change the roles of servers in a CTN. This allows more flexibility in managing the STP-only CTN, for example, by allowing an operator to move the current time server role to the backup time server, so that

the preferred time server can be serviced without disrupting the time source for the CTN. The screen captures and z/OS display commands in this topic demonstrate just such an action.

Note that other roles can be changed, such as changing the arbiter or assigning a role to a previously unassigned server in the CTN. The only requirement, as discussed in “Migrating from a mixed CTN to an STP-only CTN” on page 78, is that the assignments must be done from the intended current time server (the server that will be the stratum 1 server in the CTN after the change completes).

From the **Network Configuration** tab of the System (Sysplex) Time task for the backup time server (K28), we selected the **Backup Time Server (CPC)** radio button in the Current time server (CPC) section to assign the role of current time server to backup time server K28, as shown in Figure 45.

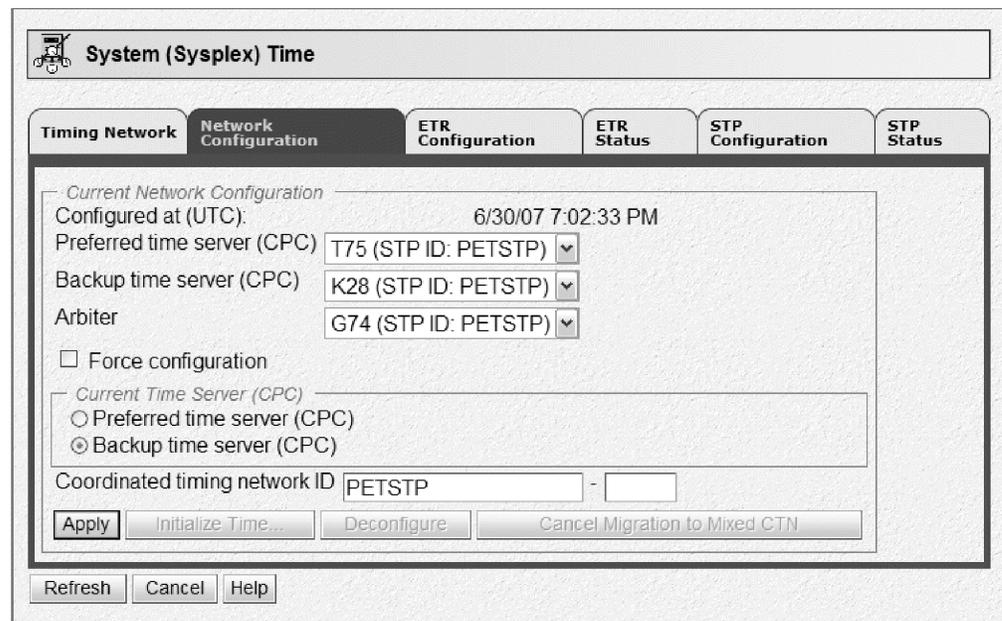


Figure 45. System (Sysplex) Time: Network Configuration panel – assigning K28 as current time server

After clicking the Apply button, we were prompted to confirm the change, as shown in Figure 46.

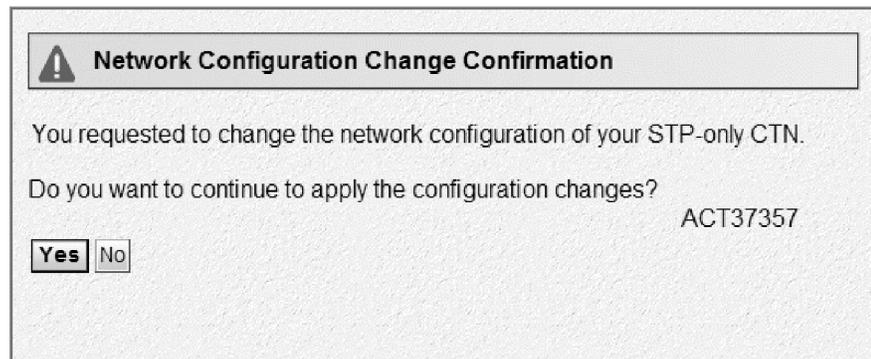


Figure 46. Network Configuration Change Confirmation panel – apply CTN role change

Figure 47 indicates successful completion of the CTN configuration change.

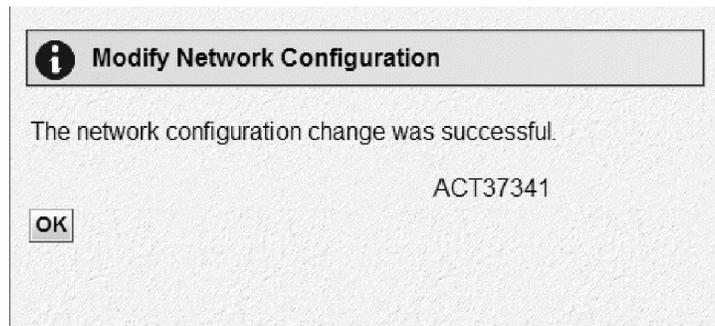


Figure 47. Modify Network Configuration panel – successful CTN role change

We then used the STP Status tab for the new current time server (K28) to verify the new CTN role assignments, as shown in Figure 48.

Figure 48 confirms that K28 is now the stratum 1 server and the System

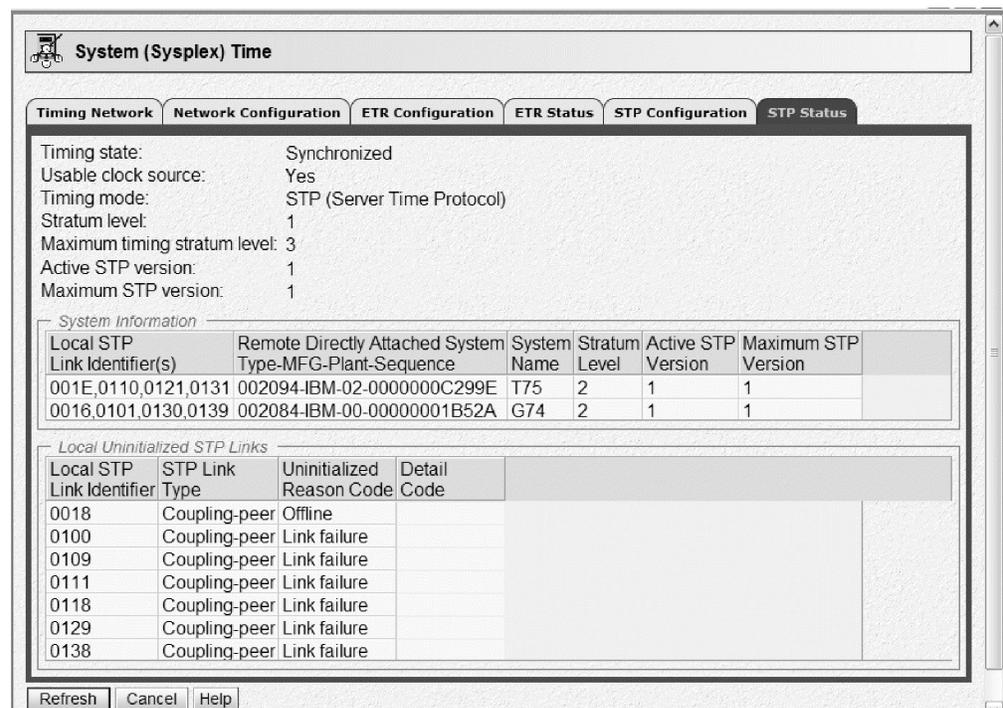


Figure 48. System (Sysplex) Time: STP Status panel with K28 as current time server

Information section shows that T75 is now a stratum 2 server and, therefore, no longer the current time server.

Finally, issuing a the DISPLAY ETR command on a z/OS image running on the preferred time server (T75) yields the following response:

```
IEA386I 18.20.13 TIMING STATUS
SYNCHRONIZATION MODE = STP
THIS SERVER IS A STRATUM 2
CTN ID = PETSTP
THE STRATUM 1 NODE ID = 002096.S07.IBM.02.00000005B96F
THIS IS THE PREFERRED TIME SERVER
NUMBER OF USABLE TIMING LINKS = 8
```

The results of this command show that this image is running in STP timing mode on a stratum 2 server that is the preferred time server and that this server has eight usable links over which it can receive STP timing signals from the stratum 1 server identified by node ID 002096.S07.IBM.02.00000005B96F, which we know to be the node ID of K28.

A DISPLAY ETR command issued to a z/OS image running on the arbiter server (G74) also confirms K28's stratum 1 role:

```
IEA386I 18.21.38 TIMING STATUS
SYNCHRONIZATION MODE = STP
THIS SERVER IS A STRATUM 2
CTN ID = PETSTP
THE STRATUM 1 NODE ID = 002096.S07.IBM.02.00000005B96F
THIS IS THE ARBITER SERVER
NUMBER OF USABLE TIMING LINKS = 8
```

Figure 49 illustrates the STP-only CTN after we have configured the backup time server (K28) as the current time server (that is, the stratum 1 server).

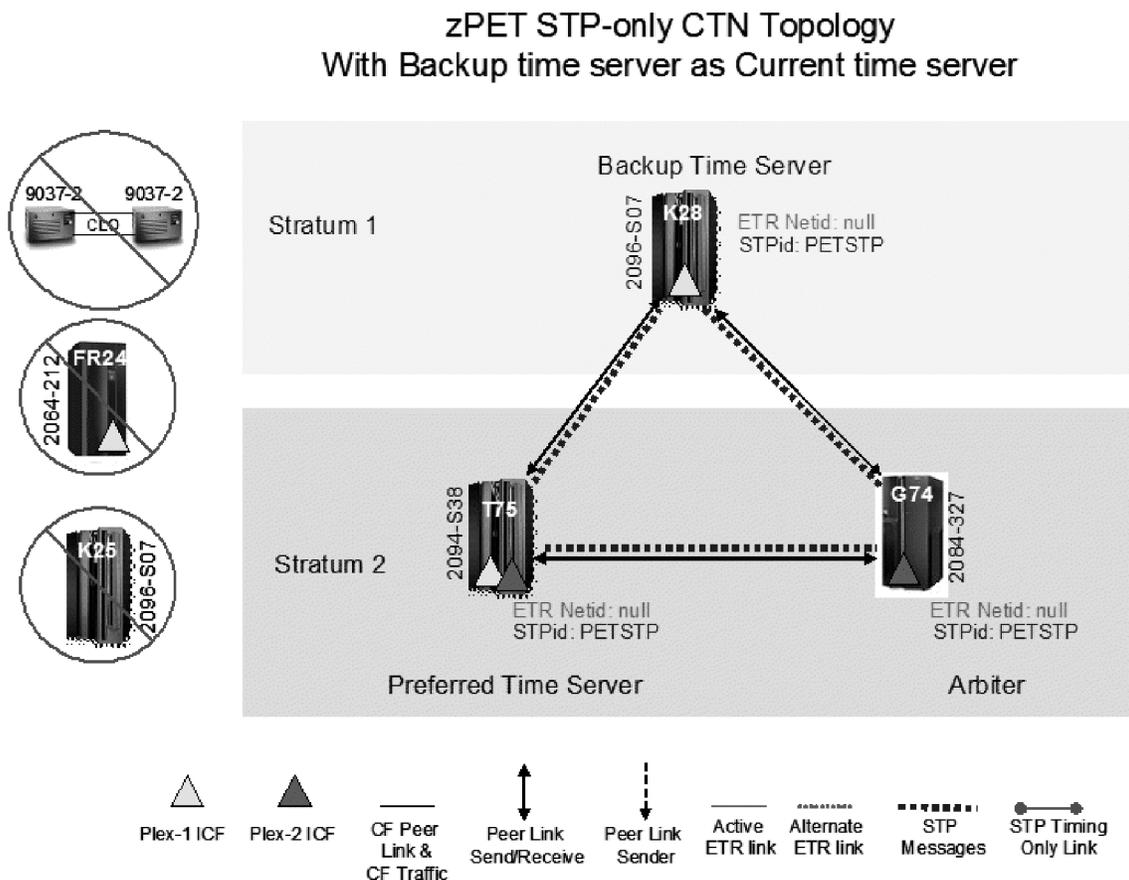


Figure 49. zPET STP-only CTN with backup time server as current time server

Reverse migration: STP-only CTN to mixed CTN

Our final STP migration task was to verify the ability to concurrently roll back the migration steps we have taken thus far. In this topic, we show the steps that we took to reverse our timing network topology from that shown in Figure 49 to the topology depicted in Figure 37 on page 79.

To perform this step, we used the **Network Configuration** tab on the System (Sysplex) Timer task for the current time server to reinstate the ETR ID portion of the CTN ID. Note that it is the operator's responsibility to ensure that the ETR ID that is entered in the CTN ID in this step is the correct ID for the ETRs to which the mixed CTN should be synchronized.

Once the reverse transition is complete, the preferred and backup time servers will be reconnected to the ETRs (that is, stratum 1, in ETR timing mode, and synchronized to the Sysplex Timer ETRs) by the STP facility, while all other servers in the CTN will remain in STP timing mode. Therefore, before starting this migration step, the **ETR Status** tabs for both the preferred and backup time servers should be used to verify that the ETR card status shows Light detected for each port and that the ETR status word state shows Semi-operational for each port. This will ensure that the ETR ports are in a state that can be used to receive Sysplex Timer signals when the servers return to Sysplex Timer synchronization.

Figure 50 shows the starting point for this reverse migration. In "Changing server roles in an STP-only CTN" on page 86, we made the backup time server (K28) the current time server, so we must start with the **Network Configuration** tab for K28.

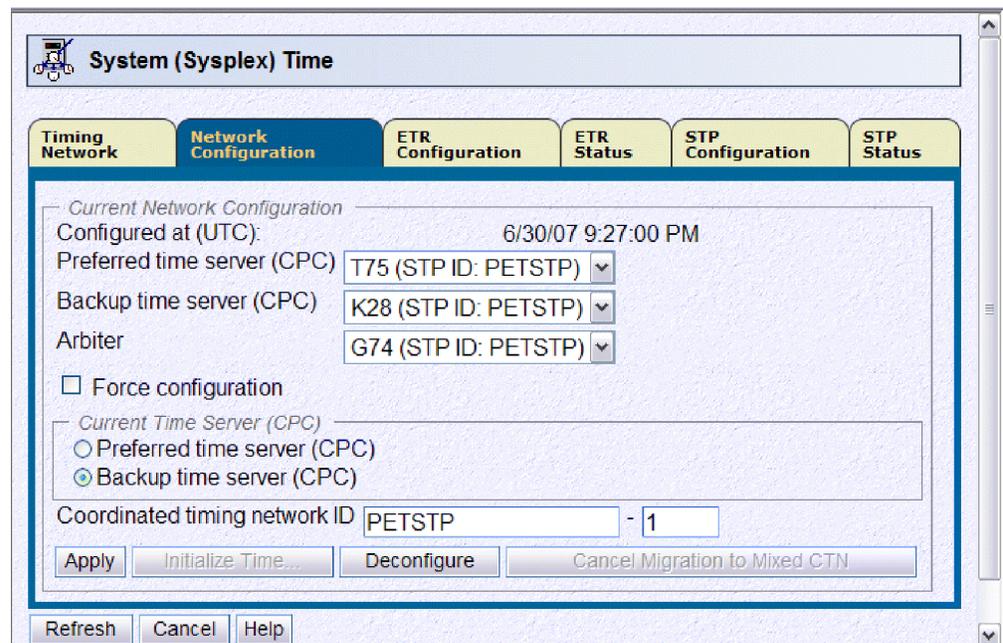


Figure 50. System (Sysplex) Time: Network Configuration panel – K28 starting reverse migration to mixed CTN

We verified the **ETR Status** tabs for both the preferred and backup time servers and entered the ETR network ID of our existing Sysplex Timer network (see Figure 4 on page 47). When we clicked the **Apply** button, we were presented with the cautionary confirmation message shown in Figure 51 on page 91.

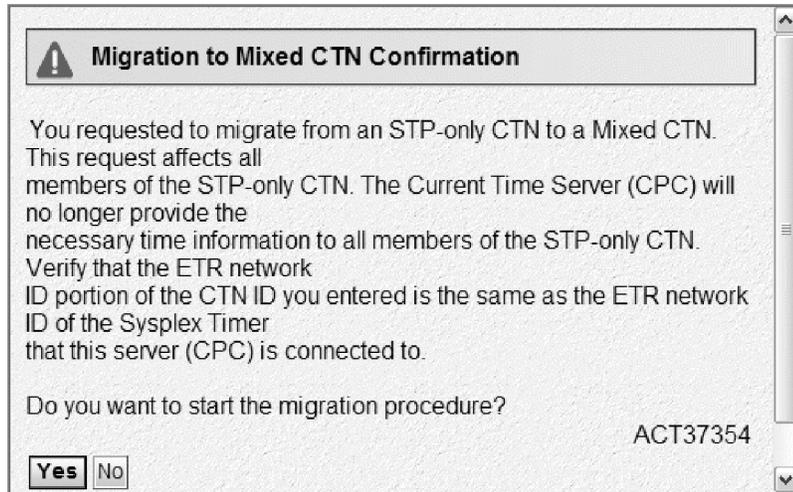


Figure 51. Migration to Mixed CTN Confirmation: Confirm migration from STP-only to mixed CTN (step 1)

Because we were certain the ETR network ID that we entered was correct, we proceeded by clicking **Yes** in response to the prompt in Figure 51. We were then presented with an additional warning before final confirmation, as shown in Figure 52. In this case, we were being told by the STP facility how long it would take for the coordinated server time of the STP-only CTN to be re-synchronized with the Sysplex Timer ETRs once the STP facility starts the CTN configuration change.

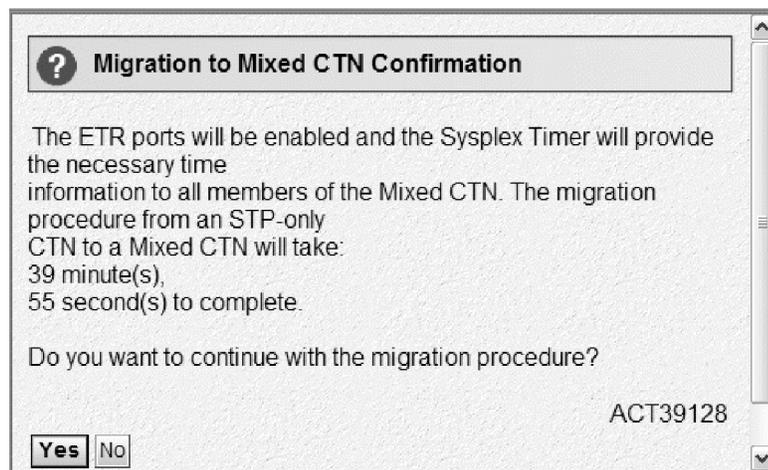


Figure 52. Migration to Mixed CTN Confirmation: Confirm migration from STP-only to mixed CTN (step 2)

After we clicked **Yes** to confirm that we wanted to proceed with the STP-only to mixed CTN configuration change, we received the final confirmation that the CTN configuration change had begun, as shown in Figure 53 on page 92.

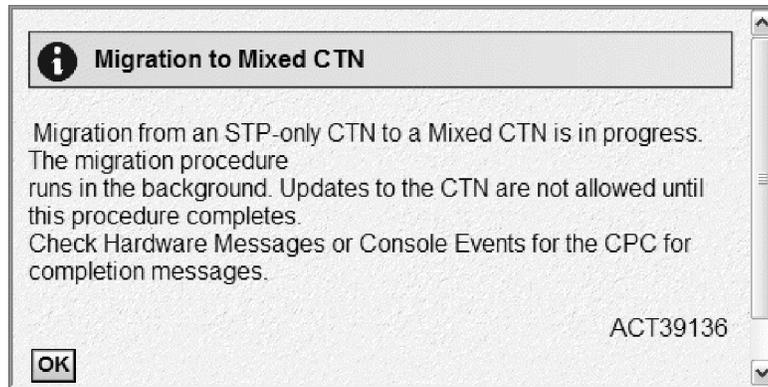


Figure 53. Migration to Mixed CTN: STP-only to mixed CTN migration in progress

Figure 53 explains that the migration has begun and continues to run in the background, and that CTN changes are not allowed until the migration completes. Figure 54 reinforces this change restriction during the migration, where we see that the **Apply** button on the **Network Configuration** tab is grayed out and a message explaining that an STP-only to mixed CTN migration is in progress. If a change is required, the **Network Configuration** tab provides the option to cancel the migration before it completes via the **Cancel Migration to Mixed CTN** button.

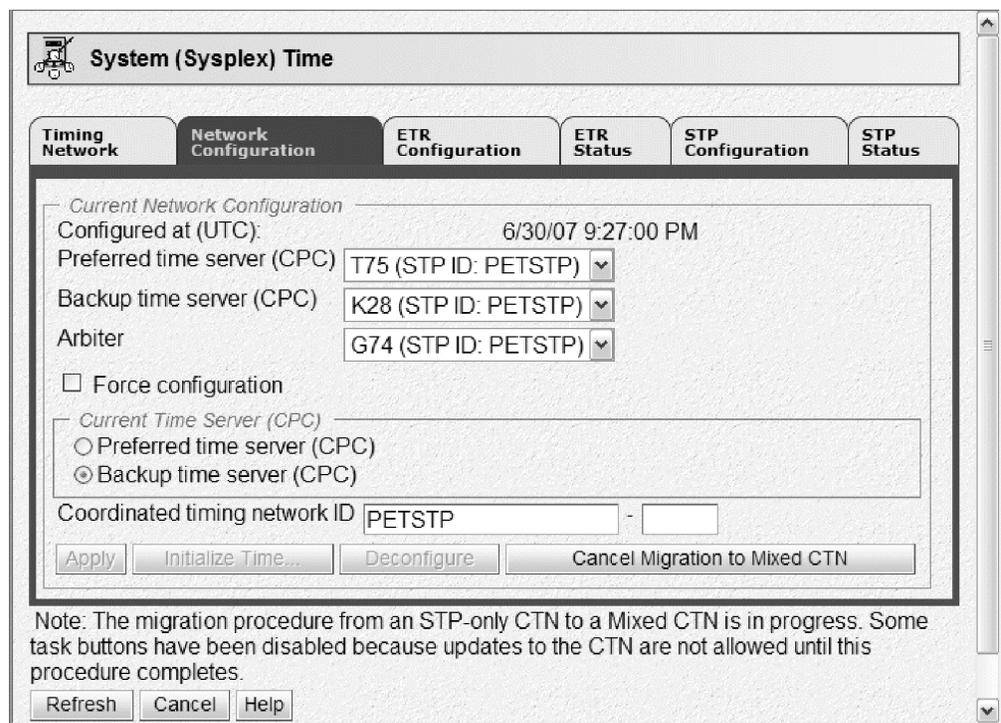


Figure 54. System (Sysplex) Time: Network Configuration panel – Migration to mixed CTN in progress

Figure 53 also explains that the HMC Hardware Messages task or the View Console Events task can be used to verify the completion on the migration. In our case, we expected z/OS messages that also confirmed and verified the completion of the migration.

Message IEA390I is issued to indicate that z/OS images have received an STP synchronization check and the TOD clock has been adjusted to keep it in synchronization with the rest of the timing network. STP synchronization checks indicate that the STP facility has processed an STP timing state change:

```
IEA390I TOD CLOCKS DYNAMICALLY ADJUSTED TO MAINTAIN STP SYNCHRONISM.
```

Each z/OS image then recognizes the CTN ID change from PETSTP to PETSTP-01 and reports message IXC438I:

```
IXC438I COORDINATED TIMING INFORMATION HAS BEEN UPDATED
FOR SYSTEM: JA0
PREVIOUS CTNID:  PETSTP
CURRENT  CTNID:  PETSTP  -01
```

As discussed in “Migrating from a mixed CTN to an STP-only CTN” on page 78, CTN ID changes might not occur simultaneously across a Parallel Sysplex and several messages were seen that indicated the temporary CTN ID mismatch during the transition window:

```
*IXC439E ALL SYSTEMS IN SYSPLEX UTCPLXJ8 ARE NOT SYNCHRONIZED
TO THE SAME TIME REFERENCE.
SYSTEM: JF0  IS USING CTNID: PETSTP
SYSTEM: JB0  IS USING CTNID: PETSTP
SYSTEM: JA0  IS USING CTNID: PETSTP  -01
SYSTEM: JB0  IS USING CTNID: PETSTP
SYSTEM: J90  IS USING CTNID: PETSTP
SYSTEM: JC0  IS USING CTNID: PETSTP
SYSTEM: JE0  IS USING CTNID: PETSTP
```

For z/OS images on our preferred time server where the Sysplex Timer connectivity was re-established by the STP facility, we saw the following set of messages:

```
IEA267I ETR PORT 0 IS NOW AVAILABLE.
IEA267I ETR PORT 1 IS NOW AVAILABLE.
IEA260I THE CPC IS NOW OPERATING IN ETR MODE.
IEA273I TOD CLOCKS DYNAMICALLY ADVANCED TO MAINTAIN ETR SYNCHRONISM.
```

Finally, when all images had completed the transition to the mixed CTN, message IXC435I reported that all images had matching CTN IDs:

```
IXC435I ALL SYSTEMS IN SYSPLEX UTCPLXJ8 ARE NOW SYNCHRONIZED 437
TO THE SAME TIME REFERENCE.
SYSTEM: JF0  IS USING CTNID: PETSTP  -01
SYSTEM: JB0  IS USING CTNID: PETSTP  -01
SYSTEM: JA0  IS USING CTNID: PETSTP  -01
SYSTEM: JB0  IS USING CTNID: PETSTP  -01
SYSTEM: J90  IS USING CTNID: PETSTP  -01
SYSTEM: JC0  IS USING CTNID: PETSTP  -01
SYSTEM: JE0  IS USING CTNID: PETSTP  -01
```

At this point, the STP-only to mixed CTN configuration change was complete and we used the System (Sysplex) Timer task as well as z/OS display commands to verify our new CTN configuration.

First, we used the **Timing Network** tab for server K28 (previously the current time server) to confirm that we were back in a mixed CTN configuration, as shown in Figure 55 on page 94.

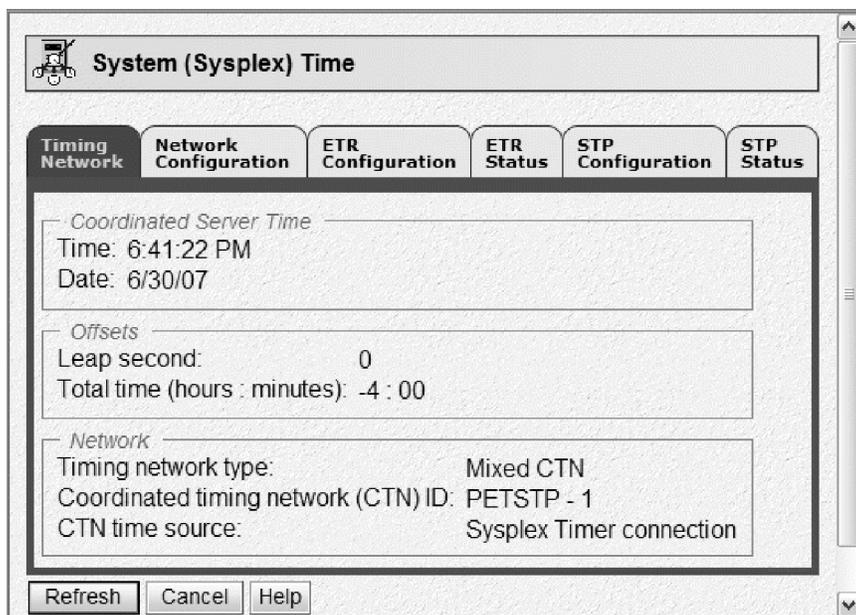


Figure 55. System (Sysplex) Time: Timing Network panel – K28 back in mixed CTN

Then, we used the **Network Configuration** tab for K28 to verify that the STP-only CTN roles were no longer assigned and the CTN ID represented a mixed CTN format, as shown in Figure 56.

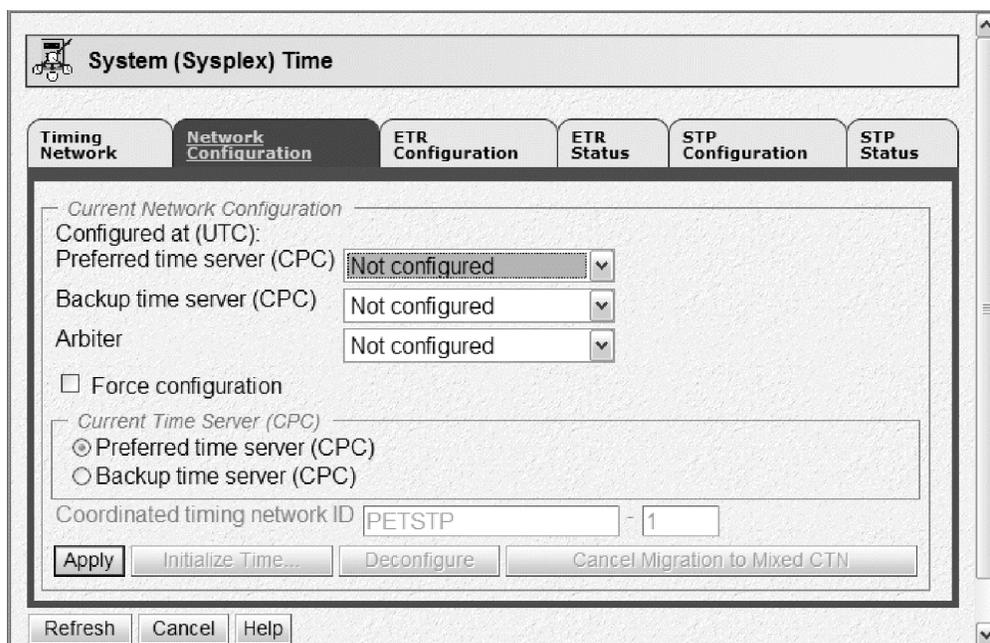


Figure 56. System (Sysplex) Time: Network Configuration panel –K28 back in mixed CTN

Next, the **ETR Configuration** tab for K28 confirmed that its ETR ports were enabled for ETR network ID 01, as shown in Figure 57 on page 95.

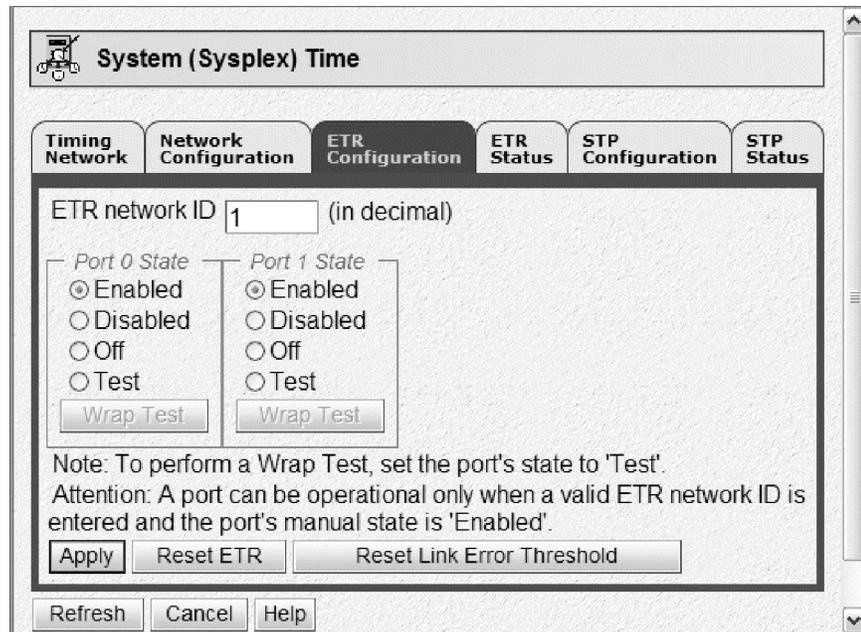


Figure 57. System (Sysplex) Time: ETR Configuration panel – K28 back in mixed CTN

Finally, we used the **STP Status** tab to verify the CTN configuration from K28's point of view, as shown in Figure 58. Figure 58 showed that K28 was at stratum 1 and synchronized in ETR timing mode, as was T75, while G74 remained at Stratum 2.

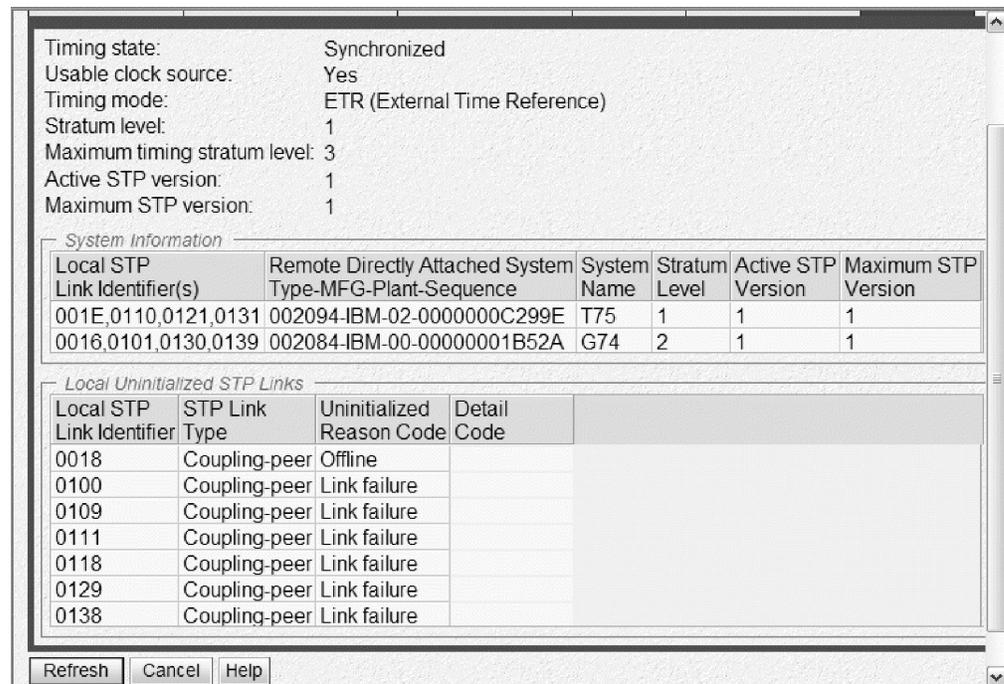


Figure 58. System (Sysplex) Time: STP Status panel – K28 back in mixed CTN

We also issued z/OS DISPLAY ETR commands to verify the mixed CTN configuration.

First, when issued to a z/OS image on the previous backup time server (T75), we saw that the server was now synchronized to the ETRs in the Mixed CTN PETSTP-01:

```
RO J80,D ETR
IEA282I 18.37.06 TIMING STATUS
SYNCHRONIZATION MODE = ETR
  CPC PORT 0 <== ACTIVE      CPC PORT 1
  OPERATIONAL                OPERATIONAL
  ENABLED                    ENABLED
  ETR NET ID=01              ETR NET ID=01
  ETR PORT=09                ETR PORT=09
  ETR ID=01                  ETR ID=00
  THIS SERVER IS PART OF TIMING NETWORK PETSTP -01
```

When issued to a z/OS image on the previous arbiter server (G74), we saw that the server was now a stratum 2 server in mixed CTN PETSTP-01 and had eight usable timing links over which it can receive STP synchronization:

```
IEA386I 18.37.03 TIMING STATUS
SYNCHRONIZATION MODE = STP
  THIS SERVER IS A STRATUM 2
  CTN ID = PETSTP -01
  NUMBER OF USABLE TIMING LINKS = 8
```

Finally, we issued a DISPLAY XCF,SYSPLEX,ALL command on a z/OS image in our Parallel Sysplex to confirm that the timing modes of the z/OS images in the Parallel Sysplex were as expected after the STP-only to mixed CTN reversal:

```
IXC335I 18.38.49 DISPLAY XCF
SYSTEM TYPE SERIAL LPAR STATUS TIME SYSTEM STATUS
J80 2094 299E 07 06/30/2007 22:28:42 ACTIVE TM=ETR
JC0 2084 B52A 0C 06/30/2007 22:28:40 ACTIVE TM=STP
JA0 2084 B52A 2A 06/30/2007 22:28:41 ACTIVE TM=STP
JB0 2084 B52A 01 06/30/2007 22:28:38 ACTIVE TM=STP
J90 2094 299E 05 06/30/2007 22:28:39 ACTIVE TM=ETR
JF0 2094 299E 06 06/30/2007 22:28:42 ACTIVE TM=ETR
JE0 2084 B52A 22 06/30/2007 22:28:41 ACTIVE TM=ETR
```

At the time of this writing, we had just begun the installation of our System z10 EC server, as described in Chapter 2, “Using the IBM System z10 Enterprise Class platform,” on page 7, and decided to leave our timing network topology in the mixed CTN configuration shown in Figure 37 on page 79. With this configuration, we have the ability to fall back to ETR-only timing or move to an STP-only CTN configuration. For now, this proves to be a suitable configuration in which to remain.

Chapter 8. Using the IBM zIIP

IBM has extended its mainframe data serving capabilities, delivered a new roadmap for the future of data serving and information on demand, previewed new DB2 function, and introduced a new specialty engine directed toward data serving workloads.

A new specialty engine, the IBM System z9 Integrated Information Processor and System z10 Integrated Information Processor (zIIP), is now available on the IBM System z10 Enterprise Class (z10 EC), System z9 Enterprise Class (z9 EC), and System z9 Business Class (z9 BC) platforms.

A zIIP is similar in concept to the System z Application Assist Processor (zAAP). Like zAAPs (but unlike CPs, ICFs and IFLs), zIIPs can do nothing on their own; they can not perform an IPL and can not run an operating system. zIIPs 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 selective workloads such as:

- ERP or CRM application serving — For applications, running on z/OS, UNIX®, Intel®, or Linux on System z that access DB2 for z/OS V8 on a System z9 or System z10 mainframe, through DRDA® over a TCP/IP connection, DB2 gives z/OS the necessary information to have portions of these SQL requests directed to the zIIP.
- Data Warehousing applications — Requests that utilize DB2 for z/OS V8 for long running parallel queries, including complex star schema parallel queries, may have portions of these SQL requests directed to the zIIP when DB2 gives z/OS the necessary information. These queries are typical in data warehousing implementations. The addition of select long running parallel queries may provide more opportunity for DB2 customers to optimize their environment for Data Warehousing while leveraging the unique qualities of service provided by System z9, System z10, and DB2.
- Some DB2 for z/OS V8 utilities — A portion of DB2 utility functions used to maintain index maintenance structures (LOAD, REORG, and REBUILD INDEX) that typically run during batch, can be redirected to zIIPs.

This topic describes what we did to configure and to prepare to exercise and test the zIIP feature on our z9 systems.

Prerequisites for IBM zIIP

The following are prerequisites for zIIP usage:

- z/OS V1R6 with JBB77S9 applied
- z/OS V1R7 with JBB772S applied
- z/OS V1R8 or higher
- DB2 V8 with the appropriate maintenance.

More detailed information about all the software and hardware prerequisites can be found in the following PSP buckets:

- 2094, 2096, and 2097 hardware device buckets
- z/OS BCP zIIP bucket

- zIIP functional PSP bucket

Also, contact your local hardware and software representatives for any additional requirements.

Configuring the IBM zIIP

We configured two zIIPs on all of the z/OS images on our System z9 EC CPC and we configured two zIIPs on our System z10 EC CPC. When you configure your z/OS logical partitions you simply specify how many logical zIIPs you want to define for each partition, just as you do for the number of standard CPs and zAAPs. When you IPL the system, z/OS determines how many zIIPs are configured and manages an additional dispatcher queue for zIIP-eligible work.

We did the following to configure our zIIPs:

1. Updated the image profiles for all our z9 EC and z10 EC partitions to define two zIIPs to each partition.

Figure 59 shows an example of the image profile for our J80 z/OS image with 2 zIIPs defined:

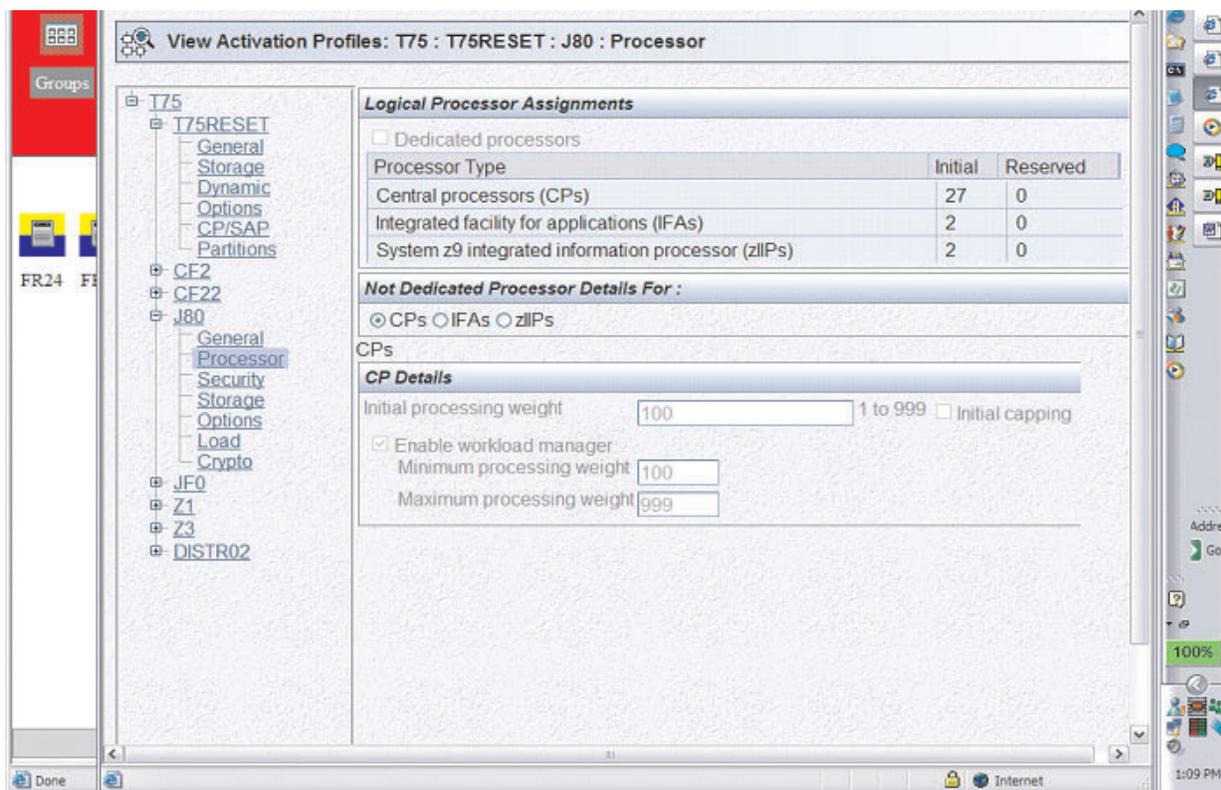


Figure 59. Image profile for our J80 z/OS image with 2 zIIPs defined

2. Deactivated, activated and IPLed the z/OS partitions to bring the zIIPs online. You can use the D M=CPU command to display the status of the zIIPs. The zIIPs appear as an integrated information processor in response to the D M=CPU command.

Response example for the D M=CPU command on system JB0:

```

-JB0D M=CPU
IEE174I 13.14.39 DISPLAY M 372
PROCESSOR STATUS
ID CPU SERIAL
00 + 0B99FF2097
01 + 0B99FF2097
02 + 0B99FF2097
03 + 0B99FF2097
04 + 0B99FF2097
05 + 0B99FF2097
06 + 0B99FF2097
07 + 0B99FF2097
08 + 0B99FF2097

11 + 0B99FF2097
12 + 0B99FF2097
13 +A 0B99FF2097
14 +A 0B99FF2097
15 +I 0B99FF2097
16 +I 0B99FF2097
CPC ND = 002097.E56.IBM.02.0000000699FF
CPC SI = 2097.742.IBM.02.00000000000699FF
CPC ID = 00
CPC NAME = H91
LP NAME = JB0 LP ID = B
CSS ID = 0
MIF ID = B

+ ONLINE - OFFLINE . DOES NOT EXIST W WLM-MANAGED
N NOT AVAILABLE

A APPLICATION ASSIST PROCESSOR (zAAP)
I INTEGRATED INFORMATION PROCESSOR (zIIP)
CPC ND CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
CPC SI SYSTEM INFORMATION FROM STSI INSTRUCTION
CPC ID CENTRAL PROCESSING COMPLEX IDENTIFIER
CPC NAME CENTRAL PROCESSING COMPLEX NAME
LP NAME LOGICAL PARTITION NAME
LP ID LOGICAL PARTITION IDENTIFIER
CSS ID CHANNEL SUBSYSTEM IDENTIFIER
MIF ID MULTIPLE IMAGE FACILITY IMAGE IDENTIFIER

```

Response example for the D M=CPU command on system J80:

```

-D M=CPU
IEE174I 07.47.11 DISPLAY M 895
PROCESSOR STATUS
ID CPU SERIAL
00 + 07299E2094
01 + 07299E2094
02 + 07299E2094
03 + 07299E2094
04 + 07299E2094
05 + 07299E2094
06 + 07299E2094
07 + 07299E2094
08 + 07299E2094
09 + 07299E2094
0A + 07299E2094
0B + 07299E2094
0C + 07299E2094
0D + 07299E2094
0E + 07299E2094
0F + 07299E2094
10 + 07299E2094
11 + 07299E2094
12 + 07299E2094
13 + 07299E2094

```

```

14 +          07299E2094
15 +          07299E2094
16 +          07299E2094
17 +          07299E2094
18 +          07299E2094
19 +          07299E2094
1A +          07299E2094
1B +A        07299E2094
1C +A        07299E2094
1D +I        07299E2094
1E +I        07299E2094

```

```

CPC ND = 002094.S38.IBM.02.0000000C299E
CPC SI = 2094.729.IBM.02.0000000000C299E
CPC ID = 00
CPC NAME = T75
LP NAME = J80          LP ID = 7
CSS ID = 0
MIF ID = 7

```

```

+ ONLINE      - OFFLINE      . DOES NOT EXIST      W WLM-MANAGED
N NOT AVAILABLE

```

```

A      APPLICATION ASSIST PROCESSOR (zAAP)
I      INTEGRATED INFORMATION PROCESSOR (zIIP)
CPC ND CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
CPC SI SYSTEM INFORMATION FROM STSI INSTRUCTION
CPC ID CENTRAL PROCESSING COMPLEX IDENTIFIER
CPC NAME CENTRAL PROCESSING COMPLEX NAME
LP NAME LOGICAL PARTITION NAME
LP ID  LOGICAL PARTITION IDENTIFIER
CSS ID CHANNEL SUBSYSTEM IDENTIFIER
MIF ID MULTIPLE IMAGE FACILITY IMAGE IDENTIFIER

```

Monitoring zIIP utilization:

There is support in RMF to provide information about zIIP utilization. This information is useful to determine if and when you need to add zIIP capacity. For more details about RMF support for zIIPs and new fields on this report, please see *z/OS RMF Report Analysis, SC33-7991*.

Here is an example of our RMF Monitor III, CPC Report that displays the use of zIIP processors (in **bold**) on our System z10 EC images:

```

RMF V1R9   CPC Capacity
Command ==>
Line 1 of 35
Scroll ==> CSR

Samples: 120   System: JB0   Date: 03/19/08   Time: 06.43.00   Range: 120   Sec

Partition: JB0   2097 Model 742
CPC Capacity: 2740   Weight % of Max: 10.0   4h Avg: 553   Group: N/A
Image Capacity: 2740   WLM Capping %: 0.0   4h Max: 606   Limit: N/A

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

*CP
DISTR01    0  49 NO  6.0    12.4  12.5    0.0    1.8    1.8
DISTR02    0   0 NO  2.0     0.0   0.0    0.0    0.0    0.0
JB0        0 608 NO 19.0    48.8  49.1    0.1   22.1   22.2
JC0        0 190 NO 16.0    18.0  18.2    0.1    6.8    6.9
TICLTST    0   0 NO  2.0     0.0   0.0    0.0    0.0    0.0
TPN        0  41 NO 14.0     4.2   4.5    0.1    1.4    1.5
Z0         0 130 NO 12.0    16.2  16.6    0.1    4.6    4.7
Z2         0 108 NO 10.0    16.3  16.5    0.0    3.9    3.9
Z4         0  33 NO 10.0     5.0   5.1    0.0    1.2    1.2
PHYSICAL
*AAP
JB0        NO  2.0    92.8  92.9    0.1   37.1   37.2
JC0        NO  2.0    64.0  64.2    0.1   25.6   25.7
TPN        NO  2.0     0.0   0.0    0.0    0.0    0.0
Z0         NO  2.0     0.0   0.0    0.0    0.0    0.0
Z2         NO  2.0    42.0  42.2    0.0   16.8   16.9
Z4         NO  2.0     6.3   6.3    0.0    2.5    2.5
PHYSICAL
*ICF
CF21       1.0    100   100    0.0   14.3   14.3
CF4        3.0    100   100    0.0   42.9   42.9
CF5        3.0    100   100    0.0   42.9   42.9
PHYSICAL
*IIP
JB0        NO  2.0     0.0   0.1    0.0    0.0    0.1
JC0        NO  2.0     0.0   0.0    0.0    0.0    0.0
TPN        NO  2.0     0.0   0.0    0.0    0.0    0.0
Z0         NO  2.0     0.0   0.0    0.0    0.0    0.0
Z2         NO  2.0     0.0   0.0    0.0    0.0    0.0
Z4         NO  2.0     0.0   0.0    0.0    0.0    0.0

```

SMF type 70.1, 72.3, 79.1 and 79.2 records contain new fields with zIIP measurements. There are also new fields in SMF type 30 records to indicate the amount of time spent in zIIP work as well the amount of time spent executing zIIP eligible work on standard processors. *z/OS MVS System Management Facilities (SMF)*, SA22-7630 provides details on the new fields.

SDSF also provides information about system zIIP utilization as well as enclave zIIP utilization. New columns on the DA display and the Enclave display have been added to provide this information. For more details about these new fields for SDSF, see *z/OS SDSF Operation and Customization*, SA22-7670.

Here is one example for the SDSF enclave display that shows zIIP utilization on our z9 EC systems:

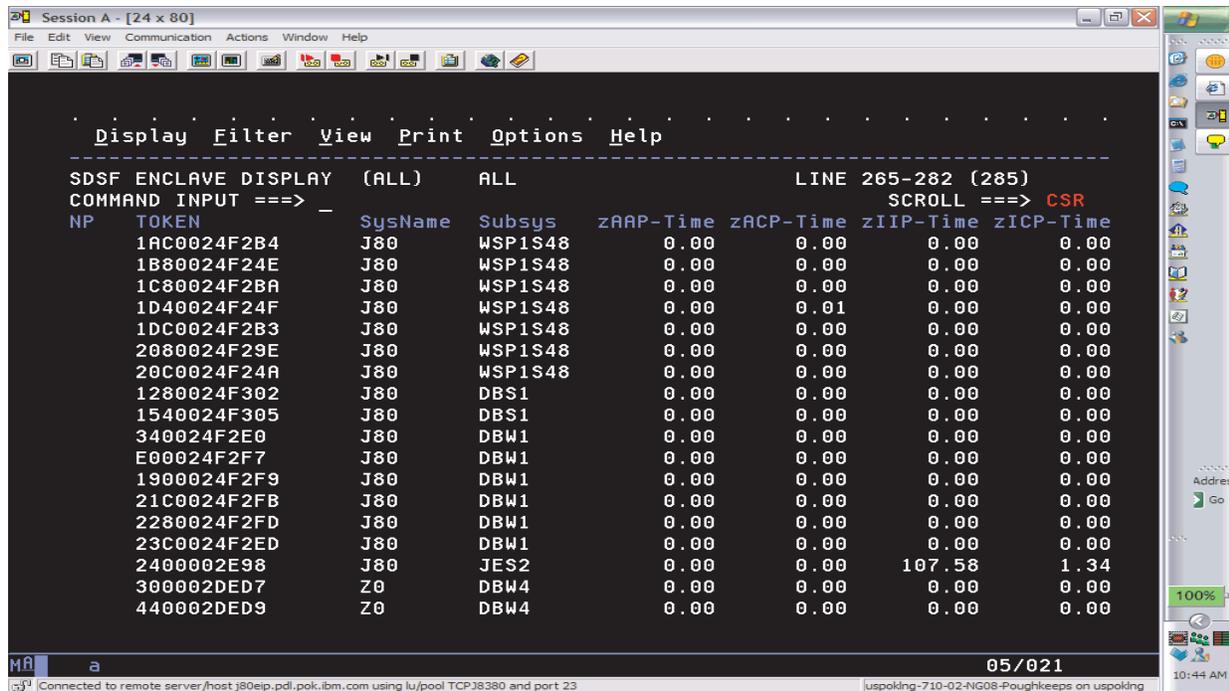


Figure 60. SDSF display showing zIIP utilization

DB2 workloads that exercise the IBM zIIP

The IBM System z9 Integrated Information Processor and System z10 Integrated Information Processor (zIIP) are designed so that specific types of DB2 programs or utilities can negotiate with z/OS to have a portion of their enclave service request block (SRB) work redirected from the general purpose central processor (CP) over to the zIIP, thereby freeing the CP for other tasks.

Those types of work which do not utilize SRBs, such as stored procedures and user-defined functions, are not eligible to offload work to the zIIP.

Currently, there are basically three situations or scenarios that may benefit from having a portion of their SQL requests redirected to the zIIP; they include the following:

1. Applications running on z/OS, UNIX, Intel, or Linux on System z that access DB2 via DRDA over a TCP/IP connection.

To test offloading portions of SQL requests using DRDA access over a TCP/IP connection to zIIP, we employed the use of the IBM Trade Performance Benchmark Sample for WebSphere Application Server V6.0 (or simply the Trade 6 workload), which may be obtained from www.software.ibm.com/webapp/iwm/web/preLogin.do?source=trade6.

Logon (or register if you are a new user), download `tradeInstall.zip` (1.7MB), and refer to the *Trade Technology* document (`tradeTech.pdf`) located in the install package for general information regarding Trade 6.

During our testing, we were able to drive substantial zIIP utilization using the Trade 6 workload and were able to monitor it via RMF Monitor III.

2. Requests that utilize DB2 for long running complex parallel queries, such as star schema parallel queries.

For this particular scenario, we made use of the following star join query which was executed after having enabled star schema parallelism:

```
SELECT COUNT(*) FROM
    ADMF001.TBFACT1 F,
    ADMF001.TBDIMN01 D1,
    ADMF001.TBDIMN02 D2,
    ADMF001.TBDIMN03 D3,
    ADMF001.TBDIMN04 D4,
    ADMF001.TBDIMN05 D5
WHERE
    F.TIME_CLOSED_KEY = D1.TIME_CLOSED_KEY AND
    F.TOD_KEY = D2.TOD_KEY AND
    F.RECEIVED_VIA_KEY = D3.RECEIVED_VIA_KEY AND
    F.CASE_KEY = D4.CASE_KEY AND
    F.CUSTOMER_KEY = D5.CUSTOMER_KEY AND
F.TIME_CLOSED_KEY = 182;
```

We noted some activity being redirected to the zIIP, but not a great deal. Note that even though star schema parallelism has been enabled and a zIIP is available for use, the DB2 Optimizer can decide that the optimal path is not to use star join, thus bypassing the zIIP. The optimizer's focus is not whether the query can take advantage of zIIP offload or not, but rather choosing the lowest cost access path.

3. Some DB2 utilities used in the maintenance of index structures that are normally executed in batch, such as the LOAD, REORG, and REBUILD INDEX utilities.

Testing the offloading of portions of the DB2 LOAD, REORG, and REBUILD INDEX utilities to zIIP entailed the creation of a batch workload comprised of three jobs, each of which performs a task specific to zIIP testing:

LOAD

Reloads tables

RBLDINDX

Rebuilds indexes

REORG

Reorgs tables

The jobs are currently chained together with LOAD executing first; LOAD then calls RBLDINDX, which in turn calls REORG. If desired, for continuous operation REORG can be set to call LOAD again. The three jobs together take about a half hour to complete. Of the three scenarios mentioned, this particular one redirected more work to the zIIP than the star schema parallel queries but less than the Trade 6 workload utilizing DRDA over TCP/IP connections.

OMEGAMON XE for z/OS 3.1.0 zIIP support

We recently installed OMEGAMON XE for z/OS 3.1.0 into our zPET environment. To learn more about OMEGAMON XE for z/OS 3.1.0 go to www.ibm.com/software/tivoli/products/omegamon-xe-zos/

If you already have OMEGAMON XE for z/OS 3.1.0 installed, you will need the following support to enable the zIIP support:

0B550: UA27609 (APAR OA15898)
M2550: UA27610 (APAR OA15899)
M5310: UA27611 (APAR OA15900)

OP360 TEP: 3.1.0-TIV-KM5-IF0001
 ITM6.1 TEP 3.1.0-TIV-KM5-ITM-IF0001

We were the first Plex with zIIPs to actually verify and use the OMEGAMON XE for z/OS 3.1.0 zIIP support. To access the OMEGAMON Classic support for zIIP, select 'C CPU' from the OMEGAMON MAIN MENU. See Figure 61. zIIP is represented by IIP.

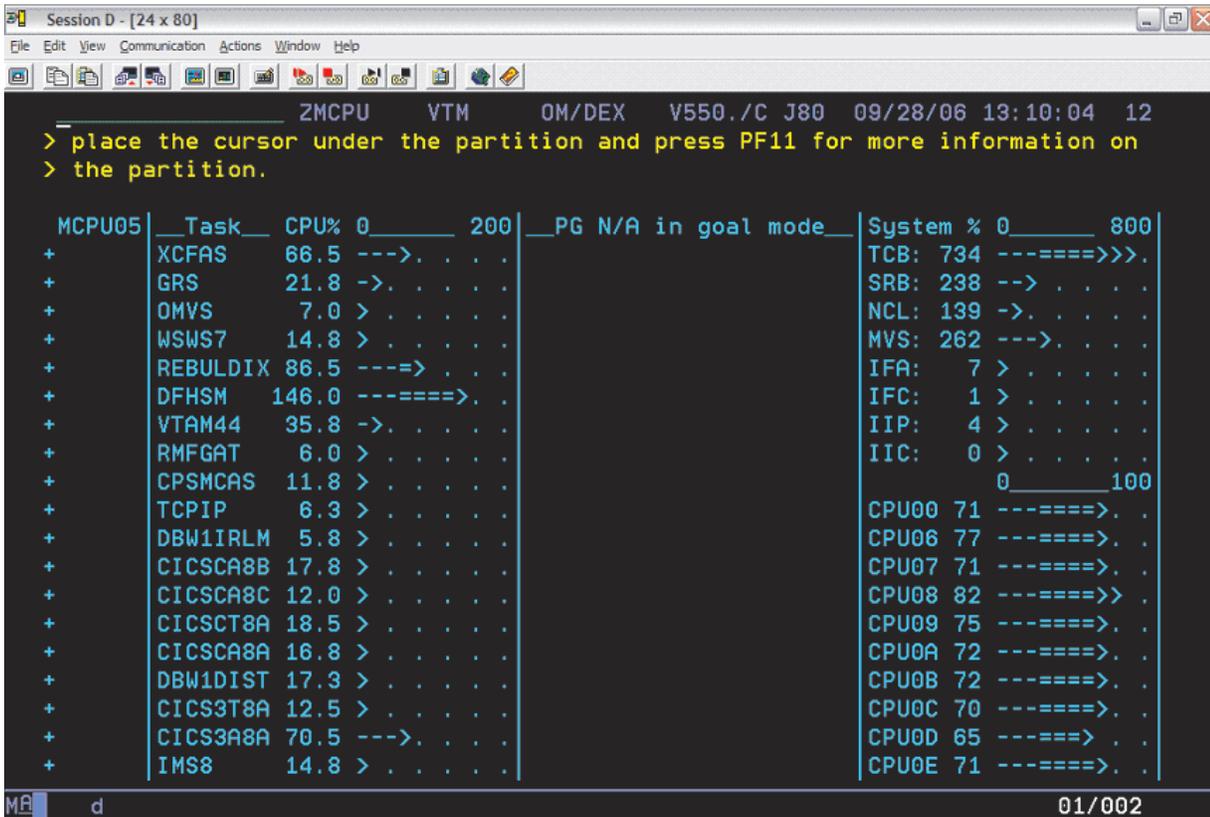


Figure 61. OMEGAMON ZMCPU screen

From your TEP server, the OMEGAMON XE for z/OS zIIP can be found in the predefined workspace System CPU Utilization. Figure 62 on page 105 and Figure 63 on page 106 show the TEP OMEGAMON XE for z/OS System CPU Utilization workspace:

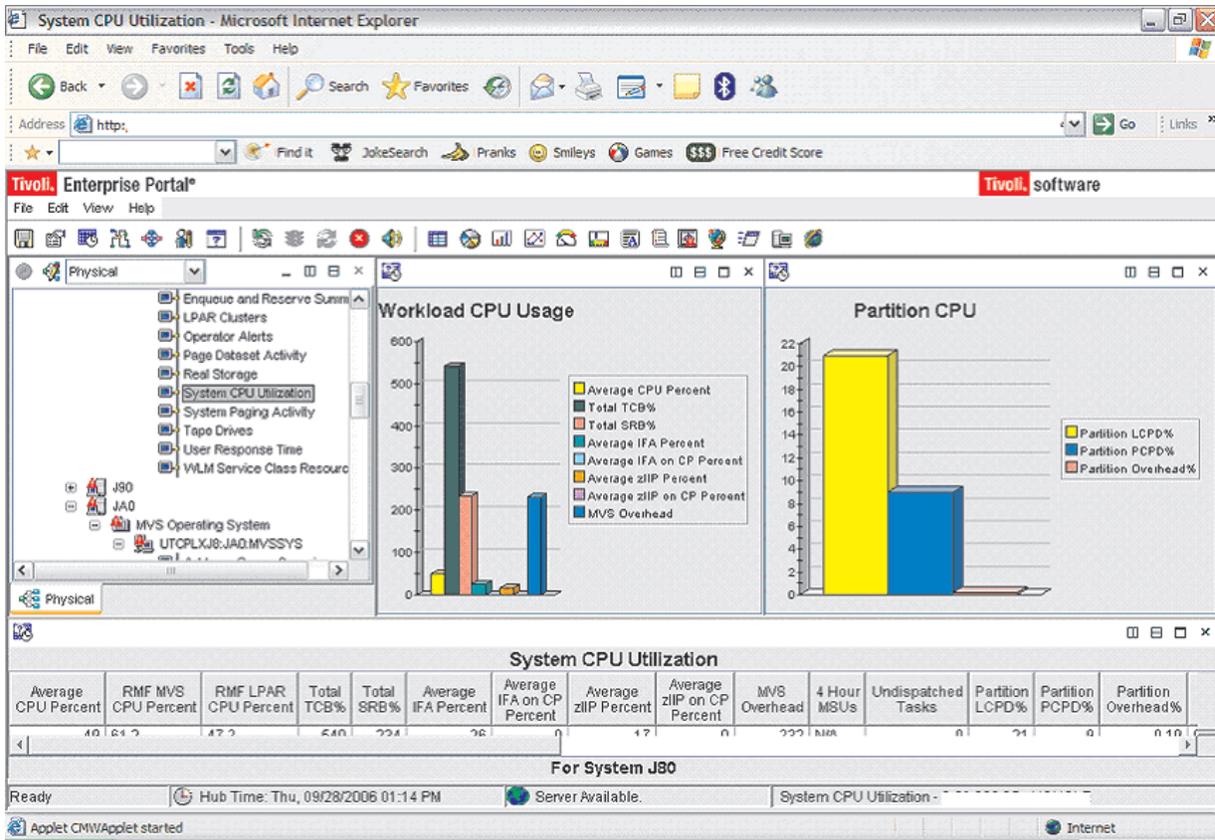


Figure 62. OMEGAMON System CPU Utilization 1

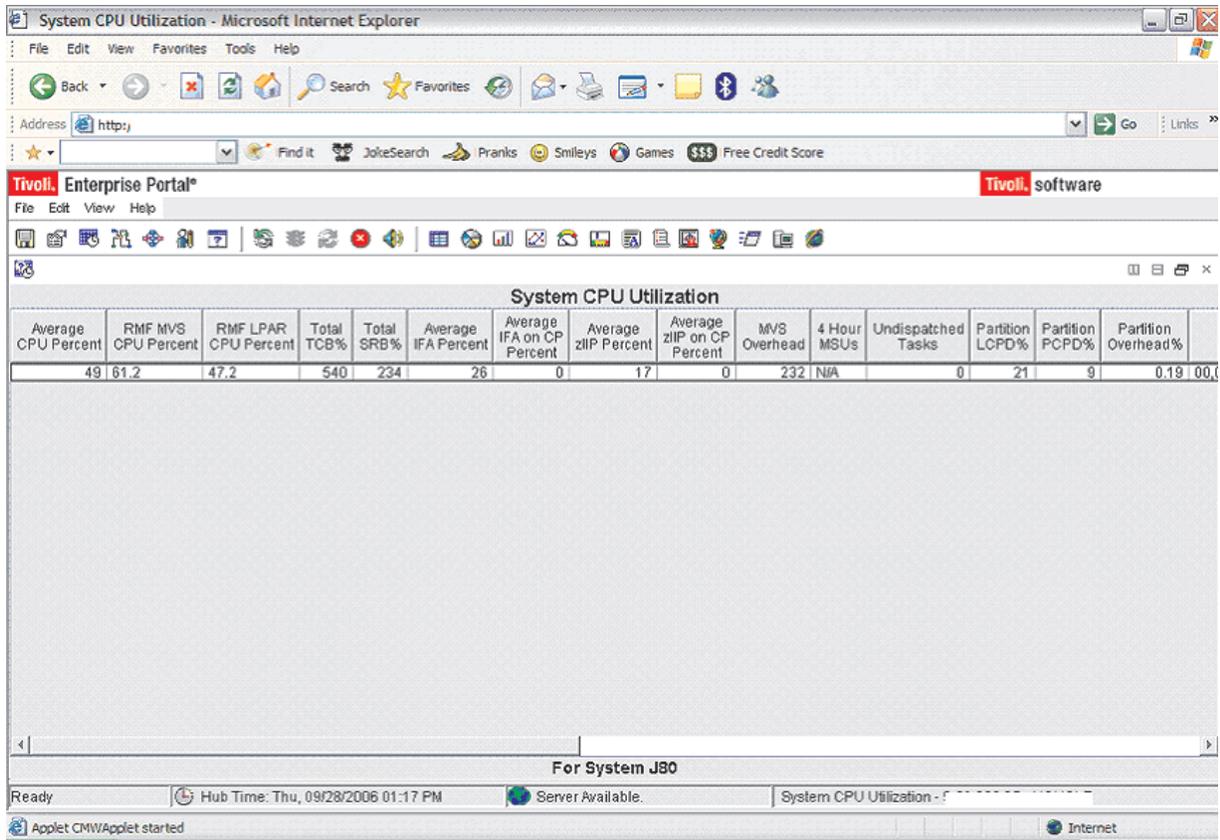


Figure 63. OMEGAMON System CPU Utilization 2

From your TEP server, the OMEGAMON XE for z/OS zIIP support can also be found in the predefined workspace Address Space Overview, as shown in Figure 64 on page 107.

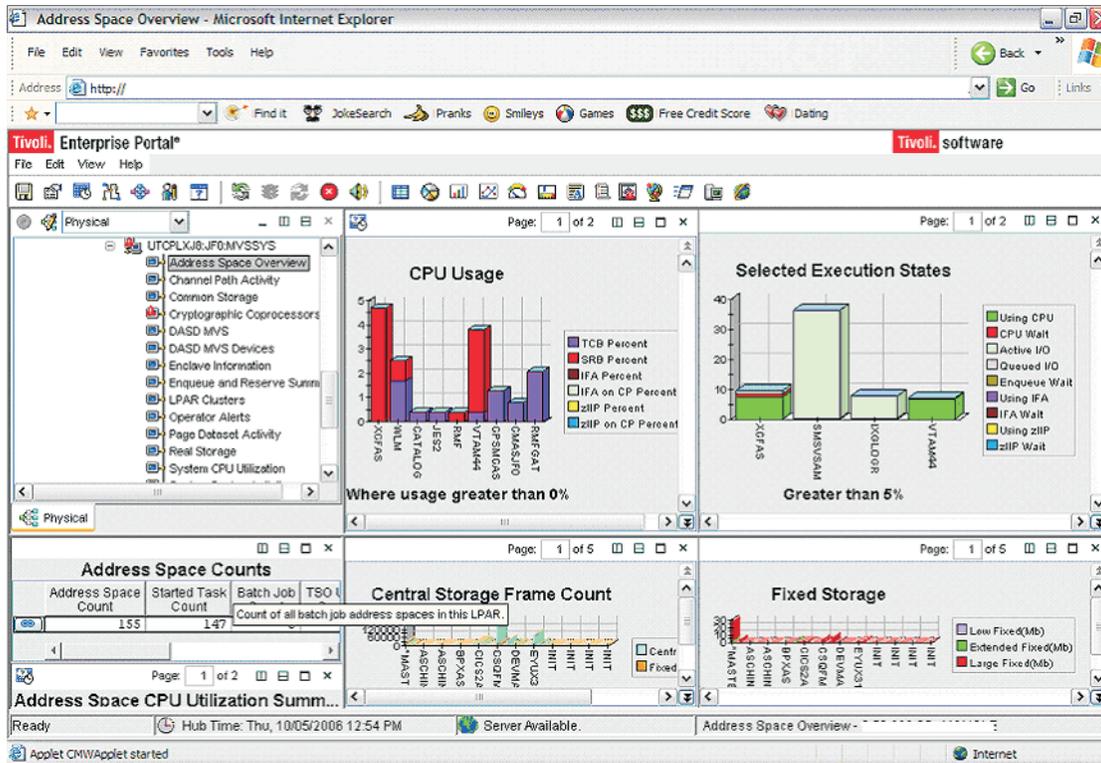


Figure 64. OMEGAMON Address Space Overview

IBM zIIP assisted IPSec

Beginning with z/OS V1R8, the IBM System z9 Integrated Information Processor and System z10 Integrated Information Processor (zIIP) can be used to handle much of the CPU-intensive processing involved in the IPSec Authentication Header (AH) and Encapsulating Security Payload (ESP) protocols. (For details, see IBM announcement 107-190, dated 18 April 2007.)

The new zIIP assisted IPSec function allows z/OS Communications Server to interact with z/OS Workload Manager to have its enclave service request block (SRB) work directed to zIIP. Since our System z9 CPC already had zIIP processors configured, we changed our existing IPSec deployment to use the zIIP processors to reduce the amount of general purpose CP consumption imposed by our IPSec workloads.

Related information: It is beyond the scope of this discussion to provide installation and configuration information related to deploying IPSec on z/OS. For information about deploying IPSec on z/OS, see the z/OS Communications Server library and *Communications Server for z/OS V1R8 TCP/IP Implementation Volume 4: Policy-Based Network Security*, SG24-7342, from the IBM Redbooks® Web site at www.ibm.com/redbooks/.

Implementing the zIIP Assisted IPsec support required that we install Communications Server TCP/IP APAR PK40178. Once we installed this enabling APAR, we performed one required configuration change and completed one strongly recommended task, as follows:

1. We added the GLOBALCONFIG ZIIP IPSECURITY statement to our TCP/IP Profile configuration.
This configuration statement is required to cause Communications Server to request that z/OS direct the IPsec enclave SRB processing to the available zIIPs. The default for this configuration statement is GLOBALCONFIG ZIIP NOIPSECURITY.
2. We created an independent enclave so that IPsec traffic would be classified and managed, within z/OS Workload Manager, differently than its owning address space (that is, it can be classified and managed differently than the TCP/IP address space). This task is optional; however, creating this enclave is strongly recommended.

We already had zIIPs in use in our environment. Thus, we already had APAR OA20045 applied. This APAR allows zIIP tuning controls to be specified in the IEAOPTxx member of PARMLIB. For our environment, we ran with the default values for the following two parameters:

IIPHONORPRIORITY

Specifying IIPHONORPRIORITY=YES allows the zIIP-eligible workload to run on standard CPs if zIIP work is not completed in a reasonable time period (see the ZIIPAWMT parameter). This is the default and recommended value. Specifying IIPHONORPRIORITY=NO disallows any zIIP-eligible work from running on standard CPs.

ZIIPAWMT

ZIIPAWMT controls how aggressive z/OS will be in requesting help from other zIIPs or CPs when IIPHONORPRIORITY=YES and all zIIPs are busy. We ran with the default value of 12 milliseconds.

While we noticed a significant benefit with the deployment of this solution, it is beyond the scope of this report to cite the performance benefits associated with our zIIP Assisted IPsec deployment. However, you can find the configuration requirements for zIIP Assisted IPsec, capacity planning information, and zIIP IPsec performance data at www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP100988.

SDM on the IBM zIIP

Most of the System Data Mover (SDM) processing associated with zGM/XRC has been running on IBM System z9 Integrated Information Processor (zIIP) in our environment. This support works in the System z9 and System z10 environments configured with a zIIP and running z/OS V1R8 or higher along with the IBM System Storage™ DS8000™. We also saw improved utilization of resources at our mirrored site.

XRC on zIIP support was added as part of the DS8000 R3 SPE. To make this function work, we installed zIIP enablement APAR OA23174 on our z/OS V1R9 systems. We used the PARMLIB enablement mechanism for controlling the use of zIIP for XRC. We specified zIIPENABLE parameter in the ANTXIN00 PARMLIB member, as follows:

Category	Parmlib parameter	Values	Dynamic / static
STARTUP	zIIPEnable	YES, NO	D (ANTAS0nn only)

When zIIPEnable is set to YES, the ANTAS000, ANTAS0nn, and ANTCL0nn address spaces are enabled for running on zIIP processors, if installed. When set to NO, these address spaces are prevented from running on zIIP processors. This parameter can be changed dynamically for ANTAS0nn by changing the value and using the XSET command to activate the change. Changes to this parameter are only recognized by XRC address spaces which are restarted subsequent to the change or for which the XSET command is used to activate PARMLIB changes. The XQUERY ENVIRONMENT(PARM) command shows the our current global setting.

Example: The following output is returned as a result of the XQUERY PET ENVIRONMENT(PARM) command. Message ANTQ8253I shows the zIIPEnable option has been set to YES.

```

ANTQ8200I XQUERY STARTED FOR SESSION(ANTAS000) ASNAME(ANTAS000) 800
ANTQ8202I XQUERY ENVIRONMENT_PARM REPORT - 001
ANTQ8251I NAME VALUE NAME VALUE
ANTQ8203I -----
ANTQ8253I zIIPEnable YES MiscHigh 15
ANTQ8253I AllowEnhancedReader NO MiscLow 2
ANTQ8253I BuffersPerStorageCon 576 MonitorOutput OFF
ANTQ8253I ChangedTracks 7500 MonitorWakeup 10000
ANTQ8253I ClusterMSession DISABLED MHLq SYS1
ANTQ8253I ClusterName SYSTEM1 NoTimeStampCount 5000
ANTQ8253I ConsistencyGroupComb 20 NumberReaderTasks (none)
ANTQ8253I DatasetDelay 75 PacingReportThreshol 10
ANTQ8253I DeadSessionDelay 45 PavByteThreshold 512500
ANTQ8253I DefaultHLq SYS1 PavVolumes 3
ANTQ8253I DefaultSessionId DEFAULT PermanentFixedPages 8
ANTQ8253I DelayTime 00.30.00 ReaderPacingLimit 33
ANTQ8253I DeviceBlockingThresh 20 ReaderPacingWindow 3
ANTQ8253I DfltWritePacingLvl 0 ReadDelay 1000
ANTQ8253I EnableREFRESHS NO ReadRecordsPriority 252
ANTQ8253I HaltAnyInit NO ReleaseFixedPages NO
ANTQ8253I HaltThreshold 1280 RequireUtility NO
ANTQ8253I HLq SYS1 ResidualLeftToRead 128
ANTQ8253I InitializationsPerPr 2 ScheduleVerify NO
ANTQ8253I InitializationsPerSe 2 SelectionAlgorithm LOAD
ANTQ8253I InitializationMethod FULL ShadowRead 10
ANTQ8253I InitializationReadWr 120 ShadowTimeoutPercent 40
ANTQ8253I IODataAreas 256 ShadowWrite 10
ANTQ8253I JournalPriority 251 StorageControlTimeou DEFAULT
ANTQ8253I LowAttention 192 SuspendOnLongBusy NO
ANTQ8253I MaxBytesTransferred 512500 TotalBuffers 25000
ANTQ8253I MaxControlTasks 128 TracksPerRead 3
ANTQ8253I MaxNumberInitializat 4 TracksPerWrite 3
ANTQ8253I MaxTotalReaderTasks 32 UtilityDevice FLOAT
ANTQ8253I MaxTracksFormatted 0 VerifyInterval 24
ANTQ8253I MaxTracksRead 64 WriteRecordsPriority 253
ANTQ8253I MaxTracksUpdated 0 WrtPacingResidualCnt 80
ANTQ8253I MinExtenderRead 55 XSWAPPrepareActive NO
ANTQ8253I MinLocalRead 0
ANTQ8203I -----
ANTQ8201I XQUERY ENVIRONMENT_PARM REPORT COMPLETE FOR SESSION(ANTAS000)

```

We used XRC enabled DS8000 system storage at our primary and mirrored site. Our primary volumes contained IMS database applications that we mirrored via XRC and IMS log streams mirrored via XRC+. We configured the coupled extended remote copy (CXRC) environment with three XRC sessions. CXRC provided the scalability that was required to support our XRC configurations.

Before CXRC, when multiple SDMs were implemented, they ran independently and data consistency was not coordinated during recovery. CXRC gave us the capability of coupling multiple XRC sessions together into a master session. For specific information about implementing the CXRC environment, see *z/OS DFSMS Advanced Copy Services*.

We performed the following two scenarios to ensure that SDM instructions could utilize zIIP processors, if available. For the first scenario, we coded the PROJECTCPU=YES parameter in the IEAOPT00 member of SYS1.PARMLIB in order to identify, using RMF, the amount of zIIP eligible work required by ANTAS0xx address spaces. Then we started the CXRC session with zIIP processors offline and monitored the XRC address spaces (ANTAS0xx) via RMF. We noticed that all the SDM address spaces were using the general purpose CPUs for the processing, as expected. Figure 65 and Figure 66 show how RMF reported the amount of zIIP eligible work by ANTAS0xx address spaces.

```

Command ==>
RMF V1R9 Processor Usage Line 1 of 154
Scroll ==> CSR

Samples: 60 System: J80 Date: 01/14/08 Time: 11.02.00 Range: 60 Sec

Jobname  Service  --- Time on CP % ---  ----- EAppl % -----
          CX Class  Total  AAP  IIP  CP  AAP  IIP
LDAPJ808 SO STCI3V50  55.3  0.0  0.0  55.3  0.0  0.0
GRS      S  SYSTEM  23.0  0.0  0.0  23.0  0.0  0.0
CATALOG  S  SYSTEM  20.3  0.0  0.0  20.3  0.0  0.0
ANTAS003 S  SYSTEM  20.3  0.0  12.5  20.3  0.0  0.0
DBWIDBM1 S  STCI2V50  18.8  0.0  0.0  18.8  0.0  0.0
XCFAS    S  SYSSTC  14.9  0.0  0.0  14.9  0.0  0.0
LDABJ804 SO STCI3V50  9.5  0.0  0.0  9.5  0.0  0.0
LDABJ803 SO STCI3V50  9.1  0.0  0.0  9.1  0.0  0.0
TCP/IP   SO SYSSTC  6.5  0.0  0.0  6.5  0.0  0.0

```

Figure 65. RMF Monitor III Processor Usage screen showing amount of zIIP eligible work by ANTAS0xx address spaces (scenario 1)

Figure 66 shows the RMF Workload Activity Report showing zIIP eligible workload by the ANTAS0xx address spaces.

```

REPORT CLASS=SDMXRC
DESCRIPTION =Report Class for SDM (XRC)

I/O--  ---SERVICE---  --SERVICE TIMES--  ---APPL %---  -----STORAGE-----
455.2  IOC  329666  CPU  51.781  CP  9.39  AVG  21581.32
1.1    CPU  9035K  SRB  30.792  AAPCP  0.00  TOTAL  43162.74
0.9    MSO  82023K  RCT  0.000  IIPCP  5.74  SHARED  1.00
0.0    SRB  5373K  IIT  1.966
0.1    TOT  96760K  HST  0.000  AAP  0.00  --PAGE-IN RATES--
0.0    /SEC  107514  AAP  0.000  IIP  0.00  SINGLE  0.0
                                           IIP  0.000  BLOCK  0.0
                                           ABRPTN  54K  SHARED  0.0
                                           TRX SERV  54K  PROMOTED  0.000  HSP  0.0

```

Figure 66. RMF Workload Activity Report showing zIIP eligible workload by the ANTAS0xx address spaces (scenario 1)

For the second scenario, we varied zIIPs online and restarted our CXRC session. We monitored RMF and verified that, once zIIPs became available, all the SDM processing moved to zIIP processors. Figure 67 on page 111 and Figure 68 on page 111 show the zIIP utilization by ANTAS0xx address spaces.

HARDCOPY		RMF V1R9	Processor Usage				Line 1 of 280	
Command ==>								
Samples: 118	System: J80		Date: 01/14/08	Time: 10.14.00		Range: 120	Sec	
	Service	--- Time on CP % ---			----- EApp1 % -----			
Jobname	CX Class	Total	AAP	IIP	CP	AAP	IIP	
CICS3A8A	SO CI2V60	184.1	0.0	0.0	185.3	0.0	0.0	
LDAPJ808	SO STCI3V50	57.7	0.0	0.0	57.7	0.0	0.0	
DBW1DBM1	S STCI2V50	47.6	0.0	0.0	47.7	0.0	0.0	
XCFAS	S SYSSTC	46.9	0.0	0.0	46.9	0.0	0.0	
ANTAS001	S SYSTEM	19.0	0.0	0.9	19.0	0.0	16.3	
CSQMSTR	S STCI2V40	27.1	0.0	0.0	27.1	0.0	0.0	
CATALOG	S SYSTEM	23.6	0.0	0.0	23.6	0.0	0.0	

Figure 67. RMF Monitor III Processor Usage screen showing amount of work processed by zIIP processors for ANTAS0xx address spaces (scenario 2)

Figure 68 shows the workload that was processed by zIIP processors for the ANTAS0xx address spaces.

```

REPORT CLASS=SDMXRC
DESCRIPTION =Report Class for SDM (XRC)

--DASD I/O--  ---SERVICE---  --SERVICE TIMES--  ---APPL %---  -----STORAGE-----
SSCHRT 1466 IOC 175663 CPU 116.647 CP 15.27 AVG 12389.83
RESP 1.3 CPU 20353K SRB 124.692 AAPCP 0.00 TOTAL 24779.69
CONN 1.1 MSO 74819K RCT 0.000 IIPCP 0.54 SHARED 0.00
DISC 0.0 SRB 21756K IIT 7.169
Q+PEND 0.2 TOT 117104K HST 0.000 AAP 0.00 --PAGE-IN RATES--
IOSQ 0.0 /SEC 130119 AAP 0.000 IIP 12.35 SINGLE 0.0
                                           IIP 111.113 BLOCK 0.0
                                           ABSRPTN 65K SHARED 0.0
                                           TRX SERV 65K PROMOTED 0.000 HSP 0.0

```

Figure 68. RMF Workload Activity Report showing work processed by zIIP processors for the ANTAS0xx address spaces (scenario 2)

Chapter 9. Using TPC-R V3.4 and Basic HyperSwap in our zPET environment

We have upgraded the IBM TotalStorage® Productivity Center for Replication (TPC-R) for System z from V3.3 to V3.4 in our z/OS integration test environment. TPC-R 3.4 provides a new availability feature called z/OS Basic HyperSwap™, which is a low-cost, single-site and high-availability disk solution.

In addition to our testing, we have also exploited TPC-R for:

- Configuring, monitoring and controlling our advanced storage replication services (DS8000)
- Peer-to-Peer Remote Copy (PPRC)
- FlashCopy®
- HyperSwap

TPC-R environment in zPET

We upgraded TPC-R V3.3 with V3.4 in our 4-way sysplex. During the migration, we just installed the latest TPC-R V3.4 driver and ran the IWNDBMIG job which made all the necessary migration updates to all of the previously created tables. We installed TPC-R V3.4 on IBM System Services Runtime Environment for z/OS and performed our testing.

We also installed TPC-R V3.4 in our 9-way sysplex.

We used the following software and hardware during our testing:

- z/OS V1R10
- TPC-R V3.4
- DB2 Version 9.1
- 2107 (DS8000) - microcode level 2.4 or higher
- 2105-800 (ESS) - LIC level 2.4.4.45 or higher
- Websphere Application Server V6.1

TPC-R V 3.4 is available in two flavors: TPC-R Basic Edition and TPC-R Full edition. TPC-R Basic Edition, which we installed in our 4-way sysplex environment, only provides the z/OS Basic HyperSwap feature. TPC-R Basic Edition requires a database: z/OS DB2 for z/OS V8.1 (or higher) and either the free IBM System Services Runtime environment (SSRE) or WebSphere Application Server for z/OS V6.1 (or higher). We chose to install it on IBM System Services Runtime environment (SSRE). However, TPC-R Full Edition, which is running in our 9-way sysplex environment, requires WebSphere Application Server for z/OS V6.1, along with DB2 for z/OS V8.1 (or higher) or the Apache Derby database.

All the DB2 application and database volumes that were designated for copy to a secondary site were part of a primary IBM System Storage DS8000 and capable of copy services functions. We also used another IBM System Storage DS8000 with copy services functions enabled as the secondary storage subsystem. Both subsystems were connected via fiber paths.

Installing and setting up TPC-R

There are a number of prerequisites for the IBM TotalStorage Productivity Center for Replication for System z. For a complete list, see the **Installing > Prerequisites** section in the TPC-R Information Center at publib.boulder.ibm.com/infocenter/tivihelp/v4r1/index.jsp?topic=/com.ibm.rm341.doc/frc_c_basichsconfig.html.

TPC-R V3.4 will not install into a WebSphere Application Server network deployment (ND) setup. It is best suited for installation into its own WebSphere Application Server base server setup.

The product install will update the product's SMP/E installed files as part of the customization. In our case, the SMP/E build is performed on a separate sysplex and the product code is copied to our zPET systems where we will actually run it.

On our target system, we did the following:

- Mounted the TPC-R SMP/E copied file system in read/write mode. Normally, we would prefer to have our SMP/E product code mounted as read-only.
- Service updates for the TPC-R product required us to re-run the setup jobs on the target system to customize the copied service update.
- We used a symbolic link to point to the current copy of the TPC-R service, rather than over-writing or replacing it on our target systems. This allows us to mount each service level at different directories.
- Because the TPC-R customization jobs create directories and copy files into the WebSphere Application Server V6.1 configuration files, we made sure the jobs had access to the WebSphere Application Server directories as well as the TPC-R files. We ran the jobs under a user ID with UID=0 and other BPX.FILEATTR.* authorities. See the program directory for full requirements.
- TPC-R also keeps various logging files within the WebSphere Application Server directories. These logs can add 200M bytes or more to the WebSphere Application Server file system. We checked the WebSphere Application Server file system to make sure we had enough space available and potentially some room for growth for when debugging is enabled.
- We checked the output of the customization jobs carefully for errors, including the z/OS UNIX files created (install_RM.log and install_RM_err.log).

z/OS Basic HyperSwap

z/OS Basic HyperSwap with TotalStorage Productivity Center for Replication V3.4 enhanced system availability in our environment. We used TPC-R V3.4 to perform the required hardware operations to establish Peer-to-Peer Remote Copy (PPRC) paths and device pair relationships. We also performed all the planned HyperSwap scenarios to switch I/O from primary logical devices in a synchronous PPRC relation to the secondary logical devices in the PPRC consistency group with no disruption to the applications using TPC-R V3.4. It allowed us to swap between primary and secondary disk volumes in the event of planned and unplanned outages, such as hardware maintenance, testing, or device failure. We initiated planned failover to a secondary for the purpose of initiating hardware maintenance on primary storage controllers, or simply to periodically test the function.

The following is an example of the messages you will see from the HyperSwap master system while initiating a planned HyperSwap:

```

IOSHM0400I 10:51:47.77 HyperSwap requested
IOSHM0424I Master status = 00000000 00000000 0000000600000000
IOSHM0401I 10:51:47.77 Planned HyperSwap started - UserExit
IOSHM0424I Master status = 00000000 00000000 0000000601000000
IOSHM0402I 10:51:47.96 HyperSwap phase - Validation of I/O co
IOSHM0501I Response from API for FC = 14, RC = 0, Rsn = 0
IOSHM0424I Master status = 00000000 80000000 0000000602000000
IOSHM0403I 10:51:48.06 HyperSwap phase - Validation of I/O co
IOSHM0404I 10:51:48.06 HyperSwap phase - Freeze and quiesce D
IOSHM0501I Response from API for FC = 17, RC = 0, Rsn = 0
IOSHM0424I Master status = 00000000 80000000 0000000603000000
IOSHM0405I 10:51:48.19 HyperSwap phase - Freeze and quiesce D
IOSHM0406I 10:51:48.19 HyperSwap phase - Failover PPRC volume
IOSHM0501I Response from API for FC = 10, RC = 0, Rsn = 0
IOSHM0417I 10:51:49.79 Response from JB0, API RC = 0, Rsn = 0
IOSHM0424I Master status = 00000000 80000000 0000000604000000
IOSHM0407I 10:51:49.79 HyperSwap phase - Failover PPRC volume
IOSHM0408I 10:51:49.79 HyperSwap phase - Swap UCBs starting
IOSHM0424I Master status = 00000000 80000000 0000000605000000
IOSHM0409I 10:51:49.90 HyperSwap phase - Swap UCBs completed
IOSHM0410I 10:51:49.90 HyperSwap phase - Resume DASD I/O star
IOSHM0424I Master status = 00000000 80000000 0000000607000000
IOSHM0413I 10:51:56.00 HyperSwap phase - Cleanup completed
IOSHM0414I 10:51:56.00 Planned HyperSwap completed
IOSHM0809I HyperSwap Configuration Monitoring stopped
IOSHM0803E HyperSwap Disabled
IOSHM0501I Response from API for FC = 0, RC = 4, Rsn = 0
IOSHM0200I HyperSwap Configuration Purge complete

```

Overall, z/OS Basic HyperSwap and TPC-R V3.4 simplified our configuration and we have been very happy with the product. Once up and running, we found the Web browser interface to be very intuitive and easy to use. TPC-R improved our operations support by providing different features, such as overwrites protection and improved monitoring and messages. It provided simple operational control of Copy Services tasks, which includes starting and suspending sessions, and also enabling, disabling, and performing HyperSwap.

You can find more information about setting up z/OS Basic HyperSwap at publib.boulder.ibm.com/infocenter/tivihelp/v4r1/index.jsp?topic=/com.ibm.rm341.doc/frc_c_basichsconfig.html.

TPC-R product documentation

See the TPC-R home page at www.ibm.com/software/tivoli/products/totalstorage-replication/ for full information about the product. The home page contains links to detailed information, including the TPC-R Information Center and product support.

Chapter 10. Testing extended address volumes

An extended address volume (EAV) is, by definition, 65 521 cylinders or larger. Extended address volumes are fully described in *z/OS DFSMS Using the New Functions*.

Our objective in testing extended address volumes was to expose these large volumes to the various VSAM workloads we have running in our test environment. This included DB2, WebSphere MQ, WebSphere, z/OS UNIX, CICS, and system type data sets.

The size of the data sets that were migrated to EAV cylinder-managed space were not huge. As long as they were bigger than the **BreakPointValue** of 21 cylinders, they were migrated to cylinder-managed space (10 cylinders is the default **BreakPointValue**). The same was true for new data set allocations. We did have one exception: our z/OS UNIX team allocated a new zFS with 165 000 primary cylinders to EAV.

Requirements for EAV

This topic lists the software and hardware requirements for extended address volumes.

z/OS software requirements for EAV

Extended address volumes require z/OS V1R10 (or higher) and ICKDSF R17.0 (see APAR PK56092).

Restriction: IMS does not support extended address volumes.

Hardware requirements for EAV

Extended address volumes require the IBM DS8000 (2107) with the following upgraded microcode levels:

- Release 4 - R10g.9b080408b (64.0.117.0)
- DS8000 Storage Manager (GUI) and DSCLI: 5.4.0.262

This release will allow you to define the new 3390 model A volume and enable EAV support.

Note: Check with IBM hardware support for the most current requirements and microcode levels.

Setting up EAV

This topic describes our hardware (DASD) and software (DFSMS™) setup for EAV.

Our DASD (DS8000) setup for EAV

Our DS8000 has 45 terabytes of usable capacity.

We configured the DS8000 using the DS8000 Storage Manager GUI. Our DS8000 is defined with 32 LCUs. Each LCU is identical: 10 base addresses and 118 hyperPAV addresses.

The base addresses are configured as:

- 1 3390-A EAV volume with 262K cylinders
- 4 3390-9 volumes, each with 65K cylinders
- 1 3390-A non-EAV volume with 10K cylinders

Notice that 10 base addresses are generated but only six base addresses are configured. Four base addresses are reserved. We did this for future growth. We also configured one 3390-A non-EAV volume so that we can dynamically expand this volume into a 3390-A EAV volume using Dynamic Volume Expansion (DVE).

Table 4 shows the DASD volume configuration for our EAV testing.

Table 4. Our DASD volume configuration for EAV testing

VOLSER	ID	Status	Base / Alias	Volume type	Storage allocation	Cylinders
P20100	0	normal	base	3390-A	standard	262 668
SAD037	1	normal	base	3390 custom	standard	65 520
XX2002	2	normal	base	3390 custom	standard	65 520
XX2003	3	normal	base	3390 custom	standard	65 520
XX2004	4	normal	base	3390 custom	standard	65 520
P2TSO6	5	normal	base	3390-A	standard	10 017

ICKDSF setup

We used ICKDSF V17 to initialize our EAV volumes. Here is an example of our ICKDSF parameters for initializing an SMS-managed EAV volume:

```
INIT UNITADDRESS(2000) DEVICETYPE(3390) PURGE NORECLAIM NOCHECK -  
VERIFY(XX2000) VOLID(P20100) VTOC(3,0,300) INDEX(1,0,30) SG
```

We chose to place the VTOC and VTOCIX at the beginning of the pack, but this is not a requirement. You can place them anywhere in track-managed space.

We defined the INDEX size of 30 tracks to accommodate the increase in the amount of data that the index contains. Also, if you choose to take the default index size, it will be computed by ICKDSF and might not be 15 tracks.

Our DFSMS setup for EAV

This topic describes our parmlib changes, and our SMS and non-SMS configuration changes for EAV.

Parmlib setup

We updated our SYS1.PARMLIB(IGDSMSxx) member to include the two new EAV parameters:

```
USEEAV(YES)  
BREAKPOINTVALUE(21) /* 10 cylinders is the default */
```

To dynamically pick up these new parameters, we issued the following MVS commands on each LPAR in our sysplex:

```
SETSMS USEEAV(YES)  
SETSYS BREAKPOINTVALUE(21)
```

To verify that the MVS commands executed successfully we issued the following MVS command:

```
D SMS,OPTIONS
```

Here is a partial snapshot of the response:

```
BLOCKTOKENSIZE = NOREQUIRE          FAST_VOLSEL = ON
USEEAV = YES                          BREAKPOINTVALUE = 21
OAMPROC = OAM
```

ISMF (SMS managed) setup

We left the BreakPointValue definitions blank in ISMF Option 6, Storage Group, so the IGDSMSxx settings were not overridden.

We then added 3390-A EAV and 3390-A non-EAV volumes to the various SMS storage groups. To force new allocations to the 3390-A volumes, we set the other volumes in the same storage groups to DISNEW status. We asked our testers to use storage classes with the guaranteed space attribute in their JCL to allocate data specifically to 3390-A volumes.

Non-SMS setup

For non-SMS 3390-A EAV and non-EAV volumes, we used the same ICKDSF parameters (except for the SG parameter) to initialize these volumes. We then added them to the SYSDA esoteric and mounted them as STORAGE.

```
D U,,,2800,1
IEE457I 13.35.47 UNIT STATUS
UNIT TYPE STATUS          VOLSER          VOLSTATE
2800 3390 0                SP0009          STRG/RSDNT

DS QD,2800,1
IEE459I 13.35.54 DEVSERV QDASD
INIT VOLSER SCUTYPE DEVTYPE          CYL SSID SCU-SERIAL DEV-SERIAL EFC
2800 SP0009 2107932 2107900 262668 2C50 0175-M9991 0175-M9991 *OK
**** 1 DEVICE(S) MET THE SELECTION CRITERIA
**** 0 DEVICE(S) FAILED EXTENDED FUNCTION CHECKING
```

Migrating to EAV

We used the following products to migrate EAV supported data sets and non-EAV supported data sets to and from EAV track-managed and cylinder-managed spaces:

- DFSMSdss™ logical data set COPY
- DFSMSshm™ MIGRATE and RECALL commands
- Transparent Data Migration Facility (TDMF™) 5.1.0 for volume moves
- zSeries Data Migration Facility (zDMF) 3.1.0 for data set moves

As you will see from the following experiences, implementing these huge volumes into our environment was straight forward, and we did not encounter any major issues.

Verifying the location of data on EAV volumes

There are several ways to verify the location of data on an EAV volume. One way is to use the DISKMAP tool. Another way is to simply run IDCAMS LISTCAT ENT against a data set and look at the VOLUME EXTENTS LOW-CCHH and HIGH-CCHH values.

In the following example, the LOW-CCHH and HIGH-CCHH values are expressed in the form **X'CCCCcccH'** where **CCCC** is the low-order 16 bits of the 28-bit cylinder number, and **ccc** is the high-order 12 bits of the 28-bit cylinder number. If the **ccc** value is greater than zero, it is allocated on cylinder-managed space.

The following is an example of the JCL for the IDCAMS LISTCAT job:

```

//LISTCAT JOB ...
//LISTC EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
LISTC ENT('DB2DBTG.DSNDBC.DBITRK10.TSLBRS10.I0001.A001') ALL

```

The following is an example of the output from the IDCAMS LISTCAT job:

```

1IDCAMS SYSTEM SERVICES                                TIME: 11:02:2
0
LISTC ENT('DB2DBTG.DSNDBC.DBITRK10.TSLBRS10.I0001.A001') ALL      0000000
0CLUSTER ----- DB2DBTG.DSNDBC.DBITRK10.TSLBRS10.I0001.A001
IN-CAT --- CATALOG.DSNDB1G
HISTORY
  DATASET-OWNER-----JCORRY      CREATION-----2008.204
  RELEASE-----2                EXPIRATION-----0000.000
SMSDATA
  STORAGECLASS -----NST          MANAGEMENTCLASS-DATABASE
  DATACLASS -----(NULL)         LBACKUP ---0000.000.0000
  BWO STATUS-----00000000        BWO TIMESTAMP---00000 00:00:00.0
  BWO----- (NULL)
RLSDATA
  LOG -----(NULL)                RECOVERY REQUIRED --(NO)      FRLOG -----
  VSAM QUIESCED -----(NO)        RLS IN USE -----(NO)
0 LOGSTREAMID----- (NULL)
  RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
  RECOVERY TIMESTAMP GMT-----X'0000000000000000'
PROTECTION-PSWD----- (NULL)      RACF----- (NO)
ASSOCIATIONS
  DATA----DB2DBTG.DSNDBD.DBITRK10.TSLBRS10.I0001.A001
0 DATA ----- DB2DBTG.DSNDBD.DBITRK10.TSLBRS10.I0001.A001
IN-CAT --- CATALOG.DSNDB1G
HISTORY
  DATASET-OWNER-----JCORRY      CREATION-----2008.204
  RELEASE-----2                EXPIRATION-----0000.000
  ACCOUNT-INFO----- (NULL)
PROTECTION-PSWD----- (NULL)      RACF----- (NO)
ASSOCIATIONS
  CLUSTER--DB2DBTG.DSNDBC.DBITRK10.TSLBRS10.I0001.A001
ATTRIBUTES
  KEYLEN-----0                AVGLRECL-----0            BUFSPACE-----
  RKP-----0                  MAXLRECL-----0            EXCPEXIT-----
  SHROPTNS(3,3)    SPEED        UNIQUE                NOERASE            LINEAR
  UNORDERED                REUSE                NONSPANNED
STATISTICS
  REC-TOTAL-----0            SPLITS-CI-----0            EXCPS-----
  REC-DELETED-----0          SPLITS-CA-----0            EXTENTS-----
  REC-INSERTED-----0          FREESPACE-%CI-----0        SYSTEM-TIMESTA
  REC-UPDATED-----0          FREESPACE-%CA-----0            X'00000000
  REC-RETRIEVED-----0        FREESPC-----0
ALLOCATION
  SPACE-TYPE-----CYLINDER      HI-A-RBA-----30965760
  SPACE-PRI-----35            HI-U-RBA-----28016640
  SPACE-SEC-----2
VOLUME
  VOLSER-----DB2A80          PHYREC-SIZE-----4096      HI-A-RBA-----
  DEVTYPE-----X'3010200F'      PHYRECS/TRK-----12        HI-U-RBA-----
1IDCAMS SYSTEM SERVICES                                TIME: 11:02:2
0 VOLFLAG-----PRIME          TRACKS/CA-----15
EXTENTS:
  LOW-CCHH-----X'79B10020'      LOW-RBA-----0            TRACKS-----
  HIGH-CCHH----X'79DA002E'        HIGH-RBA-----30965759
1IDCAMS SYSTEM SERVICES                                TIME: 11:02:2

```

zFS EAV setup and exploitation

The following steps describe our setup and exploitation of EAV for zFS:

1. Created a large zFS file system with 165 144 primary cylinders and no secondary extents on the EAV volume

This took approximately 40 minutes to complete, in our environment. Figure 69 shows the number of primary cylinders successfully assigned, as well as the EAV volume on which the zFS file system was successfully allocated and formatted.

```

                                Data Set Information
Data Set Name . . . . : OMVSPN.EAVZFS1.ZFS.DATA
General Data
Management class . . . : **None**
Storage class . . . . : **None**
Volume serial . . . . : SS0006
Device type . . . . . : 3390
Data class . . . . . : **None**
Organization . . . . . : VS
Record format . . . . : ?
Record length . . . . : ?
Block size . . . . . : ?
1st extent cylinders: 165144
Secondary cylinders : 0
Data set name type : EXTENDED
Current Allocation
Allocated cylinders : 165,144
Allocated extents . . : 1
Current Utilization
Used cylinders . . . : ?
Used extents . . . . : ?
Creation date . . . . : 2008/10/23
Referenced date . . . : 2008/10/23
Expiration date . . . : ***None***
```

Figure 69. Data Set Information panel for our zFS file system on an EAV volume

2. Exercised the zFS file system on the EAV volume using the following z/OS UNIX workloads and tools:
 - A workload that writes, reads, and removes large files (for instance, 1G, 2G in size) and measures file system activity while each task is being performed
 - A workload that attempts to fill up the zFS file systems by creating directories and files and subsequently empties the file system by removing these directories and files
3. Regression tested the related z/OS UNIX and zFS displays and utilities to ensure that displays were accurate and consistent for the zFS file system allocated on the EAV volume, for example:
 - Verified the output of the **df** and **zfsadm** commands from OMVS and from the z/OS UNIX shell
 - Verified the attributes of the mounted zFS file system using the ISPF shell (ISHELL)

There were no errors encountered with this test.

DB2 EAV 3390-A setup and exploitation

The following DB2 APARs were required to enable the support of EAV volumes:

- PK58291 (DB2 V8 PTF UK36130, DB2 V9 PTF UK36131) provides support to allow DB2-supplied administrative enablement stored procedures to access EAV
- PK58292 (DB2 V8 PTF UK35901, DB2 V9 PTF UK35902) provides DB2 Log Manager support for EAV

- PK61105 (DB2 V8 PTF UK34129, DB2 V9 PTF UK34130) provides serviceability enhancement to utilities general services

To test the DB2 exploitation of EAV volumes, we moved a DB2 database to a 3390-A EAV volume and then verified that workloads and utilities continued to function normally.

The following steps describe our setup and exploitation of EAV 3390-A volumes with DB2:

1. Selected database DBSCTL01 and partitioned tablespace TINST001 (20 parts and each partition is greater than the **BreakPointValue** of 21 cylinders)
2. Stopped the database and tablespace
3. Moved the VSAM data sets to EAV volume DB2A80 using DFDSS logical data set processing and specifying a storage class using the guaranteed space parameter, as shown in the following JCL:

```
//STEP01 EXEC PGM=ADRDSSU
//OUTPACK DD VOL=SER=DB2A80,DISP=SHR,UNIT=3390
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY DATASET(INCL -
(DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A001 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A002 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A003 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A004 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A005 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A006 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A007 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A008 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A009 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A010 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A011 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A012 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A013 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A014 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A015 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A016 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A017 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A018 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A019 -
DB2DBWG.DSNDBC.DBSCTL01.TINST001.I0001.A020)) -
OUTDD(OUTPACK) STORCLAS(DBGSPACE) -
DELETE PURGE
```

4. Verified LOGICAL COPY using ISPF 3.4:

Command - Enter "/" to select action	Message	Volume
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A002	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A003	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A004	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A005	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A006	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A007	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A008	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A009	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A010	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A011	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A012	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A013	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A014	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A015	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A016	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A017	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A018	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A019	DB2A80
	DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A020	DB2A80

5. Verified the location of the data sets in cylinder-managed space using the IEHLIST LISTVTOC format:

DATE: 2008.213 TIME: 16.53.11

CONTENTS OF VTOC ON VOL **DB2A80** <THIS IS AN SMS MANAGED VOLUME>

THERE IS A 2 LEVEL VTOC INDEX
DATA SETS ARE LISTED IN ALPHANUMERIC ORDER

FORMAT 4	DSCB	NO AVAIL	/MAX DSCB	/MAX DIRECT	NO AVAIL	NEXT ALT	FORMAT 6	LAST FMT 1	VTOC EXTENT	THIS DSCB							
VI	DSCBS	PER TRK	BLK PER TRK	ALT TRK	TRK(C-H)	(C-H-R)	DSCB(C-H-R)	LOW(C-H)	HIGH(C-H)	(C-H-R)							
81	44996	50	45	0	0	0	62	14	50	3	0	62	14	3	0	1	
NUMBER OF		MULTICYLINDER UNITS															
CYLINDERS		FIRST CYL ADDR		SPACE													
262668		65520		21													

-----DATA SET NAME-----	SER NO	SEQNO	DATE.CRE	DATE.EXP	DATE.REF	EXT	DSORG	RECFM	OPTCD	BLKSIZE	
DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A001	DB2A80	1	2006.017	00.000	2008.213	1	VS	U	80	4096	
SMS.IND	LRECL	KEYLEN	INITIAL ALLOC	2ND ALLOC	EXTEND	LAST BLK(T-R-L)	DIR.REM	PTR TO F3(C-H-R)	DSCB(C-H-R)		
S	0		CYLS	66					5	7	11
EXT.NO	LOW(C-H)	HIGH(C-H)									
0	206640	0	207500	14	----UNABLE TO CALCULATE EMPTY SPACE.						

-----DATA SET NAME-----	SER NO	SEQNO	DATE.CRE	DATE.EXP	DATE.REF	EXT	DSORG	RECFM	OPTCD	BLKSIZE	
DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A002	DB2A80	1	2006.017	00.000	2008.213	1	VS	U	80	4096	
SMS.IND	LRECL	KEYLEN	INITIAL ALLOC	2ND ALLOC	EXTEND	LAST BLK(T-R-L)	DIR.REM	PTR TO F3(C-H-R)	DSCB(C-H-R)		
S	0		CYLS	66					5	8	27
EXT.NO	LOW(C-H)	HIGH(C-H)									
0	207501	0	208235	14	----UNABLE TO CALCULATE EMPTY SPACE.						

:

-----DATA SET NAME-----	SER NO	SEQNO	DATE.CRE	DATE.EXP	DATE.REF	EXT	DSORG	RECFM	OPTCD	BLKSIZE	
DB2DBWG.DSNDBD.DBSCTL01.TINST001.I0001.A020	DB2A80	1	2006.017	00.000	2008.213	1	VS	U	80	4096	
SMS.IND	LRECL	KEYLEN	INITIAL ALLOC	2ND ALLOC	EXTEND	LAST BLK(T-R-L)	DIR.REM	PTR TO F3(C-H-R)	DSCB(C-H-R)		
S	0		CYLS	66					9	8	31
EXT.NO	LOW(C-H)	HIGH(C-H)									
0	220290	0	220961	14	----UNABLE TO CALCULATE EMPTY SPACE.						

6. Started the database and tablespace

7. Successfully reorganized partition 5 using the following control statements:

```
//SYSIN DD *
REORG TABLESPACE DBSCTL01.TINST001 PART 5
SHRLEVEL NONE REUSE ==> (to logically reset and reuse DB2-managed
                           data sets without deleting and redefining;
                           otherwise, the reorg would move to another
                           volume)
```

8. Successfully checked all indexes:

```
OUTPUT START FOR UTILITY, UTILID = CHECKI
PROCESSING SYSIN AS EBCDIC
CHECK INDEX(ALL) TABLESPACE DBSCTL01.TINST001
INDEXES WILL BE CHECKED IN PARALLEL, NUMBER OF TASKS = 9
NUMBER OF TASKS CONSTRAINED BY VIRTUAL STORAGE
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 3
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 6
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 9
- 99998 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 12
- 99998 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 15
- 99998 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 18
- 99999 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 2
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 5
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 8
- 99996 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 11
- 99999 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 14
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 17
- 99998 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 20
- 99998 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 1
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 4
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 7
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 10
- 99997 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 13
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 16
- 100000 INDEX ENTRIES UNLOADED FROM 'DBSCTL01.TINST001' PARTITION= 19
LOAD PHASE COMPLETE - ELAPSED TIME=00:01:27
1999981 ENTRIES CHECKED FOR INDEX 'NST.XINST001'
SORTCHK PHASE COMPLETE, ELAPSED TIME=00:00:01
- 1999981 ENTRIES CHECKED FOR INDEX 'NST.XINST001'
UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

9. Started workloads to verify that the tablespace is allocated:

```
DSNT360I @DBW1 *****
DSNT361I @DBW1 * DISPLAY DATABASE SUMMARY
* GLOBAL LOCKS
DSNT360I @DBW1 *****
DSNT362I @DBW1 DATABASE = DBSCTL01 STATUS = RW
DBD LENGTH = 40370
DSNT397I @DBW1
NAME TYPE PART STATUS CONNID CORRID LOCKINFO
-----
TINST001 TS 0001 RW MEMBER NAME DBW2 (CO) H-SIX,PP,I
-
TINST001 TS 0001 RW H-IS,PP,I
:
-
TINST001 TS 0005 RW MEMBER NAME DBW1 H-SIX,PP,I
-
TINST001 TS 0005 RW MEMBER NAME DBW2 (CO) H-IS,PP,I
```

```

-                               MEMBER NAME DBW5
TINST001 TS      0005 RW                               H-IS,PP,I
-                               MEMBER NAME DBW1
:

```

No errors were detected from the workloads.

DB2 non-EAV 3309-A setup and exploitation

To test DB2 support for non-EAV 3390-A volumes, we performed a test similar to the one described in “DB2 EAV 3390-A setup and exploitation” on page 121. We moved a DB2 database to non-EAV 3390-A volumes. We then verified that our workloads and utilities continued to function normally on the relocated object.

The database selected for the move was DBSCTL25, which resided on the following volumes:

DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A002	DB3A00
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A013	DB3A1A
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A010	DB3C02
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A004	DB3C04
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A006	DB3080
DB2DBWG.DSNDBD.DBSCTL25.TSSUPP25.I0001.A001	DB3083
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A012	DB3093
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A003	DB390B
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A005	DB3904
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A008	DB5F0F
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A012	DB6002
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A001	DB6005
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A010	DB6039
DB2DBWG.DSNDBD.DBSCTL25.TSVEND25.I0001.A001	DB6052
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A016	DB6054
DB2DBWG.DSNDBD.DBSCTL25.TSOPAR25.I0001.A001	DB6056
DB2DBWG.DSNDBD.DBSCTL25.XPARTS25.I0001.A001	DB6070
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A005	DB6084
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A004	DB6090
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A001	DB6113
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A007	DB6126
DB2DBWG.DSNDBD.DBSCTL25.XOPARN25.I0001.A001	DB6128
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A002	DB6139
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A009	DB6148
DB2DBWG.DSNDBD.DBSCTL25.XORDRN25.I0001.A001	DB6158
DB2DBWG.DSNDBD.DBSCTL25.XVENVE25.I0001.A001	DB6163
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A011	DB6199
DB2DBWG.DSNDBD.DBSCTL25.XPARM125.I0001.A001	DB6209
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A009	DB6216
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A014	DB6218
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A015	DB6221
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A015	DB6225
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A007	DB6242
DB2DBWG.DSNDBD.DBSCTL25.XSUPPA25.I0001.A001	DB6251
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A008	DB6267
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A011	DB6267
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A016	DB6274
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A003	DB6275
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A014	DB6276
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A006	DB6300
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A013	DB6378

The following steps describe our setup and exploitation of non-EAV 3390-A volumes with DB2:

1. Removed this database from our workload for the move
2. Moved the data sets using the following JCL:

```
//MOVEVSAM JOB ...
//STEP01 EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DBWG)
    -STOP DB(DBSCTL25) SPACENAM(*)
    END
/*
//STEP02 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    COPY DATASET(INCL(
        DB2DBWG.DSNDBC.DBSCTL25.**
    ))
    OUTDY((DB2105),(DB2185))
    LOGINDY((DB3A00),
        (DB3A1A),
        (DB3C02),
        (DB3C04),
        (DB3080),
        (DB3083),
        (DB3093),
        (DB390B),
        (DB3904),
        (DB5F0F),
        (DB6002),
        (DB6005),
        (DB6039),
        (DB6052),
        (DB6054),
        (DB6056),
        (DB6070),
        (DB6084),
        (DB6090),
        (DB6113),
        (DB6126),
        (DB6128),
        (DB6139),
        (DB6148),
        (DB6158),
        (DB6163),
        (DB6199),
        (DB6209),
        (DB6216),
        (DB6218),
        (DB6221),
        (DB6225),
        (DB6242),
        (DB6251),
        (DB6267),
        (DB6274),
        (DB6275),
        (DB6276),
        (DB6300),
        (DB6378))
    -
```

```

|          STORCLAS(GSPACE)          -
|          VOL(SRC)                   -
|          DELETE FORCE PURGE CAT      -
|          SELM(ANY) TOL(ENQF) SPHERE
|          /*
|          //STEP03 EXEC PGM=IKJEFT01
|          //SYSTSPRT DD SYSOUT=*
|          //SYSTSIN DD *
|          DSN SYSTEM(DBWG)
|          -START DB(DBSCTL25) SPACENAM(*)
|          END
|          /*
|          //

```

The move was successful; data sets now resided on the two non-EAV 3390-A volumes:

DB2DBWG.DSNDBD.DBSCTL25.TSOPAR25.I0001.A001	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A001	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A002	DB2185
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A003	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A004	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A005	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A006	DB2185
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A007	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A008	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A009	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A010	DB2185
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A011	DB2185
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A012	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A013	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A014	DB2185
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A015	DB2105
DB2DBWG.DSNDBD.DBSCTL25.TSPART25.I0001.A016	DB2185
DB2DBWG.DSNDBD.DBSCTL25.TSSUPP25.I0001.A001	DB2185
DB2DBWG.DSNDBD.DBSCTL25.TSVEND25.I0001.A001	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XOPARN25.I0001.A001	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XORDRN25.I0001.A001	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARM125.I0001.A001	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A001	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A002	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A003	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A004	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A005	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A006	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A007	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A008	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A009	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A010	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A011	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A012	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A013	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A014	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A015	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XPARPA25.I0001.A016	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XPARTS25.I0001.A001	DB2185
DB2DBWG.DSNDBD.DBSCTL25.XSUPPA25.I0001.A001	DB2105
DB2DBWG.DSNDBD.DBSCTL25.XVENVE25.I0001.A001	DB2105

- Executed the reorg using online image copy (with parameters SHRLEVEL NONE and REUSE to prevent data set relocation), runstats, load (again with

the SHRLEVEL NONE and REUSE parameters) and check index utilities against table spaces TSOPAR25, TSPART25, TSSUPP25, and TSVEND25, all of which completed successfully.

4. Reintroduced the database back into to our workload.

The workload continued to run without errors.

WebSphere MQ EAV setup and exploitation

Our approach for testing our WebSphere MQ V6 queue managers using EAV volumes was to identify BSDS, LOGCOPY, and PSID data sets that would be moved to EAV volumes. Once these data sets were moved, we verified that the queue managers successfully initialized and functioned without problems.

Working with our storage management team, we performed the following scenarios:

- Moved the BSDS, LOGCOPY, and PSID data sets for one of our queue managers to an EAV volume (cylinder-managed space)
- Moved the BSDS, LOGCOPY, and PSID data sets for another one of our queue managers to an EAV volume (track-managed space)
- Moved the one copy of the BSDS and LOGCOPY data sets for one of our queue managers to an EAV volume, leaving the other copy on non-EAV volumes
- Repro'd a BSDS on an EAV volume to produce a usable BSDS

In each of these scenarios, our queue managers were successfully initialized and continue to process our daily workloads without problems.

We ran some of the WebSphere MQ utilities against these data sets that now reside on EAV volumes to verify proper formatting:

- Ran the CSQ1LOGP log print utility against the LOGCOPY data set residing on an EAV volume
- Ran the CSQ1LOGP log print utility against the BSDS data set residing on an EAV volume
- Ran the CSQJU004 print log map utility against the BSDS data set residing on an EAV volume

Another test scenario was to format a new pageset on an EAV volume using guaranteed space with three volumes defined. The data set was defined with 600 cylinders and the result was a pageset data set allocated with 1800 cylinders (three volumes with 600 cylinders each). With the guaranteed space (GSPACE) attribute, the system will allocate the primary space on each volume. For example, if you specify a primary of 200 cylinders. and a volume count of five, then the system will allocate 200 cylinders on each of five volumes, giving a total initial allocation of 1000 cylinders. The pageset was defined to our queue manager specifying EXPAND(SYSTEM) and continues to be used today for one of our daily workloads.

Chapter 11. Using High Performance FICON for System z

High Performance FICON for System z (zHPF) channel programs are exploited by our OLTP I/O workloads—DB2, VSAM, PDSE and zFS—which transfer small blocks of fixed size data (4K blocks). The initial implementation of zHPF by the DS8000 is exclusively for I/O operations that transfer less than a single track of data.

Our main objective in exploiting zHPF was to test the usability of the z/OS external interfaces for zHPF. We did, however, execute before-and-after performance runs on our DB2 OLTP workloads exploiting zHPF, and while we did see an increase in throughput, documenting our performance measurement findings is beyond the scope of this report.

We enabled zHPF on all of our DS8000s, four in total. Data behind these four devices included: DB2, CICS, IMS, WAS, WebSphere MQ, system data, HFS, zFS, and TSO user data.

Required hardware for zHPF

The hardware requirements for zHPF in our environment include:

- System z10 CPC
- FICON channels
- DS8000 2107, with:
 - Microcode level R4.1 - R12q.9b080807a (bundle version 64.1.1.0)

Note: Check with IBM hardware support for the most current microcode levels.

- zHPF LIC feature keys enabled (turned ON)

z/OS external interfaces for zHPF

You can customize the operation of zHPF on z/OS by using the IECIOSxx parmlib member or the SETIOS system command. You can also use the DISPLAY MATRIX system command to display the zHPF status.

Our testing concentrated on using these external interfaces.

IECIOSxx parmlib support for zHPF

You can customize the IECIOSxx parmlib member to turn zHPF on or off. These changes take effect at the next IPL.

- To turn zHPF on:
ZHPF=YES
- To turn zHPF off:
ZHPF=NO

SETIOS system command support for zHPF

You can dynamically enable or disable zHPF by using the SETIOS system command.

- To dynamically turn zHPF on:
SETIOS ZHPF=YES
- To dynamically turn zHPF off:
SETIOS ZHPF=NO

Examples of system commands for zHPF

The following are some examples of the SETIOS and DISPLAY MATRIX system commands that we used to test zHPF in our environment, along with the command responses.

- Issuing the SETIOS command on a System z9 CPC:

```
SETIOS ZHPF=YES
```

```
IOS085I SETIOS. ZHPF=YES ZHPF FACILITY NOT SUPPORTED BY PROCESSOR
```

- Issuing the SETIOS command on a System z10 CPC:

```
SETIOS ZHPF=YES
```

```
IOS090I SETIOS. ZHPF UPDATE(S) COMPLETE
```

The same IOS090I message is issued in response to the SETIOS ZHPF=NO command.

- Issuing the DISPLAY MATRIX command on a DS8000 with zHPF disabled:

```
D M=DEV(2000)
```

```
IEE174I 10.25.42 DISPLAY M 932
DEVICE 2000 STATUS=OFFLINE
CHP          5C   60   62   63   64   66   6A   5F
ENTRY LINK ADDRESS  22  84  1C  70  93  A4  85  B0
DEST LINK ADDRESS  72  62  6E  B2  E3  E2  63  B1
PATH ONLINE       Y   Y   Y   Y   Y   Y   Y   Y
CHP PHYSICALLY ONLINE Y   Y   Y   Y   Y   Y   Y   Y
PATH OPERATIONAL  Y   Y   Y   Y   Y   Y   Y   Y
MANAGED          N   N   N   N   N   N   N   N
CU NUMBER        2000 2000 2000 2000 2000 2000 2000 2000
MAXIMUM MANAGED CHPID(S) ALLOWED:  0
DESTINATION CU LOGICAL ADDRESS = 00
SCP CU ND        = 002107.900.IBM.75.0000000M9991.0330
SCP TOKEN NED    = 002107.900.IBM.75.0000000M9991.0000
SCP DEVICE NED   = 002107.900.IBM.75.0000000M9991.0000
HYPERPAV ALIASES CONFIGURED = 0
FUNCTIONS ENABLED = MIDAW
```

Note that zHPF is not listed as one of the enabled functions.

- Issuing the DISPLAY MATRIX command on a DS8000 with zHPF enabled on a System z10 CPC:

```
D M=DEV(2000)
```

```
IEE174I 10.25.57 DISPLAY M 966
DEVICE 2000 STATUS=OFFLINE
CHP          5C   60   62   63   64   66   6A   5F
ENTRY LINK ADDRESS  22  84  1C  70  93  A4  85  B0
DEST LINK ADDRESS  72  62  6E  B2  E3  E2  63  B1
PATH ONLINE       Y   Y   Y   Y   Y   Y   Y   Y
CHP PHYSICALLY ONLINE Y   Y   Y   Y   Y   Y   Y   Y
PATH OPERATIONAL  Y   Y   Y   Y   Y   Y   Y   Y
MANAGED          N   N   N   N   N   N   N   N
CU NUMBER        2000 2000 2000 2000 2000 2000 2000 2000
```

```

MAXIMUM MANAGED CHPID(S) ALLOWED: 0
DESTINATION CU LOGICAL ADDRESS = 00
SCP CU ND          = 002107.900.IBM.75.0000000M9991.0330
SCP TOKEN NED     = 002107.900.IBM.75.0000000M9991.0000
SCP DEVICE NED    = 002107.900.IBM.75.0000000M9991.0000
HYPERPAV ALIASES CONFIGURED = 0
FUNCTIONS ENABLED = MIDAW, ZHPF

```

Note that zHPF is now listed as one of the enabled functions.

- Issuing the DISPLAY MATRIX command on a FICON CHPID with zHPF enabled on a System z10 CPC:

```
D M=CHP(5C)
```

```

IEE174I 10.34.57 DISPLAY M 540
CHPID 5C: TYPE=1B, DESC=FICON SWITCHED, ONLINE
DEVICE STATUS FOR CHANNEL PATH 5C
  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0004 .  .  .  .  .  .  .  .  .  +  .  .  .  .  .  .
0100 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
0101 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
0102 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
0103 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
0104 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
0105 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
0106 +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +
18CA HA HA
18CB HA HA
18CC HA HA
18CD HA HA
18CE HA HA
18CF HA HA
18FF .  HA HA
SWITCH DEVICE NUMBER = 0049
DEFINED ENTRY SWITCH - LOGICAL SWITCH ID = 49
ATTACHED ND = 006140.001.MCD.01.0000013G0041
PHYSICAL CHANNEL ID = 0130
FACILITIES SUPPORTED = ZHPF
***** SYMBOL EXPLANATIONS *****
+ ONLINE      @ PATH NOT VALIDATED  - OFFLINE      . DOES NOT EXIST
* PHYSICALLY ONLINE  $ PATH NOT OPERATIONAL
BX DEVICE IS BOXED          SN SUBCHANNEL NOT AVAILABLE
DN DEVICE NOT AVAILABLE    PE SUBCHANNEL IN PERMANENT ERROR
AL DEVICE IS AN ALIAS      UL DEVICE IS AN UNBOUND ALIAS
HA DEVICE IS A HYPERPAV ALIAS  HU HYPERPAV ALIAS UNUSABLE

```

Note that zHPF is listed as a supported facility.

Chapter 12. Testing PPRC secondary devices in subchannel set 1

We implemented the support for PPRC secondary devices in the alternate subchannel set (that is, subchannel set 1) in z/OS V1R10. It is targeted to customers that are constrained by the four-digit device number limit. This function allowed us to define PPRC secondary devices in the alternate subchannel set. This function can only be implemented on System z9 and later CPCs.

In our environment, we have a z990 CPC along with System z9 and System z10 CPCs, so we had to maintain two IODFs: one for the z990 CPC and another for the z9 and z10 CPCs. We defined all the secondary devices in the alternate subchannel set as special type 3390D devices using HCD. In order to leave the special devices that were in subchannel set 1 offline during IPLs, we added the digit 0 (zero) in column 36 of the IODF statement in our IPLPARM LOADxx member.

We observed the following characteristics after defining the secondary devices in subchannel set 1 on our System z9 and System z10 processors:

- Not visible to applications
- Secondary devices had the same device numbers as primary devices
- Five-digit device numbers
- Could not vary online or offline

We defined a HyperSwap session using TPC-R V3.4 and added special devices as PPRC secondaries in the session and performed a HyperSwap to switch the I/O to the special devices.

The following example shows one CopySet we had in our HyperSwap session:

Primary volume:

D M=DEV(08295)

```
IEE174I 11.59.30 DISPLAY M 169
DEVICE 8295 STATUS=ONLINE
CHP          52   53   54   55   56   57   58   59
ENTRY LINK ADDRESS 7541 753D 22B5 A7   03   01   40   31
DEST LINK ADDRESS 7553 754F 2235 26   83   93   91  A1
PATH ONLINE       Y   Y   Y   Y   Y   Y   Y   Y
CHP PHYSICALLY ONLINE Y   Y   Y   Y   Y   Y   Y   Y
PATH OPERATIONAL  Y   Y   Y   Y   Y   Y   Y   Y
MANAGED          N   N   N   N   N   N   N   N
CU NUMBER        8201 8201 8201 8201 8201 8201 8201 8201
MAXIMUM MANAGED CHPID(S) ALLOWED: 0
DESTINATION CU LOGICAL ADDRESS = 02
SCP CU ND        = 002107.900.IBM.75.0000000K8871.0330
SCP TOKEN NED    = 002107.900.IBM.75.0000000K8871.0200
SCP DEVICE NED   = 002107.900.IBM.75.0000000K8871.0295
HYPERPAV ALIASES CONFIGURED = 51
FUNCTIONS ENABLED = MIDAW,HS
```

Secondary volume:

```

|
|      D M=DEV(18295)
|
|      IEE174I 11.59.58 DISPLAY M 262
|      DEVICE 18295  STATUS=SPECIAL
|      CHP              52  53  54  55  56  57  58  59
|      ENTRY LINK ADDRESS 7541 753D 22B5 A7 03 01 40 31
|      DEST LINK ADDRESS 7557 7557 2236 27 A0 82 92 90
|      PATH ONLINE       Y   Y   Y   Y   Y   Y   Y   Y
|      CHP PHYSICALLY ONLINE Y   Y   Y   Y   Y   Y   Y   Y
|      PATH OPERATIONAL  Y   Y   Y   Y   Y   Y   Y   Y
|      MANAGED           N   N   N   N   N   N   N   N
|      CU NUMBER         4600 4600 4600 4600 4600 4600 4600 4600
|      MAXIMUM MANAGED CHPID(S) ALLOWED: 0
|      DESTINATION CU LOGICAL ADDRESS = 06
|      SCP CU ND          = 002107.900.IBM.75.0000000FNXR1.0300
|      SCP TOKEN NED      = 002107.900.IBM.75.0000000FNXR1.0600
|      SCP DEVICE NED     = 002107.900.IBM.75.0000000FNXR1.0605
|      HYPERPAV ALIASES CONFIGURED = 0
|      FUNCTIONS ENABLED = MIDAW,HS
|

```

For more information about this new function, see the following z/OS V1R10.0 information:

- *z/OS HCD Planning*
- *z/OS and z/VM HCM User's Guide*

Chapter 13. Using z/OS Security Server RACF

We tested the following z/OS Security Server RACF functions:

- Reorganizing the RACF databases
- Password reset granularity
- Custom fields
- RACDCERT support for 4096 bit keys

Reorganizing our RACF databases

This topic describes how and why we re-blocked and re-split our RACF databases.

About our RACF database reorganization

The need to re-block the RACF databases in our environment arose from a problem we encountered while using IBM Tivoli zSecure Admin. We received an 0C4 abend when trying to run lists of profiles. The IBM Tivoli support team pointed us to a RACF group in our database with a bad block; however, even after we used BLKUPD to remove this group, the 0C4 abend remained. We ran the IRRUT400 re-block utility against a copy of our RACF database and this cleared up the abend when running IBM Tivoli zSecure Admin against this copy.

The bad block that was causing the problem was in the SYS1.RACFP03 data set that made up our three-data-set RACF database. In order to clean up the bad block and the 0C4 abends in our production environment (as opposed to the copy of the RACF database), the IRRUT400 utility needed to be run against the actual RACF databases. IRRUT400 would also clean up fragmentation of the RACF databases that had occurred over the last several years.

We considered merging the three data sets into one unsplit, single-data-set database. The RACF development team advised us that having three data sets was better for performance when using a coupling facility because it allowed us to have 765 (255 × 3) in-storage ECSA buffers, rather than only 255 for an unsplit database, so we decided not to merge.

We took this opportunity of running the IRRUT400 utility to more evenly balance profiles across the three RACF data sets. The database had been split at U71 and U80, meaning that any profile with a name starting alphabetically prior to U71 was in the first dataset, anything between U71 and U80 was in the second dataset, and anything after U80 was in the third dataset. (For general resource profiles, the class name determines the alphabetic location of the profile. For example, STARTED profile ABC.* falls under S.)

IRRUT200 utilities showed that RACF information was not evenly distributed across the three data sets. On one of our sysplexes, the three data sets (allocated with 50 cylinders each) had the following statistics:

```
SYS1.RACFP01: 61,206 names and 76% full
SYS1.RACFP02: 1 name and 0% full
SYS1.RACFP03: 7,072 names 15% full
```

The other sysplex's RACF data sets (also allocated with 50 cylinders each) had the following statistics:

SYS1.RACFP01: 47,974 names and 63% full

SYS1.RACFP02: 1 name and 0% full

SYS1.RACFP03: 2117 names and 5% full

Testing against copies of the RACF database showed that splitting at ST and U05 would give us a fairly even distribution of profiles across the three data sets. In the future, RACF database manager activity can be analyzed and the split can be based on evenly distributing this activity.

During the reorganization process, we also renamed our databases to avoid having to vary the databases offline while performing the changes. A sysplex-wide IPL was needed to pick up a new ICHRRNG range table (which determines the splits), so a new ICHRDSNT data set names table was picked up at the same time.

Approach to reorganizing our RACF databases

The overall approach for re-blocking, re-splitting, and renaming our RACF databases was as follows:

1. Create a new ICHRRNG RACF range table to split the databases at ST and U05
2. Create a new ICHRDSNT data set names table to rename our RACF databases to the following data set names:
 - Primary: SYS1.RACF01P, SYS1.RACF02P, SYS1.RACF03P
 - Backup: SYS1.RACF01B, SYS1.RACF02B, SYS1.RACF03B
3. Run three IRRUT200 jobs to copy our three primary data sets to temporary data sets
4. Run the IRRUT400 job to copy, re-block and re-split the temporary data sets into the new primary data sets
5. Run three IRRUT200 jobs to copy the new primary data sets to the new backup data sets
6. Update parmlib to pick up the new ICHRRNG and ICHRDSNT tables
7. Perform a sysplex-wide IPL (not a rolling IPL)

Jobs and commands used during our database reorganization

Below is the sequence of commands and subcommands that we used to perform the block update. The block update removed the group XTMEGRP0 from one of the three data sets that make up our RACF database, SYS1.RACFB03. For more information about using the BLKUPD command, see *z/OS Security Server RACF Diagnosis Guide*.

1. Begin the block update process on data set SYS1.RACFB03:

```
BLKUPD 'SYS1.RACFB03'
```
2. Locate the relative block address of group XTMEGRP0:

```
LOCATE ENTRY(XTMEGRP0) CLASS(GROUP)
```

The response shows that the group XTMEGRP0 is located at relative block address 182B000.
3. Read the block at address X'182B000' for the purpose of updating it:

```
READ X'182B000' UPDATE
```
4. Point to the entry containing group XTMEGRP0 within the block:

```
DISPLAY ENTRY(XTMEGRP0) CLASS(GROUP)
```
5. Delete the entry containing the group XTMEGRP0:

- ```

DELETE
6. Update the block to reflect the changes made under the DISPLAY
subcommand:
END SAVE
7. Write the new, updated block back to the RACF database:
END SAVE
8. End the block update command:
END

```

The following jobs are all described in more detail in *z/OS Security Server RACF System Programmer's Guide*.

The following example shows our compile and link job step containing the source code for the new ICHRRNG table:

```

/*****
/*
ASSEMBLE STEP
*****/
//STEP1 EXEC HLASMCL
//C.SYSIN DD *
ICHRRNG CSECT
DC F'3'
DC XL44'00'
DC AL1(1)
B DC XL44'00'
ORG B
DC C'ST'
ORG
DC AL1(2)
C DC XL44'00'
ORG C
DC C'U05'
ORG
DC AL1(3)
END

/*
//L.SYSLMOD DD DSN=SYS1.RACF.TESTLIB,
// DISP=SHR
//L.SYSIN DD *
SETCODE AC(1)
NAME ICHRRNG (R)
/*

```

The following example shows our compile and link job step containing the source code for the new ICHRDSNT table:

```

//STEP1 EXEC HLASMCL
//C.SYSIN DD *
ICHRDSNT CSECT
DC AL1(3)
DC CL44'SYS1.RACF01P'
DC CL44'SYS1.RACF01B'
DC AL1(255)
DC XL1'8C'
DC CL44'SYS1.RACF02P'
DC CL44'SYS1.RACF02B'
DC AL1(255)
DC XL1'8C'
DC CL44'SYS1.RACF03P'
DC CL44'SYS1.RACF03B'
DC AL1(255)
DC XL1'8C'
END

/*

```

```

| //L.SYSLMOD DD DSN=RACFTST.LINKLIB,
| // DISP=SHR
| //L.SYSIN DD *
| SETCODE AC(1)
| NAME ICHRDSNT(R)
| /*

```

The following example shows one of the IRRUT200 job steps that we used in the process to copy a RACF data set:

```

| //STEP EXEC PGM=IRRUT200
| //SYSRACF DD DSN=SYS1.RACFP01,DISP=SHR
| //SYSPRINT DD SYSOUT=*
| //SYSUT1 DD DSN=SYS1.RACFP01.TEMP,DISP=SHR
| //SYSUT2 DD SYSOUT=*
| //SYSIN DD DUMMY
| /*

```

The following example shows the IRRUT400 job step that we used to re-block and re-split the RACF database:

```

| //COPY1 EXEC PGM=IRRUT400,
| // PARM='NOLOCK,FREESPACE(30),ALIGN, TABLE(ICHRRNG)'
| //SYSPRINT DD SYSOUT=*
| //INDD1 DD DSN=SYS1.RACFP01.TEMP,DISP=SHR
| //INDD2 DD DSN=SYS1.RACFP02.TEMP,DISP=SHR
| //INDD3 DD DSN=SYS1.RACFP03.TEMP,DISP=SHR
| //OUTDD1 DD DSN=SYS1.RACF01P,DISP=SHR
| //OUTDD2 DD DSN=SYS1.RACF02P,DISP=SHR
| //OUTDD3 DD DSN=SYS1.RACF03P,DISP=SHR
| //STEPLIB DD DSN=SYS1.RACF.TESTLIB,DISP=SHR
| /*

```

## Results of our RACF database reorganization

After performing this work, IRRUT200 utilities showed that RACF information was more evenly distributed across the three data sets. On one of our sysplexes, the three data sets (allocated with 50 cylinders each) had the following statistics:

```

| SYS1.RACF01P: 26,773 names and 29% full
| SYS1.RACF02P: 22,319 names and 27% full
| SYS1.RACF03P: 18,955 names and 32% full

```

The RACF data sets (also allocated with 50 cylinders each) on the other sysplex had the following statistics:

- SYS1.RACF01P: 17,433 names and 21% full
- SYS1.RACF02P: 18,451 names and 23% full
- SYS1.RACF03P: 13,988 names and 22% full

---

## Password reset granularity

In z/OS V1R10, RACF has new functions to provide additional granularity in the scope of authority for resetting user passwords. Prior to z/OS V1R10, access to the IRR.PASSWORD.RESET resource in the FACILITY class provided the only ability to delegate the ability to reset passwords; however, this did not have sufficient controls to limit the scope for which IDs or groups of IDs had the ability to reset. We tested the new password reset granularity, which adds the controls to enable password resets to be scoped by the owner of the RACF user or by scope of a group tree.

In a similar fashion, prior to z/OS V1R10, access to the IRR.LISTUSER resource in the FACILITY class allowed the LISTUSER command to be issued for all users. We tested the new function that was added to allow the LISTUSER command to be issued on a limited set of users, based on either the owner of the user profile or by the scope of the group of the user profile.

One item to note is that this new function excludes the authority to reset passwords and password phrases of users with any of the SPECIAL, OPERATIONS, AUDITOR, and PROTECTED attributes.

Prior to this z/OS release, these resource profiles controlled access for allowing the ability to reset passwords and to allow the listing of segment data for a user ID. The group name HELPDESK is used as an example in the PERMIT commands that follow.

```
RDEFINE FACILITY IRR.LISTUSER UACC(NONE)
PERMIT IRR.LISTUSER CLASS(FACILITY) ID(HELPDESK) ACCESS(READ)

RDEFINE FACILITY IRR.PASSWORD.RESET UACC(NONE)
PERMIT IRR.PASSWORD.RESET CLASS(FACILITY) ID(HELPDESK) ACCESS(READ)

SETROPTS CLASSACT(FACILITY)
```

With z/OS V1R10, these are the new facility class resources that are used for allowing access granularity for resetting passwords and the LISTUSER command:

```
IRR.PWRESET.OWNER.owner-of-profile
IRR.PWRESET.TREE.owner-of-tree
IRR.PWRESET.EXCLUDE.userid

IRR.LU.OWNER.owner-of-profile
IRR.LU.TREE.owner-of-tree
IRR.LU.EXCLUDE.userid
```

#### Example 1: Implementing the use of the IRR.PWRESET.OWNER resource profile

MVSBASE is listed in RACF as owning user IDs MVSSPT1, MVSSPT2, and MVSSPT3. To permit the HELPDESK group the ability to reset passwords for these IDs and also the ability to list the segments, the following profiles would be defined:

```
RDEFINE FACILITY IRR.PWRESET.OWNER.MVSBASE UACC(NONE) AUDIT(FAILURES(NONE) SUCCESSES(READ))
PERMIT IRR.PWRESET.OWNER.MVSBASE CLASS(FACILITY) USER(HELPDESK) ACCESS(READ)

RDEFINE FACILITY IRR.LU.OWNER.MVSBASE UACC(NONE) AUDIT(FAILURES(NONE) SUCCESSES(READ))
PERMIT IRR.LU.OWNER.MVSBASE CLASS(FACILITY) USER(HELPDESK) ACCESS(READ)
```

#### Example 2: Excluding an ID from allowing password reset or using LISTUSER

Building on Example 1, if you now wish to prevent the help desk access to reset the password or list the user segment information for user ID MVSSPT3, you would create the following profiles:

```
DEFINE FACILITY IRR.PWRESET.EXCLUDE.MVSSPT3 UACC(NONE)
DEFINE FACILITY IRR.LU.EXCLUDE.MVSSPT3 UACC(NONE)
```

#### Example 3: Permitting a group tree structure for allowing password reset or using LISTUSER

Another method of allowing password reset and LISTUSER ability is to use the tree structure. For example, groups APPLSPT, TSOSSPT, and MVSSPT are RACF groups that contain user IDs for a support organization. These groups are all owned by a group named SUPPORT. The following resource profiles would permit user IDs in the HELPDESK group the ability to do password resets and list user segment data for groups and user IDs that are owned by the SUPPORT group:

```
RDEFINE FACILITY IRR.PWRESET.TREE.SUPPORT UACC(NONE) AUDIT(FAILURES(NONE) SUCCESSES(READ))
PERMIT IRR.PWRESET.TREE.SUPPORT CLASS(FACILITY) ACCESS(READ) ID(HELPDESK)

RDEFINE FACILITY IRR.LU.TREE.SUPPORT UACC(NONE) AUDIT(FAILURES(NONE) SUCCESSES(READ))
PERMIT IRR.LU.TREE.SUPPORT CLASS(FACILITY) ACCESS(READ) ID(HELPDESK)
```

For more information on implementing and using the new features for password reset and LISTUSER, see the following documentation:

- *z/OS Security Server RACF Command Language Reference*
- *z/OS Security Server RACF Security Administrator's Guide*
- *z/OS Security Server RACF System Programmer's Guide*

---

## Custom fields in the RACF database

In z/OS V1R10, RACF has implemented an enhancement that provides additional fields in the RACF database which can be customized to hold user security information specific to an installation's needs. The new function includes:

- A RACF class named CFIELD that holds custom field definitions
- A segment in the general resource profile named CFDEF that holds custom field definitions in profiles in the CFIELD class
- Enhancements to the IRRDPI00 command for processing customized field attributes in the CFDEF segments of CFIELD profiles
- A segment in the USER profile named CSDATA that holds the custom field data

The following steps describe the process we followed for implementing this new function on our system. We defined new, custom fields for team name, department number, and component name.

1. Activate the CFIELD class.

```
SETOPTS CLASSACT(CFIELD)
```

2. Define the custom field attributes.

```
RDEF CFIELD USER.CSDATA.TEAMNAME UACC(READ) CFDEF(TYPE(CHAR) MAXLENGTH(10)
FIRST(ALPHANUM) OTHER(ALPHANUM) HELP('TEST TEAM NAME, UP TO 10 CHARS')
MIXED(NO) LISTHEAD('NAME OF TEAM IS '))
```

```
RDEF CFIELD USER.CSDATA.DEPTNUM UACC(READ) CFDEF(TYPE(CHAR) MAXLENGTH(4)
FIRST(ALPHANUM) OTHER(ALPHANUM) HELP('DEPARTMENT NUMBER, UP TO 4 CHARS')
MIXED(NO) LISTHEAD('USERS DEPT NUMBER IS '))
```

```
RDEF CFIELD USER.CSDATA.COMP UACC(READ) CFDEF(TYPE(CHAR) MAXLENGTH(20)
FIRST(ALPHANUM) OTHER(ALPHANUM) HELP('SUPPORTED COMPONENTS, UP TO 20 CHARS')
MIXED(NO) LISTHEAD('COMPONENTS SUPPORTED '))
```

3. Use the IRRDPI00 command to put the customized fields into effect.

IRRDPI00, which loads the RACF dynamic parse table, must be run in order for the newly defined custom fields to take effect. On our sysplex, this runs once when the system is IPLed via a procedure named IRRDPTAB. To run this manually, we need to have the IRRDPI00 command permitted in the TSO AUTH table (IKJTSO00). Once this is in place, we ran the command from a TSO session.

```

ALLOCATE FILE(SYSUT1) DATASET('SYS1.SAMPLIB(IRRDPSDS)') SHR
IRRDPI00 UPDATE

FREE FILE(SYSUT1)

```

- Place customized data into the fields for TSO user IDs.

```

ALTUSER FREDDY CSDATA (TEAMNAME(SEcurity) COMP(RACF) DEPTNUM(D550))
ALTUSER THOMAS CSDATA (TEAMNAME(DATABASE) COMP(DB2) DEPTNUM(D551))
ALTUSER MICHAEL CSDATA (TEAMNAME(MVSBASE) COMP(JES2) DEPTNUM(D550))

```

- List the customized data for two of the user IDs.

```

LU FREDDY CSDATA

CSDATA INFORMATION

NAME OF TEAM IS SECURITY
USERS DEPT NUMBER IS D550
COMPONENTS SUPPORTED RACF

```

```

LU THOMAS CSDATA

CSDATA INFORMATION

NAME OF TEAM IS DATABASE
USERS DEPT NUMBER IS D551
COMPONENTS SUPPORTED DB2

```

One item to note is if you decide to change a CFIELD entry attribute, other than the UACC, there is a series of steps you need to take to accomplish this. We suggest you carefully think through the attributes for the fields before implementing. For example, to change an entry attribute for our component name field, we had to perform the following steps:

- Delete the existing CFIELD definition.

```
RDEL CFIELD USER.CSDATA.COMP
```
- Define the CFIELD again.

```
RDEF CFIELD USER.CSDATA.COMP UACC(READ) CFDEF(TYPE(CHAR) MAXLENGTH(20)
FIRST(ALPHANUM) OTHER(ALPHANUM) HELP('SUPPORTED COMPONENTS, UP TO 20 CHARS')
MIXED(NO) LISTHEAD('COMPONETS SUPPORTED '))
```

- Update the dynamic parse table.

```

ALLOCATE FILE(SYSUT1) DATASET('SYS1.SAMPLIB(IRRDPSDS)') SHR
IRRDPI00 UPDATE

FREE FILE(SYSUT1)

```

- Delete the specific CSDATA entry from the user IDs affected by this change.

```

ALU FREDDY CSDATA(NOCOMP)

ALU FREDDY CSDATA(COMP(RACF))

```

For more information about implementing and using custom fields in the RACF database, see the following documentation:

- z/OS Security Server RACF Command Language Reference*
- z/OS Security Server RACF Security Administrator's Guide*
- z/OS Security Server RACF System Programmer's Guide*

---

## RACDCERT support for 4096-bit keys

In z/OS V1R10, RACF added the ability to create 4096-bit keys using the RACDCERT command. We tested this new function on images both on our System z9 and System z10 CPCs. The following examples demonstrate the commands we used to create the 4096-bit keys.

### Example 1: Create a self-signed CA certificate containing a 4096-bit key

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('RACF CA FOR 4096') O('IBM POK')
OU('IBM 4096 POK CA') L('POUGHKEEPSIE') SP('NEW YORK') C('US')) SIZE(4096)
WITHLABEL('4096CA') NOTBEFORE(DATE(2008-07-15)) NOTAFTER(DATE(2010-07-15))
```

### Example 2: Create a server certificate containing a 4096-bit key and signed by the CA in Example 1

```
RACDCERT ID(WEBSRV) GENCERT SUBJECTSDN(CN('HOST.IBM.COM') O('IBM 4096 SERVER ORG')
OU('IBM 4096 UNIT') L('TIVOLI') SP('NY') C('US')) SIZE(4096)
WITHLABEL('SERVER CERT FOR 4096 Z9') SIGNWITH(CERTAUTH LABEL('4096CA'))
```

### Example 3: Create a client certificate containing a 4096-bit key and signed by the CA in Example 1

```
RACDCERT ID(USER1) GENCERT SUBJECTSDN(CN('USER1') O('IBM PET')
OU('IBM ZPET') L('POUGHKEEPSIE') SP('NEW YORK') C('US')) SIZE(4096)
WITHLABEL('CERTIFICATE FOR USER1') SIGNWITH(CERTAUTH LABEL('4096CA'))
```

In order to test that the above certificates and keys worked, we either added them to a keyring or downloaded them to a browser, as appropriate. Using the IBM HTTP Server and a browser, we then attempted an SSL connection.

For more information about this enhancement, see *z/OS Security Server RACF Command Language Reference*.

---

## Chapter 14. Migrating to and using ICSF HCR7750

This topic describes our migration to and use of ICSF HCR7750. Our experiences include:

- “Migrating to a larger PKDS”
- “Exercising the CPACF function on the System z10 EC platform”

---

### Migrating to a larger PKDS

In order to support 4096 bit keys, the logical record length (LRECL) for the PKDS has changed in FMID HCR7750. We followed the directions under “Migrating to a larger PKDS” in *z/OS Cryptographic Services ICSF System Programmer’s Guide*.

We want to highlight the following important points:

- If you share your PKDS with down-level systems, you must install toleration APAR OA21807 in order to continue to share the PKDS.
- You must migrate the PKDS *prior* to starting ICSF on HCR7750. If you attempt to start ICSF with a PKDS that was created prior to HCR7750, ICSF startup fails with the following error messages:

```
CSFC0286 INCORRECT LRECL FOR PKDS DATASET SYS1.PKDSPLX2.
CSFM406A UNEXPECTED ERROR PROCESSING PKDS HEADER RECORD. FUNCTION = READ,
RETURN CODE = 0000000C, REASON CODE = 00002740.
CSFM407A PKDS SYS1.PKDSPLX2 IS UNAVAILABLE.
```

*z/OS Cryptographic Services ICSF System Programmer’s Guide* does a great job documenting what you need to do to migrate your existing PKDS. It provides step-by-step instructions, including sample JCL, to get you through it. Using the documentation, we were able to migrate our PKDS successfully.

---

### Exercising the CPACF function on the System z10 EC platform

We ran ICSF workloads against the new z10 EC processor in order to exercise the SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 secure hashing available to application programs through the CP Assist for Cryptographic Functions (CPACF). In order to exercise Feature code 3863, CP Assist for Cryptographic Functions (CPACF), which enables clear key DES and TDES instructions and also supports AES 128-bit, AES 192-bit and AES 256-bit encryption/description in the hardware, we ran additional ICSF workloads. All workloads ran without error.



---

## Chapter 15. Using ICSF migration health checks

ICSF provides two migration health checks for the IBM Health Checker for z/OS via APAR OA24221. These migration checks consist of the following:

- ICSFMIG7731\_ICSF\_RETAINED\_RSAKEY — Detection of the existence of retained RSA private keys on a PCICC or PCIXCC/CEX2C cryptographic card.
- ICSFMIG7731\_ICSF\_PKDS\_TO\_4096BIT — Verification that the PKDS size in an ICSF pre-HCR7750 environment is sufficiently allocated to support 4096-bit RSA keys.

Because we were already running with ICSF level HCR7750 when these migration checks came out, the ICSFMIG7731\_ICSF\_PKDS\_TO\_4096BIT check did not apply to us. We had already converted our PKDS to support 4096-bit RSA keys.

We were able to test the ICSFMIG7731\_ICSF\_RETAINED\_RSAKEY health check as we had retained keys stored on both our PCIXCCs and CEX2Cs. Of course, Health Checker for z/OS is needed to take advantage of this health check. We have been using Health Checker in our environment for some time now, so no new setup work was needed to take advantage of these health checks.

For the ICSF migration checks, you will need to install the PTF for APAR OA24221. Once we picked up the code, we did the following:

1. Navigate to the Health Checker panels by typing CK from the SDSF Primary Option Menu. This brought us into the SDSF Health Checker Display menu.
2. We issued a find command for the string ICSF where we saw, under the **NAME** column, ICSFMIG7731\_ICSF\_RETAINED\_RSAKEY. The health check is per system, so you should be able to find an ICSFMIG7731\_ICSF\_RETAINED\_RSAKEY for each system in your sysplex (as noted by scrolling over to the **SysName** column). Note that you can also use the **FILTER** option on the menu bar to filter on system and health check.
3. The migration check is shipped **INACTIVE(ENABLED)**, which you can see by looking in the **State** column. In order to activate the check, type an A in the **NP** column. The **State** column will show **ACTIVE(ENABLED)** once the activation has completed.
4. Type an S next to the ICSF health check and you will be able to browse the output.

For those systems having cards that do not contain retained keys, you will see the following message:

```
CSFH0002I Cryptographic coprocessors were examined and the (ICSF,ICSMIG7731_ICSF_RETAINED_RSAKEY)
check found no apparent RSA key use on this system.
```

For those systems having cards that contain retained keys, you will see the following messages:

```
Coprocessor
Serial Retained key label

94000081 CCA.CRT08.INT.ENC.RKL4.RETAIN
94000081 CCA.CRT08.INT.ENC.RKL8.RETAIN
94000915 CCA.CRT08.INT.ENC.RKL9.RETAIN
93000826 CCA.CRT08.INT.ENC.RKL3.RETAIN
93000826 CCA.CRT08.INT.ENC.RKL6.RETAIN
93000826 CCA.CRT08.INT.ENC.RKL7.RETAIN
```

Low Severity Exception \*

CSFH0003E Cryptographic coprocessors were examined and found to possess retained RSA Keys.

At this point, you know where your retained keys reside and can make appropriate plans to convert them to an alternative key strategy.

Additional information for this new function is available at <ftp://ftp.software.ibm.com/eserver/zseries/zos/icsf/pdf/oa24221.pdf>.

---

## Chapter 16. Using Network Authentication Service (Kerberos)

Integrated Security Services Network Authentication Service for z/OS is the IBM z/OS program based on Kerberos Version 5 and GSS.

---

### Password phrase support

The z/OS Integrated Security Services Network Authentication Service (Kerberos) has been enhanced in z/OS V1R10 to allow for password phrase support. This topic discusses our experiences with enabling the Network Authentication Service for password phrase support.

We used the following documentation to help us with the enablement and exploitation:

- *z/OS Integrated Security Services Network Authentication Service Administration*
- *z/OS Security Server RACF Command Language Reference*

### Enabling password phrase support

We experienced a few problems while enabling a Kerberos principal with a password phrase. We could have avoided all of these problems if we had first read *z/OS Security Server RACF Command Language Reference*.

- **Password phrase must be in single quotation marks**

We did not place the password phrase in single quotation marks when we issued the ALTUSER command. For instance, we issued the following command:  
ALTUSER LDAPSRV PHRASE(LDAPSRVPASSWORD) NOEXPIRED KERB(KERBNAME(LDAP/HOST.IBM.COM))

This resulted in the following messages:

```
ENTER Password Phrase -
ldapsrvpassword
INVALID STRING, ldapsrvpassword
REENTER THIS OPERAND+ -
PHRASE:
```

*z/OS Security Server RACF Command Language Reference* states that the “password phrase you define is a text string of up to 100 characters and must be enclosed in single quotation marks.”

- **RACF user ID must have a password**

The RACF user ID that we began with did not have a password. We issued the following ALTUSER command again, this time with the password phrase enclosed in single quotation marks:

```
ALTUSER LDAPSRV PHRASE('LDAPSRVPASSWORD') NOEXPIRED KERB(KERBNAME(LDAP/HOST.IBM.COM))
```

This resulted in the following messages:

```
PHRASE OPERAND IGNORED
```

Sure enough, the next paragraph in *z/OS Security Server RACF Command Language Reference* states:

Every user that you assign a password phrase must have a password. If you attempt to remove the password from a user with a password phrase, or add a password phrase for a user with no password, the PHRASE operand is ignored and an error message issued.

- **Password phrase must not contain the RACF user ID**

We switched to a RACF user ID that did have a password and issued the following ALTUSER command to add a password phrase:

```
ALTUSER BUEHL1 PHRASE('BUEHL1PASSWORD') NOEXPIRED KERB(KERBNAME(BUEHL1))
```

This resulted in the following message:

```
NEW PASS PHRASE REJECTED BY RACF RULES
```

*z/OS Security Server RACF Command Language Reference* states the following syntax rules for password phrases:

- Maximum length: 100 characters
  - Minimum length:
    - 9 characters, when ICHPWX11 is present and allows the new value
    - 14 characters, when ICHPWX11 is not present
  - Must not contain the user ID (as sequential uppercase or sequential lowercase characters)
  - Must contain at least 2 alphabetic characters (A-Z, a-z)
  - Must contain at least 2 non-alphabetic characters (numerics, punctuation, or special characters)
  - Must not contain more than 2 consecutive characters that are identical
- If a single quotation mark is intended to be part of the password phrase, you must use two single quotation marks together for each single quotation mark.

We then issued the following ALTUSER command and the password phrase was successfully added:

```
ALTUSER BUEHL1 PHRASE('PASSWORD1PHRASE2') NOEXPIRED KERB(KERBNAME(BUEHL1))
```

We issued the LISTUSER command to verify that the Kerberos password phrase was set:

```
LU BUEHL1 NORACF KERB
```

```
USER=BUEHL1
```

```
KERB INFORMATION
```

```

```

```
KERBNAME= bueh11
```

```
KEY FROM= PHRASE
```

```
KEY VERSION= 001
```

```
KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256
```

Notice that the KEY FROM value is PHRASE.

This completed our enablement of a Kerberos principal with a password phrase.

## Verifying the Kerberos principal's password phrase

We used **ftp** to verify that our Kerberos principal's password phrase would work as expected.

We first issued the **kinit** command to obtain the Kerberos credentials:

```
KINIT BUEHL1
```

We replied to the password prompt with our password phrase:

```
password1phrase2
```

We issued the **klist** command to verify that the **kinit** command did work and that we received the Kerberos credentials. We received the following response:

```
Ticket cache: FILE:/var/skrb/creds/krbcred_8863f9e0
Default principal: buehl1@IBM.COM
```

```
Server: krbtgt/IBM.COM@IBM.COM
Valid 2008/07/22-16:14:22 to 2008/07/23-02:14:22
```

We then issued the **ftp** command to confirm the use of the Kerberos credentials that were obtained:

```
ftp host.ibm.com
```

```
IBM FTP CS V1R10
FTP: using TCP/IP
Connecting to: host.ibm.com xx.yy.99.999 port: 21.
220-FTPD1 IBM FTP CS V1R10 at HOST.IBM.COM, 21:32:46 on 2008-07-22.
220 Connection will close if idle for more than 5 minutes.
>>> AUTH GSSAPI
334 Using authentication mechanism GSSAPI
>>> ADAT
235 ADAT=YIGQBgkqhkiG9xIBAgICAG+BgDB+oAMCAQWhAwIBD6JyMHCgAwIBEqJpBGfgh
sn0iRGawk8M/AecyvWuTtuEnE7P4ZET80K77UMdGo4iR2c8Dp+C8EV2QLpK+CkhjwKoZe6
Xwu4M7gKfG7JwoWnI/nRoHcyshWQtveNdL09zSv7u02XRq1KK+b1PxrZf2oiJT1/r
Authentication negotiation succeeded
NAME (host.ibm.com:BUEHL1):
buehl1
>>> USER buehl1
230-User BUEHL1 is an authorized user
230 BUEHL1 is logged on. Working directory is "BUEHL1.".
Command:
get /tmp/ftp.test.file (REPLACE
>>> PORT 9,99,99,999,15,80
200 Port request OK.
>>> RETR /tmp/ftp.test.file
125 Sending data set /tmp/ftp.test.file
250 Transfer completed successfully.
45 bytes transferred in 0.010 seconds. Transfer rate 4.50 Kbytes/sec.
Command:
quit
>>> QUIT
221 Quit command received. Goodbye.
```

The **ftp** transaction worked as expected.

This completed our verification of using a password phrase with a Kerberos principal.



---

## Chapter 17. Using LDAP Server

The 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. There are two versions of the LDAP server available:

1. Integrated Security Services (ISS) Server
2. IBM Tivoli Directory Server (IBM TDS)

We address the following topics related to using LDAP Server:

- “IBM TDS plug-in support”
- “Using TDS differentiation, currency, and certification validation” on page 153
- “Using LDAP server wait for DB2 during startup” on page 154
- “Using TDS password phrase support” on page 156
- “Using LDAP support for RACF custom fields” on page 158
- “Using SHA and MD5 encrypted passwords” on page 160

---

### IBM TDS plug-in support

z/OS IBM Tivoli Directory Server (IBM TDS) has been enhanced in the z/OS V1R10 release to allow for plug-in support. This topic discusses our experiences with enabling IBM TDS for plug-in support.

We used the following documentation to help us with the enablement and exploitation:

- *IBM Tivoli Directory Server Administration and Use for z/OS*
- *IBM Tivoli Directory Server Plug-in Reference for z/OS*

### Enabling IBM TDS plug-in support

We used the plug-in sample described in *IBM Tivoli Directory Server Plug-in Reference for z/OS*. This allowed us to verify the plug-in as well as the shipped sample and documentation. The documentation was very good but we did deviate in a couple places—in creating the data set for the sample plug-in and in enabling the sample plug-in.

#### Creating the data set for the sample plug-in

We created a data set for the plug-in to do the initial testing. Once the testing was complete, we moved the plug-in to an existing permanent data set that was already APF authorized and in the LNKLST. We did this to prevent adding to the list of APF authorized data sets and remove the steplib in the IBM TDS server's started task JCL.

We performed the following steps:

1. We created the plug-in data set GLD.PLUGIN.SIEALNKE from the SYS1.SIEALNKE data set specifications.
2. We then issued the following command to temporarily enable the APF authorization:

```
SETPROG APF,ADD,DSNAME=GLD.PLUGIN.SIEALNKE,VOL=SP0009
```

3. We verified that the GLD.PLUGIN.SIEALNKE data set was APF authorized by issuing the following command:

```
D PROG,APF
```

4. We added the following lines to the IBM TDS server's started task JCL to enable tracing for the plug-in and to add the STEPLIB:

```
// PARMS='-D PLUGIN',

// STEPLIB DD DSN=GLD.PLUGIN.SIEALNKE,DISP=SHR
```

### Enabling the sample plug-in

Our /usr/lpp directory is read only. This meant that the **make** and **c** files needed to be moved into a directory that is writeable. We issued the following commands to move these two files.

```
cp /usr/lpp/ldap/examples/plugin_sample.c /write_dir
cp /usr/lpp/ldap/examples/makefile.plugin /write_dir
```

After we had the files copied, we made the update to **makefile.plugin** as stated in *IBM Tivoli Directory Server Plug-in Reference for z/OS*.

From here on, we did not deviate from the documentation.

## Verifying IBM TDS plug-in support

We checked both the tracing that resulted from our setting the plug-in tracing parm in our IBM TDS startup procedure and the log file written by the plug-in sample. Both were fine.

Overall, we did not find any problems with the documentation, the sample, or the plug-in.

## Cleaning up after our testing

After our testing was complete, we moved the sample to a data set that was already set up with APF authorization and in the LNKLST.

We then did the following to account for moving the sample:

1. We removed the following lines from our IBM TDS started task JCL to remove tracing for the plug-in and to remove the STEPLIB:

```
// PARMS='-D PLUGIN',

// STEPLIB DD DSN=GLD.PLUGIN.SIEALNKE,DISP=SHR
```

2. Our systems are IPLed at least once a week, which removed our temporary APF authorization of the GLD.PLUGIN.SIEALNKE data set. If we had needed to remove the APF authorization prior to our next IPL, we would have used the following command:

```
SETPROG APF,DELETE,DSNAME=GLD.PLUGIN.SIEALNKE,VOL=SP0009
```

After making these changes, we stopped IBM TDS and started it back up to pick up the changes. We then executed the tests again to confirm that the logging was still taking place, and it was.

---

## Using TDS differentiation, currency, and certification validation

IBM Tivoli Directory Server (IBM TDS) was enhanced in the z/OS V1R10 release to enhance the LDAP SASL EXTERNAL bind support to optionally validate that a public key certificate is associated with a RACF user so that the user will be able to perform SDBM operations after a SASL bind.

### Implementing TDS differentiation, currency, and certification validation

Before using this new function, you must utilize RACF skills to use the RACF RACDCERT utility to set up the association of a certificate with a RACF user ID. See *z/OS Security Server RACF Security Administrator's Guide* for detailed steps.

We modified the LDAP server configuration data set to add the new options:

```
sslAuth serverClientAuth
sslCipherSpecs 15104
sslMapCertificate replace fail
sslKeyRingFile LDAPZ104
```

The `sslKeyRingFile LDAPZ104` option identifies the RACF key ring which contains certificates. Use RACF skills and expertise to create the key ring.

RACF skill was also needed to associate RACF key ring LDAPCL1 to user ID *user1* which will be used to perform SDBM operations.

### Verifying certificate validation

We used `rlogin` to log in to the test machine with user ID *user1* and issued the following `ldapsearch` command:

```
ldapsearch -v -L -h host_ip -p host_port \
-Z \
-K LDAPCL1 \
-b "cn=racfsdbm, c=US" \
-s base \
-S EXTERNAL \
"objectclass=*
```

When the LDAP server receives a client request, it first verifies the client certificate. If the certificate is valid and an associated RACF user ID is found, the server creates a distinguished name (DN) based on the user ID and the SDBM suffix. Because we set the **sslMapCertificate replace fail** option in the LDAP server configuration data set, the server then replaced the bind DN that was created from the subject name in the certificate with this mapped DN. The server uses the mapped DN when checking authorization for LDAP operations.

The resulting output was:

```
ldap_ssl_client_init(LDAPCL1, NULL, 0, &failureReasonCode)
ldap_ssl_init(host_ip, host_port, NULL)
filter pattern: objectclass=*
returning: ALL
filter is: (objectclass=*)
dn: cn=racfsdbm, c=US
objectclass: RACFBASE
objectclass: EXTENSIBLEOBJECT
cn: RACFSDBM
1 matches
```

We also used this feature to modify the RACF user password. To do this, we first created a file, pw.file, as follows:

```
dn: racfid=user1,profiletype=USER,cn=racfsdbm, c=US
changetype: modify
add: x
racfpassword: PWD4TEST
```

Then, we again used **rlogin** to log in to the test machine with user ID *user1* and issued the following **ldapmodify** command:

```
ldapmodify -v -h host-ip -p host_port \
-Z \
-K LDAPCL1 \
-S EXTERNAL \
-f pw.file
```

The resulting output was:

```
ldap_ssl_client_init(LDAPCL1, NULL, 0, &failureReasonCode)
ldap_ssl_init(host-ip, host_port, NULL)
add racfpassword:
 PWD4TEST
modifying entry racfid=user1,profiletype=USER,cn=racfsdbm, c=US
modify complete
```

Now the password for RACF user *user1* had been modified.

---

## Using LDAP server wait for DB2 during startup

IBM Tivoli Directory Server (IBM TDS) was enhanced in the z/OS V1R10 release to allow the LDAP server to wait for DB2 during startup. LDAP users can configure LDAP such that it will wait for DB2 to fully initialize before the LDAP server configures DB2-based backends and starts accepting requests.

### Implementing LDAP server wait for DB2 during startup

There are two new global configuration options: **db2StartUpRetryLimit** and **db2StartUpRetryInterval**, which specify, respectively, the maximum number of retry attempts in order to establish the first DB2 connection and the interval (in seconds) before each retry attempt.

**Example:** During startup, if the LDAP server finds that DB2 is not available, the following values cause the LDAP server to try up to three times to establish a connection with DB2, waiting ten seconds before each attempt:

```
db2StartUpRetryLimit 3
db2StartUpRetryInterval 10
```

### Verifying LDAP server wait for DB2 during startup

We tested this function with different configurations. We will describe two scenarios to illustrate how we used this function.

#### Scenario 1

With the following settings in the LDAP server configuration data set, the LDAP server will attempt to retry five times, waiting 10 seconds before each attempt. We stopped DB2 when we executed this test, so the LDAP server failed to start up after five attempts.

We used the following settings for this scenario:

```
db2StartupRetryLimit 5
db2StartupRetryInterval 10
srvStartupError Terminate
```

The following job log shows the execution results:

```
080701 22:06:59.224760 GLD1172E Error code -1 received for ODBC function SQLAllocHandle(DBC).
080701 22:06:59.225774 GLD1171E Native return code -99999, SQL state 58004, SQL message: {DB2 FOR
OS/390}{ODBC DRIVER} SQLSTATE=580
04 ERRLOC=2:170:4;
RRS "IDENTIFY" failed using DB2 system:DBXG,
RC=08 and REASON=00f30002
0080701 22:06:59.225951 GLD1248W Unable to connect to DB2; will attempt retry 1 of 5 in 10 seconds.
080701 22:07:09.002146 GLD1248W Unable to connect to DB2; will attempt retry 2 of 5 in 10 seconds.
080701 22:07:19.003117 GLD1248W Unable to connect to DB2; will attempt retry 3 of 5 in 10 seconds.
080701 22:07:29.002194 GLD1248W Unable to connect to DB2; will attempt retry 4 of 5 in 10 seconds.
080701 22:07:39.002249 GLD1248W Unable to connect to DB2; will attempt retry 5 of 5 in 10 seconds.
080701 22:07:49.004064 GLD1172E Error code -1 received for ODBC function SQLAllocHandle(DBC).
080701 22:07:49.007722 GLD1171E Native return code -99999, SQL state 58004, SQL message: {DB2 FOR
OS/390}{ODBC DRIVER} SQLSTATE=580
04 ERRLOC=2:170:4;
RRS "IDENTIFY" failed using DB2 system:DBXG,
RC=08 and REASON=00f30012
0080701 22:07:49.013423 GLD1249E Unable to start DB2 monitor.
080701 22:07:49.016047 GLD1101A Unable to load the database backends.
080701 22:07:49.017229 GLD1007I LDAP server is stopping.
```

## Scenario 2

With the following settings in the LDAP server configuration data set, the LDAP server will attempt to retry 10 times, waiting 60 seconds before each attempt. We stopped DB2 before executing this test, but we started DB2 after LDAP server had tried to connect to DB2 three times. The LDAP server successfully started after DB2 startup.

We used the following settings for this scenario:

```
db2StartupRetryLimit 10
db2StartupRetryInterval 60
srvStartupError Terminate
```

The following job log shows the execution results:

```
080701 22:26:01.878045 GLD1172E Error code -1 received for ODBC function SQLAllocHandle(DBC).
080701 22:26:01.880209 GLD1171E Native return code -99999, SQL state 58004, SQL message: {DB2 FOR
OS/390}{ODBC DRIVER} SQLSTATE=580
04 ERRLOC=2:170:4;
RRS "IDENTIFY" failed using DB2 system:DBXG,
RC=08 and REASON=00f30002
080701 22:26:01.880285 GLD1248W Unable to connect to DB2; will attempt retry 1 of 10 in 60 seconds.
080701 22:27:01.003747 GLD1248W Unable to connect to DB2; will attempt retry 2 of 10 in 60 seconds.
080701 22:28:01.002192 GLD1248W Unable to connect to DB2; will attempt retry 3 of 10 in 60 seconds.
080701 22:29:03.499690 GLD3340I Found 8 frequent values for column PEID of table
LDAPZ104.DIR_ENTRY. Table cardinality is 38016 and least frequent value has a frequency of 1.
080701 22:29:03.512183 GLD3340I Found 10 frequent values for column AEID of table
LDAPZ104.DIR_DESC. Table cardinality is 76039 and least frequent value has a frequency of 1.
080701 22:29:03.525250 GLD3340I Found 37 frequent values for column ATTR_ID of table
LDAPZ104.DIR_SEARCH. Table cardinality is 1102126 and least frequent value has a frequency of 2.
080701 22:29:03.534577 GLD3340I Found 100 frequent values for column ATTR_ID,VALUE of table
LDAPZ104.DIR_SEARCH. Table cardinality is 1102126 and least frequent value has a frequency of 1.
080701 22:29:04.153300 GLD1004I LDAP server is ready for requests.
080701 22:29:04.247906 GLD1059I Listening for requests on 192.168.25.39 port 4389.
```

With this setting, if we had configured DB2 to start upon system IPL, then we could have also set LDAP to start upon system IPL. Then, even if the system starts LDAP first and DB2 is not yet available, LDAP will try to connect to DB2 several times according to our settings. It gives DB2 time to start, so LDAP can successfully start after DB2 startup.

---

## Using TDS password phrase support

In z/OS V1R9, RACF added support for enveloping a password phrase in a PKCS7 envelope and extended the R\_admin API to allow the envelope to be retrieved. In z/OS V1R10, LDAP can extract the RACF password phrase envelope and return it to the user so that the phrase can be push to other repositories.

In z/OS V1R10, LDAP enhances SDBM support of RACF change logging to support the changes attribute when a change log entry is created for a password phrase change. LDAP also supports using password phrases for a simple bind, both via SDBM and native authentication

## Implementing TDS password phrase support

Using RACF, we enabled the user ID *user1* with a password phrase envelope. We then used the `ldapsearch` command to verify it, as follows:

```
ldapsearch -h host_ip -p host_port \
 -D racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US -w password \
 -b racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US \
 objectclass=*
```

The command response included the following:

```
racfhavepassphraseenvelope=YES
```

We also verified that the user ID *user1* was enabled with a password phrase envelope by issuing the LISTUSER command from a TSO session:

```
lu user1
```

The command response included the following:

```
PHRASE ENVELOPED=YES
```

Next, we again used RACF to enable the LDAP server's startup ID to extract the password phrase envelope. This entailed a series of steps, which are well documented in *z/OS Security Server RACF Security Administrator's Guide*.

## Verifying TDS password phrase support

We used the following `ldapsearch` command to retrieve the password phrase envelope, which is base-64 encoded:

```
ldapsearch -h host_ip -p host_port \
 -D racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US -w password \
 -b racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US \
 objectclass=* racfPassPhraseEnvelope
```

The output looks like the following:

```
dn: racfid=user1,profiletype=USER,sysplex=UTCPLXJ8,o=IBM,c=US
racfhavepassphraseenvelope:: MIIFKQYJKoZIhvcNAQcDoIIFGjCCBRYCAQAxcg8wgcwCAQAwNTAw MQ
swCQYDVQQGEwJ1czEMMAoGA1UEChMDaWJtMRMwEQYDVQQDEwpsQUVUQ0FMRFEQAgEFMA0GCsGSIb3DQ
EBAQUABIGAMYqTZ5gDZ0FAWJzfH5rTgwMbn3hrdHPHrGqQE6/ra3+KwTyHCfuG650DC1Yyt12M602m02
eYgLSp61Mob3SgxBmgdz2GbLCD00tvuvVx++RrgwqYtn8mW/R/hDzc08rRSAC1hYufBmR8CKLICPPVEM
gBz14Qz997MT9/ug5irFwwggQ9BgkqhkiG9w0BBwIwFAYIKoZIhvcNAwECA0Qy84RFd0ugIIEGDhbUL
YppoW8iL4e00FnfismJNrQ51ytY7JX4HdXQwz0VY1YaJxFCBEvpS3e/xmi knVE84bnE/M+DLdDAGVFAP
s67WD0w3gzvEtF0f0ByYq/o+ma064IFZZeKN362P1VTFfkyEBzUz0UWu1a7qDXbFCDBEH00y1g/ss+d
NnnSr3/Sc6en060TN07pHU0xE1MDhEs9YPHiGGWxqIxrFEmMbEeuhNuK/8pjQ9qhbcmWIhZ+Di485GA5
uyQH+IahtGoHdJ+hqj9ajFaPnnDbI6Jl rjMdtzgc1ICeNWJVHwX18Zb/TJ+wH2RvBUeniPYiUnc3qCRp
G0oLzMX72Rz1hh14tPv6BzxhlzXppeVie5HAq9mvyM3A35vktC5L3Lxj8D6SaFmsBPXGCxncE1ocH4T8
U0DPs90fuw/ikKwN09LNVsVWZY2Mp673dSZob3v0KarHx742U5u0Q3J99QWbEKbUQj0cFr45eM1ztDP
tqjL5xs1PkouMzJ7tcdED7jJWpcjXnVJIHb3135UYfsUCIREi8mondH9yMS00pnQdGhSngLkLI6yKTO
```

```
Zn10QiFSTMCi8hxxxDMnkRJNBw8e0tuQAINI44oPm1hN1TIsfYJrk31JrnbM6lq1h01xv+Kq6/ko8DNH
NA3LT8mmi xOgXA6X0ZxyBwj6MuwkAqun9iUqUUQy4Di vGNhDBIyWUfud+any34Kp2BzhizhahA0T/cON
YTsDGSwS0tYsbvbBriRkME+yhhNoXLRqnfBo/741KbatNcFmHSZOI17881bKJmIAoXEBpQcNLQ2xY7fQ
eo9wzC3cDH6BFgyIZMqQOac8Vttjcqz/xP+L0ARcTtgD9a5v1bFZSjJ6FASxrw61KL1Gt0teocMsQkTl
SN4v5phNpds30QRcIzxxIR3etFGP/0x91R4uWijW779GmrFOQ1VcD0+YUfag0m+esw40uLa0Xw5SqBke
UP+zzJE1IKH+5u/gKsArbI+fGTVwSqKi3SmFONgUq7qDqJ2R8FLA6prodSi0k0hL50DIjKfCznw7e5D8
2be0SUpoewiq43IWE0TakaWRymXy5VZNe4bYOW6ZevSL5FMVZDpWcAIPx5pjCSTOfmMhZjcLhqLIKVCV
7xoiAILIgcTp0QMutYGwtkgDunK8pKIDvWeAATvQy/LF1GU5Evd1iie6GonyEpIdfAuUT9ob0jpe/Vde
LA7zYL+/b899iWsLUt3aXsoNzuMmVxB00PtUudDKuvQmf99qPKgAdz7aCmf0Bim8s9cv1qFxiKIaguWZ
d9Dvh0CPAkYx0zvDkE4FOZr+cGCDV99Z0h4xsVAAWSjq3U1066SLJle/Y1ZNbe3o5ZIMjk9SjKudz5n
Eh4/+3SyJhbj1UWnRcW0sWT2P/uVUav7Q=
```

We used long password phrases (that is, where the length of the `-w` option exceeds eight characters) to do LDAP operations, such as the following:

```
ldapsearch -L -h host_ip -p host_port \
-D racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US -w ThisIsLongPassword \
-b racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US \
objectclass=* racfPassPhraseEnvelope
```

We also configured the LDAP server to support native authentication, as follows:

1. Configured the LDAP server configuration data set with the following options:

```
useNativeAuth all
nativeUpdateAllowed on
```

2. Created ldif file `native.ldif` with the following contents:

```
dn: cn=test1, o=Your Company, c=US
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: ibm-nativeAuthentication
cn: test1
sn: test1
ibm-nativeId: user1
```

3. Added the new entry to the LDAP server:

```
ldapadd -h host_ip -p host_port -D "cn=ldap administrator" -w password -f native.ldif
```

We were then able to search the SDBM with long password phrases, as in the following example:

```
ldapsearch -h host_ip -p host_port \
-D "cn=test1, o=Your Company, c=US" -w ThisIsLongPassword \
-b racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US \
objectclass=*
```

In our test environment, we also configured the LDAP server to support changelog, so that when we modify a password phrase, it will be recorded in the changelog GDBM.

We did the following to enable changelog support:

1. Ensured that GDBM was configured on the LDAP server, such as:

```
listen ldap://:pc
database GDBM GLDBGD31/GLDBGD64
databaseDirectory /J80/ldapdata/ldabj807/gdbm
```

2. Created an ldif file `test1mod4.ldif` with the following contents:

```
racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US
racfpassphrase=ThisIsLongNewPasswordForTest
racfattributes=noexpired
```

3. Issued the following command to change the password phrase:

```
ldapmodify -h host_ip -p host_port \
-D racfid=user1,profiletype=user,sysplex=UTCPLXJ8,o=IBM,c=US \
-w OldLongPassword -f test1mod4.ldif
```

We were then able to search the changelog GDBM using a command such as the following:

```
ldapsearch -h j90 -p 7389 -D "cn=ldap administrator" -w secret -b "cn=changelog" "objectclass=*"
```

The command response looks like the following:

```
changeNumber=3,cn=changelog
objectclass=top
objectclass=changeLogEntry
objectclass=ibm-changeLog
changenumber=3
changetype=modify
targetdn=RACFID=USER1,PROFILETYPE=USER,SYSPLEX=UTCPLXJ8,O=IBM,C=US
changes=replace: racfPassPhrase
racfPassPhrase: *ComeAndGetIt*
```

---

## Using LDAP support for RACF custom fields

In z/OS V1R10, RACF provides a new user and group segment, CSDATA, that can contain any number of user-defined fields. In defining custom fields, you specify the name for each field, the type of the field, and other attributes. LDAP can retrieve data from these custom fields.

### Implementing LDAP support for RACF custom fields

We did the following to implement LDAP support for RACF custom fields:

1. Configured custom fields in the RACF database, as follows:

```
rdef cfield user.csdata.cfctest uacc(none) cfdef(type(char) maxlength(10)
first(alphanum) other(alphanum) help('test team name, up to 10 chars')
mixed(no) listhead('name of team is '))

setr classact(cfield)
```

For details about completing this step, see *z/OS Security Server RACF Security Administrator's Guide*.

2. Ran IRRDPI00 CHECK.
3. Issued the following command to add the custom field to the RACF profile for user ID *user1*:

```
altuser user1 csdata (cfctest(zpet))
```

4. Configured LDAP server to recognize the new RACF segment, as follows:

- a. Modified the schema to add an attribute definition consisting of an **attributetypes** value and an **IBMAttributetypes** value, according to the custom field defined in step 1. Created an ldif file, *custom\_field.ldif*, with the following contents:

```
dn: cn=schema
changetype:modify
add: attributetypes
attributetypes: (
 cfctestOID
 NAME 'cfctest'
 DESC 'test team name'
 EQUALITY caseIgnoreMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 SINGLE-VALUE
 USAGE userApplications
```

```

)
IBMAttributetypes: (
 cftestOID
 ACCESS-CLASS sensitive
 RACFFIELD ('USER-CSDATA-CFTEST' 'char')
)

```

- b. Added the new attribute definition to the LDAP server:

```
ldapadd -h host_ip -p host_port -D "cn=ldap administrator" -w password -f custom_field.ldif
```

## Verifying LDAP support for RACF custom fields

We did the following to verify the LDAP support for RACF custom fields:

1. Checked the schema to ensure that the LDAP server recognizes the new attribute by issuing the following command and looking for the cftestOID that was defined in “Implementing LDAP support for RACF custom fields” on page 158:

```
ldapsearch -h host_ip -p host_port -s base -b "cn=schema" "objectclass=subschema"
```

The command response contained the following:

```

...
attributetypes=(cftestOID NAME ('cftest') DESC 'test team name' EQUALITY caseIgnoreMatch
SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.)
...

```

2. Searched the SDBM to ensure that the LDAP server can retrieve the RACF custom field:

```
ldapsearch -h host_ip -p host_port \
 -D racfid=user1,profiletype=user,cn=racfsdbm,c=us -w webadm \
 -b racfid=user1,profiletype=user,cn=racfsdbm,c=us \
 objectclass=*
```

The command response contained the following information (which is what we defined with the ALTUSER command in “Implementing LDAP support for RACF custom fields” on page 158):

```

...
cftest=ZPET
...

```

We also verified that we could modify the custom field as a normal attribute, as follows:

1. Created an ldif file, `modify_cf.ldif`, containing the following:

```

dn: racfid=user1,profiletype=user,cn=racfsdbm,c=us
changetype:modify
replace: cftest
cftest: great_zpet

```

2. Issued the `ldapmodify` command to modify the custom field:

```
ldapmodify -h host_ip -p host_port \
 -D racfid=user1,profiletype=user,cn=racfsdbm,c=us -w password \
 -f modify_cf.ldif
```

3. Issued the `ldapsearch` command to check the new, custom field value:

```
ldapsearch -h host_ip -p host_port \
 -D racfid=user1,profiletype=user,cn=racfsdbm,c=us -w password \
 -b racfid=user1,profiletype=user,cn=racfsdbm,c=us \
 objectclass=* cftest
```

The command response was:

```

racfid=USER1,profiletype=USER,cn=racfsdbm, c=US
cftest=GREAT_ZPET

```

---

## Using SHA and MD5 encrypted passwords

In z/OS V1R10, LDAP provides a new feature to handle and parse tagged **userPassword** attribute values on add, modify and retrieve operations, where the encryption tag is in UTF-8 followed by the binary hash which is base64 encoded. This is a compatibility feature to other LDAP servers, such as LDAP server on Linux for System z.

### Implementing SHA and MD5 encrypted passwords

We modified the LDAP server configuration data set, as follows:

```
pwSearchOutput base64
...
pwEncryption MD5
```

The **pwEncryption** option specifies the encryption method to use when storing the **userPassword** attribute values in the backend of the directory. The value can be: *none*, *crypt*, *MD5*, *SHA*, *DES:keylabel*, *AES:keylabel*, and so on.

The **pwSearchOutput** option specifies the format for MD5 and SHA encrypted **userPassword** attribute values when retrieved on a search operation. The value can be binary or base64.

### Verifying SHA and MD5 encrypted passwords

We then added one entry into that LDAP server with the **userPassword** attribute, such as `userPassword: secret`. We then searched that entry and received the following response:

```
cn=test1, o=test MD5 encryption
sn=test1
userpassword={MD5}Xr4i10zQ4PC0q3aQ0qbuaQ==
```

The **userpassword** value is base-64 encoded with prefix UTF-8 tag (MD5).

We then verified the compatibility with the LDAP server on Linux for System z. To do so, we transferred this entry ldif file to the LDAP server on Linux and then used the **idsldapadd** tool to added this entry into the Linux LDAP server, as follows:

```
idsldapadd -v -D cn=administrator -w password -f ldif_file

ldap_init(localhost, 389)
add objectclass:
 BINARY (20 bytes) organizationalPerson
 BINARY (6 bytes) person
 BINARY (3 bytes) top
add cn:
 BINARY (5 bytes) test1
add sn:
 BINARY (5 bytes) test1
add userpassword:
 BINARY (29 bytes) {MD5}Xr4i10zQ4PC0q3aQ0qbuaQ==
Operation 0 adding new entry cn=test1,o=IBM,c=US
```

Then, we made this user a TAM user:

```
pdadmin sec_master> user import test1 cn=test1,o=IBM,c=US
pdadmin sec_master> user modify test1 account-valid yes
```

We then logged in to TAM using the password *secret*, as follows:

```
pdadmin sec_master> login -a test1 -p secret
```

|  
|

It told us the password *secret* had been successfully uploaded into the Linux LDAP server.



---

## Chapter 18. Using the Cryptographic Services PKI Services

For z/OS V1R10, there were a few updates for PKI Services that we exploited.

---

### IP version 6 support

For z/OS V1R10, PKI Services enabled support for IP version 6 (IPv6). There were a few ways we verified this functionality. Before any testing can be done to verify this support, you must verify that IPv6 is functional in your environment. For more information about IPv6, see *z/OS Communications Server: IPv6 Network and Application Design Guide*.

The pkiserv.tmpl file now has fields for IPv6 support. We verified this function by updating the **AltIPAddr** field in the pkiserv.tmpl file. We then recycled the server for these changes to take effect.

The next step was to verify IPv6 support in the end-user Web pages. We did this by requesting a certificate via the Web-based certificate request form. The **alternate IP address** field now allows for an IPv6 entry to be assigned. During our certificate request, we updated this field with a valid IPv6 address and then continued with our request process. Once the request was completed, we were able to verify that the **alternate IP address** field was valid with the new IPv6 address.

---

### UTF-8 support

For z/OS V1R10, PKI Services allows a certificate that contains two-byte characters in its fields to be the PKI Services' CA cert, as long as those characters can be mapped to code page 1047. This allows a user to create certificates with two-byte characters. To verify this new functionality we generated a new CA certificate that contained two-bit, UTF-8 characters. After this CA was successfully created, we started the PKI server to verify that the CA is valid. The server started error free and the CA certificate was recognized.

---

### 4096-bit key support

There were a few more updates that we verified during our Web-based certificate request process. We verified the new **Distinguished name qualifier** and **Domain component** fields during our certificate process requests by adding the appropriate values to these fields during the request process. See the table in *z/OS Cryptographic Services PKI Services Guide and Reference* for a complete list of variable fields and values.



---

## Chapter 19. Using System SSL

This topic describes our experiences testing the following System SSL functions:

- “Using System SSL CPACF hardware support”
- “Using System SSL 4096-bit hardware support” on page 166
- “Enhancements to gskkyman” on page 167

---

### Using System SSL CPACF hardware support

We tested the exploitation of System SSL using the CP Assist for Cryptographic Function (CPACF) on both z/OS V1R9 with APAR OA22451 and z/OS V1R10. The CPACF is a set of cryptographic functions available on all CPs for z890, z990, z9 BC, z9 EC, and z10 EC hardware. We were specifically interested in the SHA-224, SHA-256, SHA-384 and SHA-512 algorithms. In order to perform this function, you need System SSL APAR OA22451.

In order to see what hardware functions are available to System SSL on the CPACF, we issued the System SSL DISPLAY CRYPTO command, as shown in the following examples.

**Example:** The System SSL DISPLAY CRYPTO command issued from a z/OS image on our z10 EC:

```
-F GSKSRVR,D CRYPTO
GSK01009I Cryptographic status
Algorithm Hardware Software
DES 56 56
3DES 168 168
AES 256 256
RC2 -- 128
RC4 -- 128
RSA Encrypt 4096 4096
RSA Sign 4096 4096
DSS -- 1024
SHA-1 160 160
SHA-2 512 512
```

**Example:** The System SSL DISPLAY CRYPTO command issued from a z/OS image on our z9 EC:

```
-F GSKSRVR,D CRYPTO
GSK01009I Cryptographic status
Algorithm Hardware Software
DES 56 56
3DES 168 168
AES 128 256
RC2 -- 128
RC4 -- 128
RSA Encrypt 4096 4096
RSA Sign 4096 4096
DSS -- 1024
SHA-1 160 160
SHA-2 256 512
```

As you can see from the Hardware column, SHA-2 row, the z10 EC hardware supports up to and including the SHA-512 algorithm. From the z9 EC display, the

hardware supports up to and including the SHA-256 algorithm. It also shows that software will be used on the z9 EC if an algorithm of SHA-384 or SHA-512 is requested.

Using the panels in **gskkyman**, we requested that certain algorithms be used when creating the certificates. Specifically, when creating the certificate request, we were prompted for the signature digest type. (For example, SHA-512 signature digest type).

To verify that the CPACF hardware was exploited, we used an FTP server and FTP client residing on z/OS to create a secure connection. Using each certificate with the various SHA types specified, we attempted to create a secure connection. This was successful. In all cases, we verified that hardware was used when it should be. When the hardware was not available, software was used.

---

## Using System SSL 4096-bit hardware support

We tested the exploitation of 4096 bit keys through System SSL on our z/OS V1R9 systems and z/OS V1R10 systems running on our z10 EC and z9 EC CPCs. In order to perform this function, you need the following:

- System SSL APAR OA22481 (if running on z/OS V1R9)
- ICSF HCR7750 or higher
- Crypto Express2 Coprocessor running Nov. 2007 or later version of Licensed Internal Code (LIC)

In order to verify that the 4096-bit support is available on the hardware, we issued the System SSL DISPLAY CRYPTO command, as shown in the following examples.

**Example:** The System SSL DISPLAY CRYPTO command issued from a z/OS image on our z10 EC:

```
-F GSKSRVR,D CRYPTO
GSK01009I Cryptographic status
Algorithm Hardware Software
DES 56 56
3DES 168 168
AES 256 256
RC2 -- 128
RC4 -- 128
RSA Encrypt 4096 4096
RSA Sign 4096 4096
DSS -- 1024
SHA-1 160 160
SHA-2 512 512
```

**Example:** The System SSL DISPLAY CRYPTO command issued from a z/OS image on our z9 EC:

```
-F GSKSRVR,D CRYPTO
GSK01009I Cryptographic status
Algorithm Hardware Software
DES 56 56
3DES 168 168
AES 128 256
RC2 -- 128
RC4 -- 128
RSA Encrypt 4096 4096
RSA Sign 4096 4096
DSS -- 1024
SHA-1 160 160
SHA-2 256 512
```

If you look in the Hardware column in both displays, you can see that 4096-bit support is available on the hardware for both RSA Encrypt and RSA Sign.

Using the panels in **gskkyman**, we created a certificate with a 4096-bit key. Specifically, when creating the certificate request, we were prompted for the key size. We chose option 3, Certificate with 4096-bit RSA key.

We then attempted to use the certificate through the HTTP server to create a secure connection. This was successful.

---

## Enhancements to gskkyman

In z/OS V1R10, System SSL provides enhancements to the **gskkyman** utility to help you better manage your **gskkyman** environment. In the past, you had to invoke the **gskkyman** panels in order to display information about certificates and tokens. In z/OS V1R10, you can now display this information using the command line interface. Following are some examples of these displays which we used during our testing.

**Example:** To display the certificate with label *Z1 Self Signed* that exists in key database IMWEBZ1.kdb, we issued the following command:

```
gskkyman -dc -k IMWEBZ1.kdb -l 'Z1 Self Signed'
```

The output from the command follows (we had to enter the key database password):

```
Enter database password (press ENTER to cancel):
```

```
Label:
```

```
<Z1 Self Signed>
```

```
Trusted:
```

```
Yes
```

```
Version:
```

```
3
```

```
Serial number:
```

```
484eaac40006f197
```

```
Issuer's Name:
```

```
<CN=host.ibm.com,OU=z1 server unit,O=z1 server org,L=Tivoli,ST>
<=NY,C=US>
```

```
Subject's Name:
```

```
<CN=host.ibm.com,OU=z1 server unit,O=z1 server org,L=Tivoli,ST>
<=NY,C=US>
```

```
Effective Date:
```

```
2008/06/10 16:24:36
```

```
Expiration Date:
```

```
2013/12/01 16:24:36
```

```
Signature algorithm:
```

```
sha512WithRsaEncryption
```

```
Issuer unique ID:
```

```
None
```

```
Subject unique ID:
```

```
None
```

```
Public key algorithm:
```

```
rsaEncryption
```

```
Public key size:
```

```
1024
```

```
Public key:
```

```
30 81 89 02 81 81 00 DC 0E 80 4C 46 F7 78 D2 42
2D 63 FA 32 37 82 10 BA 34 CD 51 4E C7 20 0A 36
69 0A CA 6E 73 4A 39 0A 5E 63 2B 9B AE 5B 06 7F
57 04 4D EE A5 79 93 9A 69 21 41 AA 84 2C C2 A5
ED 9A B6 1E D2 A7 C1 A2 E0 87 43 33 1C 78 8E 90
9E BF C8 82 88 BA FC F7 A9 23 5B 2F F7 1C DF E6
12 50 BF 7B EC 3C 39 E5 12 95 C4 D6 11 5E 6F B8
```

```
|
| 0A B6 7C 07 DE DC FB DA 9B 3F 5E 3C DC 10 47 DF
| CC C5 D1 57 61 96 37 02 03 01 00 01
| Private key:
| Yes
| Default key:
| No
| Certificate extensions:
| 3
```

| An additional enhancement to **gskkyman** is the ability to display the expiration date of the key database via the panels or the command line interface. When opening up a key database in **gskkyman**, there is now an expiration date field which displays the key database expiration date. We tested that this information can also be obtained via command mode, as shown in the following example.

| **Example:** To display information for key database file IMWEBZ1.kdb, we entered the following command:

```
| gskkyman -dk -k IMWEBZ1.kdb
```

| The output from the command follows (we had to enter the key database password):

| Enter database password (press ENTER to cancel):

```
| Database: /keys/IMWEBZ1.kdb
| Expiration: None
| Record length: 5000
```

| For more information about these enhancements, see *z/OS Cryptographic Services System SSL Programming*.

---

## Chapter 20. Using z/OS UNIX System Services

The following topics describe our experiences with z/OS UNIX System Services (z/OS UNIX):

- “z/OS UNIX enhancements in z/OS V1R10”
- “z/OS UNIX tools: Service to display a z/OS UNIX directory” on page 172
- “z/OS UNIX health checks: USS\_PARMLIB\_MOUNTS and USS\_CLIENT\_MOUNTS” on page 179
- “z/OS zFS enhancements” on page 183

---

### z/OS UNIX enhancements in z/OS V1R10

The following topics describe our test experiences with z/OS UNIX enhancements in z/OS V1R10:

- “Password phrase enhancements for rlogin, shell”

#### Password phrase enhancements for rlogin, shell

z/OS V1R10 provides **rlogin** password phrase support for improved system security. Password phrases provide an alternative to traditional passwords that are restricted to eight characters in length. The length of a password phrase can range from 9 to 100 bytes and allows for a larger character set (made up of mixed case letters, numbers, and special characters, including blanks) for authentication. Check with your system administrator to determine if your system and security product supports password phrases.

With z/OS V1R10, the z/OS UNIX shell and utilities allow for either a password or password phrase. A password is recognized when its length ranges from 1 to 8 characters, while a password phrase is recognized when its length ranges from 9 to 100 characters. The shell and utilities do not validate the length of a password or password phrase; they will be validated by the security product. The shell and utilities simply pass the user’s input to the security product for verification. Also, with z/OS V1R10, **rlogin** continues to prompt for a password and accepts a password phrase in the current password string parameter.

**Note:** For details about password phrase support, see *z/OS Security Server RACF General User’s Guide*. For syntax rules for password phrases, see *z/OS Security Server RACF Security Administrator’s Guide*.

We successfully tested the following shell and utility commands and tools:

- **rlogin** — log in to a z/OS UNIX system from a remote system
- **passwd** — change user passwords
- **su** — change the user ID associated with a session

#### Examples of password phrase exploitation with the rlogin command

The following examples demonstrate our testing of password phrases using the **rlogin** command:

1. Password phrase with a length of 14 characters

We used **rlogin** to log in as user *pet001* with a password phrase of *test@12345pass*.

```
|
| # rlogin host_ip -l pet001
| FOMR0226 pet001's Password: <<-- enter password phrase
| 128:/u/pet001 $
```

The remote login was successful.

## 2. Password phrase with a length of 0 characters

We used **rlogin** to log in as user *pet001* with an expired password phrase and then attempted to change the password to a null (character length of zero) password phrase.

```
|
| # rlogin host_ip -l pet001
| FOMR0226 pet001's Password:
| FOMR0229 Password expired
| FOMR0230 Enter new password: <<-- enter null (character length 0) password phrase
| FOMR0231 Re-enter new password:
| FOMR0232 You entered an invalid password
| FOMR0230 Enter new password:
```

A password phrase with a character length of zero is not valid.

## 3. User ID with both a password and password phrase

We used **rlogin** to log in as user *pet001* with an expired password and password phrase, and then changed the password and password phrase.

```
TSO ALU PET001 PASSWORD(HIIBM) PHRASE('HIIBM@123456789')
```

```
|
| # rlogin host_ip -l pet001
|
| FOMR0226 pet001's Password: <<-- entered current password (HIIBM)
| FOMR0229 Password expired
| FOMR0230 Enter new password: <<-- entered new password (goibm)
| FOMR0231 Re-enter new password:
| 128:/u/pet001 $ exit
| Connection closed.
|
| # rlogin host_ip -l pet001
|
| FOMR0226 pet001's Password: <<-- entered current password phrase (HIIBM@123456789)
| FOMR0229 Password expired
| FOMR0230 Enter new password: <<-- enter new password phrase (ibmtest@123pass)
| FOMR0231 Re-enter new password:
| 128:/u/pet001 $
```

The expired password and password phrase were successfully changed.

## Examples of password phrase exploitation with the **passwd** and **su** commands

The following examples demonstrate our testing of password phrases using the **passwd** and **su** commands.

**Examples:** The following examples use password phrases with the **passwd** command.

### 1. Change the login password phrase to a new password phrase with a length of 100 characters (new password phrase:

```
ABCDEF0123456789NOPQRS@#*$!() aCbDTuVwXyZ0123456789!()#@*$
%&abcdefghijklmnop0123456789nopqrs @$!0).
```

```
91:/u/pet001 $ passwd -u pet001
Updating password for user: pet001
Enter current password:
Enter new password:
Enter new password again:
```

The password phrase was successfully changed.

2. Change the login password phrase to a new password phrase with a length of 101 characters, which exceeds the limit of 100 characters (new password phrase: ABCDEFGHIJKLM0123456789NOPQRS0#\*\$!()aCbDTuVwXyZ0123456789!()#@\*\$%&abcdefghijklmnopqrs @\$!09).

```
95:/u/pet001 $ passwd -u pet001
Updating password for user: pet001
Enter current password:
Enter new password:
Enter new password again:
Password was not changed: EDC5121I Invalid argument. (errno2=0x090C02A8)
```

The password phrase was not changed because the new password phrase is too long.

3. Change the login password phrase to a new password phrase that does not meet the password phrase requirements for the installation's security product (new password phrase: TESTPET001IBMPASS, which contains the user ID, *pet001*, within the phrase).

```
126:/u/pet001 $ passwd -u pet001
Updating password for user: pet001
Enter current password:
Enter new password:
Enter new password again:
Password was not changed: EDC5169I Password is invalid. (errno2=0x090C0000)
```

The password phrase was not changed because the new password phrase violates the requirements defined by the security product.

**Examples:** The following examples use password phrases with the **su** command.

1. Change user ID with password phrase with a length of 20 characters, including blanks and special characters (password phrase: test@pass1 phrase\$ F).

```
/u/pet001 $ su - pet002
FSUM5019 Enter the password for pet002:
/u/pet002 $
```

2. Change user ID with password phrase with a length of 101 characters (password phrase: ABCDEFGHIJKLM0123456789NOPQRS0#\*\$!()aCbDTuVwXyZ0123456789!()#@\*\$%&abcdefghijklmnopqrs @\$!09).

```
128:/u/pet001 $ su - pet002
FSUM5019 Enter the password for pet002:
FSUM5033 su: Invalid password entered: reason code = 090C02A7.
```

3. Change to user ID with read access to BPX.SURROGAT and verify that no password phrase is needed. The setup for this test is:

- Define user ID *pet003* with password phrase *hiibm@70712345*.
- Define BPX.SRV.pet003 surrogat class, permitting user ID *pet001*.
- Under bin shell (under userid *pet001*), issue **su -s pet003** and verify that it returns successfully.
- Verify that user identity has been changed to *pet003* by issuing the `whoami` shell command, which should return the new user id, *pet003*.

```
128:/u/oizavi $ su - pet001
FSUM5019 Enter the password for pet001:
128:/u/pet001 $ su -s pet003
1:/u/pet001 $ whoami
PET003
2:/u/pet001 $
```

```

91:/u/pet002 $ su -s pet003
FSUM5027 su: User is not a surrogate of "pet003". <<-- expected
92:/u/pet002 $

```

---

## z/OS UNIX tools: Service to display a z/OS UNIX directory

In z/OS V1R9, the ISPF option 3.17 allowed you to display a z/OS UNIX directory list and process the files in that directory. However, there was no service available to display a directory list from an ISPF application.

z/OS V1R10 provides a new ISPF service, called DIRLIST, that allows an application to display a z/OS UNIX directory list. The caller also has the ability to control the format of the data displayed in the list and to process line commands entered against entries in the list.

*z/OS ISPF Services Guide* describes the syntax for calling the DIRLIST service as a command or as a function call; however, we repeat them here in the interest of helping you understand the examples that will follow.

DIRLIST command invocation format:

```

ISPEXEC DIRLIST PATH(path-var)
 [CONFIRM(YES|NO)]
 [CONFDRD(YES|NO)]
 [PANEL(panel-name)]
 [COLS(column-list)]
 [FIXCOLS(YES|NO)]
 [LCMDS(line-command-list)]

```

DIRLIST call invocation format:

```

CALL ISPLINK ('DIRLIST ', path-var
 ,['YES ' | 'NO ']
 ,['YES ' | 'NO ']
 ,[panel-name]
 ,[column-list]
 ,['YES ' | 'NO ']
 ,[line-command-list];

```

## Examples of DIRLIST using the command invocation format

We tested the DIRLIST command invocation format with both a REXX EXEC and a CLIST.

### Example of calling DIRLIST from REXX using the command invocation format

This example shows the command invocation of the DIRLIST service from a REXX EXEC, which will display the directory list for the /SYSTEM/etc directory. The list will show columns for permissions, file type, and modified date. The line command processor, UDLCMD, will be invoked for line commands /, E, A, VI, and LL. See Figure 70 on page 174 for the resulting display.

```

/* REXX */
/* The invocation DIRLIST will display the directory list for */
/* /SYSTEM/tmp. The list will show columns for Permissions, File */
/* Type and Modified Date. The line command processor UDLCMD will */
/* be invoked for line commands /, E, A, VI and LL. */
TRACE C
ADDRESS ISPEXEC
this_system = mvsvr("sysname")
SAY 'THIS SYSTEM =' THIS_SYSTEM
DIR = '/' || THIS_SYSTEM || '/etc'

```

```

DIRLIST_STR = "DIRLIST PATH(" || | ,
 "DIR"
 ") COLS(PE,10,TY,4,MO,10) LCMDS(UDLCMD,A,VI,LL,E,/) "
ADDRESS ISPEXEC DIRLIST_STR
SAY 'DIRLIST RC=' RC
EXIT

```

### Example of calling DIRLIST from a CLIST using the command invocation format

This example shows the command invocation of the DIRLIST service from a CLIST, which will display the directory list for the /SYSTEM/etc directory. The list will show columns for permissions, file type, and modified date. The line command processor, UDLCMD, will be invoked for line commands /, E, A, VI, and LL.

```

/* CLIST which invokes the DIRLIST service to
/* display the directory list for /SYSTEM/etc.
/* The list will show columns for Permissions, File Type and
/* Modified Date. The line command processor UDLCMD will be
/* be invoked for line commands /, E, A, VI and LL.
PROC 0
/* CONTROL LIST
/* CONTROL LIST CONLIST SYMLIST MSG
/* CONTROL MAIN
CONTROL ASIS
ISPEXEC CONTROL ERRORS RETURN
SET THIS_SYSTEM = &SYSNAME
WRITE &STR(THIS_SYSTEM =) &THIS_SYSTEM
SET DIR = &STR(/)&THIS_SYSTEM&STR(/etc)
ISPEXEC DIRLIST PATH(DIR) COLS(PE,10,TY,4,MO,10) LCMDS(UDLCMD,A,VI,LL,E,/)
SET DIRLISTRC = &LASTCC
WRITE &STR(DIRLIST RC=) &DIRLISTRC
IF &DIRLISTRC > 0 THEN +
 DO
 WRITE &NRSTR(&ZERRSM)
 WRITE &NRSTR(&ZERRLM)
 END
EXIT CODE(0)

```

### DIRLIST results from the REXX example

Figure 70 on page 174 shows the directory list display resulting from the REXX EXEC shown in “Example of calling DIRLIST from REXX using the command invocation format” on page 172.

z/OS UNIX Directory List					
Pathname . : /SYSTEM/etc					
Command	Filename	Message	Permission	Type	Modified
	.		rw-rw-rw-	Dir	2007/10/09
	..		rw-rw-rw-	Dir	2008/03/26
	.envfile		rw-rw-rw-	File	2002/09/13
	.nfsc		rw-r--r--	File	2006/11/29
	alias		rw-----	File	1998/02/02
	auto.master		rw-----	File	2006/02/24
	banner		rw-----	File	2002/12/16
	csh.cshrc		rw-r--r--	File	2004/10/12
	dce		rw-rw-rw-	Dir	1998/02/02
	ehwinfo		rw-r--r--	File	1998/02/02
	envcs2		rw-----	File	2002/05/14
	filedrive.table		rw-r--r--	File	2006/11/13
	ftps.data		rw-rw-rw-	File	1999/01/15
	gateways.tpn		rw-----	File	1999/04/02
	hosts.bak		rw-r--r--	File	1998/02/02
	httpd.conf		rw-r--r--	File	1998/02/02
	httpd.conf.0401		rw-r--r--	File	2004/01/05
	httpd.conf.0401		rw-r--r--	File	2004/01/14

Figure 70. DIRLIST display from a REXX EXEC

## Validating the DIRLIST command line processor

The following scenarios illustrate some of our testing to validate the DIRLIST command line processor. Each scenario presents the initial DIRLIST display panel showing the line command that we executed, trace information from the UDLCMD command line processor, and the resulting panel display.

### Validate the / line command

This scenario illustrates validating the / line command, which invokes the Directory List Actions panel.

z/OS UNIX Directory List					
Pathname . : /SYSTEM/etc					
Command	Filename	Message	Permission	Type	Modified
	.		rw-rw-rw-	Dir	2007/10/09
	..		rw-rw-rw-	Dir	2008/03/26
	.envfile		rw-rw-rw-	File	2002/09/13
	.nfsc		rw-r--r--	File	2006/11/29
	alias		rw-----	File	1998/02/02
/	auto.master		rw-----	File	2006/02/24
	banner		rw-----	File	2002/12/16
	csh.cshrc		rw-r--r--	File	2004/10/12
	dce		rw-rw-rw-	Dir	1998/02/02
	ehwinfo		rw-r--r--	File	1998/02/02
	envcs2		rw-----	File	2002/05/14
	filedrive.table		rw-r--r--	File	2006/11/13

```

UDLCMD entered
Parm 1 is
Parm 2 is
Parm 3 is
Parm 4 is
Parm 5 is
Line command is /
Pathname is /TPN/etc/auto.master
File type is File
Extended attributes ---
Modified date/time 2006/02/24 14:04:17
Created date/time 2002/04/01 15:27:33
File size = 191
File size + 123 = 314
Processing / line command

```

Directory List Actions

File ----- /SYSTEM/etc/auto.master

DIRLIST Action

1. Edit	9. Rename
2. Edit - ASCII	10. Copy Out
3. View	11. Copy In
4. View - ASCII	12. Information
5. Browse	13. Modify Mode Fields
6. New	14. Modify Extended Attrs
7. Directory List	15. Execute command
8. Delete	16. Refadd

Select a choice and press ENTER to process data set action.

### Validate the E line command

This scenario illustrates validating the E line command, which invokes the EDIT Entry panel.

z/OS UNIX Directory List

Pathname . : /SYSTEM/etc

Command	Filename	Message	Permission	Type	Modified
	.		rwxrwxrwx	Dir	2007/10/09
	..		rwxrwxrwx	Dir	2008/03/26
	.envfile		rwxrwxrwx	File	2002/09/13
	.nfsc		rw-r--r--	File	2006/11/29
	alias		rw-----	File	1998/02/02
e	auto.master		rw-----	File	2006/02/24
	banner		rw-----	File	2002/12/16
	csh.cshrc		rwxr-xr-x	File	2004/10/12
	dce		rwxrwxrwx	Dir	1998/02/02
	ehwisinf		rwxr-xr-x	File	1998/02/02

```

UDLCMD entered
Parm 1 is
Parm 2 is
Parm 3 is
Parm 4 is
Parm 5 is

```

```

Line command is E
Pathname is /SYSTEM/etc/auto.master
File type is File
Extended attributes --s-
Modified date/time 2006/02/24 14:04:17
Created date/time 2002/04/01 15:27:33
File size = 191
File size + 123 = 314
Letting ISPF do E line command

```

EDIT Entry Panel

More:    +

Object Name:  
/SYSTEM/etc/auto.master  
\* No workstation connection  
Initial Macro . . .  
Profile Name . . .           (Blank defaults to Type)  
Format Name . . .  
Panel Name . . . .           (Leave blank for default)  
Record Length . . .

Options  
Confirm Cancel/Move/Replace  
EDIT Mixed Mode  
EDIT host file on Workstation  
Preserve VB record length

### Validate the A line command

This scenario illustrates validating the A line command, which is an invalid option, so the line command should fail.

z/OS UNIX Directory List

Pathname . . : /SYSTEM/etc

Command	Filename	Message	Permission	Type	Modified
	.		rw-rw-rw-r	Dir	2007/10/09
	..		rw-rw-rw-r	Dir	2008/03/26
	.envfile		rw-rw-rw-r	File	2002/09/13
	.nfsc		rw-r--r--	File	2006/11/29
	alias		rw-----	File	1998/02/02
a	auto.master		rw-----	File	2006/02/24
	banner		rw-----	File	2002/12/16
	csh.cshrc		rw-r-xr-x	File	2004/10/12
	dce		rw-rw-rw-r	Dir	1998/02/02

```

UDLCMD entered
Parm 1 is
Parm 2 is
Parm 3 is
Parm 4 is
Parm 5 is
Line command is A
Pathname is /TPN/etc/auto.master
File type is File
Extended attributes --s-
Modified date/time 2006/02/24 14:04:17

```

Created date/time 2002/04/01 15:27:33  
 File size = 191  
 File size + 123 = 314  
 \*\*\*

z/OS UNIX Directory List		Line command failed	
Command	Filename	Message	Permission Type Modified
	.		rw-rw-rw- Dir 2007/10/09
	..		rw-rw-rw- Dir 2008/03/26
	.envfile		rw-rw-rw- File 2002/09/13
	.nfsc		rw-r--r-- File 2006/11/29
	alias		rw----- File 1998/02/02
a	auto.master		rw----- File 2006/02/24
	banner		rw----- File 2002/12/16
	csh.cshrc		rw-r-xr-x File 2004/10/12
	dce		rw-rw-rw- Dir 1998/02/02
	ehwisinf		rw-r-xr-x File 1998/02/02
	envcs2		rw----- File 2002/05/14
	filedrive.table		rw-r--r-- File 2006/11/13
	ftps.data		rw-rw-rw- File 1999/01/15

### Validate the VI line command

This scenario illustrates validating the VI line command, which is an invalid option, so the line command should fail.

z/OS UNIX Directory List		Line command failed	
Command	Filename	Message	Permission Type Modified
	.		rw-rw-rw- Dir 2007/10/09
	..		rw-rw-rw- Dir 2008/03/26
	.envfile		rw-rw-rw- File 2002/09/13
	.nfsc		rw-r--r-- File 2006/11/29
	alias		rw----- File 1998/02/02
vi	auto.master		rw----- File 2006/02/24
	banner		rw----- File 2002/12/16
	csh.cshrc		rw-r-xr-x File 2004/10/12
	dce		rw-rw-rw- Dir 1998/02/02
	ehwisinf		rw-r-xr-x File 1998/02/02

UDLCMD entered  
 Parm 1 is  
 Parm 2 is  
 Parm 3 is  
 Parm 4 is  
 Parm 5 is  
 Line command is VI  
 Pathname is /SYSTEM/etc/auto.master  
 File type is File  
 Extended attributes --s-  
 Modified date/time 2006/02/24 14:04:17  
 Created date/time 2002/04/01 15:27:33  
 File size = 191

```
File size + 123 = 314
shell cmd is vi /SYSTEM/etc/auto.master
VI rtc = 0

```

z/OS UNIX Directory List		Line command failed	
Command	Filename	Message	Permission Type Modified
	.		rw-rw-rw- Dir 2007/10/09
	..		rw-rw-rw- Dir 2008/03/26
	.envfile		rw-rw-rw- File 2002/09/13
	.nfsc		rw-r--r-- File 2006/11/29
	alias		rw----- File 1998/02/02
vi	auto.master		rw----- File 2006/02/24
	banner		rw----- File 2002/12/16
	csh.cshrc		rw-r--r-- File 2004/10/12
	dce		rw-rw-rw- Dir 1998/02/02
	ehwisinf		rw-r--r-- File 1998/02/02
	envcs2		rw----- File 2002/05/14
	filedrive.table		rw-r--r-- File 2006/11/13

### Validate the LL line command

This scenario illustrates validating the LL line command, which executes the ls -l shell command.

z/OS UNIX Directory List			
Command	Filename	Message	Permission Type Modified
	.		rw-rw-rw- Dir 2007/10/09
	..		rw-rw-rw- Dir 2008/03/26
	.envfile		rw-rw-rw- File 2002/09/13
	.nfsc		rw-r--r-- File 2006/11/29
	alias		rw----- File 1998/02/02
ll	auto.master		rw----- File 2006/02/24
	banner		rw----- File 2002/12/16
	csh.cshrc		rw-r--r-- File 2004/10/12
	dce		rw-rw-rw- Dir 1998/02/02
	ehwisinf		rw-r--r-- File 1998/02/02
	envcs2		rw----- File 2002/05/14
	filedrive.table		rw-r--r-- File 2006/11/13
	ftps.data		rw-rw-rw- File 1999/01/15

```
UDLCMD entered
Parm 1 is
Parm 2 is
Parm 3 is
Parm 4 is
Parm 5 is
Line command is LL
Pathname is /SYSTEM/etc/auto.master
File type is File
Extended attributes --s-
Modified date/time 2006/02/24 14:04:17
Created date/time 2002/04/01 15:27:33
```

```

File size = 191
File size + 123 = 314
shell cmd is ls -l /SYSTEM/etc/auto.master
rtc = 0
out.0 = 1
-rw----- 1 LORAIN0 sys1 191 Feb 24 2006 /SYSTEM/etc/auto.master
err.0 = 0

```

z/OS UNIX Directory List					
Pathname . : /SYSTEM/etc					
Command	Filename	Message	Permission	Type	Modified
	.		rwxrwxrwx	Dir	2007/10/09
	..		rwxrwxrwx	Dir	2008/03/26
	.envfile		rwxrwxrwx	File	2002/09/13
	.nfsc		rw-r--r--	File	2006/11/29
	alias		rw-----	File	1998/02/02
	auto.master	LL done - ok!	rw-----	File	2006/02/24
	banner		rw-----	File	2002/12/16
	csh.cshrc		rwxr-xr-x	File	2004/10/12

## z/OS UNIX health checks: USS\_PARMLIB\_MOUNTS and USS\_CLIENT\_MOUNTS

z/OS V1R10 provides two new z/OS UNIX health checks:

- **USS\_PARMLIB\_MOUNTS** — addresses certain mount failures with file systems that are identified in the BPXPRMxx member used at initialization
- **USS\_CLIENT\_MOUNTS** — identifies file systems in a shared file system that are being accessed using function shipping when they are identified as being locally accessible

These checks are run on individual systems.

For more information about the health checks, see *IBM Health Checker for z/OS: User's Guide*.

### USS\_PARMLIB\_MOUNTS check

The **USS\_PARMLIB\_MOUNTS** check generates an exception when a file system in a **ROOT** or **MOUNT** statement specified in the BPXPRMxx parmlib members used during initialization fails to mount. During file system initialization (IPL or OMVS RESTART), one or more BPXPRMxx MOUNT commands might fail. The appropriate BPXF\* error message is issued, but might scroll off the operator console and the failure might not be noticed or addressed in a timely manner by the operations staff. The impact of such failures can increase with time.

This health check will provide a timely mechanism to highlight missing file systems at the completion of OMVS initialization. This check will also re-run automatically after the failing file system is successfully mounted and after issuing the **MODIFY BPXOINIT,FILESYS=REINIT** system command.

These messages can occur in the **USS\_PARMLIB\_MOUNTS** check:

```
BPXH003I
```

```
|
| BPXH059I (new)
| BPXH061E (new)
| BPXH062I (new, no exceptions)
```

Note that, at this time, mount failures due to duplicate MOUNT statements with different attributes, such as mount point or mode, will not be flagged by this check.

**Example:** These mounts in the BPXPRM00 parmlib member, which have the same file system name but different mount points, and the associated mount failure message, will not be identified by this health check:

```
|
| MOUNT statements in BPXPRM00:
| MOUNT FILESYSTEM('OMVSSPN.LOCAL.FS') TYPE(ZFS)
| MODE(RDWR) MOUNTPOINT('/local')
|
| MOUNT FILESYSTEM('OMVSSPN.LOCAL.FS') TYPE(ZFS)
| MODE(RDWR) MOUNTPOINT('/localamzfs')
```

Mount failure message not identified by the health check:

```
|
| BPXF237I FILE SYSTEM OMVSSPN.LOCAL.FS 477
| WAS ALREADY MOUNTED ON PATHNAME
| /local.
```

**Example:** Here is an example of the USS\_PARMLIB\_MOUNTS health check output from our Z4 system where no problems are found:

```
|
| CHECK(IBMUSS, USS_PARMLIB_MOUNTS)
| START TIME: 08/04/2008 09:14:40.906602
| CHECK DATE: 20070809 CHECK SEVERITY: HIGH
```

**BPXH003I** z/OS UNIX System Services was initialized using MVS=(00,Z4), where each 2-character item is a BPXPRMxx suffix.

**BPXH062I** All file systems specified by ROOT and MOUNT statements in the BPXPRMxx parmlib members used to configure z/OS UNIX System Services are mounted.

```
|
| END TIME: 08/04/2008 09:14:40.909717 STATUS: SUCCESSFUL
```

**Example:** Here is an example of the mount failures and the USS\_PARMLIB\_MOUNTS health check output when the following mount failures were found:

```
|
| BPXF008I FILE SYSTEM OMVSSPT.NONEXIST.READ.HFS 646
| WAS NOT MOUNTED.
| THE MOUNT POINT SPECIFIED IN BPXPRM00 DOES NOT EXIST.
| BPXF008I FILE SYSTEM OMVSSPT.NONEXIST.WRITE.HFS 647
| WAS NOT MOUNTED.
| THE MOUNT POINT SPECIFIED IN BPXPRM00 DOES NOT EXIST.
| BPXF008I FILE SYSTEM OMVSSPT.NONEXIST.READ.ZFS 648
| WAS NOT MOUNTED.
| THE MOUNT POINT SPECIFIED IN BPXPRM00 DOES NOT EXIST.
| BPXF008I FILE SYSTEM OMVSSPT.NONEXIST.WRITE.ZFS 649
| WAS NOT MOUNTED.
| THE MOUNT POINT SPECIFIED IN BPXPRM00 DOES NOT EXIST.
```

Health check USS\_CLIENT\_MOUNTS now shows:

```
|
| CHECK(IBMUSS, USS_PARMLIB_MOUNTS)
| START TIME: 08/04/2008 11:31:36.702284
| CHECK DATE: 20070809 CHECK SEVERITY: HIGH
```

**BPXH003I** z/OS UNIX System Services was initialized using OMVS=(00,Z4), where each 2-character item is a BPXPRMxx suffix.

**BPXH059I** The following file systems are not active:

-----  
File System: OMVSSPT.NONEXIST.READ.HFS  
Parmlib Member: BPXPRM00  
Path: /nonexistreadhfs  
Return Code: 00000081  
Reason Code: 1288005C

File System: OMVSSPT.NONEXIST.WRITE.HFS  
Parmlib Member: BPXPRM00  
Path: /nonexistwritehfs  
Return Code: 00000081  
Reason Code: 1288005C

File System: OMVSSPT.NONEXIST.READ.ZFS  
Parmlib Member: BPXPRM00  
Path: /nonexistreadzfs  
Return Code: 00000081  
Reason Code: 1288005C

File System: OMVSSPT.NONEXIST.WRITE.ZFS  
Parmlib Member: BPXPRM00  
Path: /nonexistwritezfs  
Return Code: 00000081  
Reason Code: 1288005C

\* High Severity Exception \*

**BPXH061E** One or more file systems specified in the BPXPRMxx parmlib members are not mounted.

Explanation: During the USS\_PARMLIB\_MOUNTS check, one or more file systems that were specified in the BPXPRMxx parmlib members used for initialization were found not to be active.

System Action: The system continues processing.

Operator Response: Report this problem to the system programmer.

System Programmer Response: Review the return code and reason code in the summary message and determine why the file systems are not active. Correct the problem using documented procedures. After the problem has been corrected, mount each file system using one of the following procedures:

Ask a superuser to enter the corrected information using the TSO/E MOUNT command or the mount shell command. If the statement in error was the ROOT statement, specify '/' as the mount point.

Alternatively, the SET OMVS=(xx) system command can be issued, where "xx" is the last two characters of a BPXPRMxx parmlib member that contains the MOUNT statement(s) to re-process.

Problem Determination: See BPXH059I in the message buffer.

Source: z/OS UNIX System Services

Reference Documentation:

For information on modifying BPXPRMxx see:  
Customizing z/OS UNIX in z/OS UNIX System Services Planning  
BPXPRMxx in z/OS MVS Initialization and Tuning Reference

For information on using the DISPLAY OMVS, MF command see:  
DISPLAY in MVS System Command Reference in z/OS MVS System  
Commands

Automation: N/A

Check Reason: BPXPRMxx parmlib mount failures can cause outages if not  
handled in a timely manner.

END TIME: 08/04/2008 11:31:36.709648 STATUS: EXCEPTION-HIGH

## USS\_CLIENT\_MOUNTS check

The USS\_CLIENT\_MOUNTS check is used to verify that mounted file systems that  
can be locally accessed are not function shipping (shared file system check). In a  
shared file system configuration, a file system that should be accessible through a  
local mount might be function-shipping to the remote file system server.  
Performance is not optimal in this situation. This health check provides a specific  
notification of this situation.

These messages can occur in the USS\_CLIENT\_MOUNTS check:

BPXH003I  
BPXH063I (new)  
BPXH065E (new)  
BPXH066I (new, no exceptions)

**Example:** Here is an example of the USS\_CLIENT\_MOUNTS health check output  
from our Z4 system where no problems are found:

```
CHECK(IBMUSS,USS_CLIENT_MOUNTS)
START TIME: 06/09/2008 11:25:38.422855
CHECK DATE: 20070809 CHECK SEVERITY: MEDIUM
```

**BPXH003I** z/OS UNIX System Services was initialized using OMVS=(00,Z4), where  
each 2-character item is a BPXPRMxx suffix.

**BPXH066I** All file systems that can be locally mounted in the shared file  
system configuration are accessed locally.

END TIME: 06/09/2008 11:25:38.423232 STATUS: SUCCESSFUL

**Example:** Here is an example of the USS\_CLIENT\_MOUNTS health check output  
from our Z1 system where a problem was found. Access to the DASD where these  
file systems resided was not active.

```
CHECK(IBMUSS,USS_CLIENT_MOUNTS)
START TIME: 06/09/2008 12:11:25.771862
CHECK DATE: 20070809 CHECK SEVERITY: MEDIUM
```

**BPXH003I** z/OS UNIX System Services was initialized using OMVS=(00,Z1),  
where each 2-character item is a BPXPRMxx suffix.

**BPXH063I** The following file systems are available through a remote owner  
system:

```

File System: D10.V1R10.USSE1011.R0.ZFS
Mount Mode: READ
PFS Type: ZFS
File System: D10.V1R10.USSE1011.R0.HFS
Mount Mode: READ
PFS Type: HFS
```



## Sysplex root migration from HFS to zFS

In z/OS V1R10, z/OS UNIX System Services provides the ability to migrate a SYSPLEX ROOT mount point to a new file system. We used this function to migrate our sysplex root HFS file system mounted at / to a zFS file system without having to take down our sysplex.

Before proceeding, however, the new sysplex root must be created and populated with the mount points and symbolic links that the active sysplex root contains. We accomplished this by performing the following steps:

1. Allocated a new zFS file system
2. Mounted the new zFS file system at a temporary mount point
3. Used the **copytree** utility to clone the contents of the active root to the new file system:  

```
copytree -os / /tempdir
```
4. Verified that the contents of the new sysplex root was the same as the active sysplex root
5. Changed the permission bits for the new file system base directory from 700 to 755, which is how our active sysplex root runs. We needed to make this change because new file systems created on our system have permission bits of 770 and the **copytree** utility does not change bits for the base directory of the target file system. We made this change using the **chmod** command.
6. Unmounted the new file system before using the new command to switch roots.

**Note:** Another method to create a cloned root is to use the **pax** command. After you have mounted the newly created file system at the temporary directory, you would enter the following command:

```
pax -wr -pe -XCM ./ /tempdir
```

We then proceeded to use the new command to switch the root to the new file system:

```
F OMVS,NEWROOT=NEW.SYSPLEX.ROOT.ZFS,COND=YES|NO
```

Here are some examples of what we attempted and the results we observed.

**Attempt 1:** Issued the following command to switch, and it failed due to the root being mounted as RDWR. The mode of the active sysplex root must be READ. We then used the ISHELL panel to remount the root file system in READ mode.

```
F OMVS,NEWROOT=OMVSSPN.SYSPLEX.ROOT2.HFS,COND=YES
BPXF243E F OMVS,NEWROOT COMMAND HAS BEEN TERMINATED DUE TO THE 243
FOLLOWING REASON(S):
CURRENT SYSPLEX ROOT FILE SYSTEM IS MOUNTED RDWR
CURRENT SYSPLEX ROOT HAS FUNCTION SHIPPING CLIENTS
```

**Attempt 2:** Issued the same command again to switch the root, which failed again with return code 00000072, reason code 124F0626. This reason code indicates that there is activity on the root mount point.

```
BPXF245I LIST OF ACTIVITIES IN THE CURRENT SYSPLEX ROOT FILESYSTEM: 730
PATH NAME: /
BPXF245I LIST OF ACTIVITIES IN THE CURRENT SYSPLEX ROOT FILESYSTEM: 333
PATH NAME: /
BPXF245I LIST OF ACTIVITIES IN THE CURRENT SYSPLEX ROOT FILESYSTEM: 733
PATH NAME: /
```

```
|
| BPXF244E F OMVS,NEWROOT COMMAND FAILED. 334
| RETURN CODE = 00000072, REASON CODE = 124F0626
| IEF196I IGD104I OMVSSPT.SYSPLEX.ROOT2.HFS RETAINED,
| IEF196I DDNAME=SYS00155
|
```

**Attempt 3:** Issued the switch command again, this time using COND=NO, which processes the root switch unconditionally. While this affected active connections to the root that were previously identified, the switch was successful.

```
|
| F OMVS,NEWROOT=OMVSSPT.SYSPLEX.ROOT2.HFS,COND=NO
| IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00156
| IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00033
| IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00106
| IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00668
| BPXF245I LIST OF ACTIVITIES IN THE CURRENT SYSPLEX ROOT FILESYSTEM: 258
| PATH NAME: /
| BPXF245I LIST OF ACTIVITIES IN THE CURRENT SYSPLEX ROOT FILESYSTEM: 760
| PATH NAME: /
| BPXF245I LIST OF ACTIVITIES IN THE CURRENT SYSPLEX ROOT FILESYSTEM: 338
| PATH NAME: /
| BPXF245I LIST OF ACTIVITIES IN THE CURRENT SYSPLEX ROOT FILESYSTEM: 813
| PATH NAME: /
| BPXF246I THE SYSPLEX ROOT FILE SYSTEM MIGRATION PROCESSING 339
| COMPLETED SUCCESSFULLY.
|
```

Note the following points about this new function to switch to a new root file system:

- You need to ensure the new root file system contains all mount points that are active in the current root. While a directory in the sysplex root can be a mount point (when a file system is mounted on a directory), only those directories that are active mount points will be compared for ensuring that they exist in the new root. This is also true for symbolic links, as they also need to exist in the new root file system, or the switch attempt will fail.
- The UID, GID, and permissions for the new file system root directory need to be the same or the switch will fail with message:  
BPXF243E , NEW SYSPLEX ROOT UID, GID OR MODE IS INVALID
- Update your BPXPRMxx member with the new sysplex root file system after the switch is completed.
- The switch function can also be used to switch a zFS sysplex root to zFS or any combination of HFS/zFS file system types.
- The current SYSPLEX ROOT must be mounted as READ only.
- This new command is only supported for a sysplex where HFS sharing is being used (BPXPRMxx contains the SYSPLEX(YES) parameter).

For more information about this new function for migrating the sysplex root, see *z/OS Migration*.

## zFS format authorization

It is a two-step process to create your zFS aggregates via JCL. First, you define the aggregate and then you format it. In order to format the aggregate, you use the IOEAGFMT utility. Until now, in order to run IOEAGFMT, you needed either a UID of 0 or READ authority to the SUPERUSER.FILESYS.PFSCTL resource profile in the UNIXPRIV class. On the other hand, all you needed to create an HFS was ALTER authority to the data set profile.

Starting with z/OS V1R9, zFS will work just like HFS and allow users with ALTER authority to the data set profile to run the IOEAGFMT utility against that data set. This enhancement is also rolled back to z/OS releases V1R7 and V1R8.

Another zFS utility, IOEAGSLV, is also included in this change. Users with UPDATE authority to the data set profile now will be able to run this utility.

Note that, in actuality, all that is required to run the IOEAGFMT utility is UPDATE authority to the data set profile. However, since ALTER authority is required to define a VSAM linear data set and to set the zFS bit in the catalog, we will say that overall you need ALTER authority to create a zFS, just like you do to create an HFS.

There are other ways to create zFS aggregates as well. For instance, you can use the ISHELL panels or the **zfsadm** shell commands. Both of these methods would then use the zFS APIs to define and format the zFS aggregate. Note that, as of today, the zFS APIs are not changed as part of this enhancement. In order to format a zFS aggregate using these APIs, you still need a UID of 0 or READ authority to the SUPERUSER.FILESYS.PFSCCTL profile in the UNIXPRIV class. Our team opened Marketing Requirement MR0608072541 to request that the zFS APIs support this change as well.

## Aggregate full message from zFS

One of the zFS RAS enhancements in z/OS V1R9 is a new file system full message. We wanted to mention it here so that you are not surprised by it since it is displayed regardless of the aggregate full option you configured in your environment. The message is:

```
IOEZ00551I Aggregate AggrName ran out of space.
```

It will be issued no more than every 10 minutes for the same aggregate. If dynamic aggregate grow is on, zFS will attempt to grow it.

## zFS AUDITID

The new zFS AUDITID support changes the AUDITID supplied for zFS files so that it is unique per file. It also allows the zFS AUDITID to map back to the original path name of the file. This support, already available with HFS, brings zFS functionality closer to HFS functionality.

This is how the zFS AUDITFID is determined today:

```

 4 4 8
zFS AUDITID +-----+-----+-----+
 | inode | uniq | 0 |
zFS AUDITID (old) +-----+-----+-----+
```

This is how the HFS AUDITFID is determined today:

```

 1 6 3 4 2
HFS AUDITID +--+-----+-----+-----+-----+
 |01| volser | TTR | inode |uniq|
HFS AUDITID +--+-----+-----+-----+-----+
```

Here is how the zFS AUDITFID is determined after the new support:

```

 6 4 4 2
zFS AUDITID +-----+-----+-----+-----+
 | volser | CCHH | inode |uniq|
zFS AUDITID (new) +-----+-----+-----+-----+
 \-----/
 V
 AUDITFID
```

The AUDITFID section is also called the aggregate identification piece. If this aggregate is mounted, the 10-byte AUDITFID can be displayed with the following command:

```
zfsadm aggrinfo -aggregate aggrname -long
```

There are a few different methods that you can use to set the new AUDITID:

- **During aggregate mount**

There is a new option for the IOEFSPRM member:

```
convert_auditfid=on | off
```

The default value is off. If it is on and an aggregate is mounted RDWR, then the AUDITID is automatically changed to the new format upon mount.

You can dynamically turn this option on or off with the **zfsadm config -convert\_auditfid [on | off]** command.

You can display the current setting with the **zfsadm config -convert\_auditfid** command.

- **Using IOEAGFMT and the zfsadm format command**

A new **-newauditfid** option is available for IOEAGFMT and the **zfsadm format** command. If this option is specified, the aggregate will be formatted with the new AUDITFID.

- **Using zfsadm setauditfid -aggregate *aggrname* [-force | -old]**

If the aggregate already contains the new form of the zFS AUDITFID and you want to change it to a new zFS AUDITFID, you must specify the **-force** option. The zFS AUDITFID will be based on the VOLSER and the CCHH of the first extent unless you specify **-old**. In that case, the zFS AUDITFID will be set to binary zeros.

For more information, see *z/OS Distributed File Service zSeries File System Administration*.

## **zFS read-only mount recovery**

We implemented and tested a new zFS feature for z/OS V1R9 which deals with file system recovery and the ability for zFS to automatically manage recovery after a system failure. This feature handles the recovery of zFS logs for when a zFS aggregate (file system) that had been mounted read/write at the time of a system failure, is attempted to be remounted in read-only mode. Prior to this release, if this was to occur, the attempted mount would fail with zFS reason code EFxx6271, which means zFS was not able to run log recovery since the aggregate needed to be mounted in read/write mode.

### **Implementing read-only mount recovery**

There is a new parameter you can place in the zFS configuration settings in parmlib member IOEFSPRM:

```
romount_recovery on
```

This feature can also be implemented dynamically using the **zfsadm config** command as well so you can activate this feature without having to recycle zFS. The command would be:

```
zfsadm config -romount_recovery on
```

The following message is issued when zFS attempts to perform log recovery for an aggregate but, because it is mounted read-only, it is not able to accomplish it. Prior to z/OS V1R9, the only way to recover the zFS filesystems was to mount in read/write mode.

```
BPMX002I FILE SYSTEM OMVSSPT.LARGEZFS.ZFS WAS 955
NOT MOUNTED. RETURN CODE = 0000008D, REASON CODE = EF096271
```

### Testing read-only mount recovery

For our testing of this new function, we used one of our test workloads and a zFS file system mounted as read/write. This workload performs reads and writes to directories in the file system. We then coded this file system to be mounted as read-only in the system's BPMXPRMxx member. With the workload active, we crashed the system hard by issuing an XCF command to vary it out of the sysplex.

Upon re-IPL of the system, when zFS was started, the file system was temporarily placed in read/write mode "under the covers" so it could perform the log recovery. Once the recovery was complete, the mode for the file system was read-only. Prior to this new function, the file system would not have mounted due to needed log recovery and zFS would have issued the BPMX002I message.

The following messages are an example of what you will see in the system logs after an aggregate has been automatically recovered via this new setting:

```
IOEZ00397I recovery statistics for OMVSSPN.LARGEZFS.ZFS:
IOEZ00391I Elapsed time was 39260 ms
IOEZ00392I 1926 log pages recovered consisting of 263274 records
IOEZ00393I Modified 2174 data blocks
IOEZ00394I 233885 redo-data records, 0 redo-fill records
IOEZ00395I 2 undo-data records, 0 undo-fill records
IOEZ00396I 0 not written blocks
```

---

## Chapter 21. Migrating to CICS Transaction Server for z/OS, Version 3.2

This topic describes our experiences during the migration from CICS Transaction Server for z/OS (CICS TS) Version 3.1 to Version 3.2 in our Parallel Sysplex environment. It is not intended to be a step-by-step procedure because each migration is unique due to the CICS configuration and features installed.

During the migration to CICS TS 3.2, we used the following documentation:

- *Program Directory for CICS Transaction Server for z/OS V3.2.0*
- *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1*
- *CICS Transaction Server for z/OS V3.2 Installation Guide*
- *CICS Transaction Server for z/OS V3.2 Messages and Codes*
- *CICS Transaction Server for z/OS V3.2 Operations and Utilities Guide*
- *CICS Transaction Server for z/OS CICSplex SM V3.2 Messages and Codes*

---

### Overview of migrating to CICS TS 3.2

Our goal with all of our migrations is to follow the path of a typical customer. We migrated slowly across our test sysplex and within the workloads on that sysplex. This created a mix of releases within a system as well as across the CICSplex<sup>®</sup>. We did this to test as many coexistence and operational combinations as possible. After we completed the migration steps for the CICSplex on our test sysplex, we followed a similar migration strategy for the CICSplex across our pseudo-production sysplex.

Figure 71 on page 190 illustrates the four main application groups in our Parallel Sysplex.

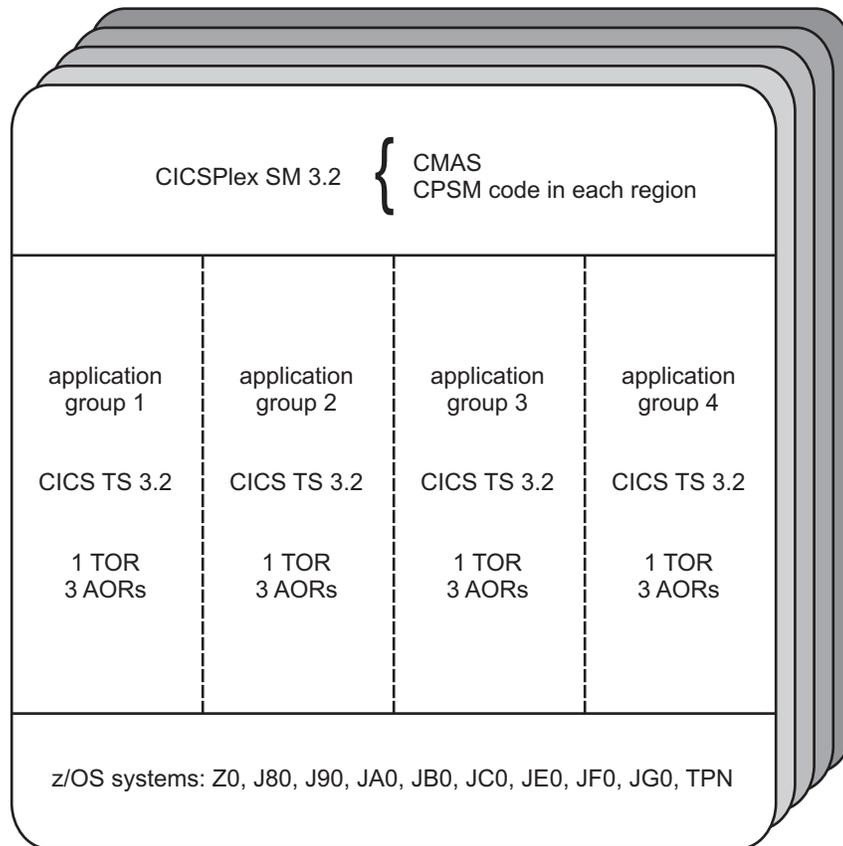


Figure 71. Our CICS TS configuration

The application groups process the following types of work:

- Application group 1: IMS/DBCTL
- Application group 2: CICS/VSAM (both RLS and non-RLS)
- Application group 3: DB2 and WebSphere MQ
- Application group 4: cryptographic functions, Java™, C++, and others

## Preparing to migrate to CICS TS 3.2

Before we began the actual migration process, we did some preparatory work in the following areas:

- **Backing up our data**

Even though we created new files and data sets, as a precaution, we first took backups of all of the CSDs and data repositories.

- **Defining aliases**

We defined new catalog aliases for CICS TS 3.2.

- **Loading the CICS TS 3.2 product libraries**

We set up and ran the SMPE jobs to load the CICS TS 3.2 product libraries. We then ran a copy job to bring the build libraries over to our production systems.

- **Allocating supporting data sets**

We created copies of all our supporting libraries (JCL, SYSIN, TABLEs, and so on). We reviewed these and updated accordingly with all the necessary CICS TS 3.2 changes.

- **Customizing the CICS region data sets**

We customized the jobs in *hlq.SDFHINST(DFHDEFDS)* and *hlq.SEYUINST(EYUDEFDS)* to define all of the region data sets and submitted them a number of times. Depending on your environment, you might need to alter the default file sizes. We increased the file sizes for the DFHGCD, DFHINTRA and DFHTEMP data sets. Being a test organization, we also increased the sizes of our auxtrace data sets.

- **Reviewing and reassembling tables**

We reviewed and reassembled any tables we had modified.

- **Updating SYS1.PARMLIB and APF-authorizing program libraries**

In SYS1.PARMLIB, we updated LINK list and LPA list. We APF-authorized the following program libraries:

- *hlq.SDFHAUTH*
- *hlq.SDFHLINK*
- *hlq.SDFJAUTH*
- *hlq.SEYUAUTH*
- *hlq.SEYULINK*

In SYS1.PROCLIB, we reviewed and updated all our procs for the CICS TS 3.2 changes.

---

## Migrating CICSplex SM

With this release of CICS TS 3.2, the CICSplex SM (CPSM) installation is integrated with the CICS install. Previously, it was a separate process. You can now modify the DFHISTAR job to change both the CICS and CICSplex SM installation parameters.

If you are migrating to CICSplex SM for the first time, CPSM consists of the following parts on each system:

- CMAS (CICS-managed address space)
- CPSM code running in each CICS region (MAS), which communicates with the CMAS, sometimes referred to as the *agent* code.

In order for a CMAS and a MAS to communicate, they must be running the same version of CPSM.

You can run CPSM at different release levels across a CICSplex, as long you follow the rules documented in *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1* under the section “Running CICSplex SM Version 3.2 and earlier releases concurrently.” By adhering to these rules, we ran the following combinations at the same time across our CICSplex/sysplex during our migration:

- 3.2 CMAS with 3.2 MAS
- 3.2 CMAS with 3.1 MAS
- 3.1 CMAS with 3.1 MAS

**Note:** DFHIRP must be at the highest level of the code in a system image, and the version of DFHIRP for CICS TS 3.2 can only be used on z/OS 1.7 and higher.

As previously announced, in this release of CICS, the CICSplex SM TSO end user interface (EUI, and its CAS) is no longer supported and has been removed. Those functions have been moved to the Web User Interface (WUI). All of these changes are discussed in *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1*.

---

## Migrating the CMASs

We did the following to migrate the CMASs:

1. Defined a new CSD
2. Upgraded the CSD with CPSM 3.2 level resource definitions and the CICS startup group list. We did this by running the DFHCSDUP utility with the UPGRADE command, as documented in *CICS Transaction Server for z/OS V3.2 Operations and Utilities Guide*.
3. Reviewed our CICS resource definition tables, which we had updated earlier
4. Converted the CPSM data repository to the CPSM 3.2 level by running the EYU9XDUT utility, as documented in *CICS Transaction Server for z/OS V3.2 Installation Guide*
5. Reviewed the JCL in the EYUCMAS member for any changes to the CMAS startup procedure. We added MEMLIMIT=NOLIMIT to our procedures, as documented in *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1*, and updated the data set names to use our new high-level qualifiers.
6. Updated the MAS startup procedures to point to the new CPSM data sets in order to identify the new CMAS code to the MAS regions

**Note:** The EDASALIM system initialization parameter default value has changed. The default is now 34MB. If you have created your SIT using previously supplied defaults, you should update the table to use the new CICS-supplied defaults.

Our CMASs were then ready to start. Remember that the maintenance point CMAS must be the first CMAS to migrate to the new release.

---

## Migrating the MASs

We reviewed the steps documented in *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1* to migrate the MASs. Many of the steps are similar to the steps we followed to migrate the CMASs.

We did the following to migrate the MASs:

1. Defined a new CSD and copied our application groups from the old CSD
2. Upgraded the CSD for CICS TS 3.2. We also removed groups for previous releases of CPSM from the group list.

**Note:** If you are also changing z/OS release levels, the LE definitions (on CSD) may need to be updated too.

3. Reviewed our CICS resource definition tables, which were updated earlier
4. Copied the JCL for our MAS startup procedures, added MEMLIMIT=NOLIMIT and changed the library names to use our new high-level qualifiers
5. Reviewed the LE libraries we had in RPL concatenation

Note that CICS no longer supports the DCT macro as a means of defining transient data queues. These must be defined in the CSD using TDQUEUE resource definitions.

Before CICS TS version 2.3, JVM profiles were stored in a PDS member. In CICS TS 2.3 and later, they are stored in a file system directory pointed to by the JVMPROFILEDIR system initialization parameter. If you are migrating from a

release prior to CICS TS 2.3, you will need to make the appropriate JAVA changes. We keep our JVM profiles outside the file system shipped with CICS TS 3.2, so that they would not be overridden with a CICS TS 3.2 file system at maintenance time. Note that a number of options in the JVMProfile have changed in CICS TS 3.2. See *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1* for details.

Our MASs were then ready to start.

---

## Migrating the CICSplex SM Web User Interface

As documented in *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1*, both the CICSplex SM Web User Interface (WUI) and the CMAS it connects to must be at the highest level of CICSplex SM within the CICSplex. This means that both must be at the same level as the maintenance point CMAS.

Because the CICS system that acts as the WUI is just another MAS, we used the same steps as for migrating a MAS:

1. Migrated the MAS that acts as the WUI
2. Upgraded the WUI CSD, as above
3. Added new WUI server initialization parms AUTOIMPORTDSN and AUTOIMPRORTMEM, as documented in *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1*
4. Migrated the contents of the WUI server repository (EYUWREP), as documented in *CICS Transaction Server for z/OS V3.2 Migration from CICS Transaction Server 3.1*

---

## Experiences with migrating to CICS TS 3.2

With the exception of the usual typos and unrelated sysplex issues, this migration went well. The removal of the CPSM TSO End User Interface was a big change for us. Even though the CPSM WUI has been available for a few releases, we still used the TSO interface for certain summary views. We are still learning the WUI and plan to take more advantage of features like customized views in the future.



---

## Chapter 22. Migrating to CICS TG V7.0

We migrated our CICS Transaction Gateway (CICS TG) setups from V6.1 to V7.0. Our migration was very simple and straight forward and our CICS TG continues to run solidly.

The basic setups for CICS TG V7.0 are much the same as for V6.x. See the “Migration to CICS Transaction Gateway V6.1” topic in our December 2007 test report for details about our CICS TG V6.x setups.

For our migration to CICS TG V7.0, we used the following documentation resources:

- CICS TG home page, at [www.ibm.com/software/htp/cics/ctg/](http://www.ibm.com/software/htp/cics/ctg/)
- CICS TG library, at [www.ibm.com/software/htp/cics/ctg/library/](http://www.ibm.com/software/htp/cics/ctg/library/)

---

### Migrating the CICS TG daemon to V7

The CICS Transaction Gateway supports communication with CICS TG resource adapters of the same level or an earlier level. To maintain this compatibility, we migrated our CICS TG daemons to V7 before the client side code (such as WebSphere Application Server resource adapters).

---

### CICS TG daemon statistics

New with CICS TG V7.0 is a robust set of statistics that can provide better real-time monitoring of the CICS TG daemons.

Statistics can be viewed using the CICS TG administration interface, either by using the MVS system command `MODIFY jobname,APPL=STATS,options` or by enabling the statistics interface on the CICS TG daemon and using the CICS TG statistics application programming interface (API).

The statistics API is written in C. CICS TG V7.0 provides a sample C program that you can use to test the statistics interface and use as a model for your own monitoring program. See the CTGSTAT1 sample in the `ctg70hlq.SCTGSAMP` data set.

We enabled the statistics interface on our CICS TG daemons and use them in conjunction with OMEGAMON XE for CICS TG V4.1.0.

The statistics API is used by OMEGAMON to obtain data.

---

### Port requirements for the statistics interface

The statistics interface requires a unique port number for each daemon.

The statistics interface must be enabled in the CICS TG daemon. This can be done by using the configuration tool or by editing the configuration file.

In our environment, we use multiple CICS TG daemons on each system. We use TCP/IP port sharing to allow the multiple daemons to listen for incoming requests on the same port, thus providing duplexing for backup and load balancing.

| When the statistics interface is enabled, another listener is started on each CICS TG  
| daemon for statistics requests. Since the statistics are unique to each daemon, this  
| port cannot be shared by multiple daemons; each must be configured for a unique  
| port.

| There are a number of ways that this port can be specified and overridden.  
| Generally, it is specified in the CICS TG initialization file or as a parameter to the  
| **ctgstart** command.

| We use a single, common CICS TG initialization file (ctg.ini) for all of our  
| daemons. We updated our ctg.ini file to set a default port number  
| (statsport=2016). We updated the JCL procs for additional daemons to specify a  
| different port as a parameter to the **ctgstart** command (-statsport=2017). This  
| parameter can also be passed as a parameter on the start command used to start  
| the CICS TG daemon.

| The method you choose will depend on your startup procedures, but be aware of  
| implications such as changes needed for system automations.

---

## | **Reserving the statistics interface port**

| Another consideration is that you may want to reserve the port used by the  
| statistics port(s) used by the CICS TG daemons. We reserve all of the ports used by  
| our daemons to prevent others from inadvertently obtaining these ports. This is  
| done in our TCPIP.PROFILE members in the PORT statements.

---

## Chapter 23. Migrating to IMS Version 10.1

This topic discusses our experiences with migrating our production 8-way IMS data sharing group (composed of members IMS8, IMS9, IMSA, IMSB, IMSC, IMSE, IMSF and IMS0) from IMS V9 to IMS Version 10. It is not intended to be a step by step procedure because each migration is unique due to the IMS configuration and features selected.

We used the following documentation to plan the migration:

- *IMS Version 10 Release Planning Guide*
- *IMS Version 10 System Definition Reference*
- *IMS Version 10 Implementation Guide: A Technical Overview* (IBM Redbook)
- IMS Version 10 Information Center

---

### Migration and coexistence software

Before starting the migration, we installed the following service:

- **DBRC migration SPE**  
IMS V9 PK06147 / UK18490 allows IMS V9 systems to use RECONs which have been upgraded to IMS V10.
- **Operations management coexistence SPE**  
IMS V9 PK27280 / UK18913 service provides IMS Operations Manager coexistence support for higher level command registration lists.
- **System management coexistence SPE**  
IMS V9 SPE PK30189 / UK22059 coexistence service to handle input from destination DFSOMAPI  
  
In a Shared Queues environment, a message inserted to a transaction from an input destination of DFSOMAPI can be processed by an IMS V9 system. If the application program inserts a message back to the IOPCB, the ISRT back to the IOPCB is not allowed and is rejected with a STATUS AD.
- **Global online change SPE**  
IMS V9 PK23402 / UK20811 coexistence APAR for Global OLC to enable IMS V9 and future IMS releases to coexist in an OLCSTAT data set.
- **Resource consistency checking coexistence SPE**  
IMS V9 PK32970 / UK24486 coexistence APAR allows resource consistency checking to be enabled for IMS V9 systems.  
  
Resource consistency checking is not performed in IMS V10.

---

### Staging our migration

We performed the following staged migration of our IMS V9 systems:

1. We migrated one of our IMS systems from IMS V9 to V10 on an IBM System z9 Business Class (z9 BC) server.
2. We migrated another IMS system to V10 on an IBM System z10 Enterprise Class (z10 EC) server.
3. We migrated another IMS system to V10 on an IBM eServer zSeries 990 server (z990) server.
4. We then migrated the remaining IMS systems to V10.

After each step of the migration, we ran all of our workloads and performed regression tests. The migration was completed in approximately two months.

---

## FPCTRL system definition macro eliminated

The FPCTRL system definition macro has been eliminated and will be ignored during system definition. Because we use the IMS Fast Path feature, we added FP=Y into our DFSPBxxx members.

---

## IMS Exits

We relinked several IMS exits.

### DBRC SCI registration exit routine (DSPSCIX0)

DSPSCIX0 was updated to support parallel access to the RECON data sets. This added the DBRC group ID.

Here is how we defined our IMSplex and Group ID in DSPSCIX0:

```
DC CL(DSNL)'RECON1.PROD' RECON name
DC CL(PNL)'PROD ' IMSplex name
DC CL(GIL)'001' Sharing Group ID
DC XL(RCL)'00000000' RC00 = use the IMSplex name
DC CL(DSNL)'RECON2.PROD' RECON name
DC CL(PNL)'PROD ' IMSplex name
DC CL(GIL)'001' Sharing Group ID
DC XL(RCL)'00000000' RC00 = use the IMSplex name
DC CL(DSNL)'RECON3.PROD' RECON name
DC CL(PNL)'PROD ' IMSplex name
DC CL(GIL)'001' Sharing Group ID
DC XL(RCL)'00000000' RC00 = use the IMSplex name
```

### OTMA routing exits (DFSYPX0 and DFSYDRU0)

The OTMA descriptors are enhanced in IMS Version 10 to allow you to define OTMA routing information in the DFSYDTx member of the IMS.PROCLIB data set. For details about how to do this, see *IMS Version 10 System Definition Reference*.

We chose not to implement this function at this time so we are still using DFSYPX0 and DFSYDRU0 exits.

---

## IMS Java migration

There are several changes in IMS V10 for Java. These changes are documented in the topic, "Specific migration considerations," in the IMS V10 Information Center.

### Summary of changes for Java in IMS V10

- Java application programs that run in JMP or JBP regions require the IBM 31-bit SDK for z/OS, Java 2 Technology Edition, Version 5 or later.
- The **Dibm.jvm.shareable.application.class.path** and the **Dibm.trusted.middleware.class.path** statements are both replaced by the **Dibm.java.class.path** statement.
- The `imsjava.jar` file is not supported in IMS Version 10. Instead, use the following `.jar` files that pertain to your environment, as shown in Table 5 on page 199:

Table 5. Applicable .jar files for various Java application environments in IMS Version 10

Java application environment	.jar file
All	imsjavaBase.jar
Java message processing (JMP) region and Java batch processing (JBP) region	imsjavaTM.jar
JDBC only	imsJDBC.jar
WebSphere Application Server for z/OS only	imsBJCA.jar
WebSphere Application Server distributed only	imsRDSClient.jar

## Service applied

We applied APAR PK56411 which fixes the following problem:

While utilizing JDK 5.0 support with IMS V10, an IMS Java Dependent Region (JDR) encounters an ABENDU0101. The abend is issued from DFSRCJB0 / DFSRCJM0. In addition to the abend, the following error message is also issued:  
DFSJVM00: Exception: java.lang.NoClassDefFoundError:

## Migration procedure

The following DFSJVM members were updated for our Java applications:

- DFSJVMMS changes:
  - Removed the `-Dibm.jvm.shareable.application.class.path` and `-Dibm.jvm.trusted.middleware.class.path` statements and replaced them with the `-Djava.class.path` statement
  - Changed the `-Djava.class.path` from using the `/imsjava/current/imsjava.jar` file to the new jar files: `/imsjava/current10/imsjavaBase.jar`, `/imsjava/current10/imsJDBC.jar`, and `/imsjava/current10/imsjavaTM.jar`
- DFSJVMEV changes:
  - Set the LIBPATH to our IBM 31-bit SDK for z/OS, Java 2 Technology Edition, Version 5

---

## IMS syntax checker

Invoked the syntax checker, as follows:

```
EXEC 'MVSBUILD. IMS110. SDFSEXEC(DFSSCRT)' 'HLQ(MVSBUILD. IMS110)'
```

When checking our DFSPBxxx members, it flagged the ISIS parameter. ISIS = 0, 1 and 2 is no longer valid. We updated our DFSPBxxx members with:  
ISIS=R

---

## Migrating IMS V9 RECON data sets to V10

IMS V9 RECONS can be upgraded to IMS Version 10 by executing the DBRC utility (DSPURX00) and using the CHANGE.RECON UPGRADE command with an IMS V10 SDFSRESL library. Before doing the upgrade, we applied the IMS V9 SPE PK06147 which permits IMS V9 systems to use IMS V10 RECONS.

We then made backups of our RECON data sets.

## RECON status prior to migration

The following display shows our IMS V9 RECON status prior to migration:

```

RECON
RECOVERY CONTROL DATA SET, IMS V9R1
DMB#=2102 INIT TOKEN=02249F1302471F
NOFORCER LOG DSN CHECK=CHECK44 STARTNEW=NO
TAPE UNIT=3400 DASD UNIT=SYSDA TRACEOFF SSID=***NULL**
LIST DLOG=NO CA/IC/LOG DATA SETS CATALOGED=YES
MINIMUM VERSION = 9.1 CROSS DBRC SERVICE LEVEL ID= 00000
REORG NUMBER VERIFICATION=NO
LOG RETENTION PERIOD=00.001 00:00:00.0
COMMAND AUTH=NONE HLQ=***NULL**
SIZALERT DSNUM=15 VOLNUM=16 PERCENT= 95
LOGALERT DSNUM=3 VOLNUM=16

```

TIME STAMP INFORMATION:

TIMEZIN = %SYS

OUTPUT FORMAT: DEFAULT = LOCORG NONE PUNC YY  
CURRENT = LOCORG NONE PUNC YY

**IMSPLEX = PROD**

-DDNAME-	-STATUS-	-DATA SET NAME-
RECON1	COPY1	RECON1.PROD
RECON2	COPY2	RECON2.PROD
RECON3	SPARE	RECON3.PROD

**Note:** We have an IMSplex defined named PROD. There is no Group ID defined in the IMS V9 RECON.

## RECON upgrade

We used the DBRC CHANGE.RECON UPGRADE command to migrate our IMS V9 data sets to V10:

```

DSP1123I UPGRADE0 DBRC REGISTERED WITH IMSPLEX PROD USING EXIT
 IMS VERSION 10 RELEASE 1 DATA BASE RECOVERY CONTROL
 CHANGE.RECON UPGRADE
DSP0251I RECON COPY 1 UPGRADE IS BEGINNING
DSP0252I RECON COPY 1 UPGRADED SUCCESSFULLY
DSP0251I RECON COPY 2 UPGRADE IS BEGINNING
DSP0252I RECON COPY 2 UPGRADED SUCCESSFULLY
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 08.060 11:09:13.1
 IMS VERSION 10 RELEASE 1 DATA BASE RECOVERY CONTROL
DSP0211I COMMAND PROCESSING COMPLETE
DSP0211I HIGHEST CONDITION CODE = 00

```

**Note:** If you have an IMSplex name defined in your RECON, make sure you have PK55384 / UK30890 IMS V10 fix installed.

## RECON status after migration

LIST.RECON STATUS after the upgrade to IMS V10:

```

RECON
RECOVERY CONTROL DATA SET, IMS V10R1
DMB#=2112 INIT TOKEN=02249F1302471F
NOFORCER LOG DSN CHECK=CHECK44 STARTNEW=NO
TAPE UNIT=3400 DASD UNIT=SYSDA TRACEOFF SSID=***NULL**
LIST DLOG=NO CA/IC/LOG DATA SETS CATALOGED=YES
MINIMUM VERSION = 9.1 CROSS DBRC SERVICE LEVEL ID= 00001
REORG NUMBER VERIFICATION=NO

```

```
LOG RETENTION PERIOD=00.001 00:00:00.0
COMMAND AUTH=NONE HLQ=**NULL**
ACCESS=SERIAL LIST=STATIC
SIZALERT DSNUM=15 VOLNUM=16 PERCENT= 95
LOGALERT DSNUM=3 VOLNUM=16
```

TIME STAMP INFORMATION:

TIMEZIN = %SYS

OUTPUT FORMAT: DEFAULT = LOCORG NONE PUNC YY  
CURRENT = LOCORG NONE PUNC YY

**IMSPLEX = PROD GROUP ID = 001**

-DDNAME-	-STATUS-	-DATA SET NAME-
RECON1	COPY1	RECON1.PROD
RECON2	COPY2	RECON2.PROD
RECON3	SPARE	RECON3.PROD

**Note:** Group ID is now defined in the IMS V10 RECON.

---

## IRLM support in IMS V10

IRLM 2.2 is the only IRLM shipped and supported with IMS V10.

IRLM 2.2 connections to IRLM 2.1 are supported:

- IRLM 2.1 is supported with IMS Version 9
- IRLM 2.1 is not supported with IMS Version 10

IRLM 2.2 with IMS Version 10 can connect to IRLM 2.1 with IMS V9.

As we migrated each IMS to V10, we also migrated the associated IRLM from V2.1 to V2.2.

---

## Upgrading IMS utilities

We needed to upgrade several of our IBM IMS utilities because our current versions were not compatible with IMS V10:

- IMS HP Pointer Checker 2.1 upgraded to HP Pointer Checker 2.2, 5655-K53
- IMS HP Change Accumulation 1.2 to HP Change Accumulation 1.4, 5655-F59
- IMS Database Recovery Facility 2.1 to Database Recovery Facility 3.1, 5655-N47

You can find a complete list of IMS tools and V10 compatibility by going to [www.ibm.com/software/data/db2imstools/](http://www.ibm.com/software/data/db2imstools/) and then clicking **Support > IMS Tools and IMS Compatibility > IMS Version 10**.



---

## Chapter 24. Migrating to IMS Transaction Manager Resource Adapter V10.2

We migrated the IMS Connector for Java V9.2.0.4 resource adapter used in our WebSphere Application Servers to IMS Transaction Manager Resource Adapter (IMS TM RA) V10.2 level.

Along with the new level comes another name change. Formerly known as the IMS Connector for Java Resource Adapter, it is now called the IMS Transaction Manager Resource Adapter (IMS TM RA). The latest version of the IMS TM RA is available from their home page, at [www.ibm.com/software/data/ims/ims/components/tm-resource-adapter.html](http://www.ibm.com/software/data/ims/ims/components/tm-resource-adapter.html).

We also used the following documentation for our migration:

- IMS documentation library, at [www.ibm.com/software/data/ims/library/](http://www.ibm.com/software/data/ims/library/)
- IMS for z/OS Information Center, at [publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp). Navigate to **IMS Version 10 > IMS SOA Integration Suite > IMS TM Resource Adapter**.

---

### Evaluating applications for potential migration

The IMS TM Resource Adapter V10 is based on J2EE Connector Architecture (JCA) 1.5. You should review your applications to see if they need to be migrated to the JCA 1.5 specification levels. In addition, this version will only run on WebSphere Application Server V6.x or higher. See the IMS TM Resource Adapter home page for a complete list of supported releases, application development and runtime environments.

We used the tools in Rational® Application Developer (RAD) V7 to migrate our applications to the latest JCA 1.5 specifications. We also migrated the applications to the J2EE 1.4 specifications in the process. The RAD migration wizards made this task very simple.

---

### Updating applications that use conversational transactions

Some of the tolerances to the handling of conversational transactions have been tightened for IMS V10. In our case, we have security enabled on our IMS V10 region and would receive a security violation error because our applications were calling conversational transactions in IMS, but were not explicitly ending these transactions before the connection was reused.

To correct this, we needed a small update to our application to check whether the conversation was ended prior to closing the connection. To do this, the code now calls the `interactionSpec.getConvEnded()` method. When it is false (meaning that the conversation is not marked as ended), we now explicitly set `setInteractionVerb(IMSInteractionSpec.SYNC_END_CONVERSATION)` to end the conversation, as shown in the following code snippet:

```
 :
 :
 interaction.execute(interactionSpec, inputRecord, outputRecord);
 :
 :

 // Added check for end of conversation.
```



---

## Chapter 25. Using JZOS

Most of the IBM Java for z/OS JDKs that are currently available include a JZOS function package. This package includes a Java batch launcher and Java APIs for system services specific to z/OS.

We used the following documentation in our testing:

- Java on z/OS home page, at [www.ibm.com/servers/eserver/zseries/software/java/](http://www.ibm.com/servers/eserver/zseries/software/java/)
- JZOS Java batch launcher and toolkit overview, at [www.ibm.com/servers/eserver/zseries/software/java/products/jzos/overview.html](http://www.ibm.com/servers/eserver/zseries/software/java/products/jzos/overview.html)
- *Java Stand-alone Applications on z/OS, Volume II* (IBM Redbook), at [www.redbooks.ibm.com/abstracts/sg247291.html](http://www.redbooks.ibm.com/abstracts/sg247291.html)
- *JZOS Cookbook*, at [www.alphaworks.ibm.com/tech/zosjavabatchtk/](http://www.alphaworks.ibm.com/tech/zosjavabatchtk/)

---

### JZOS batch launcher

We have been using the JZOS Java batch launcher for some time now and continue to increase our usage of it. It helps provide an MVS look and feel by encapsulating our Java applications within a batch job or JCL procedure or both. This provides a comfortable environment for those who are not as familiar with z/OS UNIX or Java. Java output and system output are easily sent to the job log, eliminating the need to view output in the z/OS UNIX file system—a real convenience that our MVS folks appreciate. Although you can use BPXBATCH in a similar fashion, the JZOS batch launcher is tailor-made for Java applications.

---

### MVS operations and JZOS

In conjunction with the batch launcher, we also incorporate JZOS in our Java applications to interact with the MVS console. In addition to allowing Java applications to write to the MVS console, our Java applications can now easily integrate the handling of MVS START, MODIFY, and STOP commands. This has been particularly helpful with our long-running Java processes. MODIFY commands allow dynamic changes and status reporting to the MVS operators. STOP commands are intercepted to allow a graceful shutdown of the application, avoiding a hard cancel.

---

### JZOS installation and setup

Although the JZOS toolkit is provided with the IBM JDKs, there are additional steps needed to set it up. See the instructions that come with the JDKs for information about JZOS installation and setup.

To use some of the JZOS functions, you will need to have the JZOS load modules in an MVS data set, in addition to the JDK code packaged in the z/OS Unix file system. Sample JCL jobs and procedures are provided. We used these as templates and modified them as needed for our environments.

For more information, see the JZOS Java Launcher and Toolkit Overview, at [www.ibm.com/servers/eserver/zseries/software/java/products/jzos/overview.html](http://www.ibm.com/servers/eserver/zseries/software/java/products/jzos/overview.html).

---

## JZOS Cookbook

We would also like to point out the JZOS Cookbook, which is available from the alphaWorks® site. This publication provides a tutorial of the JZOS functions as it develops and deploys a sample application. Many of these examples can be used as a basis for your own projects.

---

## Chapter 26. 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.

The following topics describe our experiences:

- “Using WebSphere MQ shared queues and coupling facility structures”
- “Enabling WebSphere MQ Security” on page 210
- “Enabling higher availability for WebSphere MQ” on page 212
- “Using WebSphere Message Broker” on page 213
- “MQCICS — WebSphere MQ-CICS adapter/bridge workload” on page 214

---

### 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.

Much of our discussion here focuses on our experience with the usage and behavior of the coupling facility structures that support shared queues as well as using shared channels in a distributed environment with queue managers running V6.0.

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 and *WebSphere MQ for z/OS System Setup Guide*, SC34-6583, 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/](http://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/](http://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/](http://www.ibm.com/redbooks/)

### Our queue sharing group configuration

We currently have two queue sharing groups: one with three members and another with five 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 five-member production queue sharing group. All of the queue managers in the group run WebSphere MQ for z/OS Version 6.0.

## Our coupling facility structure configuration

We defined our MQ coupling facility structures to use four coupling facilities (CF1, CF2, CF3, and CF4) as defined in the PREFLIST in the structure definitions. (See “Coupling facility details” on page 315 for details about our coupling facilities.)

The following is the structure definition for our CSQ\_ADMIN structure:

```
STRUCTURE NAME(MQGPCSQ_ADMIN)
 INITSIZE(24064)
 MINSIZE(18668)
 DUPLEX(ENABLED)
 SIZE(30740)
 ALLOWAUTOALT(YES)
 PREFLIST(CF1,CF2,CF3,CF4)
 REBUILDPERCENT(1)
 FULLTHRESHOLD(85)
```

We also have the following five message structures defined to support different workloads:

- MSGQ1 — for the batch stress workload and system shared queues
- CICS — for the CICS bridge application
- EDSW — for the IMS bridge application
- WMQI — for the WebSphere Message Broker applications
- BOOK – for our BookStore workload (uses DB2, WMQ and WebSphere Application Server)

The following is the structure definition for the message structure that supports the MQ-CICS bridge workload:

```
STRUCTURE NAME(MQGPCICS)
 INITSIZE(15872)
 DUPLEX(ENABLED)
 SIZE(20480)
 ALLOWAUTOALT(YES)
 PREFLIST(CF1,CF4,CF2,CF3)
 REBUILDPERCENT(1)
 FULLTHRESHOLD(85)
```

The other four 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, EDSW, WMQI, BOOK and MSGQ1 structures are recoverable and backed up daily.

## Recovery behavior with queue managers using 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 prelist, 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.

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.

## 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 00C51nnn. 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 queuing. 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 for members of the queue sharing group. 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:

- 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 messages could be:

```
CSQ5003A !MQJA0 CSQ5CONN Connection to DB2 using DBWG pending, no active DB2
CSQ5026E !MQJA0 CSQ5CONN Unable to access DB2, RRS is not available
```

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 DBW3
```

- MQ abend reason codes that indicate a DB2 failure start with 00F5nnnn.

#### 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(24064) and MINSIZE(18668) for CSQ\_ADMIN in the CFRM policy to accommodate the increase in usage.

---

## Enabling WebSphere MQ Security

We recently went through the task of enabling MQ security for the z/OS queue managers in our zPET environment. WebSphere MQ provides an interface to an external security manager which, in our case, is Resource Access Control Facility (RACF). When we decided to enable security for our queue managers, we took a step back to determine the best approach for our environment. Our simple approach to controlling security was to use queue-sharing group level of security for our queue managers that were members of a queue-sharing group and queue manager level of security for the rest of the queue managers in our environment which are not members of queue-sharing groups.

Referencing the *System Setup Guide* section “Using RACF classes and profiles,” we first verified that the WebSphere MQ classes were activated in RACF. As in most customer environments we then used our ‘test plex’ as our starting point for enabling MQ security. Our ‘test plex’ consists of 3 z/OS images each running a queue manager at V6.0. These 3 queue managers are all members of the queue-sharing group MQGT. Since all 3 queue managers are members of the same queue-sharing group we decided to use queue-sharing group level of security. We started by defining a basic set of profiles to each of the WebSphere MQ classes. We recently installed a new queue manager on our test sysplex. We implemented queue manager level of security because this queue manager is not part of the queue-sharing group.

### Reference material

We found the following reference material useful when working with WebSphere MQ Security:

- WebSphere MQ for z/OS Security (Technical Conference) which is a good overview located at:

<http://www.gse.org.uk/wg/racf/docs/apr2005/GSE-%20WebSphere%20MQ%20zOS%20Security.pdf>

- **WebSphere Message Broker (WMB):** which outlines the necessary authority required by the broker. Search for "Authorization required" and then select "Summary of required access (z/OS)" at:  
<http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp>
- **WebSphere MQ Explorer:** which outlines the necessary authority required by the MQ Explorer. Search for "Authorization to use WebSphere MQ Explorer" at:  
<http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp>
- **SDSF:** The following links document the necessary authority required by SDSF:
  - **Communications:**  
[http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/ISF4CS50/3.7?SHELF=ISF4BK50&DT=20050707140821](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ISF4CS50/3.7?SHELF=ISF4BK50&DT=20050707140821)
  - **WebSphere:**  
[http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/ISF4CS50/7.29?SHELF=ISF4BK50&DT=20050707140821](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ISF4CS50/7.29?SHELF=ISF4BK50&DT=20050707140821)
  - **SDSF Customization Wizard:** provides assistance in defining security for SDSF's use of MQ:  
<http://www-03.ibm.com/servers/eserver/zseries/zos/wizards/sdsf/sdsfv1r1/>

Once we had the basic profiles defined we started to enable security for each queue manager one at a time, resolving problems as they arose. We used RACF groups to grant authorities instead of individual userids which should make maintaining this security much easier. After enabling security for our test sysplex, we moved on to our production sysplex. Our production sysplex consists of nine z/OS images each running a queue manager at V6.0. Of these nine queue managers, four of them are members of the same queue-sharing group MQGP. For the four queue managers that are members of a queue-sharing group we implemented security using the queue-sharing group level of authority. For the other queue managers we implemented the queue manager level of security.

## Problems encountered

Following are some of the problems we encountered:

### 1. WebSphere V6 Explorer:

- a. After enabling security for our z/OS queue managers our connection to these queue managers using the WebSphere MQ Explorer were rejected with the following error:

```
ICH408I USER(dodaro) GROUP() NAME(???) 932
 LOGON/JOB INITIATION - USER AT TERMINAL NOT RACF-DEFINED
 IRR012I VERIFICATION FAILED. USER PROFILE NOT FOUND.
```

The userid was being sent to the host in 'lower case' and RACF was rejecting it. We installed fix pack 6.0.1.1 (U200247) for WebSphere MQ V6.0 to resolve this problem.

- b. With security enabled for our z/OS queue managers our connection was rejected with the following error:

```
ICH408I USER(DODARO) GROUP(SYS1) NAME(#####) 015
 MQGT.AMQ.BF0F023EF3019DB9 CL(MQQUEUE)
 PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
 ACCESS INTENT(READ) ACCESS ALLOWED(NONE)
```

The queue being created was using the incorrect prefix 'AMQ.\*\*' instead of 'AMQ.MQEXPLORER.\*\*'. APAR IC50201 will resolve this issue.

2. **Mixed case' queue names:** After enabling security in our environment we ran into a situation trying to access one of our queues. WebSphere MQ supports

'mixed case' for their queue names. We had a queue named 'Trade3BrokerTestQueue'. When we attempted to access this queue we received the following RACF error:

```
ICH408I USER(WAS5SSR3) GROUP(WASSRGP) NAME(WAS 5 APPSVR SR 3
MQGT.Trade3BrokerTestQueue CL(MQQUEUE)
PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
ACCESS INTENT(READ) ACCESS ALLOWED(NONE)
```

RACF currently does not allow defining 'mixed case' profiles for the MQ classes. To get around this situation we created a profile named 'MQGT.T\*' and granted the necessary authority to this profile. Until RACF supports 'mixed case' profiles we would suggest that if you use 'mixed case' that your queue name is prefixed with enough characters in 'upper case' (for example TRADE3BrokerTestQueue) which will allow you to properly protect your queues.

3. **WebSphere Message Broker and WebSphere Application Server:** After enabling security for our z/OS queue managers we experienced problems when connecting to our queue managers from these applications when the userid being sent to the host was in 'lower or mixed case' and subsequently was rejected by RACF. This was the case with the WMB toolkit running on Windows® and connecting to z/OS config mgr.. Here we changed the userid on Windows to be in 'uppercase'. This was also the case for WebSphere Application Server when the JMS resource was defined using a 'lowercase' userid causing the listener not to start. Again, here we were able to get around this problem by changing the JMS resource definition in WebSphere Application Server to use an 'uppercase' userid.

MQJMS2013: invalid security authentication supplied for MQQueueManager at startup.

```
CSQ8MSTR has: ICH408I USER(setup) GROUP() NAME(???)
LOGON/JOB INITIATION - USER AT TERMINAL NOT RACF-DEFINED
```

---

## Enabling higher availability for WebSphere MQ

With an ever increasing dependence on the infrastructure to perform critical business processes, the availability of this infrastructure is becoming increasingly more important. In an effort to provide high availability for our WebSphere MQ deployment, we recently converted our queue managers over to using unique application-instance dynamic virtual IP addresses (DVIPA).

Specifically, each of our queue manager CHINIT JCL streams now use the MODDVIPA utility to create (upon initialization) and delete (upon shutdown) unique application-activated dynamic virtual IP addresses. By utilizing application specific IP addresses, connectivity to each queue manager is dynamically re-established regardless of where in the sysplex the queue manager is restarted, either manually or via automatic restart manager (ARM).

To avoid failures when trying to create the DVIPAs prior to the DYNAMICVIPAs block being processed, message EZD1214I Initial Dynamic VIPA processing has been added to the TCP/IP initialization. This gives you a reliable message that can be used to automate the activation of an application-activated DVIPA. See APAR PK14941 for further details.

Here is a sample of the first step in our CHINIT JCL to create the DVIPA(xx.xx.xx.xx):

```
DVIPA(xx.xx.xx.xx):
//TCPDVP EXEC PGM=MODDVIPA,REGION=0K,TIME=1440,
// PARM='POSIX(ON) ALL31(ON)/-p TCPIP -c xx.xx.xx.xx'
```

Here is a sample of the last step in our CHINIT JCL to delete the DVIPA(xx.xx.xx.xx):

```
//TCPDVP EXEC PGM=MODDVIPA,REGION=0K,TIME=1440,
// PARM='POSIX(ON) ALL31(ON)/-p TCPIP -d xx.xx.xx.xx'
```

For additional information, see:

- *WebSphere MQ in a z/OS Parallel Sysplex Environment*, SG24-6864
- *z/OS Communications Server: IP Configuration Guide*, SC31-8775

---

## Using WebSphere Message Broker

Our current WebSphere Message Broker configuration consists of three brokers and one configuration manager on z/OS in our test sysplex. The other sysplex also contains three brokers but with the configuration manager on Windows. All six brokers are at the V6.1 level.

### Updating the Retail\_IMS workload for workload sharing and high availability

In an effort to make our broker domain more complex and introduce workload sharing and high availability to our workloads, we created another broker for a total of three brokers on our production systems. We altered our Retail\_IMS workload to utilize all three brokers (workload sharing) and to continue processing if one or two brokers go down (high availability). We did this by making our application request queue shared. This was possible because each of the broker's queue managers was already a member of the queue sharing group.

#### Description of the workload

The Retail\_IMS workload uses WebSphere Application Server to host a Web front end (HTML page and Java servlet) to receive information from the user. This information is rolled up into a message and placed on the RETAIL.IMS.IN queue, which a broker message flow is monitoring. The message flow extracts some fields from the message, adds an IMS header, and puts the new message on the RETAIL.IMS.OUT queue. The Java servlet then takes the message from the RETAIL.IMS.OUT queue and passes the IIH header to an HTML page for display. Any failure in message processing will result in a message being placed on the RETAIL.IMS.FAIL queue.

Thus, this workload uses three queues:

1. RETAIL.IMS.IN — holds the input message to the message flow
2. RETAIL.IMS.OUT — holds the output message from the message flow when normal processing occurs
3. RETAIL.IMS.FAIL — holds the output message from the message flow when abnormal processing occurs

#### Changes to the workload

The MQReply node in the message flow enables you to have multiple candidates for the output queue without having multiple message flows (one for each client). For this reason we decided to make the output queues non-shared and use unique names per client. Figure 72 on page 214 shows what our message flow looks like now.

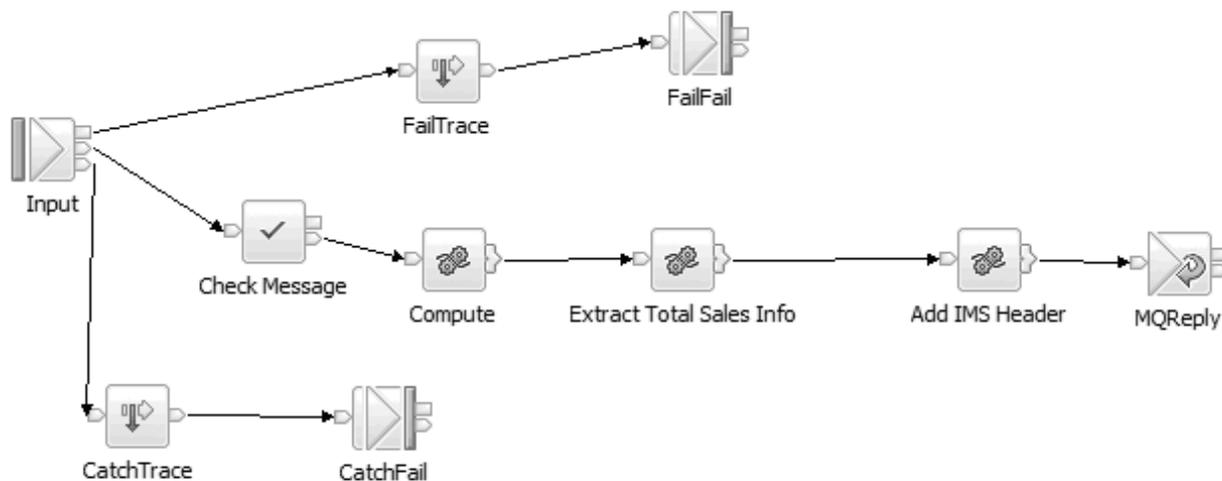


Figure 72. Message flow in our Retail\_IMS workload

We made the following changes to the queue definitions for the workload. We created unique output queues for each client putting request messages to the input queue. A WMQ cluster was created between all queue managers that are local to either a broker or a WebSphere Application Server instance where the client application was deployed. The application servers connect through bindings mode to their local queue managers. These queue managers are often not members of the queue sharing group, therefore we created the WMQ cluster to ease communication.

```

QUEUE (RETAIL.IMS.OUTJ90)
TYPE(QLOCAL)
QSGDISP(QMGR)
STGCLASS(WMQI)
CFSTRUCT()
CLUSTER(MQBROKER.CLUSTER)

```

As a result of these changes, the Retail\_IMS workload now utilizes all three of our brokers, alternating between brokers for each transaction by using the round robin functionality of MQ clustering. Additionally, if one of the two brokers fails, all of the messages are then processed by the other broker, and if the failed broker returns, the messages again alternate in a round-robin manner between the brokers. The end result is a workload that utilizes workload sharing and provides some level of high availability. Additional information about this topic can be found in the technical article, *WebSphere Business Integration Message Broker and high availability environments*, at [www.ibm.com/developerworks/websphere/library/techarticles/0403\\_humphreys/0403\\_humphreys.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0403_humphreys/0403_humphreys.html).

## MQCICS — WebSphere MQ-CICS adapter/bridge workload

Our MQCICS workload is a Java application that places a request message containing the name of a CICS transaction and required parameters. These transactions can be received by CICS either through the WebSphere MQ-CICS Bridge or the WebSphere MQ-CICS Adapter, depending on which process gets triggered by the request queue. The request queue is monitored by one or more CICS regions. After the request has been processed, the CICS region puts a

message on the specified reply queue. Our Java application runs either through z/OS UNIX or WebSphere Application Server on z/OS.

When we first started running this workload, we had WebSphere MQ V5 Release 3.1, where each bridge monitor task needed its own request queue. This limitation was removed with WebSphere MQ V6.

For variety in our test environment, we configured a shared queue solution for transactions to be processed by the WebSphere MQ CICS Adapter. Meanwhile, we test our WebSphere MQ CICS Bridge setup with an MQ cluster configuration.

## WebSphere MQ-CICS bridge monitor using clustered queues

In our first workload environment, we have one or more systems running the request applications to a Web front end being hosted by WebSphere Application Server. The queue where the requests are going to is being monitored by one WebSphere MQ-CICS bridge monitor on either of four queue managers. The CICS region that picks up the request then sends a reply to the queue manager being monitored by the client application. Figure 73 demonstrates the cluster environment.

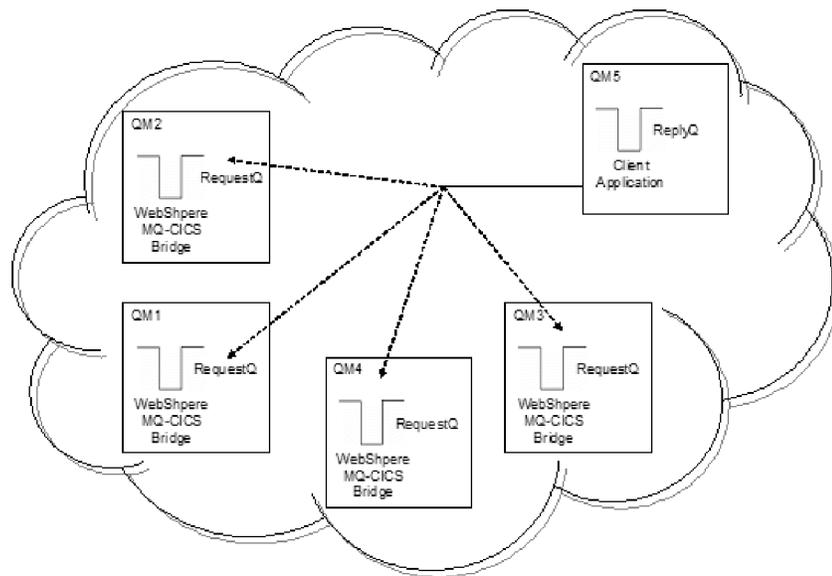


Figure 73. Our MQ cluster configuration for the WebSphere MQ-CICS bridge

## WebSphere MQ-CICS adapter using shared queues

For our second workload environment shown in Figure 74 on page 216, we use a shared queue environment and the transactions are processed through the WebSphere MQ-CICS adapter. All three queues (request, reply, and initiation) are shared. All members of the queue sharing group have a CICS region monitoring the queue.

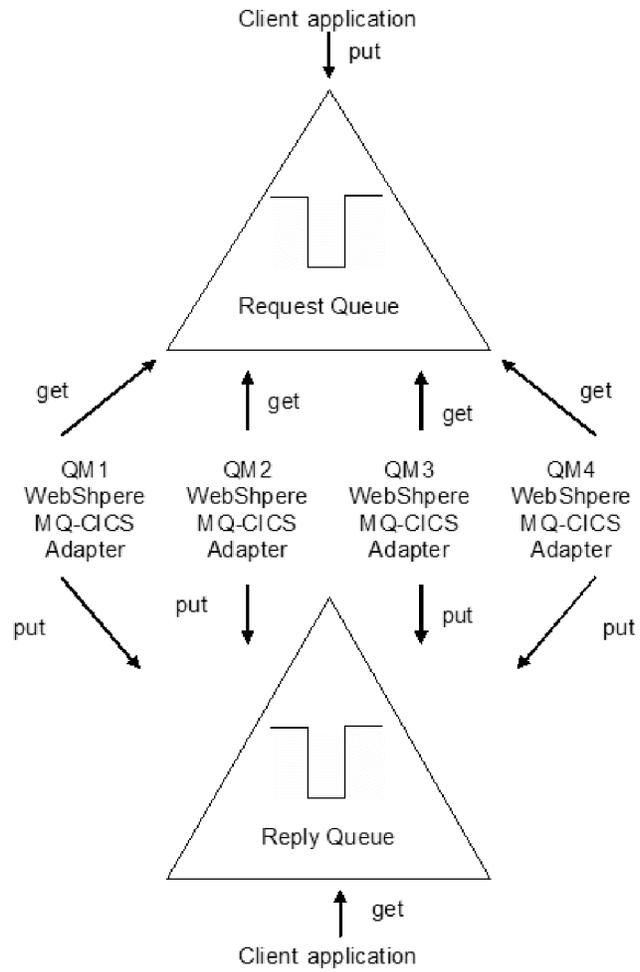


Figure 74. Our queue sharing group configuration for our WebSphere MQ-CICS adapter workload

---

## Chapter 27. Using IBM WebSphere Application Server for z/OS

The following topics describe our experiences using IBM WebSphere Application Server for z/OS and related products. We had previously migrated most of our WebSphere Application Server for z/OS V6.0 cells to WebSphere Application Server for z/OS V6.1 on z/OS V1R9.

**Note:** References to WebSphere Application Server for z/OS V6.x appear in the text as “WebSphere for z/OS V6.x” or simply “V6.x.”

---

### About our z/OS V1R10 test environment running WebSphere Application Server

The following topics 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 V1R10 WebSphere test environment

The following topics provide an overview of our z/OS V1R10 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.

#### Our 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 V1R9, our WebSphere test environment includes the following products:

- WebSphere Application Server for z/OS Version 6.1, service level cf140750.17
- WebSphere Application Server for z/OS Version 6.0.2, service level cf170648.05
- WebSphere Studio Workload Simulator V1.0
- WebSphere MQ for z/OS V6
- WebSphere Message Broker V6
- WebSphere Service Registry and Repository for z/OS Version 6.1
- WebSphere Process Server for z/OS Version 6.1
- DB2 V9.1 with JDBC
- CICS TS 3.2
  - CICS Transaction Gateway (CICS TG) V7
- IMS V10 with IMS Connect
  - IMS Transaction Manager Resource Adapter (IMS TM RA) V10.2
- TPC-R V3.4

**Software products on the workstation platform:** Software products on the workstation platform: On our workstations, we use the following tools to develop and test our Web applications:

- Rational Application Developer Version 7.0.0.5
- IBM Rational Developer for System z Version 7.1.1
- WebSphere Studio Workload Simulator V1.0

- Application Server Toolkit (AST) V6.1.1.2
- zPMT Configuration Tool for WebSphere for z/OS V6.1.0

## **Our current WebSphere Application Server for z/OS configurations and workloads**

The following are our current WebSphere Application Server for z/OS configurations and workloads.

**Configuration update highlights:** We made the following updates to our test and production configurations:

- Migrated cells to WebSphere Application Server for z/OS V6.1
- Migrated to CICS Transaction Gateway (CICS TG) V7
- Migrated to IMS Transaction Manager Resource Adapter (IMS TM RA) V10.2
- Installed TPC-R application into our WebSphere Application Server P1 cell (in a single, non-clustered J2EE server)
- Added three new cells to support WPS, WSRR, and a new application, LGI
- Expanded cell T1 to span from one system to three (Z1, Z2, and Z4)

*Our test and production configurations:* In our environment, we have fully migrated most of our WebSphere for z/OS V6.0.2 cells to WebSphere for z/OS V6.1. Our current setup contains nine cells: T1, T2, T3, Q2, A1, A2, and A3 on our PLEX2 systems, and P1 and QP on our PLEX1 systems. The Q2 and QP cells host WebSphere Application Server for z/OS applications used by the MQ team. The A1, A2, and A3 cells are used for the LGI SOA application. All cells are configured as network deployment cells. The Q2 and QP cells continue to run using WebSphere for z/OS V6.0.2, while all others are now at V6.1.

Our T1 cell is configured as follows:

- Resides entirely on one of our test systems (Z1)
- Contains seven different J2EE servers, each running different applications (as described below)

Our T2 cell is configured as follows:

- Resides entirely on one of our test systems (Z2)
- Contains seven different J2EE servers, each running different applications (as described below)

Our T3 cell is configured as follows:

- Resides entirely on one of our test systems (Z3)
- Contains seven different J2EE servers, each running different applications (as described below)

Our P1 cell is configured as follows:

- Spans three production systems in our sysplex (J80, JB0 and JF0)
- Contains six different clusters, each of which spans all three 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/T2 cell. Initially, we configure and deploy applications on a test J2EE server in the T1 and/or T2 cell and then deploy them to the corresponding server cluster in the P1 cell.

Our Q2 cell is configured as follows:

- Spans two systems in our PLEX2 sysplex (Z2 and Z3)
- Contains two different clusters, each of which spans both systems. Each cluster contains two J2EE servers—one J2EE server per system.
- Each cluster hosts various applications that connect WebSphere Application Server for z/OS to MQ as used by the MQ team.

Our QP cell is configured as follows:

- Spans two production systems in our sysplex (JC0 and J90)
- Contains two different clusters, each of which spans both systems. Each cluster contains two J2EE servers—one J2EE server per system.
- Each cluster hosts various applications that connect WebSphere Application Server for z/OS to MQ as used by the MQ team.

The A1, A2 and A3 cells were created to support the LGI SOA application. Details on the setup and application can be found in Chapter 30, “Deploying a secure SOA solution,” on page 231.

Our A1 cell is configured as follows:

- Spans four systems in our sysplex (Z1, Z2, Z3, and Z4)
- Contains one cluster which spans all four systems. Each cluster contains one J2EE server—one J2EE server per system.
- Each cluster corresponds to one of the single J2EE servers in our cell.
- WebSphere Process Server 6.1 is installed into this cluster.

Our A2 cell is configured as follows:

- Spans four systems in our sysplex (Z1, Z2, Z3, and Z4)
- Contains four 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 cell.

Our A3 cell is configured as follows:

- Spans four systems in our sysplex (Z1, Z2, Z3, and Z4)
- Contains one cluster which spans all four systems. Each cluster contains one J2EE server—one J2EE server per system.
- Each cluster corresponds to one of the single J2EE servers in our cell.
- WebSphere Service Registry and Repository 6.1 is installed into this cluster.

*Our Web application workloads:* The following applications run in the J2EE servers on our T1, T2 and P1 cells:

- J2EE server 1 runs our workload monitoring application. The application accesses only z/OS UNIX System Services files.
- J2EE server 2 runs our bookstore application, accessing DB2 and WebSphere MQ
- J2EE server 3 runs the Trade6 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

The following application runs in the J2EE Server on our T2 and T3 cells in addition to the above six applications:

- J2EE server 7 runs our zBank application used for security testing and accessing DB2

Figure 75 shows the server address spaces in our P1 cell.

**Note:** The wsp1s1 cluster is not shown in the diagram.

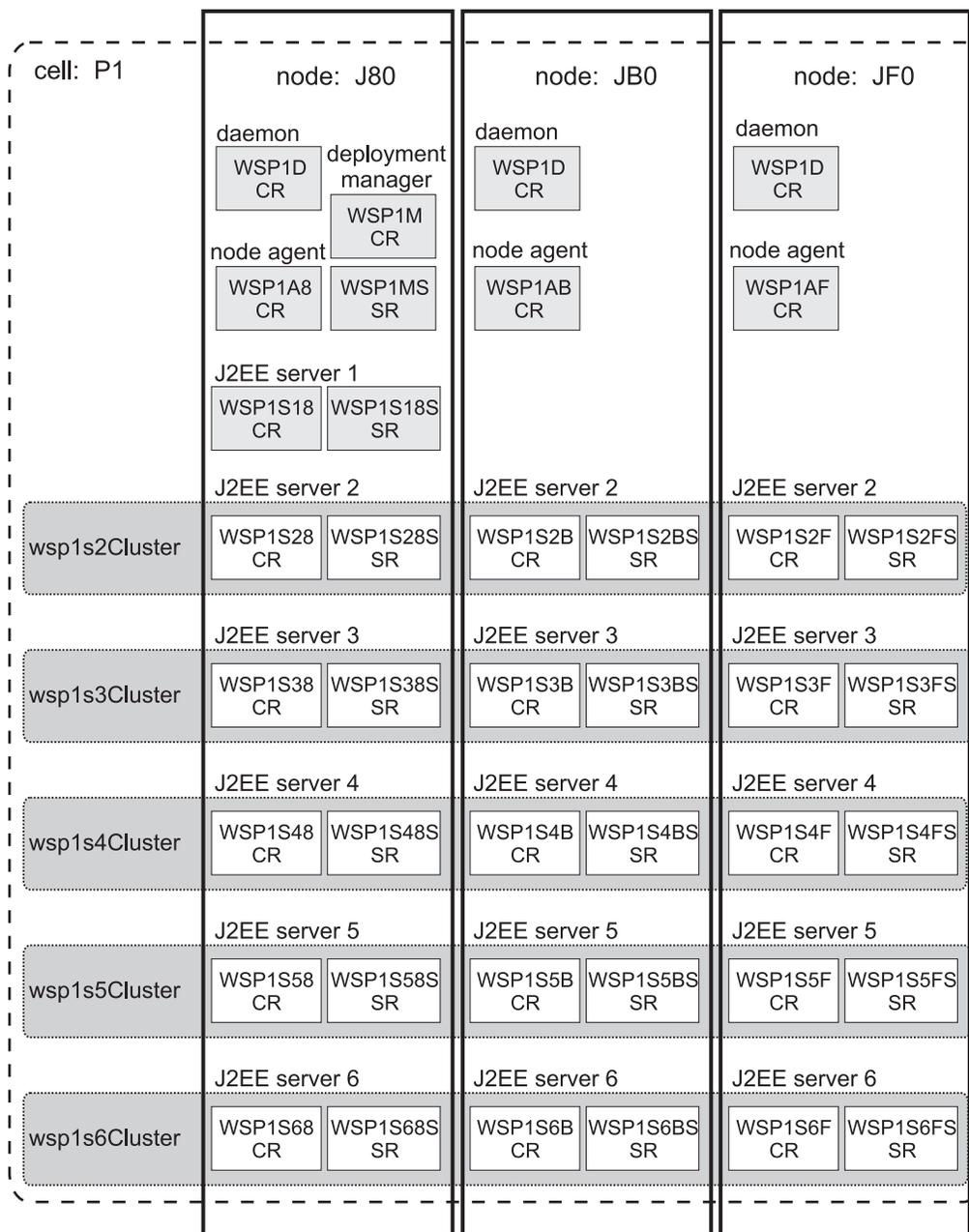


Figure 75. Our WebSphere for z/OS V6 configuration

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:

$ttccs[n]y[S]$

where:

	<i>tt</i>	Team identifier:
		<b>WS</b> WebSphere team
		<b>MQ</b> MQ team
		<b>PS</b> Process Server team
	<i>cc</i>	Cell identifier:
		<b>T1</b> Test cell 1
		<b>T2</b> Test cell 2
		<b>P1</b> Production cell 1
		<b>QP</b> MQ team production cell
		<b>Q2</b> MQ team test cell
		<b>A1</b> WPS cell
		<b>A2</b> WPS test application cell
		<b>A3</b> WSRR cell
	<i>s[n]</i>	Server type. For J2EE server control regions and server regions, <i>n</i> is the instance number of the server within the node:
		<b>A</b> Node agent
		<b>D</b> Daemon
		<b>M</b> Deployment manager
		<b>Sn</b> J2EE server control region, instance <i>n</i>
	<i>y</i>	System identifier:
		<b>1</b> Z1 (test)
		<b>2</b> Z2 (test)
		<b>3</b> Z3 (test)
		<b>4</b> Z4 (test)
		<b>8</b> J80 (production)
		<b>9</b> J90 (production)
		<b>B</b> JB0 (production)
		<b>C</b> JC0 (production)
		<b>F</b> JF0 (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 WebSphere production cell 1 J2EE server server region 1 on system J80.

Server short names are specified in upper case. Server long names are the same as the short names, but are specified in lower case.

---

## Other changes and updates to our WebSphere test environment

The following topics describe other changes and updates to our WebSphere test environment:

- “Migrating to WebSphere Application Server for z/OS V6.1”
- “Migrating to CICS Transaction Gateway V7” on page 222
- “Migrating to IMS Transaction Manager Resource Adapter V10.2” on page 222
- “Passing DB2 client information to the server” on page 222
- “Installed TPC-R V3.4” on page 226

### Migrating to WebSphere Application Server for z/OS V6.1

We have migrated most of our WebSphere Application Server V6.0.2 cells to V6.1. Overall, our migrations to V6.1 have been very smooth. The process is very similar to the migration from V5.1 to V6.0.2. It still requires a good bit of planning and

work to migrate to V6.1 from V6.0.2. Careful review of all the latest documents in the WebSphere InfoCenter is highly recommended.

While we did have some problems in our initial V6.1 migrations, we successfully migrated from our WebSphere Application Server for z/OS V6.0.2 with service level CF180704 to WebSphere Application Server for z/OS V6.1 at service level CF50625. Many fixes have been included and it is recommended to apply the latest service updates, including those for the V6.0.2 configuration from which you are migrating, prior to starting.

One issue we ran into is now addressed in APAR PK48599. It was due to our setups using z/OS UNIX symbolic links within our WebSphere configurations and the configurations using IMS or CICS resource adapters. If your setups have neither, you need not worry about this. We used the local fix described in Tech Doc #21257063 (available at [www.ibm.com/support/docview.wss?rs=404&uid=swg21257063](http://www.ibm.com/support/docview.wss?rs=404&uid=swg21257063)) during our migrations, but the formal fix for this APAR should be available by press time and is the recommended way to go.

The Legacy RRS connector for DB2 JDBC access is no longer supported on WAS V6.1. Many of the JDBC resources were still defined using this JDBC provider. To help migrate these to DB2 JDBC JCC resources, we used a utility available from the WebSphere Application Server support web pages. The utility is well documented, easy to use, and ran very well for us. See the “JDBC Migration White Paper and Utility for DB2 on z/OS” available at [www.ibm.com/support/docview.wss?rs=404&context=SS7K4U&q1=RRS&uid=swg27007826&loc=en\\_US&cs=utf-8&lang=en](http://www.ibm.com/support/docview.wss?rs=404&context=SS7K4U&q1=RRS&uid=swg27007826&loc=en_US&cs=utf-8&lang=en). The utility can be run either against the Version 6.0.2 setup prior to migration to Version 6.1 or after on the Version 6.1 setup. We chose to update our provider prior to migrating to V6.1 to prevent errors after the migration.

See the following references for more information:

- WebSphere Application Server for z/OS V6.1 Information Center, at [publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp)
- WebSphere Integration Test team’s report on migrations to WebSphere Application Server V6.1 from various levels, which you can find at [publibz.boulder.ibm.com/epubs/pdf/e0z1r100.pdf](http://publibz.boulder.ibm.com/epubs/pdf/e0z1r100.pdf)
- Also see the following WebSphere Integration Test report, which you can find at [publibz.boulder.ibm.com/epubs/pdf/e0z1r111.pdf](http://publibz.boulder.ibm.com/epubs/pdf/e0z1r111.pdf)

## **Migrating to CICS Transaction Gateway V7**

We made changes to the WebSphere environment when migrating to CICS Transaction Gateway (CICS TG) V7. See Chapter 22, “Migrating to CICS TG V7.0,” on page 195 for specific information about these changes.

## **Migrating to IMS Transaction Manager Resource Adapter V10.2**

We made changes to the WebSphere environment when migrating to IMS Transaction Manager Resource Adapter (IMS TM RA) V10.2. See Chapter 24, “Migrating to IMS Transaction Manager Resource Adapter V10.2,” on page 203 for specific information about these changes.

## **Passing DB2 client information to the server**

We tested the DB2-only methods provided by the DB2 Universal JDBC Driver that can be used to provide extra information about the client to the server. This can really help make your DB2 administrator’s life a bit easier!

## Passing client information to DB2 from WebSphere Application Server datasources

One of the common complaints we often hear from our DB2 database administrators is that they can't tell where a thread is coming from and what application it's from.

While Type 2 connections provide a bit more detail, such as the local address space initiating the connection, it still leaves our DBAs scratching their heads saying, "What app is that?" With Type 4 (TCP/IP) connections, it becomes even harder.

The DB2 Universal JDBC Driver provides DB2-only methods that you can use to provide extra information about the client to the server. WebSphere Application Server makes it easy to add these to your DB2 datasources.

For full details, see the DB2 Information Center at [publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp). Look for the topic titled "Providing extended client information to the DB2 server with the DB2 Universal JDBC Driver" under "Advanced JDBC application programming concepts."

Table 6 shows, from the above reference, the methods that you can use to pass additional client information.

*Table 6. DB2 Universal JDBC driver methods for passing client information to the server*

Method	Information provided
setDB2ClientUser	User name for a connection
setDB2ClientWorkstation	Client workstation name for a connection
setDB2ClientApplicationInformation	Name of the application that is working with a connection
setDB2ClientAccountingInformation	Accounting information

For our J2EE applications running in WebSphere Application Server (managed servers), these methods can be set as properties on the DB2 datasource.

Prior to setting these priorities, a DB2 DISPLAY THREAD command showed the following for a connection from this server:

```

NAME ST A REQ ID AUTHID PLAN ASID TOKEN
SERVER RA * 22 db2jcc_appli DB2USR DISTSERV 011B 99099
V437-WORKSTATION=Z2EIP.PDL.POK.IB, USERID=db2usr,
APPLICATION NAME=db2jcc_application
V445-G90C14A2.GB60.BFD666FD252E=99099 ACCESSING DATA FOR
::FFFF:xx.yy.20.162

```

Using the WebSphere Application Server admin console Web application, we set the custom priorities for the JDBC datasource for our application as shown in Table 7.

*Table 7. Custom priorities that we set for the JDBC datasource for our application*

Datasource property	Our value set	Our usage
clientWorkstation	WST2S22_3	Address space or workstation name initiating the connection. A suffix (_3) is used if the app/server has more than one datasource.
clientApplicationInformation	zipSeriesStore	Application using the datasource

Table 7. Custom priorities that we set for the JDBC datasource for our application (continued)

Datasource property	Our value set	Our usage
clientAccountingInformation	BookStoreEJDB2Entity	Datasource resource name (same as used in WebSphere Application Server admin console)

After the server was updated with the changes, the client information is now sent and the DISPLAY THREAD command now shows the following information:

```

NAME ST A REQ ID AUTHID PLAN ASID TOKEN
SERVER RA * 31 db2jcc_appli DB2USR DISTSERV 011B 157752
V437-WORKSTATION=WST2S22_3, USERID=db2usr,
 APPLICATION NAME=zipSeriesStore
V445-G90C14A2.G6F9.BFD66C17E48A=157752 ACCESSING DATA FOR
 ::FFFF:xx.yy.20.162

```

The clientAccountingInformation is not shown when using the DISPLAY THREAD command unless you add the DETAIL option (for instance: DIS,THD(\*),DETAIL). See “Example: Thread detail output for a Type 4 connection from WebSphere Application Server” on page 225 for an example showing the use of the DETAIL option.

### Additional notes and experiences with passing DB2 client information

We have the following additional notes and experiences to share about passing DB2 client information:

- While we could set the clientUser property, we found this a bit confusing on the dis,thd side. In the following example, we set the clientUser property to BookStore\_Search. The actual user ID that is used for the connection (SETUP) is displayed in the first line of the output. The value specified for the clientUser property shows up in the second line (USERID=BookStore\_Search). This made it a bit more confusing as the clientUser property is not associated with any user ID and is only informational, but the display output has the USERID= shown along with this. Not setting this property, the actual user ID is displayed, so the two lines of output match.

```

NAME ST A REQ ID AUTHID PLAN ASID TOKEN
SERVER RA * 41 db2jcc_appli DB2USR DISTSERV 011B 12627
V437-WORKSTATION=WST2S22_3, USERID=BookStore_Search,
 APPLICATION NAME=zipSeriesStore
V445-G90C14A2.G574.BFD671E125DA=12627 ACCESSING DATA FOR
 ::FFFF:xx.yy.20.162

```

- In many of our applications we have multiple datasources defined and used. We found it helpful to add some additional information to the clientWorkstation or clientApplicationInformation property to help discern between them, since these values show up in the DISPLAY THREAD output. In our examples, we have added a suffix to the clientWorkstation property that identifies to us the third resource defined for this application (WST2S22\_3) . While this bit of information doesn’t always help our DB2 administrators, it does help our WebSphere Application Server administrators.
- For standalone (non-managed) Java applications that access DB2, you need to code these methods to enable. See the sample code in the DB2 Information Center.

For further information, see the WebSphere Application Server Information Center at [publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp). Look for the topic titled, “Passing client information to a database.”

### Example: Thread detail output for a Type 4 connection from WebSphere Application Server

The following is an example of the response from the DIS,THD(\*),DETAIL command for a Type 4 connection from WebSphere Application Server, before setting any of the clientxxx properties:

```
NAME ST A REQ ID AUTHID PLAN ASID TOKEN
SERVER RA * 48 db2jcc_appli DB2USR DISTSERV 011B 99099
V437-WORKSTATION=Z2EIP.PDL.POK.IB, USERID=db2usr,
 APPLICATION NAME=db2jcc_application
V441-ACCOUNTING=JCC03010Z2EIP.PDL.POK.IBM.
 ',X'00'
V436-PGM=NULLID.SYSLN200, SEC=1, STMNT=0
V445-G90C14A2.GB60.BFD666FD252E=99099 ACCESSING DATA FOR
 (1)::FFFF:xx.yy.20.162
V447--INDEX SESSID A ST TIME
V448--(1) 446:2912 W S2 0634511043933
```

The following is an example of the response from the DIS,THD(\*),DETAIL command after setting the clientWorkstation, clientApplicationInformation, and clientAccountingInformation properties for the datasource:

```
NAME ST A REQ ID AUTHID PLAN ASID TOKEN
SERVER RA * 68 db2jcc_appli DB2USR DISTSERV 011B 157752
V437-WORKSTATION=WST2S22_3, USERID=db2usr,
 APPLICATION NAME=zipSeriesStore
V441-ACCOUNTING=BookStoreEJBDB2Entity
V436-PGM=NULLID.SYSLN200, SEC=1, STMNT=0
V445-G90C14A2.G6F9.BFD66C17E48A=157752 ACCESSING DATA FOR
 (1)::FFFF:xx.yy.20.162
V447--INDEX SESSID A ST TIME
V448--(1) 446:1785 W S2 0634511195815
```

### Example: Thread detail output for a Type 2 connection from WebSphere Application Server

The following is an example of the response from the DIS,THD(\*) command for a Type 2 connection from WebSphere Application Server, before setting any of the clientxxx properties:

```
NAME ST A REQ ID AUTHID PLAN ASID TOKEN
RRSAF TD 4 WST2S22S DB2USR ?RRSAF 01B0 56848
```

The following is an example of the response from the DIS,THD(\*) command after setting the clientWorkstation, clientApplicationInformation, and clientAccountingInformation properties for the datasource:

```
NAME ST A REQ ID AUTHID PLAN ASID TOKEN
RRSAF TD 4 WST2S22S DB2USR ?RRSAF 01B0 20436
V437-WORKSTATION=WST2S22_3, USERID=*,
 APPLICATION NAME=zipSeriesStore
```

The DIS,THD(\*),DETAIL command displays the value of the clientAccountingInformation property:

```
NAME ST A REQ ID AUTHID PLAN ASID TOKEN
RRSAF TD 4 WST2S22S DB2USR ?RRSAF 01B0 20436
V437-WORKSTATION=WST2S22_3, USERID=*,
 APPLICATION NAME=zipSeriesStore
V441-ACCOUNTING=BookStoreEJBDB2Entity
```

## Installed TPC-R V3.4

We made changes to the WebSphere environment when we installed TPC-R V3.4 in our zPET environment. For specific information about these changes, see Chapter 9, “Using TPC-R V3.4 and Basic HyperSwap in our zPET environment,” on page 113.

---

## 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 documentation, at [www.ibm.com/software/webservers/appserv/zos\\_os390/library/](http://www.ibm.com/software/webservers/appserv/zos_os390/library/)
- IBM WebSphere Application Server, Version 6.0 Information Center, at [publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp)
- IBM WebSphere Application Server, Version 6.1 Information Center, at [publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp)
- IBM DB2 Information Center, available at [publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp)
- IBM Techdocs (flashes, white papers, and others), at [www.ibm.com/support/techdocs/](http://www.ibm.com/support/techdocs/)
- *Java 2 Platform Enterprise Edition Specification*, available at <http://java.sun.com/products/j2ee/>
- IBM CICS Transaction Gateway documentation, at <http://www.ibm.com/software/ts/cics/library/>
- IBM HTTP Server for OS/390 documentation, at <http://www.ibm.com/software/webservers/htpservers/library/>
- IBM WebSphere Studio Workload Simulator documentation, at [www.ibm.com/software/awdtools/studioworkloadsimulator/library/](http://www.ibm.com/software/awdtools/studioworkloadsimulator/library/)

---

## Chapter 28. Installing and configuring WebSphere Process Server for z/OS

This topic describes our WebSphere Process Server (WPS) installation and configuration. It is not intended as a complete installation guide but, rather, a summary of what we did to add WPS to our suite of WebSphere products.

WPS requires that a WebSphere Application Server (WAS) environment be defined. Shell scripts are run after the WAS cell is defined to install WPS into that WAS environment. WPS was installed in our environment for the Secure SOA Solution project, described in Chapter 30, “Deploying a secure SOA solution,” on page 231. Additional details about our WAS and WPS setup for this application can be found in the IBM Techdocs library white paper titled *The Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS* at [www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300](http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300).

We also used the following additional documentation to help us install and configure WebSphere Process Server for z/OS:

- WebSphere Process Server 6.1 for z/OS Information Center, at [publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/welcome\\_wpsz.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/welcome_wpsz.html)
- *Introducing the zPMT Configuration Tool for WebSphere z/OS*, at [www.ibm.com/support/techdocs/atmastr.nsf/fe582a1e48331b5585256de50062ae1c/b6253a5c0a76275686257206001093bd](http://www.ibm.com/support/techdocs/atmastr.nsf/fe582a1e48331b5585256de50062ae1c/b6253a5c0a76275686257206001093bd)
- Application Server Toolkit (AST) Overview presentation, at [publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was\\_v6/was/6.1/DevelopmentTools/WASv61\\_ASTOverview/player.html](http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/DevelopmentTools/WASv61_ASTOverview/player.html)

---

### WPS installation and configuration

We created a new WebSphere Application Server 6.1 network deployment server configuration with a deployment manager node. Then we created a new empty WebSphere Application Server node on which to install WebSphere Process Server. At the time of the install, we had WAS version 6.1 at service level 6.1.0.14. The following APARs are required:

- APAR BK56876 to WAS 6.1 service level 6.1.0.14. FMID(H28W610)
- APAR AK59140 for WPS 6.1 FMID(HWPS610)

Our WAS environment for WPS is defined as a network deployment cell to make use of the clustering capabilities for improving the availability and the scalability. The WAS cell was created by using zPMT and the Application Server Toolkit (AST) to generate and upload the installation JCL to z/OS.

The WAS cell consists of a deployment manager and nodes on every system in our test sysplex. A cluster was defined with one server per LPAR. The cell is called A1 and is documented in “Our current WebSphere Application Server for z/OS configurations and workloads” on page 218.

After the WebSphere Application Server cell was successfully created, we proceeded to run the WPS installation scripts and define the DB2 resources as

instructed in the WPS documentation. Before running any scripts we backed up our WAS cell ZFS data sets so we could restore should there be a problem. Then the following WPS installation scripts were run:

- **zWPSInstall.sh** — modifies a WebSphere Application Server profile to install WebSphere Process Server
- **zWPSConfig.sh** — configures a server as a process server, enabling the server to handle business processes

The **zWPSConfig.sh** configuration script generates data definition language (DDL) scripts that you can use to create the DB2 database objects for the configuration. We used SPUFI to run the DDL to define the database resources required by WPS.

We validated our WPS configuration using the Business Process Choreographer sample application BPCIVTApp. Instructions can be found in section “Verifying that Business Process Choreographer works” in the WebSphere Process Server 6.1 Information Center.

---

## WPS security

Our WAS cell was created with security disabled. It was enabled later, after successful startup and testing of WPS. The following were enabled using RACF as the security server:

- Administrative security
- Application security
- Bus security
- WebSphere MQ to SIBus security with SSL

The RACF EJBROLE class is used to secure access to various levels of function in WPS. The following WPS authorization roles are defined in our environment using the RACF EJBROLE class:

- BPEAPIUser
- BPESystemAdministrator
- BPESystemMonitor
- WebClientUser
- JMSAPIUser
- TaskAPIUser
- TaskSystemAdministrator
- TaskSystemMonitor
- EscalationUser
- WBIOperator

Details about the bus security and WebSphere MQ SSL setup can be found in the IBM Techdocs library white paper titled *The Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS*, at [www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101300](http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101300).

---

## Chapter 29. Installing and configuring WebSphere Service Registry and Repository for z/OS

This topic describes our WebSphere Service Registry and Repository (WSRR) installation and configuration. It is not intended as a complete installation guide but, rather, a summary of what we did to add WSRR to our suite of WebSphere products.

WSRR requires that a base WebSphere Application Server (WAS) environment be defined. Shell scripts are run after the WAS cell is defined to install WSRR as an application into that WAS environment. WSRR was installed in our environment for the LGI application. Additional details about our WAS and WSRR setup for this application can be found in the IBM Techdocs library white paper titled *The Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS* at [www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300](http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300).

We also used the following additional documentation to help us install and configure WebSphere Process Server for z/OS:

- WebSphere Service Registry and Repository for z/OS Version 6.1 Information Center, at [publib.boulder.ibm.com/infocenter/sr/v6r1/index.jsp](http://publib.boulder.ibm.com/infocenter/sr/v6r1/index.jsp)
- *Introducing the zPMT Configuration Tool for WebSphere z/OS*, at [www.ibm.com/support/techdocs/atmastr.nsf/fe582a1e48331b5585256de50062ae1c/b6253a5c0a76275686257206001093bd](http://www.ibm.com/support/techdocs/atmastr.nsf/fe582a1e48331b5585256de50062ae1c/b6253a5c0a76275686257206001093bd)
- Application Server Toolkit (AST) Overview presentation, at [publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was\\_v6/was/6.1/DevelopmentTools/WASv61\\_ASTOverview/player.html](http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/DevelopmentTools/WASv61_ASTOverview/player.html)

---

### WSRR installation and configuration

We created a new WebSphere Application Server 6.1 network deployment server configuration with a deployment manager node. Then we created a new empty WebSphere Application Server node on which to install WebSphere Service Registry and Repository. At the time of the install, we had WAS version 6.1 at service level 6.1.0.14. We applied the following APARs:

- PK65697: WSRR INSTALL FAILS IF DATABASE NAME IS GREATER THAN 8 CHARACTERS

Our WAS environment for WSRR is defined as a network deployment cell to make use of the clustering capabilities for improving the availability and the scalability. The WAS cell was created by using zPMT and the Application Server Toolkit (AST) to generate and upload the installation JCL to z/OS.

The WAS cell consists of a deployment manager and nodes on every system in our test sysplex. A cluster was defined with one server per LPAR. The cell is called A3 and is documented in "Our current WebSphere Application Server for z/OS configurations and workloads" on page 218.

After the WebSphere Application Server cell was successfully created, we proceeded to run the WSRR installation scripts and define the DB2 resources as

instructed in the WSRR documentation. Before running any scripts we backed up our WAS cell ZFS data sets so we could restore should there be a problem.

There are three options for installing WSRR. You can deploy to the following:

- A stand-alone WSRR
- Federated nodes
- A WebSphere Application Server cluster

We deployed to a WAS cluster.

For more information about running the installation script, see the section “Deploying WSRR for z/OS” in the WebSphere Service Registry and Repository for z/OS Version 6.1 Information Center.

We validated our WSRR configuration by logging on to WSRR using the non-secure URL:

```
http://http-server-host-name:http-port-number/ServiceRegistry
```

After we verified a successful installation, we enabled security for WSRR.

---

## WSRR security

Our WAS cell was created with security disabled. It was enabled later, after successful startup and testing of WSRR. The following were enabled using RACF as the security server:

- Administrative security
- Application security
- Bus security

The RACF EJBROLE class is used to secure access to various levels of function in WSRR. The following WSRR authorization roles are defined in our environment using the RACF EJBROLE class:

- Administrator
- User

Additional details about WSRR security for our application can be found in the IBM Techdocs library white paper titled *The Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS*, at [www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300](http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300).

---

## Chapter 30. Deploying a secure SOA solution

The zPET team worked jointly with the IBM Software Group (SWG) Federated Integration Test (FIT) team on a project—the Mixed Platform Stack project—that demonstrated that an SOA solution that is deployed to a distributed platform could also be deployed to a z/OS platform with little or no modification to the application. The project considered the key integration points between products running on different platforms and tested them to ensure interoperability. In addition, the project included particular focus on security configuration. At the end, we verified that the products, platforms, and applications involved in our scenario could be deployed successfully in both environments. We were also able to conduct extensive tests in order to verify the correct behavior.

The high-level objectives of the Mixed Platform Stack project include the following:

- Create and maintain a secure solution on the customer's platform of choice (z/OS or z/OS and AIX®).
- Ensure that all products in the solution work well together, regardless of platform specifics.
- Demonstrate that the deployment of SOA solutions to z/OS does not degrade existing production applications.

The project successfully demonstrates that an SOA solution can be deployed to the z/OS platform as well as the z/OS-AIX mixed platform. The scenario makes use of both J2EE and legacy technologies. It demonstrates the integration between these technologies on multiple platforms.

---

### The SOA solution scenario

Our scenario is based on a merger between two fictional insurance companies, Lord General Insurance (LGI) and DirectCar (DC). LGI is a large, established company with z/OS skills, infrastructure, and applications. DC is a new Internet-based company with AIX based skills, infrastructure, and applications.

Following the merger, the company implemented an insurance quote and policy system based on a service-oriented architecture (SOA). The quote and policy system includes the following characteristics:

- A direct internet channel for end users
- An Enterprise Service Bus (ESB) to transform and route messages to the various backend systems
- Two distinct backend systems (DC and LGI), both capable of performing the business logic required to issue insurance quotes and policies.
- A common business process used to handle the offline background checks required for final acceptance of an insurance policy
- Web services to perform specific application functions

This solution was implemented and tested in two environments. One was a mixed platform environment that included AIX and z/OS. This environment was created and tested in the FIT Lab. The other was a pure z/OS environment where all infrastructure and application components ran on z/OS. We tested the z/OS-only solution in our environment.

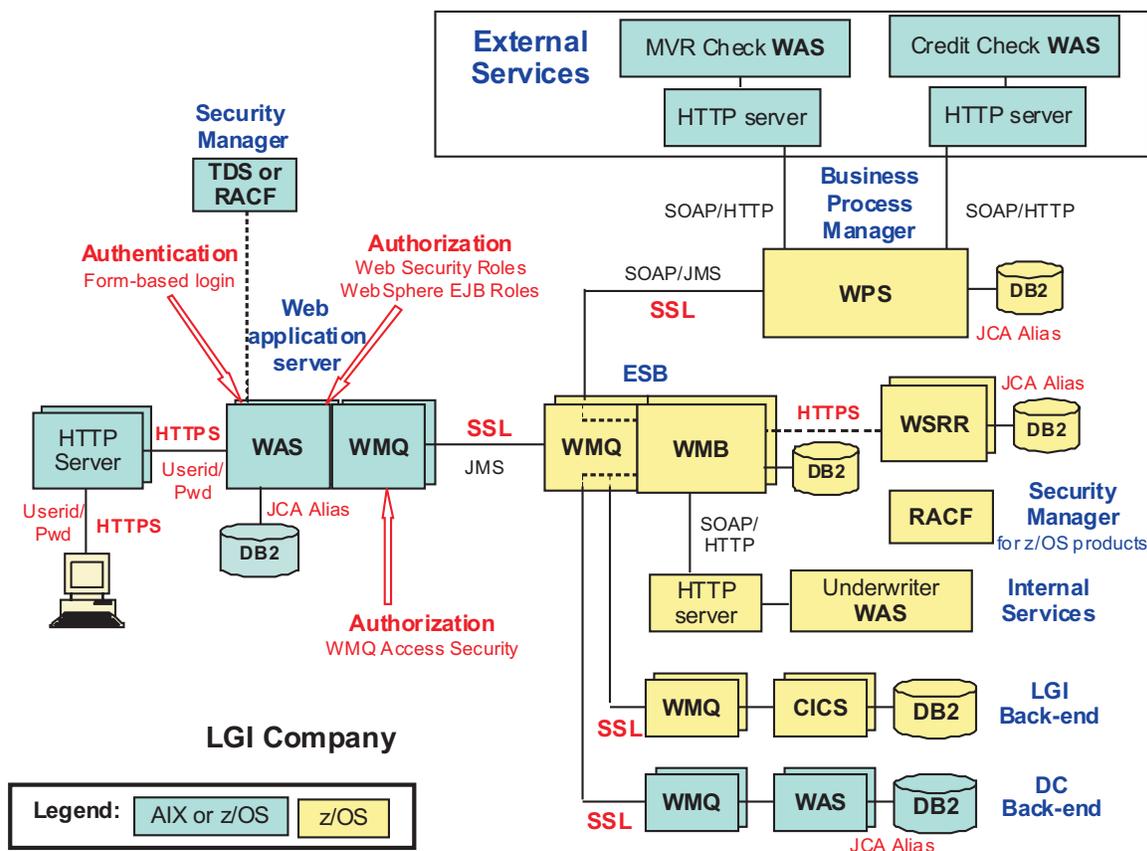


Figure 76. Components of our secure SOA scenario

In order to achieve high availability, reliability, and scalability, our solution takes advantage of the Parallel Sysplex capabilities within each product. This allows us to run any component on any available logical partition (LPAR) and with the highest quality of service.

WebSphere Application Server (WAS) clusters provide workload balancing and failover. The WAS environment is critical to our solution. It is needed for various application components and the WebSphere Process Server (WPS) and WebSphere Service Registry and Repository (WSRR) product infrastructure. The WAS environment is configured in a network deployment topology. This allows the z/OS workload manager (WLM) to manage the WAS workload across multiple LPARs. Clustering the servers across the various LPARs provides the highest availability possible.

DB2 data sharing on z/OS provides maximum availability, reliability, and recovery. This is not only important for the application components; many of the products used in the solution require DB2 themselves. WebSphere MQ (for shared queues), WebSphere Message Broker, WebSphere Process Server, and WebSphere Service Registry and Repository each require DB2.

WebSphere MQ shared queues on z/OS provide the highest availability and scalability. Combined with WAS clusters and CICSplex, our messages can be processed anywhere in the sysplex. We use a combination of WebSphere MQ features to provide the most robust, secure, and highly-available environment

| possible. Components use shared, clustered, and local queues along with cluster  
| channels and intragroup queuing for connectivity between queue managers.  
| Multiple WebSphere Message Brokers in a broker domain using WebSphere MQ  
| clustering and shared queue facilities provide high availability and workload  
| balancing for our ESB.

| For more details about this project, you can read the full experience report, *The*  
| *Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS*, at  
| [www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300](http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101300).



---

## Part 2. System z Platform Evaluation Test for Linux virtual servers

The System z Linux Virtual Servers Platform Evaluation Test (LVS PET) team focuses on integration testing of the Linux on System z and Linux virtual server aspects of our computing environment.

We address such topics as:

- Any significant updates to our environment since our last test report
- Our current test efforts and results
- Where we are headed from here and what we hope to report on in upcoming test reports



---

## Chapter 31. About our Linux virtual server environment

In this edition of our test report, we discuss the continued implementation of system management architectures throughout our environment based on the core infrastructure demonstrated in previous report editions. As you may recall, the previous edition discussed how systems management encompasses many smaller items including software management, data management, and security.

In order to continue our emphasis on software management, we continue our established process of documenting necessary upgrades to our middleware and operating systems to maintain current supported levels. We have also begun to explore the critical aspects of performance management and capacity management in our integration testing laboratory. To expand in the area of security management, we plan to integrate several Tivoli products that we are currently investigating.

In addition, we have before us a unique opportunity scheduled for our laboratory this fall: We plan to physically move our data center to a new location in the fourth quarter. We look at this as an opportunity to further test some of the availability management (HA and RAS-BR) work we have investigated in previous test report editions. We have documented some of the design and planning practices that we have put in place to ease this transition. The follow on report to this will discuss in more detail the lessons learned and best practices for that migration, including any data management tasks related to unforeseen problems caused by the migration.

The information that we present describes the sample configurations and deployments that we executed in our integration test laboratory as well as the recommendations and best practices discovered by our team. Though we focus on modeling customers who perform workloads based on Web technologies, many of the tools, configurations, and recommendations apply to anyone deploying large numbers of virtual Linux servers on the System z platform.

We employ a fairly large number of discreet Linux images spread across several hardware platforms exercising a diverse array of middleware products and solutions. Yet, the team managing and exploiting this environment is modest in size. Therefore, it is essential that the operation of the LVS PET complex be highly efficient, and that implies an emphasis on systems management.

The topics in this test report discuss the highlights of the last 6months of testing done in the development lab by the LVS PET team in accordance with our systems management, and availability policies.

---

### Fundamental goals and priorities

There are many ways to approach each systems management discipline. The approaches we have chosen should, as with all of our activities, align with customer trends and IBM strategies. In addition, the products and themes chosen for this edition make sense within the context of our existing environment. With that in mind, the following are the strategic priorities which guided our implementation choices:

- **Autonomic management and single point of control:** An important goal for both this test laboratory and IBM's customers is to enable a small team to keep the environment operating and running efficiently, 24x7x365, with minimal effort. This means the system should be as self-managing as possible. A related goal is to get as close as possible to a single point of monitoring and control. Approaches should address both native LPAR environments and z/VM<sup>®</sup> environments.
- **Preference for packaged (rather than home-grown) solutions:** We strive to use tools that are widely available, rather than home-grown alternatives. This includes both IBM tools and popular open source solutions. In this report, you will see that many of the data management tasks are handled by IBM products, while overall systems management is handled by a combination of IBM tools and tools provided by the Linux distributors.

An additional objective of the research behind this test report is to develop and validate a set of recommended best practices for monitoring and managing the systems and data in the environment we have constructed over the last several years.

---

## Staged implementation

Our LVS PET team is relatively small compared to many large IT organizations. As such, attempting to simultaneously implement all of the possible combinations of IBM and vendor management products is simply not practical. We have, as usual, staged our systems management implementation by opting to focus on one or two solutions for each of our system management discipline categories. The following is a rough ontology for systems management:

1. Availability management
2. Performance, capacity, and accounting management
3. Security management
4. Data management and system programmer tasks

As our returning readers know, we have previously spent a great deal of time on availability management, including our test reports focused on reliability, availability, and serviceability along with business resilience, as well as the previous editions covering essential high availability mechanisms. This test report edition will not deal with availability management nor performance and accounting management. Rest assured that each have been considered for future editions of our test report, so stay tuned.

---

## About our environment

Over time, the infrastructure presented in our test reports has evolved and become substantial in size. Some critical infrastructure such as automated backup procedures had been missing from our reports, as well as information on how we keep system software current.

Luckily, there are many systems management offerings available from IBM that can help with these tasks. Some tools, such as the backup utilities, are long established in the industry, while other tools, such as IBM Director, are more recent product offerings. This test report serves to outline these evolutionary steps in the test lab environment and reflects our continued commitment to performing integration testing in a customer-like way based on your feedback.

In addition to our current IT (production) environment, we have duplicated a small subset of systems to create a test environment where we will run various middleware on pre-GA Linux versions. This environment, called MDAT (Middleware Driven Acceptance Test), is an extension to our current distribution testing as well as a precursor to implementing new operating system versions and middleware products into our IT environment.

---

## Our workloads

This topic explains the workloads that we execute in our lab. It is here to provide context, but in no way limits the scope of workload applicable for the management tools that we will explain later.

The workloads we have selected for execution in our test environments run on IBM WebSphere Application Server. For variety, we execute two types of workloads concurrently:

- **Trade6** is designed to simulate a corporation that places stock trades and orders. The Trade6 workload exercises WebSphere Application Server, as well as DB2 on Linux in addition to the usual networking and security infrastructure. This particular workload exploits JDBC, including session-based servlets and EJB components.
- **Bookstore** is modeled in spirit after major on-line book retailers. The Bookstore application exercises WebSphere Application Server, WebSphere MQ, and DB2 z/OS data sharing group. This workload includes a Web-based portal that enables users to browse for and order books. Bookstore exploits JDBC, session-based servlets, EJB components, and MQSeries, and is populated with an extremely large data set from the Library of Congress.

In addition, we have instrumented our Web applications for service level accounting metrics to better model some of our customers. We intend to use these metrics to perform baseline measurements to track potential regressions in performance. To complement that instrumentation, we have implemented a new workload driver which replaces our previous workload simulator products to better ensure that we have continuous end-to-end transaction processing capability while performing our routine administration tasks, such as maintenance, upgrades, and other systems management tasks.

In addition, we have been expanding our stock trading simulation workloads to contain end-to-end cryptographic security. This workload will enable us to better monitor and test the cryptographic assist functionality found on modern mainframes. The IBM Trade application that we have been using for some time is a suite of performance benchmarking utilities used by the WebSphere Application Server performance team which simulate an online stock trading application front end. Together, these utilities, or primitives, can be used to isolate the performance of a single J2EE component under load and expose how that component impacts the overall behavior of the application server environment. HTTP session handling, JDBC reading and writing, servlet to EJB component interaction, and many other features can all be closely examined using the Trade application.

The Trade application was developed as a general performance benchmarking program; it was not developed to support security or exercise the JAAS invoking layers of WebSphere Application Server. However, in order to better integrate and secure the Trade application, an IBM test team added a new primitive. This change allows Trade to retrieve user credential information passed to it from WebSEAL.

In the J2EE Core environment, when a client's Web browser requests to access the Trade application, WebSEAL challenges with a prompt for a secure user ID and password. After this challenge, WebSEAL acquires the credentials for the user (user name and group) from LDAP. WebSEAL augments the client's original HTTP request with these credentials by adding iv-cred information into the HTTP header. Once this step is complete, entry into the J2EE Core domain is established. Afterwards, Trade can perform application authorization for its various functions based on the user and group set in the HTTP header.

This more complex environment will help us determine regressions, suggest usability improvements, and better exercise the hardware and software platforms that we test.

## Overall configuration

Figure 77 shows the logical flow of a transaction.

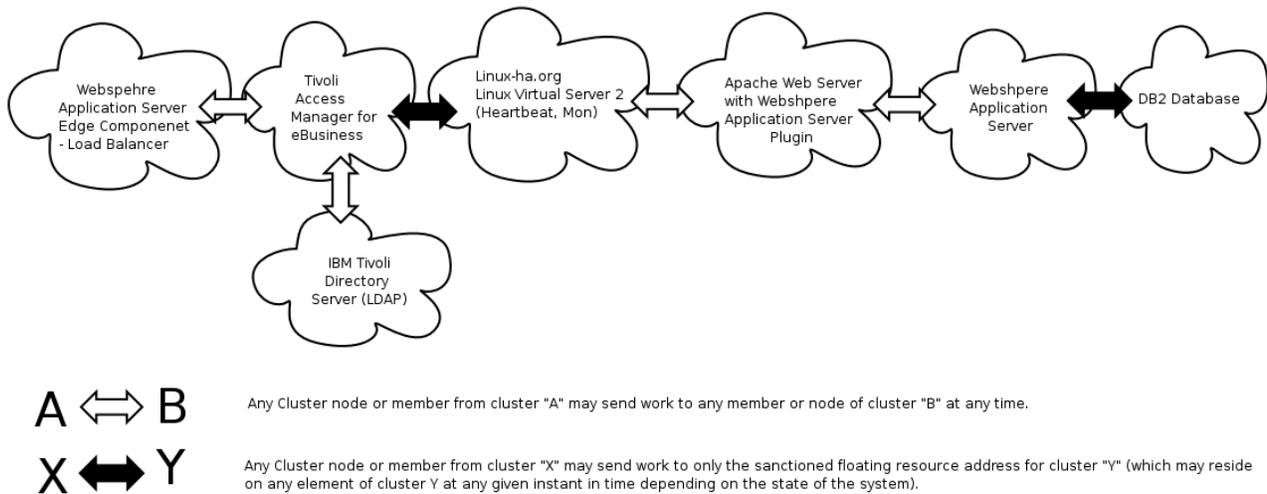


Figure 77. LVS PET application configuration: Logical transaction flow between application clusters

The clouds depict application clusters. Each cluster has members that are spread across two z/VM LPARs on two different CPCs. Cluster members of LVS Director, Apache, and WebSphere Application Server are also spread across native Linux LPARs in addition to their z/VM hosted peers, as shown in Figure 78 on page 241.

Figure 78 on page 241 shows only the production workload systems that have been in place for some time.

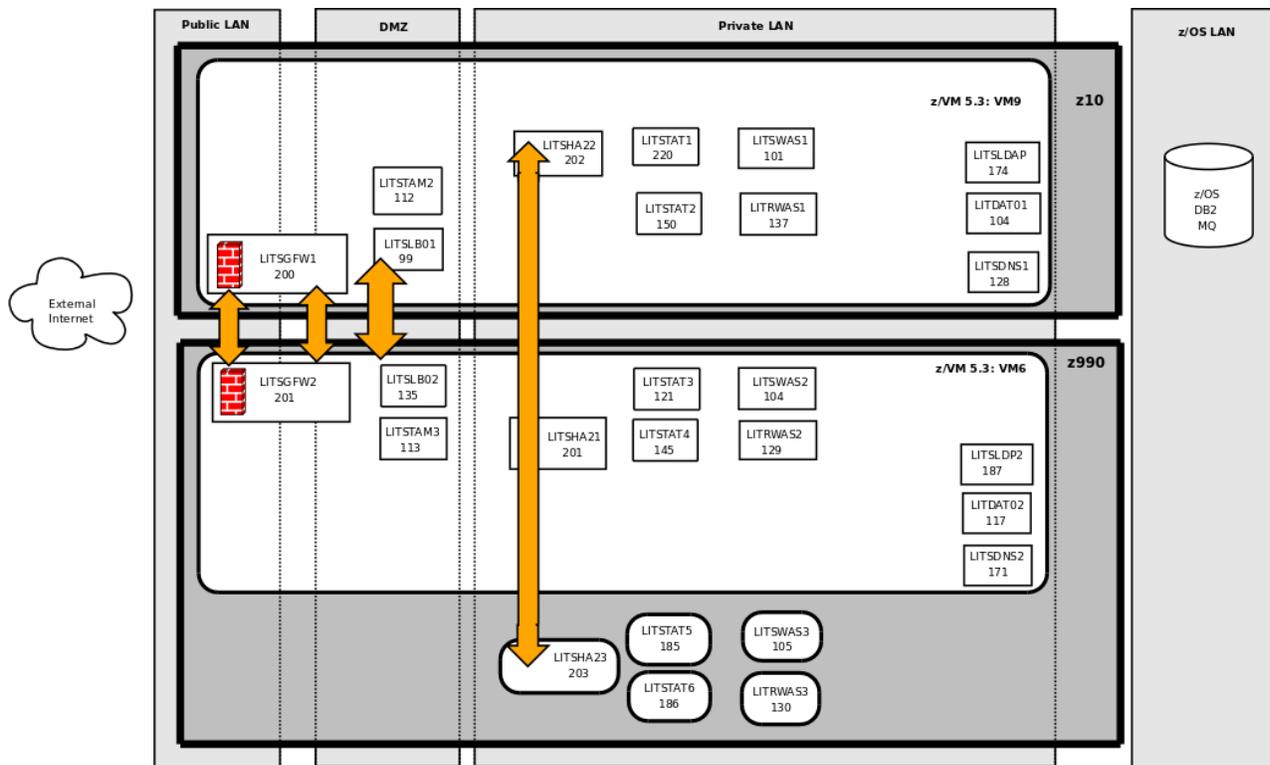


Figure 78. LVS PET system configuration: z/VM LPARs hosting Linux virtual servers on two CPCs

For a typical new transaction, the flow to access both applications is:

1. Client initiates application request from the “outside” world.
2. The request is handled by the firewall and passed to the Tivoli Access Manager for e-Business (TAME) WebSEAL cluster address.
3. IBM WebSphere Application Server Network Deployment Edge Component Load Balancer handles spraying the request to a member of the TAME WebSEAL cluster.
4. WebSEAL then asks the end user for authentication and authorization information.
5. The end user enters the authentication and authorization information.
6. WebSEAL checks against the LDAP user registry for authentication and authorization. If OK, then WebSEAL passes the request to the Apache cluster address.

**Note:** WebSEAL itself has the capability to load balance among Apache Web servers. If you are configuring Apache HA, you can do so without the Linux Virtual Server Director layer. Because we are a test team, we chose to have WebSEAL go to the Apache cluster address so that we could test Linux-ha.org’s Linux Virtual Server and heartbeat components.

7. The Linux Virtual Server Director handles spraying the request to an available Apache server.
8. The WebSphere Application Server Plug-in installed on the Apache server transfers the request to an available member in the WebSphere Application Server cluster.



Table 8. Our production Linux system names, IP addresses, and usages (continued)

Host name	IP address	Usage
litdat02	192.168.71.117	DB2 UDB - Director
litdcon1	192.168.71.104	Primary DB2 <sup>®</sup> Connect <sup>™</sup>
litdcon2	192.168.71.136	Secondary DB2 Connect
litdir00	192.168.71.249	IBM Director
lithub	192.168.75.192	Network hub
litlog1	192.168.71.110	Log server
litrsmb1	192.168.71.108	Samba server
litwas1	192.168.71.137	WebSphere Application Server - Red Hat
litwas2	192.168.71.129	WebSphere Application Server - Red Hat
litwas3	192.168.71.130	WebSphere Application Server - Red Hat (native LPAR)
litwas4	192.168.71.138	WebSphere Application Server Network Deployment - Red Hat
litsdns1	192.168.71.128	Primary DNS server
litsdns2	192.168.71.171	Secondary DNS server
litsgfw1	192.168.75.200	Primary StoneGate Firewall
litsgfw2	192.168.75.201	Secondary StoneGate Firewall
litsha21	192.168.71.201	Linux-HA Version 2 - Linux Virtual Server Director
litsha22	192.168.71.202	Linux-HA Version 2 - Linux Virtual Server Director
litsha23	192.168.71.203	Linux-HA Version 2 - Linux Virtual Server Director (native LPAR)
litslb01	192.168.74.99	Primary load balancer
litslb02	192.168.74.135	Backup load balancer
litsldap	192.168.71.114	Primary LDAP server
litsldp2	192.168.71.115	Secondary LDAP server
litsprxy	192.168.71.113	Proxy server
litstam2	192.168.74.112	IBM Tivoli Access Manager for e-business WebSEAL
litstam3	192.168.74.113	IBM Tivoli Access Manager for e-business WebSEAL
litstat1	192.168.71.220	Apache Web Server with WebSphere Application Server Plug-in
litstat2	192.168.71.150	Apache Web Server with WebSphere Application Server Plug-in
litstat3	192.168.71.121	Apache Web Server with WebSphere Application Server Plug-in
litstat4	192.168.71.145	Apache Web Server with WebSphere Application Server Plug-in
litstat5	192.168.71.185	Apache Web Server with WebSphere Application Server Plug-in (native LPAR)
litstat6	192.168.71.186	Apache Web Server with WebSphere Application Server Plug-in (native LPAR)

Table 8. Our production Linux system names, IP addresses, and usages (continued)

Host name	IP address	Usage
litstems	192.168.71.109	Tivoli Enterprise Monitoring Server
litsteps	192.168.71.119	Tivoli Enterprise Portal Server
litstsm	192.168.71.177	Tivoli Storage Manager server
litswas1	192.168.71.101	WebSphere Application Server - SUSE
litswas2	192.168.71.102	WebSphere Application Server - SUSE
litswas3	192.168.71.105	WebSphere Application Server - SUSE (native LPAR)
litswas4	192.168.71.106	WebSphere Application Server Network Deployment - SUSE
littam01	192.168.74.120	IBM Tivoli Access Manager for e-business WebSEAL

## Test (MDAT) system names and usages

Table 9 lists our test (MDAT) systems along with their host names, IP addresses, and usages. Throughout this test report, both in general discussion and in examples, we might reference these systems either by their host names or their IP addresses.

Table 9. Our MDAT system names, IP addresses, and usages

Host name	IP address	Usage
mdtswasp	192.168.71.77	WebSphere Portal Server
mdtswas1	192.168.71.73	WebSphere Application Server - Trade / HCM
mdtrwas2	192.168.71.74	WebSphere Application Server - Trade / QuickSec
mdtrihs1	192.168.71.71	IHS Server (RedHat)
mdtsihs1	192.168.71.72	IHS Server
mdtrdb21	192.168.71.75	DB2 - Trade database
mdtsdb22	192.168.71.76	DB2 - HCM database

---

## Chapter 32. Software management

Software management is a key component to the health of an enterprise. Ensuring your systems are properly updated and upgraded to remain supported is a pivotal part of day-to-day systems administration. This topic discusses our experiences in the lab with Linux distribution upgrades and the associated caveats we have run into.

---

### Maintenance strategy and methodology

Whenever applying system maintenance, it is best to perform thorough system backups and ensure you are following the recommended vendor procedures for service. In our highly available environment, we always ensure backups are performed, and only apply the update to one of the cluster systems (secondary, failover, standby) after removing it from the active configuration. How we perform our backups is detailed later on in this test report. The primary system allows the environment to remain available, but note that during the update process, two-node cluster configurations are no longer HA.

In each of the enterprise Linux distributions that we test, we have found that applying maintenance once a month has been sufficient for our needs. Naturally, your strategy will need to comply with regulations and procedures outlined in your data center guidelines and protocols.

---

### Base operating system upgrades

This topic discusses our experiences with various base operating system upgrades.

#### Upgrading the operating system on the Tivoli Storage Manager server

In our June 2008 test report, we documented the process of installing and configuring an IBM Tivoli Storage Manager (TSM) server instance running on a Linux installation on the IBM System z platform. Since then, we have upgraded to a newer release of the TSM code which required an initial upgrade of the underlying host operating system from SUSE Linux Enterprise Server 9 to SUSE Linux Enterprise Server 10. We chose to upgrade the TSM server's operating system while leaving the TSM product itself at the TSM 5.3 level. The operating system upgrade went normally, as we documented in the chapter, "Linux software management," in our June 2008 test report.

Though most of this procedure is as expected for an in-place upgrade of a SUSE Linux Enterprise Server system, there were some details peculiar to this upgrade, due to the unique hardware driver requirements, that bear mentioning. Because we upgraded to a new version of the operating system, that means we are on a new kernel level, which means we need a new IBM Tape driver to support the new kernel version.

The new SCSI tape drivers from IBM are distributed as a source (src) rpm, which means that the packages can be specifically built to work with whatever kernel level is running on the system. This removes the delay that used to occur between a Linux kernel becoming available from a distributor and IBM's release of the drivers for that kernel version.

You will either need the GCC (GNU Compiler Collection) and the requisite kernel development packages installed on the production server, or you will have to set up a separate build server and keep it at the identical kernel level as the production server.

We chose to install the GCC and kernel development packages on the TSM server itself, but this may not be acceptable in some production shops. We downloaded the `lin_tape` drivers and daemon for our 3583 library from [www.ibm.com/support/docview.wss?uid=ssg1S4000522&rs=577](http://www.ibm.com/support/docview.wss?uid=ssg1S4000522&rs=577).

The `rpmbuild` command is used to build a binary `lin_tape` rpm from the source rpm. You can then apply this binary rpm to as many systems as needed.

We used the following `rpmbuild` command to build a binary `lin_tape` rpm from the source rpm:

```
litstsm:~/tape_drivers # rpmbuild --rebuild lin_tape-1.15.0-1.src.rpm.bin
Installing lin_tape-1.15.0-1.src.rpm.bin
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.22184
+ umask 022
+ cd /usr/src/packages/BUILD
+ cd /usr/src/packages/BUILD
+ rm -rf lin_tape-1.15.0
+ /usr/bin/gzip -dc /usr/src/packages/SOURCES/lin_tape-1.15.0.tgz
+ tar -xf -
+ STATUS=0
+ '[' 0 -ne 0 ']'
+ cd lin_tape-1.15.0
++ /usr/bin/id -u
+ '[' 0 = 0 ']'
+ /bin/chown -Rhf root .
++ /usr/bin/id -u
+ '[' 0 = 0 ']'
+ /bin/chgrp -Rhf root .
+ /bin/chmod -Rf a+rX,u+w,g-w,o-w .
+ exit 0
Executing(%build): /bin/sh -e /var/tmp/rpm-tmp.44514
+ umask 022
```

```

|
| + cd /usr/src/packages/BUILD
|
| + /bin/rm -rf /var/tmp/lin_tape-1.15.0-1-root-root
|
| ++ dirname /var/tmp/lin_tape-1.15.0-1-root-root
|
| + /bin/mkdir -p /var/tmp
|
| + /bin/mkdir /var/tmp/lin_tape-1.15.0-1-root-root
|
| + cd lin_tape-1.15.0
|
| ++ echo s390x-suse-linux
|
| ++ cut -f 1 -d -
|
| + p=s390x
|
| + '[' s390x == i386 ']'
|
| + '[' s390x == i586 ']'
|
| + '[' s390x == i686 ']'
|
| + '[' s390x == ppc64 ']'
|
| + '[' s390x == powerpc ']'
|
| + '[' s390x == s390 ']'
|
| + '[' s390x == s390x ']'
|
| + proc=zSeries
|
| + '[' s390x == ia64 ']'
|
| + '[' s390x == x86_64 ']'
|
| + cp -af lin_tape_359X_zSeries.ReadMe lin_tape_359X.ReadMe
|
| + cp -af lin_tape_Ultrium_zSeries.ReadMe lin_tape_Ultrium.ReadMe
|
| + make KERNEL=2.6.16.60-0.21-default PROC=s390x driver
|
| make -C /lib/modules/2.6.16.60-0.21-default/build SUBDIRS=/usr/src/packages/BUILD/lin_tape-1.15.0
| PWD=/usr/src/packages/BUILD/lin_tape-1.15.0 clean
|
| make[1]: Entering directory `~/usr/src/linux-2.6.16.60-0.21-obj/s390/default'
|
| make -C ../../../../linux-2.6.16.60-0.21 0=~/linux-2.6.16.60-0.21-obj/s390/default clean
|
| make[1]: Leaving directory `~/usr/src/linux-2.6.16.60-0.21-obj/s390/default'
|
| rm -rf *.tgz *.ko bldtmp .** .**
|
| mkdir bldtmp
|
| make KERNEL=2.6.16.60-0.21-default compileclean lin_tape.ko

```

```

|
| make[1]: Entering directory `~/usr/src/packages/BUILD/lin_tape-1.15.0'
|
| rm -f *.o
|
| export PWD
|
| make -C /lib/modules/2.6.16.60-0.21-default/build SUBDIRS=/usr/src/packages/BUILD/lin_tape-1.15.0
| PWD=/usr/src/packages/BUILD/lin_tape-1.15.0 modules
|
| make[2]: Entering directory `~/usr/src/linux-2.6.16.60-0.21-obj/s390/default'
|
| make -C ../../../../linux-2.6.16.60-0.21 0=../../linux-2.6.16.60-0.21-obj/s390/default modules
|
| CC [M] /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape_scsi_config.o
|
| CC [M] /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape_scsi_tape.o
|
| CC [M] /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape_scsi_trace.o
|
| CC [M] /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape_ioctl_tape.o
|
| CC [M] /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape_ioctl_changer.o
|
| CC [M] /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape_extra_ioctl.o
|
| LD [M] /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape.o
|
| Building modules, stage 2.
|
| MODPOST
|
| CC /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape.mod.o
|
| LD [M] /usr/src/packages/BUILD/lin_tape-1.15.0/lin_tape.ko
|
| make[2]: Leaving directory `~/usr/src/linux-2.6.16.60-0.21-obj/s390/default'
|
| make[1]: Leaving directory `~/usr/src/packages/BUILD/lin_tape-1.15.0'
|
| mv lin_tape.ko bldtmp/lin_tape-2.6.16.60-0.21-default.ko
|
| + exit 0
|
| Executing(%install): /bin/sh -e /var/tmp/rpm-tmp.4644
|
| + umask 022
|
| + cd /usr/src/packages/BUILD
|
| + cd lin_tape-1.15.0
|
| + rm -rf /var/tmp/lin_tape-1.15.0-1-root-root
|
| + install -D -m 644 bldtmp/lin_tape-2.6.16.60-0.21-default.ko /var/tmp/lin_tape-1.15.0-1-root-root/
| lib/modules/2.6.16.60-0.21-default/kernel/drivers/scsi/lin_tape.ko
|
| + install -D -m 644 98-lin_tape.rules /var/tmp/lin_tape-1.15.0-1-root-root/etc/udev/rules.d/
| 98-lin_tape.rules

```

```

|
| + install -D -m 755 udev.get_lin_tape_id.sh /var/tmp/lin_tape-1.15.0-1-root-root/sbin/
| udev.get_lin_tape_id.sh
|
| ++ uname -m
|
| + PROC=s390x
|
| + p=s390x
|
| + '[' s390x == i386 ']'
|
| + '[' s390x == i586 ']'
|
| + '[' s390x == i686 ']'
|
| + install -D -m 700 lin_tape /var/tmp/lin_tape-1.15.0-1-root-root/etc/init.d/lin_tape
|
| + cd /var/tmp/lin_tape-1.15.0-1-root-root
|
| + ln -sf /etc/init.d/lin_tape /usr/sbin/rclin_tape
|
| + cd -
|
| /usr/src/packages/BUILD/lin_tape-1.15.0
|
| + /usr/lib/rpm/brp-lib64-linux
|
| sf@suse.de: if you find problems with this script, drop me a note
|
| + RPM_BUILD_ROOT=/var/tmp/lin_tape-1.15.0-1-root-root
|
| + export RPM_BUILD_ROOT
|
| + test -x /usr/sbin/Check -a 0 = 0 -o -x /usr/sbin/Check -a '!' -z /var/tmp/lin_tape-1.15.0-1-root-root
|
| + echo 'I call /usr/sbin/Check...'
|
| I call /usr/sbin/Check...
|
| + /usr/sbin/Check
|
| Checking permissions and ownerships - using the permissions files
|
| /tmp/Check.perms.Q14510
|
| setting /home to root:root 0755. (wrong owner/group 501:300 permissions 0777)
|
| setting /usr/src/packages/RPMS/s390x/ to root:root 1777. (wrong permissions 0755)
|
| + /usr/lib/rpm/brp-compress
|
| + /usr/lib/rpm/brp-symlink
|
| Processing files: lin_tape-1.15.0-1
|
| Executing(%doc): /bin/sh -e /var/tmp/rpm-tmp.84686
|
| + umask 022

```

```

| + cd /usr/src/packages/BUILD
| + cd lin_tape-1.15.0
| + DOCDIR=/var/tmp/lin_tape-1.15.0-1-root-root/usr/share/doc/packages/lin_tape
| + export DOCDIR
| + rm -rf /var/tmp/lin_tape-1.15.0-1-root-root/usr/share/doc/packages/lin_tape
| + /bin/mkdir -p /var/tmp/lin_tape-1.15.0-1-root-root/usr/share/doc/packages/lin_tape
| + cp -pr lin_tape_Ultrium.ReadMe /var/tmp/lin_tape-1.15.0-1-root-root/usr/share/doc/packages/lin_tape
| + cp -pr lin_tape_359X.ReadMe /var/tmp/lin_tape-1.15.0-1-root-root/usr/share/doc/packages/lin_tape
| + cp -pr COPYING COPYING.LIB /var/tmp/lin_tape-1.15.0-1-root-root/usr/share/doc/packages/lin_tape
| + exit 0
|
| Checking for unpackaged file(s): /usr/lib/rpm/check-files /var/tmp/lin_tape-1.15.0-1-root-root
|
| Wrote: /usr/src/packages/RPMS/s390x/lin_tape-1.15.0-1.s390x.rpm
|
| Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.84686
|
| + umask 022
|
| + cd /usr/src/packages/BUILD
|
| + cd lin_tape-1.15.0
|
| + rm -rf /var/tmp/lin_tape-1.15.0-1-root-root
|
| + exit 0
|
| Executing(--clean): /bin/sh -e /var/tmp/rpm-tmp.84686
|
| + umask 022
|
| + cd /usr/src/packages/BUILD
|
| + rm -rf lin_tape-1.15.0
|
| + exit 0
|
| litstsm:~/tape_drivers #

```

Note the line that starts with **Wrote:** above, as this indicates where the binary rpm is located. We then installed the rpm using the normal **rpm** process:

```
rpm -ivh /usr/src/packages/RPMS/s390x/lin_tape-1.15.0-1.s390x.rpm
```

Then, to get the driver loaded, we used the **modprobe** command:

```
modprobe lin_tape
```

We also installed the **lin\_taped** daemon package. The **lin\_taped** daemon will handle extracting tape error log dumps.

```
rpm -ivh lin_taped-1.15.0-sles10.s390x.rpm

chkconfig --add lin_tape

/etc/init.d/lin_tape start
```

After installing these packages and loading the driver and daemon, we restarted TSM and ran it for a week to make sure that the underlying operating system and tape drivers were sound. After we were confident that there were no observable regressions, we moved on to the TSM product upgrade, which we discuss in “Tivoli Storage Manager server upgrade” on page 258.

## Upgrading WebSphere Application Server prior to an operating system upgrade

In our June 2008 test report, we documented how we upgraded the operating system on our systems to a current, supported level and verified that the applications running on the older 6.0.x level of WebSphere Application Server on those systems were not affected. We then went ahead and upgraded WebSphere Application Server to its current, supported level (6.1.x). We also wanted to try the reverse: upgrading the application server first, then upgrading the operating system. We decided to leave a handful of systems at the prior operating system release (Red Hat Enterprise Linux 4 and SUSE Linux Enterprise Server 9) and upgraded the applications first. The WebSphere Application Server 6.0.x level that we were working with was supported on the older operating systems so, as expected, everything still functioned properly.

## z/VM 5.3 to z/VM 5.4 transition notes

When we moved from z/VM 5.3 to z/VM 5.4, it appears that the default for one of the IBM Directory Maintenance for z/VM (DirMaint™) to Resource Access Control Facility (RACF) integration parameters changed between the z/VM 5.4 Early Support Program (ESP) and general availability (GA) time. As we created Linux machines using DirMaint, we were not getting the associated RACF entries created anymore. After some investigation, we found that the z/VM default user group was changed at some point between the beginning of the ESP and GA to a user group that does not exist on our systems. To remedy this, we simply edited the DirMaint CONFIGRA DATADVH file and placed all the Linux users into the SYS1 group. This may not be appropriate for some production environments.

In the CONFIGRC DATADVH file on DirMaint’s C-disk, we altered the following line to set the DFLTGRP value to SYS1:

```
RACF_ADD_USER_DEFAULTS= UACC(NONE) DFLTGRP(SYS1)
```

---

## Middleware upgrades

Our middleware product upgrades included WebSphere Application Server Network Deployment and Tivoli Storage Manager.

## Upgrading our application servers and deployment manager

We upgraded WebSphere Application Server Network Deployment from version 6.0.2.27 to version 6.1.0.0. Our environment is made up of the deployment manager server LITRWAS4 and the application servers LITWAS1, LITRWAS3, LITSWAS1, LITSWAS2, LITSWAS3.

## Planning the upgrade

We decided to start the upgrade process with our deployment manager, LITRWAS4, for a number of reasons, in addition to IBM's recommendation that you start with the deployment manager. For example, by starting with the deployment manager, we were able to become familiar with the upgrade process without risk of damage to one of our application servers that process the work in our simulated production shop. In addition, if we did inadvertently damage the deployment manager, the application servers would continue to process requests. Secondly, the deployment server is easier to upgrade because it does not have any applications that need to be migrated. This simplifies the process and reduces the stress of the initial upgrade thus making the deployment manager a good candidate with which to start.

The source for our upgrade was provided to us on an FTP server, so we had to download the source locally to our system as a tar file. (Your upgrade might be provided on a CD or DVD that you can mount on a network file system.) The tar file was expanded to /opt/WAS61 and the install script was placed in /opt/WAS61/WAS. Before we could start the upgrade, we needed to shut down the deployment manager on LITRWAS4.

## Starting the upgrade

The upgrade is started by running the `./install` command from the installation directory, /opt/WAS61/WAS in our case. This launched the installation/upgrade GUI. After accepting the software license agreement, we were prompted to install the sample applications. We choose not to install these samples because this was an established environment with our own applications in place; therefore, we saw no benefit in loading the samples.

Next we were prompted to select the components we would like to install. For LITRWAS4, we selected the Deployment Manger.

The remaining step in the upgrade of the deployment manager was straight forward. We followed the prompts in the GUI and the installation completed without any errors or complications. After the upgrade completed, we used the version checker provided with the upgrade to verify that the deployment manager was at the correct level by issuing the command: `versionInfo.sh`

**Note:** This example was captured after we placed additional service on the 6.1 installation, so it displays a level of 6.1.0.15 rather than 6.1.0.0.

```
[root@litwas4 ~]# /opt/IBM/WebSphere/AppServer1/bin/versionInfo.sh
```

```
WVER0010I: Copyright (c) IBM Corporation 2002, 2005; All rights reserved.
WVER0012I: VersionInfo reporter version 1.15.1.14, dated 11/17/06
```

```

IBM WebSphere Application Server Product Installation Status Report

```

```
Report at date and time October 31, 2008 10:49:21 AM EDT
```

```
Installation
```

```

Product Directory /opt/IBM/WebSphere/AppServer1
Version Directory /opt/IBM/WebSphere/AppServer1/properties/version
DTD Directory /opt/IBM/WebSphere/AppServer1/properties/version/dtd
Log Directory /opt/IBM/WebSphere/AppServer1/logs
Backup Directory /opt/IBM/WebSphere/AppServer1/properties/version/nif/backup
TMP Directory /tmp
```

```
Product List
```

```

ND installed
```

```
Installed Product
```

```

Name IBM WebSphere Application Server - ND
Version 6.1.0.15
ID ND
Build Level cf150808.12
Build Date 2/28/08

```

```
End Installation Status Report
```

```

[root@litrwas4 ~]#
```

After verifying that the level was correct, we updated our startup scripts to point to the newly installed code in `/opt/IBM/WebSphere/AppServer1/profiles/RHELDmgr/bin`. Our startup script resides in `/etc/init.d` and contains the following.

```
[root@litrwas4 init.d]# cat wasstart
/opt/IBM/WebSphere/AppServer1/profiles/RHELDmgr/bin/startManager.sh
```

We were then ready to test our scripts and verify that the deployment manager starts automatically after an IPL. We started by verifying the shutdown and startup scripts with the commands:

```
service wasstart
service wasstop
```

Next, we verified that the entire IPL process was working by simply shutting down LITRWAS4 and re-IPLing the Linux system. As expected, the deployment manager started and communication to the application servers was established. This was verified by launching the Deployment Manager Integrated Solutions Console Web interface and displaying the application server nodes, as shown in Figure 80 on page 254.

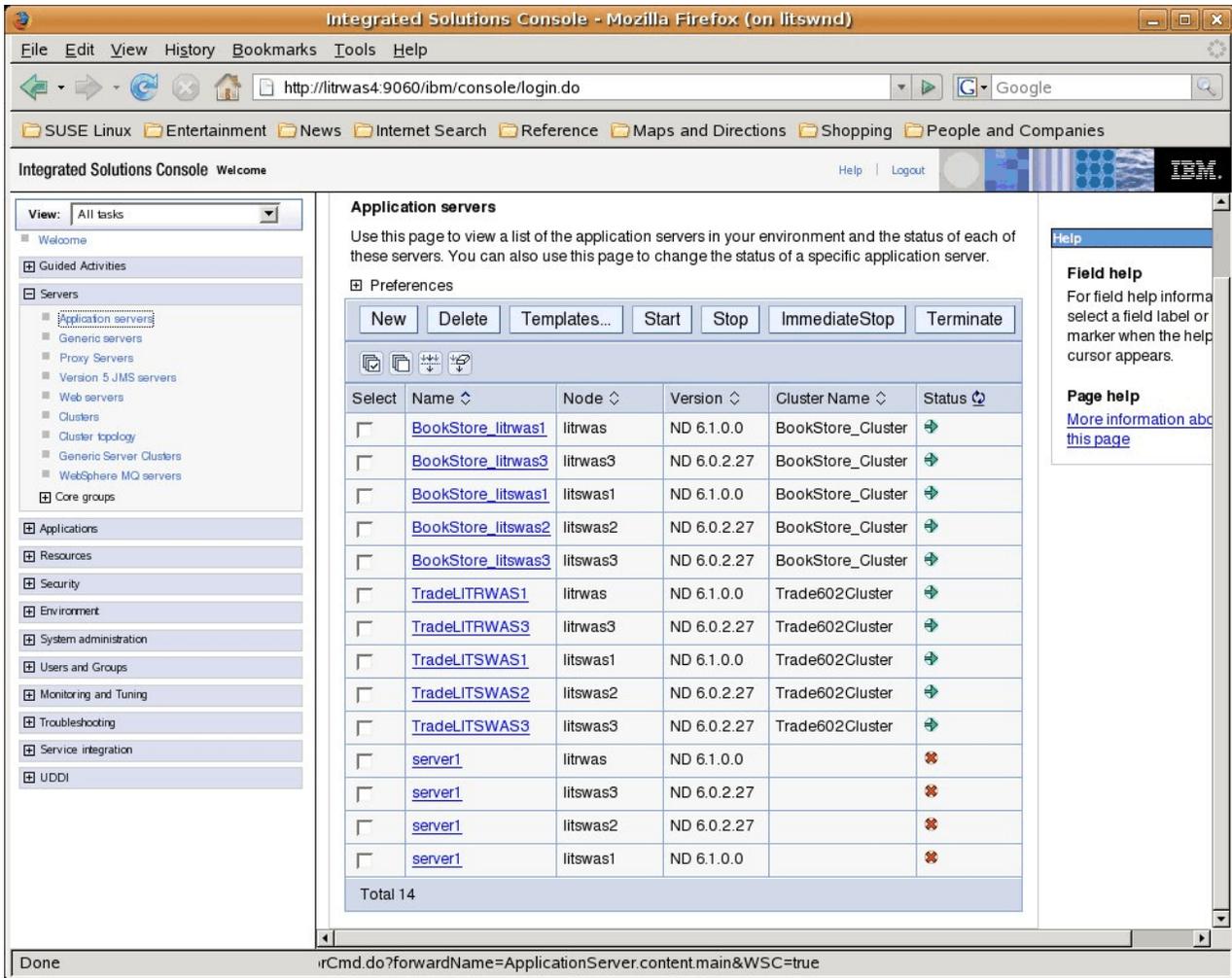


Figure 80. Example of the Integrated Solutions Console display

**Note:** This screen capture was taken after some of the application servers were upgraded to version 6.1.0.0.

Now that we had verified that the upgrade of the deployment manager was working as expected and communicating with the application servers, it was time to tackle the task of upgrading one of the application servers. Because all of our application servers are identical, it did not matter which one we chose to do first.

### Upgrading the application servers

Preparation for upgrading the application server was identical to that of the deployment manager. We downloaded and expanded the installation tar file and shut down any running WebSphere applications running on the Linux guest.

The upgrade tools for the application server are also identical to those used for the deployment manager. We started with the same `./install` script run from the installation directory. Once the install/upgrade GUI started, we again accepted the software license agreement, reviewed the system prerequisites checklist, did not select to install the sample applications, and accepted the default directory of `/opt/IBM/WebSphere/AppServer1` for all the same reasons as we did with the deployment manager.

This is where the upgrade process for the application server begins to differ from that of the deployment manager.

The next panel in the install is the “WebSphere Application server environments” selection screen. This time, we selected the application server and pressed **Next**.

The next panel requested the federation details about the deployment manager in which the application server will connect. Ours is LITRWAS4 and we use the standard port, 8879, as shown in Figure 81.

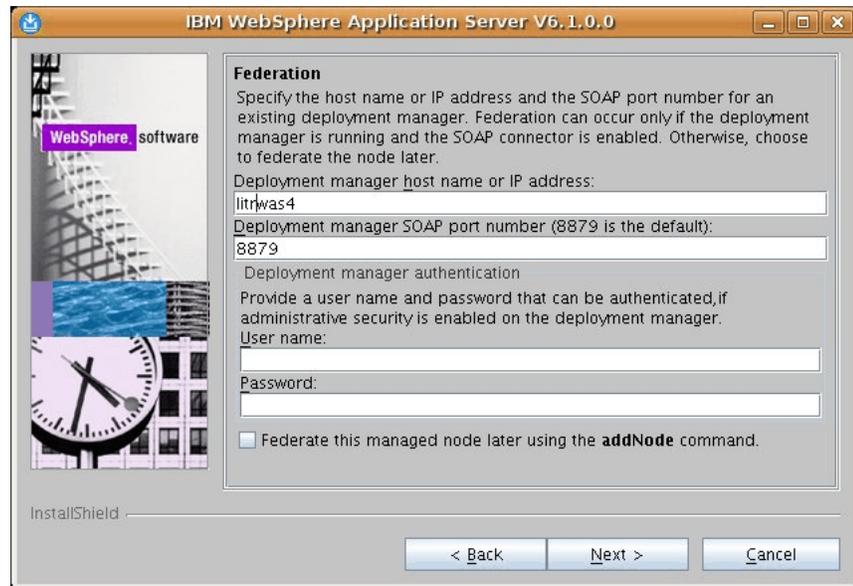


Figure 81. WebSphere Application Server V6.1.0.0 Federation panel

From this point, we followed the remaining panels to complete the installation. However, unlike the upgrade of the deployment manager, when the application server installation completes, the FIRSTSTEP migration tool is launched. We ran into some complications with the node name when using FIRSTSTEP. For this reason, we suggest that you do not use FIRSTSTEP and cancel it when it launches.

## Migrating the application server

As previously mentioned, we recommend that you cancel the FIRSTSTEP migration tool and manually launch the migration tool provided with the install. We will describe how we used the migration tool for our application server.

The migration tool GUI, `migration.sh`, resides within the `bin` directory of the product installation path specified during install. In our case, this is `/opt/IBM/WebSphere/AppServer1/bin`. When running the migration tool, we found it was easiest to `cd` to this directory first before starting the migration with the `./migration.sh` script.

```
litrwas1:/opt/IBM/WebSphere/AppServer1/bin # ./migration.sh
```

After starting the migration, you are presented with the welcome panel. Click **Next** to display the “Detected versions of WebSphere Application Server” panel. This panel, shown in Figure 82 on page 256, displays the pre-migration version of WebSphere Application Server, 6.0.2.27 in our case. (Yours might differ depending on the level of service you have applied.)

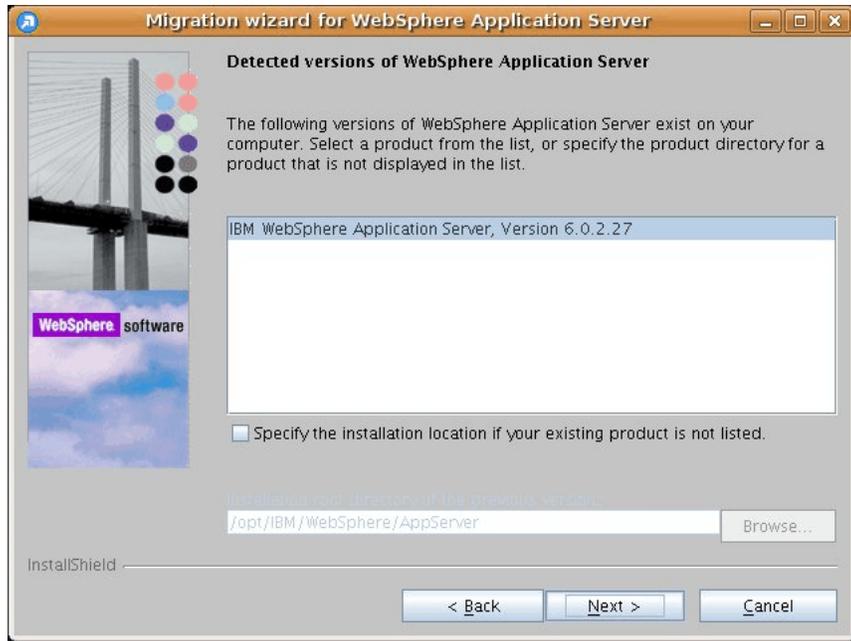


Figure 82. Migration wizard for WebSphere Application Server: Detected versions of WebSphere Application Server panel

After confirming that the pre-migration version level and directory path are correct, click **Next** to display the “Source Profile selection” panel. If the correct source profile is not displayed or available in the drop down list, click the back button to verify that the correct path has been selected on the previous panel.

After selecting the proper source profile, click **Next** to go to the “Target profile selection” panel. On the “Target profile selection” panel, select **Create new profile** and click **Next**. This displays the “Profile Creation parameters” panel, as shown in Figure 83 on page 257.

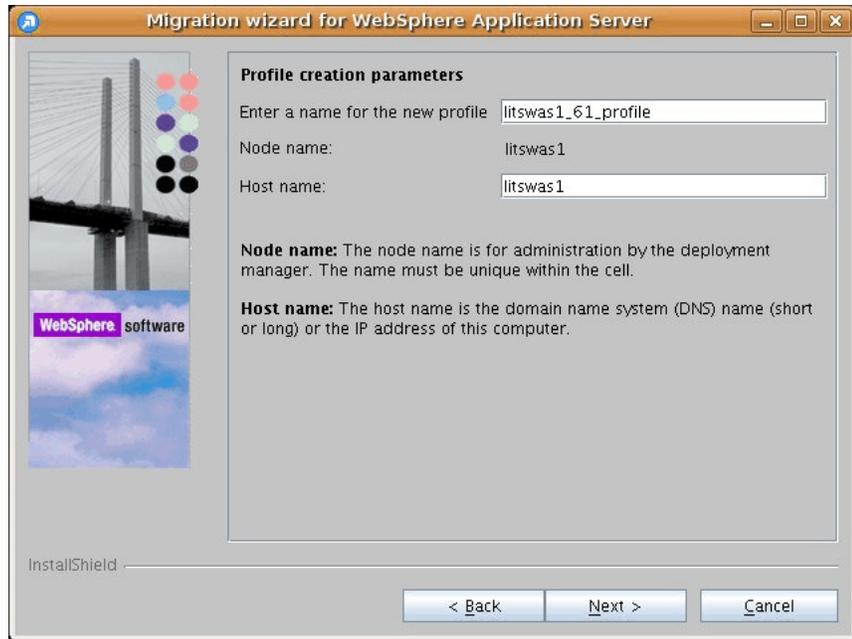


Figure 83. Migration wizard for WebSphere Application Server: Profile creation parameters panel

It is critical that you review the values in the **Node name** and the **Host name** fields on this panel. On our system the host name defaulted to the fully qualified host name of `litswas1.lab710.poughkeepsie.ibm.com`. We needed to shorten this to just the host name of `litswas1`, dropping the `lab710.poughkeepsie.ibm.com` to get the migration to work properly. If we did not do this, we encountered a node name mismatch with the deployment manager and the migration failed to complete.

From this point on, the migration went flawlessly. For your reference, contains a list of the remaining migration panels that we went through.

Table 10. Remaining panels and actions for the `migration.sh` GUI

Panel name	Action
Deployment manager verification	Verified that the Version 6.1 deployment manager is running and pressed <b>Next</b>
Migration backup directory	Used <code>/var/backup_was_profiles</code> because that file system had the most free space
Application migration settings	Selected <b>Migrate and install the applications</b>
Application migration settings	Selected <b>Install the application in the default directory of the target version</b>
Port value assignment	Selected <b>Use the port values assigned to the previous (source) installation</b>
Additional migration options	Checked <b>Migrate to support script compatibility</b>
Status panels	The remaining panels are just for status. We reviewed each panel and then continued to the next status panel.

This completed the migration of the application server. It was now time to clean up some loose ends. Just like with the deployment manger, we verified that the level of the application server was correct and updated our startup scripts to point to the newly installed code.

| To test that the application server startup scripts worked correctly, we simply shut  
| down the Linux image and re-IPLed it. As expected, the application server started  
| and was able to communicate with the deployment manager. We verified this by  
| launching the Deployment Manager Integrated Solutions Console Web interface  
| and displaying the application server nodes as we did in Figure 80 on page 254.

### **Cleaning up after the upgrades**

| After we had verified that the application servers and deployment server are  
| working properly, we wanted to remove the old version 6.0 code to save space. To  
| be safe, we backed up the deployment manager and application servers using  
| z/VM Backup and Restore Manager. (See our discussion of z/VM Backup and  
| Restore Manager in our June 2008 test report.)

| After the backup completed, we deleted /opt/IBM/WebSphere/AppServer and its  
| subdirectories. To verify that everything was still working, we shut down and  
| re-IPLed the Linux systems to verify that all services started and ran correctly  
| without intervention.

### **Tivoli Storage Manager server upgrade**

| We upgraded Tivoli Storage Manager from version 5.2.3 to version 5.5 by following  
| the directions in the IBM Tivoli Storage Manager Version 5.5 information center at  
| [publib.boulder.ibm.com/infocenter/tivihelp/v1r1/topic/com.ibm.itsmfdt.doc/  
| b\\_install\\_guide\\_linux23.htm#t\\_srvr\\_lnx\\_upg\\_overvu](http://publib.boulder.ibm.com/infocenter/tivihelp/v1r1/topic/com.ibm.itsmfdt.doc/b_install_guide_linux23.htm#t_srvr_lnx_upg_overvu). The upgrade went smoothly,  
| as described in the documentation, with no unexpected difficulties.

---

## Chapter 33. Systems management

Our recent systems management efforts have focused on:

- Installing an open source VPN server
- Altering the FSTAB and kernel to use disk by-path identification, instead of the default disk by-id
- Migrating from the ReiserFS3 file system to the Ext3 file system

---

### Installing an open source VPN server

One of the new items we have chosen to take on for systems management was the installation of a virtual private network (VPN) server which enables us to dial directly into our backend, secured systems from our local workstations.

To set up our VPN server, we chose to use Linux running on an Intel workstation wired into the private network (which connects to the numerous back-of-house Linux systems that reside on our mainframe). We chose to deploy the OpenVPN server. You can read more about the OpenVPN solution at [openvpn.net/](http://openvpn.net/).

Additionally, we opted to select a form of authentication known as Public Key Infrastructure, or PKI. PKI is a security system that uses a public or private key to authenticate the identity of people (or possible teams or entire organizations) in order to complete the secure exchange of electronic messages over a non-secure medium, such as the Internet.

To use PKI, you must also use a Certificate Authority (CA). The CA must be trusted by all parties involved and is agreed to ahead of time. The CA issues an electronic document known as a certificate which is directly associated with a key pair belonging to someone whose identity it has already verified. Later in the PKI process, other users or operating systems may rely on the authenticity of the key holder's identity based on the key they present in conjunction with their certificate. The certificates granted by the CA contain information about the holder's public key (and associated expiration date), as well as the digital signature of the CA itself.

To implement this solution for VPN connectivity we followed the instructions at [openvpn.net/index.php/documentation/howto.html#pki](http://openvpn.net/index.php/documentation/howto.html#pki).

We began by generating a CA for the OpenVPN server and generating a unique certificate and key for each user that is then signed by the server's CA. A user's access is granted based on possession of both the user's certificate and key. Access is revoked by adding the user's certificate to a revocation list which resides on the OpenVPN server.

Our OpenVPN server configuration file was located at `/etc/openvpn/ltic_server_openvpn.conf` and was configured to contain the following information:

```
Which local IP address should OpenVPN
listen on? (optional)
```

```
local 9.12.20.225
```

```

|
| # Which TCP/UDP port should OpenVPN listen on?
| # If you want to run multiple OpenVPN instances
| # on the same machine, use a different port
| # number for each one. You will need to
| # open up this port on your firewall.
|
| port 1195
|
|
| # TCP or UDP server?
|
| ;proto tcp
|
| proto udp
|
|
| # "dev tun" will create a routed IP tunnel,
| # "dev tap" will create an ethernet tunnel.
| # Use "dev tap" if you are ethernet bridging.
| # If you want to control access policies
| # over the VPN, you must create firewall
| # rules for the the TUN/TAP interface.
| # On non-Windows systems, you can give
| # an explicit unit number, such as tun0.
| # On Windows, use "dev-node" for this.
| # On most systems, the VPN will not function
| # unless you partially or fully disable
| # the firewall for the TUN/TAP interface.
|
| dev tap0
|
| ;dev tun
|
|
| # SSL/TLS root certificate (ca), certificate
| # (cert), and private key (key). Each client
| # and the server must have their own cert and
| # key file. The server and all clients will
| # use the same ca file.
| #
| # See the "easy-rsa" directory for a series
| # of scripts for generating RSA certificates
| # and private keys. Remember to use
| # a unique Common Name for the server
| # and each of the client certificates.
| #
| # Any X509 key management system can be used.
|
| # OpenVPN can also use a PKCS #12 formatted key file
|
| # (see "pkcs12" directive in man page).
|

```

```

|
| #ca keys/ca.crt
|
|
|
| ca /etc/openvpn/keys/ca.crt
|
| cert /etc/openvpn/keys/openvpn-server.crt
|
| key /etc/openvpn/keys/openvpn-server.key # This file should be kept secret
|
|
|
| # Diffie hellman parameters.
| # Generate your own with:
| # openssl dhparam -out dh1024.pem 1024
| # Substitute 2048 for 1024 if you are using
| # 2048 bit keys.
|
| dh /etc/openvpn/keys/dh1024.pem
|
|
|
| # Maintain a record of client <-> virtual IP address
| # associations in this file. If OpenVPN goes down or
| # is restarted, reconnecting clients can be assigned
| # the same virtual IP address from the pool that was
| # previously assigned.
|
| ifconfig-pool-persist /var/log/openvpn/ipp.txt
|
|
|
| # Configure server mode for ethernet bridging.
| # You must first use your OS's bridging capability
| # to bridge the TAP interface with the ethernet
| # NIC interface. Then you must manually set the
| # IP/netmask on the bridge interface, here we
| # assume 10.8.0.4/255.255.255.0. Finally we
| # must set aside an IP range in this subnet
| # (start=10.8.0.50 end=10.8.0.100) to allocate
| # to connecting clients. Leave this line commented
| # out unless you are ethernet bridging.
|
| server-bridge 192.168.71.29 255.255.255.0 192.168.71.231 192.168.71.242
|
|
|
| # Push routes to the client to allow it
| # to reach other private subnets behind
| # the server. Remember that these
| # private subnets will also need
| # to know to route the OpenVPN client
| # address pool (10.8.0.0/255.255.255.0)
| # back to the OpenVPN server.
|
| push "route 192.168.0.0 255.255.0.0"

```





```

|
|
| # It's a good idea to reduce the OpenVPN
| # daemon's privileges after initialization.
| #
| # You can uncomment this out on
| # non-Windows systems.
|
| user nobody
|
| group nogroup
|
|
| # The persist options will try to avoid
| # accessing certain resources on restart
| # that may no longer be accessible because
| # of the privilege downgrade.
|
| persist-key
|
| persist-tun
|
|
| # Output a short status file showing
| # current connections, truncated
| # and rewritten every minute.
|
| status /var/log/openvpn/openvpn-status.log
|
|
| # By default, log messages will go to the syslog (or
| # on Windows, if running as a service, they will go to
| # the "\Program Files\OpenVPN\log" directory).
| # Use log or log-append to override this default.
| # "log" will truncate the log file on OpenVPN startup,
| # while "log-append" will append to it. Use one
| # or the other (but not both).
|
| ;log openvpn.log
|
| ;log-append openvpn.log
|
|
| ;log /var/log/openvpn/openvpn.log
|
| log-append /var/log/openvpn/openvpn.log
|
|
| # Set the appropriate level of log
| # file verbosity.
| #

```

```
| # 0 is silent, except for fatal errors
| # 4 is reasonable for general usage
| # 5 and 6 can help to debug connection problems
| # 9 is extremely verbose
```

```
| verb 4
```

```
| # Silence repeating messages. At most 20
| # sequential messages of the same message
| # category will be output to the log.
```

```
| mute 5
```

```
| With the server side configuration in place, we moved on to the client
| configuration. Our client configuration resides on each client that needs to connect
| into our VPN server.
```

```
| The contents of our client configuration file, as tested, are:
```

```
| client
|
| remote 9.12.20.225 1195
|
| resolv-retry infinite
|
| proto udp
|
| dev tap
|
| pull
|
| nobind
|
| persist-key
|
| persist-tun
|
| up /etc/openvpn/update-resolv-conf
|
| down /etc/openvpn/update-resolv-conf
|
| mute-replay-warnings
|
| cd /etc/openvpn
|
| ca /etc/openvpn/keys/ca.crt
|
| cert /etc/openvpn/keys/brenneman1.crt
|
| key /etc/openvpn/keys/brenneman1.key # This file should be kept secret
|
| tls-auth /etc/openvpn/keys/ta.key 1 # This file is secret
|
| remote-cert-tls server
|
| comp-lzo
|
| verb 5
```

```
mute 5

user nobody

group nobody

fast-io

log-append /var/log/openvpn/openvpn.log

status /var/log/openvpn/openvpn-status.log 60
```

There is a Windows client that we have tested using the above server configuration. The client is available from [openvpn.se/download.html](http://openvpn.se/download.html). We opted to select this package:

Installation Package (Both 32-bit and 64-bit TAP driver included):

```
openvpn-2.0.9-gui-1.0.3-install.exe
```

On Linux systems, there are no unique client or server packages. The OpenVPN package contains both client and server code, and the appropriate code is run depending on the contents of the config file.

We have found this VPN server to be a valuable mechanism to connect into our backend systems. We have used this technology as a backup when the normal mechanism for connecting to our systems is not available. Additionally, direct connectivity of our workstations allows us to more easily test and diagnose issues residing on our private, secure, network infrastructure.

---

## Altering the FSTAB and kernel to use disk-by-path by default

Introduced with SUSE Linux Enterprise Server 10 SP1, the installation disk identification default was changed to use unique disk identifiers. On the Marist Mainframe Linux forum ([www.marist.edu/htbin/wlvindex?LINUX-390](http://www.marist.edu/htbin/wlvindex?LINUX-390)) alone, many people have raised a number of issues concerning this option when specified in a mainframe environment. For instance, if you move file systems around frequently, or if you clone your virtual Linux instances by copying their disks, you can get into trouble very quickly with the default disk-by-id setup.

We have made a request to Novell to change the default setting to by-path in future releases of SUSE Linux Enterprise Server. To address the immediate business needs in the interim, we have converted all of our systems to disk-by-path identification. The following procedure documents the straightforward mechanism for doing so.

Here is an example of what the by-id identification would look like in `/etc/zipl.conf`:

```
parameters = "root=/dev/disk/by-id/ccw-IBM.750000000M1881.2c20.32-part1 TERM=dumb"
```

If your system was built this way, you can follow these steps to change it to by-path identification:

1. Make a backup copy of `fstab` and `zipl.conf`.
2. Issue the following command to verify the disk by-path name:  

```
ls -l /dev/disk/by-path/
```
3. Modify `/etc/zipl.conf` to use by-path names. For example:

```
parameters = "root=/dev/disk/by-path/ccw-0.0.0201-part1 TERM=dumb"
```

4. Have the boot configuration pick up the changes:

```
mkinitrd
zipl
```

5. Change all by-id entries in `/etc/fstab` to by-path entries, as well. For example:

```
/dev/disk/by-path/ccw-0.0.0201-part1 / ext3 defaults 1 1
```

6. Reboot to pick up your changes.

If you are performing a new install, you can click **Fstab Options** on the Edit Partition panel and set the **Mount in `/etc/fstab`** radio button group to device name.

---

## Migrating from ReiserFS v3 to the ext3 file system

In recent months, we have completed the process of converting our ReiserFS v3 file systems to the ext3 format. There are two major motivators for this. The first involves future support and our intent to align our systems with the forward roadmap as outlined by Novell, and the second is our migration away from mod3 disks to minidisks obtained from a disk pool (which we will discuss later). Since we were copying data anyway for the disk conversion and capacity expansion, we felt the time was right to convert to the ext3 file system format.

ReiserFS v3 was, for some time, the default file system in the Novell SUSE Linux distribution. As specified on the Novell file system FAQ page at [www.novell.com/linux/filesystems/faq.html](http://www.novell.com/linux/filesystems/faq.html), there will be a strategic shift to the ext3 file system for SUSE Linux Enterprise 11 in response to customer preference. Though ReiserFS v3 file system will still be available as a file system option for new SUSE Linux Enterprise 11 file systems, we felt that converting our data would better align us with the future defaults. Novell has stated that ReiserFS v3 will be supported and maintained for the full lifetime of the SUSE Linux Enterprise 11 platform.

In addition, users updating from SUSE Linux Enterprise 10 to SUSE Linux Enterprise 11 will not experience any problems. SUSE Linux Enterprise 11 will automatically detect and use the existing file system. No conversion is needed.

The following is a reproduction of the relevant segments of the Novell file system FAQ page taken at the time of this writing:

### **Why has Novell decided to make ext3 the default file system in SUSE Linux Enterprise 11 instead of remaining with ReiserFS or choosing OCFS2?**

This change is a response to recent customer demand. Novell is seeing increasing numbers of customers who prefer ext3 to ReiserFS, and our choice of default for SUSE Linux Enterprise 11 reflects this. Today, ext3 and ReiserFS v3 are mostly on par: ext3 has very recently gained some scalability enhancements (h-trees) and online expansion support (in SUSE Linux Enterprise Server 10). Furthermore, ext3 now features more than three years of journaling hardening, which makes it competitive with ReiserFS v3. OCFS2 is a parallel cluster file system designed for specific workloads, but it is unlikely that it will have root/boot support by SUSE Linux Enterprise 11. Finally, Novell is following the development of ext4, and expects it to become a solid nextgeneration enterprise file system.

### **Is Novell abandoning ReiserFS?**

| Not at all. Novell continues to support and improve ReiserFS v3. It is also  
| included in mainline kernel distributions and will continue to receive  
| enterprise support in future SUSE Linux Enterprise distributions, including  
| SUSE Linux Enterprise 11. Customers who deploy SUSE Linux Enterprise 9 or  
| SUSE Linux Enterprise 10—and determine that ReiserFS is best for their  
| companies' applications or service use cases—should use ReiserFS. This will  
| continue to be the case in SUSE Linux Enterprise 11, as users will still be able  
| to create new file systems with it. ReiserFS v3 will be supported and  
| maintained for the full lifetime of the SUSE Linux Enterprise 11 platform.  
| Novell has always recommended using the best file system for each  
| application or service, and ReiserFS v3 is one of several supported file  
| systems in SUSE Linux Enterprise 10 and 11. In addition, users updating from  
| SUSE Linux Enterprise 10 to SUSE Linux Enterprise 11 will not experience  
| any problems. SUSE Linux Enterprise 11 will automatically detect and use the  
| existing file system. No conversion is needed.

---

## Chapter 34. Capacity management

Capacity management is an effective way to manage an IT infrastructure. The intent of capacity management is to allow the current and future business capacity requirements of your enterprise to be met in a cost-efficient fashion. Though there are multiple subspecies of capacity management, we will focus on disk capacity management. We have found that even enterprise distributions, for all their desire to remain slim and efficient, require more installation space than the traditional 3390 mod 3 disk. We will explain a tactical strategy to deal with our changing disk needs. In addition, this step will enable us to integrate with IBM Director disk pool use in the not-too-distant future.

---

### Setting up DASD groups and automatic allocation in DirMaint

Until now, many of our Linux systems have been using dedicated DASD or full-pack minidisks using the DEVNO directive on the MDISK statement in the z/VM directory. This required us to manually track which devices are in use and which are available for use. Although we do our best to keep the tracking database up to date, there have been times where reality and the database get out of sync, causing overwrites or inefficient use of forgotten DASD space. Because both the DEVNO and DEDICATED devices do not show up in a DISKMAP report, we needed to write some REXX execs to search for these volumes and manually compare the results with our tracking database—not a very efficient use of our time.

With the addition of DirMaint and storage groups, we no longer have to manually track which z/VM user or Linux guest is using which volume. We simply add the volumes to storage groups and tracking is done for us. DirMaint can generate a report telling us which volumes are in which storage groups.

We will discuss how—with a little forethought and planning—you can implement storage groups that will allow you to easily track your storage allocation and prevent orphaned DASD from being wasted due to poor tracking and prevent allocated DASD from being overwritten.

### Planning the DASD storage groups

Our environment has three distinct types of work:

1. The integration test environment, which is the basis for this test report and the one you are the most familiar with
2. Our Linux distribution test environment, where pre-released versions of Red Hat Enterprise Linux and SUSE Linux Enterprise are tested, along with a small hosting environment where we set up Linux guests for other IBM test teams
3. System volumes used for product installation and infrastructure

With storage demands differing greatly between all the environments we manage, it has been difficult to wisely plan our DASD growth, resulting in inaccurate estimates for future needs. Since we know that each environment has different properties with respect to DASD usage, we decided to go with three storage pools: One for the steady growth of the IT environment, another for the dynamic nature of both the distribution test and hosted environment, and a third for the relatively stable z/VM infrastructure. We named our pools, ITPOOL, DISTRO and GENERAL, respectively.

## Preparing DASD for use in a storage group

With the ever increasing demands for larger and larger volumes for our Linux guests and applications (such as DB2), we needed a pool that could accommodate these larger requests and still efficiently use the remaining space on the volume for smaller requests. In fact, our standard Linux installation has grown from 3339 cylinders to 10 017 cylinders. With the increased performance of the newer 2105 and 2107 DASD and the high data transfer rates obtained with FICON®, we are able to define storage pools made up of 3390 volumes with over 63 000 cylinders and still meet our performance requirements. In addition, with all our systems primarily using minidisks from storage groups, reclamation of disk space from a system that is deleted from the z/VM directory happens without any manual intervention, preventing orphaned packs due to poor housekeeping.

Now that we have decided on the type and size DASD we are going to use in the storage group, we have a couple of things we need to do to prepare it for use in a storage group. These involve formatting and labeling the volumes as PERM use.

The following steps describe how we prepared our DISTRO storage group:

1. We knew that we wanted to add thirty 63 441-cylinder volumes to this group. The 2105 Enterprise Storage Server® we plan to use is made up of six strings of DASD, each with 52 devices on the 9000 controller. In other words, we have devices 9000-9033, 9100-9133, 9200-9233, ..., 9500-9533, and each device has four FICON paths.
2. We decided to use the volume labels DTxxxx for the volumes in the DISTRO group. Rather than labeling the devices sequentially, we elected to stripe the volume labels across all six controllers. We started with the first free device, 9019 and labeled it DT0001. The next volume, DT0002, would be on device 9109, followed by DT0003 on device 9219, and so forth.  
  
We did this because, on our group definition, we will specify (ALLOCATE ROTATING). This way, add requests processed by DirMaint will allocate minidisks in a round robin fashion. If the volume labels were sequentially labeled across the device range, then allocation would load one controller first before moving to the next, and would have an overall negative impact on I/O performance.
3. To save time, rather than format all 30 volumes using CPFMTXA, we formatted and labeled a single volume and then used a REXX exec to invoke the FlashCopy service to copy that volume over to the remaining volumes and update the volume labels.

Once completed, our DASD volumes were ready to be added to the DISTRO storage group.

## Defining the pool in the EXTENT CONTROL file

Storage groups are defined in the extent control file, named EXTENT CONTROL, owned by DirMaint. Use the DIRM SEND EXTENT CONTROL command to request a copy of the extent control file to be sent to your reader, as in this example:

```
dirm send extent control
DVHXTM1191I Your SEND request has been sent for processing.
Ready; T=0.01/0.01 14:43:51
 From LTICVM6(DIRMAINT): DVHREQ2288I Your SEND request for BEYER at * has
 From LTICVM6(DIRMAINT): DVHREQ2288I been accepted.
RDR FILE 0255 SENT FROM RSCS PUN WAS 0380 RECS 0101 CPY 001 A NOHOLD NOKEEP
DMTAXM104I File (1384) spooled to BEYER -- origin LTICVM6(DIRMAINT) 11/12/08 14:47:33 EDT
 From LTICVM6(DIRMAINT): DVHREQ2289I Your SEND request for BEYER at * has
 From LTICVM6(DIRMAINT): DVHREQ2289I completed; with RC = 0.
```

After the file arrives in your reader, receive it to your local A-disk for editing.

Now that we have the extent control file, we need to update the REGIONS, GROUPS, and DEFAULTS sections of the file. Although the order of the sections is not important, we leave them in the original order.

The first thing we need to update is the DEFAULTS section at the very bottom. In this section, you define the number of cylinders available for each type of DASD that will be used on your system. Since each of our volumes will have 63 411 cylinders, we added the line 3390-63K 63441, as shown in the following example:

```
:DEFAULTS.
 * IBM supplied defaults are contained in the DEFAULTS DATADVH file.
 * The following are customer overrides and supplements.
 *
 *DASDType Max-Size
 3390 3339
 3390-30K 30051
 3390-63K 63441
:END.
```

With the DEFAULTS section updated, it is now time to add the volumes we are going to use in our storage group to the REGIONS section of the extent control file. The purpose of the REGIONS section is to define which areas of the DASD will be used for automatic allocation. As you can see in the example below, the REGIONS section is not limited to volumes that will be listed in the GROUPS section of the extent control file. For our volumes, by setting the START location to 0001, we have told DirMaint that it can use all but the first cylinder of each volume for allocation.

```

:REGIONS.
 *RegionId VolSer RegStart RegEnd Dev-Type Comments
 VMPP00 VMPP00 START END 3390-63K
 VM809E VM809E START END 3390-03
 * ADD DISTRO POOL VOLUMES HERE
 DT0001 DT0001 0001 END 3390-63K
 DT0002 DT0002 0001 END 3390-63K
 DT0003 DT0003 0001 END 3390-63K
 DT0004 DT0004 0001 END 3390-63K
 DT0005 DT0005 0001 END 3390-63K
 DT0006 DT0006 0001 END 3390-63K
 DT0007 DT0007 0001 END 3390-63K
 :
 DT0026 DT0026 0001 END 3390-63K
 DT0027 DT0027 0001 END 3390-63K
 DT0028 DT0028 0001 END 3390-63K
 DT0029 DT0029 0001 END 3390-63K
 DT0030 DT0030 0001 END 3390-63K
```

With the region for each volume defined, we can now add volumes to the GROUPS section of the extent control file. The GROUPS section is where you define which regions will be part of a particular group for automatic minidisk allocation. The example below is our definition for the DISTRO group. Note that we specified the allocation to be rotating, as described earlier, and that multiple GROUPS can be specified in this section.

```
:GROUPS.
 *GroupName RegionList
 USR191 (ALLOCATE ROTATING)
```

```

| USR191 REG000
| *GROUPNAME DISTRO TEST POOL
| DISTRO (ALLOCATE ROTATING)
| DISTRO DT0001 DT0002 DT0003 DT0004 DT0005
| DISTRO DT0006 DT0007 DT0008 DT0009 DT0010
| DISTRO DT0011 DT0012 DT0013 DT0014 DT0015
| DISTRO DT0016 DT0017 DT0018 DT0019 DT0020
| DISTRO DT0021 DT0022 DT0023 DT0024 DT0025
| DISTRO DT0026 DT0027 DT0028 DT0029 DT0030

```

There are two final steps we need to take to make the newly defined DISTRO group available for use with DirMaint. We need to write the extent control file back to DirMaint by using the DIRM FILE EXTENT CONTROL command, and reload (activate) the extent control file by using the DIRM RLDEXTN command, as shown in the following example:

**dirm file extent control**

```

| PUN FILE 0262 SENT TO RSCS RDR AS 0381 RECS 0105 CPY 001 0 NOHOLD NOKEEP
| DVHXTM1191I Your FILE request has been sent for processing.
| Ready; T=0.01/0.01 14:45:55
| From LTICVM6(DIRMAINT): DVHREQ2288I Your FILE request for BEYER at * has
| From LTICVM6(DIRMAINT): DVHREQ2288I been accepted.
| From LTICVM6(DIRMAINT): DVHRCV3821I File EXTENT CONTROL E1 has been
| From LTICVM6(DIRMAINT): DVHRCV3821I received; RC = 0.
| From LTICVM6(DIRMAINT): DVHREQ2289I Your FILE request for BEYER at * has
| From LTICVM6(DIRMAINT): DVHREQ2289I completed; with RC = 0.

```

**dirm rldextn**

```

| DVHXTM1191I Your RLDEXTN request has been sent for processing.
| Ready; T=0.01/0.01 14:46:26
| From LTICVM6(DIRMAINT): DVHREQ2288I Your RLDEXTN request for BEYER at * has
| From LTICVM6(DIRMAINT): DVHREQ2288I been accepted.
| From LTICVM6(DIRMAINT): DVHILZ3510I Starting DVHINITL with directory:
| From LTICVM6(DIRMAINT): DVHILZ3510I USER DIRECT E
| From LTICVM6(DIRMAINT): DVHILZ3510I DVHINITL Parms: BLDMONO BLDDASD BLDLINK
| From LTICVM6(DIRMAINT): DVHIZD3528W One or more DASD volume control files
| From LTICVM6(DIRMAINT): DVHIZD3528W (VM312A) were created using default
| From LTICVM6(DIRMAINT): DVHIZD3528W values for device characteristics -
| From LTICVM6(DIRMAINT): DVHIZD3528W EWLM02 0191
| From LTICVM6(DIRMAINT): DVHREQ2289I Your RLDEXTN request for BEYER at * has
| From LTICVM6(DIRMAINT): DVHREQ2289I completed; with RC = 0.

```

We are now ready to use our newly defined DASD group for automatic allocation of minidisks with the AUTOG operation of the DirMaint AMDISK command.

For more detailed information on setting up the EXTENT CONTROL file, see *z/VM Directory Maintenance Facility Tailoring and Administration Guide*.

## Allocating minidisks with DirMaint

Minidisk can be allocated with the DIRM AMDISK command in command line mode or with panels to assist you. We will discuss the command line option. To allocate a 10 017-cylinder minidisk at virtual address 201 for user LINUX001 from the DISTRO storage group, we would issue the following DirMaint command:

```
DIRM FOR LINUX001 AMD 201 3390 AUTOG 10017 DISTRO
```

In addition to all the benefits we have discussed so far, using automatic disk allocation with DirMaint allows you to quickly write REXX execs to automate a number of processes that involve minidisk creation and deletion. One such exec we created was to allow our Linux guests to send a message to Operations Manager to allocate a new volume of any size within the limitations we set in the automation

script. Doing this allows our Linux users to allocate new disk space without having to give them authorization to issue any DirMaint commands. For more information of setting up Operations Manager for z/VM, see our December 2007 test report.

## Tracking storage group utilization

One of the key factors in using automation minidisk allocation was to better track our minidisk allocations. This is very simple with the DirMaint DIRM FREE command. For example, let us say we need to know the utilization of our DISTRO storage group. We can check it at any time with the DIRM FREE G= DISTRO. If we wanted the report to include all of our storage groups we would replace DISTRO with an asterisk (\*).

The following is an example of the DIRM FREE G= DISTRO command on our system:

```
DIRM FREE G= DISTRO
```

```
FREEXT G= DISTRO
```

GROUP	REGION	VOLUME	START	SIZE	(END)	OWNER	ADDR	SA
DISTRO	DT0001	DT0001	63411	19	63429	.FREE.	0000	*
DISTRO	DT0002	DT0002	63275	155	63429	.FREE.	0000	*
DISTRO	DT0003	DT0003	63361	69	63429	.FREE.	0000	*
DISTRO	DT0004	DT0004	63337	93	63429	.FREE.	0000	*
DISTRO	DT0005	DT0005	57324	6106	63429	.FREE.	0000	*
DISTRO	DT0006	DT0006	60308	3122	63429	.FREE.	0000	*
DISTRO	DT0007	DT0007	60563	2867	63429	.FREE.	0000	*
DISTRO	DT0008	DT0008	60308	3122	63429	.FREE.	0000	*
DISTRO	DT0009	DT0009	61917	1513	63429	.FREE.	0000	*
DISTRO	DT0010	DT0010	60452	2978	63429	.FREE.	0000	*
DISTRO	DT0011	DT0011	60587	2843	63429	.FREE.	0000	*
DISTRO	DT0012	DT0012	61977	1453	63429	.FREE.	0000	*
DISTRO	DT0013	DT0013	60417	3013	63429	.FREE.	0000	*
DISTRO	DT0014	DT0014	60307	3123	63429	.FREE.	0000	*
DISTRO	DT0015	DT0015	60363	3067	63429	.FREE.	0000	*
DISTRO	DT0016	DT0016	60236	3194	63429	.FREE.	0000	*
DISTRO	DT0017	DT0017	62078	1352	63429	.FREE.	0000	*
DISTRO	DT0018	DT0018	60452	2978	63429	.FREE.	0000	*
DISTRO	DT0019	DT0019	58740	4690	63429	.FREE.	0000	*
DISTRO	DT0020	DT0020	62013	1417	63429	.FREE.	0000	*
DISTRO	DT0021	DT0021	58739	4691	63429	.FREE.	0000	*
DISTRO	DT0022	DT0022	62968	462	63429	.FREE.	0000	*
DISTRO	DT0023	DT0023	55509	7921	63429	.FREE.	0000	*
DISTRO	DT0024	DT0024	61730	1700	63429	.FREE.	0000	*
<b>DISTRO</b>	<b>DT0025</b>	<b>DT0025</b>	<b>55371</b>	<b>8059</b>	<b>63429</b>	<b>.FREE.</b>	<b>0000</b>	<b>* ◀</b>
DISTRO	DT0026	DT0026	61630	1800	63429	.FREE.	0000	*
DISTRO	DT0027	DT0027	61897	1533	63429	.FREE.	0000	*
DISTRO	DT0028	DT0028	61786	1644	63429	.FREE.	0000	*
DISTRO	DT0029	DT0029	56897	6533	63429	.FREE.	0000	*
DISTRO	DT0030	DT0030	62938	492	63429	.FREE.	0000	*

As you can see, our DISTRO group is starting to fill up. In fact, we currently cannot define a minidisk greater than 8059 cylinders. This is because the largest free block of space on a volume is only 8059 cylinders on volume DT0025. This tells us that it is time to define a few more volumes to the storage group. This simply involves labeling a few more DASD volumes and updating the EXTENT CONTROL file with the new volume information, as we described in "Defining the pool in the EXTENT CONTROL file" on page 270.

## Planning for growth

With the addition of storage pools, we have been able to accurately generate DASD allocation reports for all three storage groups, making our planning easier and more accurate. For example, in our integration test environment, we found that growth occurs at a steady, predictable rate as we expand the number of applications and servers. In our distribution test and hosting environments, we see wide fluctuations in usage depending on the type of testing being performed. System volumes and infrastructure remain fairly consistent.

By tracking our use and keeping good records, we will even be able to accurately plan how much DASD is needed for future growth in our distribution testing and hosting environments, based on historical trends.

Check out our next test report, coming in 2009, for details about how we migrated from using DEVNO and DEDICATED DASD to minidisks with AUTOG allocation.

---

## Chapter 35. Security management

For the first part of our security management upgrade, we decided to update our aging Tivoli Access Manager policy server and WebSEAL server infrastructure.

Overview of the Tivoli Access Manager products:

- **Tivoli Access Manager policy server**

Tivoli Access Manager policy server maintains the master authorization database for the management domain as well as the policy databases associated with other secure domains that you might decide to create. This server is key to the processing of access control, authentication, and authorization requests.

- **Tivoli Access Manager WebSEAL server**

Tivoli Access Manager WebSEAL is a security manager for Web-based resources. WebSEAL is a high performance, multithreaded Web server that applies fine-grained security policy to the protected Web object space.

Note that there is no built in high availability (HA) for any of the TAM components. For the IT environment, we use WebSphere Load Balancer as a front end to the TAM WebSEAL servers to make them highly available.

---

### Upgrading the Tivoli Access Manager policy server

As always, before we began our upgrade procedure, we performed a backup in the case of a failure. We did this on each of our Tivoli Access Manager systems (including both the policy servers and WebSEAL instances).

To actually perform the backup and upgrade process, we followed the “Upgrade Guide” found on the Tivoli Information Center at [publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp).

The first step was to stop the policy server so we can subsequently perform a backup:

```
littam01:/AM61/v6.1 # pd_start stop
Stopping the: Access Manager policy server.
Stopping the: Access Manager authorization server.
```

### Backing up the policy server and WebSEAL servers

The syntax for doing the backup of a halted policy server is as follows:

```
littam01:~ # pdbackup ?
Usage:
pdbackup -action backup
 -list <path to list> [-path <archive path>] [-file <archive>]

pdbackup -action restore
 -file <path to archive> [-path <restore directory>]

pdbackup -action extract
 -file <path to archive> -path <directory for extraction>

pdbackup -help -option1 -option2...
```

At this time, we also chose to make a backup of our WebSEAL systems.

The following examples show how we backed up the policy server and WebSEAL servers.

**Backing up the policy server (LITTAM01):** We issued the following command to back up the Tivoli Access Manager policy server, LITTAM01:

```
littam01:~ # pdbackup -action backup -list /opt/PolicyDirector/etc/pdbackup.lst -path /tmp -file LITTAM01.PD.BACKUP
```

The output of this operation was written to /tmp/msg\_\_pdbackup.log.

**Backing up WebSEAL server 1 (LITSTAM2):** We issued the following command to back up WebSEAL server 1, LITSTAM2:

```
litstam2:~ # pdbackup -action backup -list /opt/PolicyDirector/etc/pdbackup.lst -path /tmp -file LITSTAM2.PDWEB.BACKUP
```

The output of this operation was written to /tmp/msg\_\_pdbackup.log.

**Backing up WebSEAL server 2 (LITSTAM3):** We issued the following command to back up WebSEAL server 2, LITSTAM3:

```
litstam3:~ # pdbackup -action backup -list /opt/PolicyDirector/etc/pdbackup.lst -path /tmp -file LITSTAM3.PDWEB.BACKUP
```

The output of this operation was written to /tmp/msg\_\_pdbackup.log.

## Upgrading the policy server (LITTAM01)

The following activities describe how we upgraded the Tivoli Access Manager policy server, LITTAM01.

To begin, we verified the currently installed version of our policy server instance by using variants of the **rpm** command:

```
littam01:~ # rpm -qa |grep PD-6.0*
PDlic-PD-6.0.0-0
PDRTE-PD-6.0.0-0
PDMgr-PD-6.0.0-0
PDAclD-PD-6.0.0-0
```

The Tivoli Access Manager base components are dependent on other packages. We used the **rpm** command to verify their levels:

```
littam01:/AM61/v6.1 # rpm -qa |grep gsk
gsk7bas-7.0-3.17
```

```
littam01:~ # rpm -qa |grep TivSec
TivSecUtl-TivSec-6.0.0-0
```

```
littam01:~ # rpm -qa |grep idsldap
idsldap-clt32bit60-6.0.0-2
idsldap-cltbase60-6.0.0-2
idsldap-cltjava60-6.0.0-2
```

We performed our upgrade for the policy server using the native rpm installation utility. We upgraded the prerequisite packages first, starting with upgrading the Global Security Kit to version 7:

```
littam01:/AM61/v6.1 # rpm -Uvh gsk7bas-7.0-4.11.s390.rpm
Preparing... ##### [100%]
 1:gsk7bas ##### [100%]
```

Next we installed the Tivoli Directory Server client packages. Before doing so, we removed the existing v6.0 idsldap client packages using the **rpm -qa | grep idsldap | xargs rpm -e** command. The existing v6.0 idsldap client packages cannot be upgraded to v6.1.

```

littam01:/AM61/v6.1 # rpm -ivh idsldap-cltbase61-6.1.0-6.s390.rpm
Preparing... ##### [100%]
1:idsldap-cltbase61 ##### [100%]
littam01:/AM61/v6.1 # rpm -ivh idsldap-clt32bit61-6.1.0-6.s390.rpm
Preparing... ##### [100%]
1:idsldap-clt32bit61 ##### [100%]
littam01:/AM61/v6.1 # rpm -ivh idsldap-cltjava61-6.1.0-6.s390.rpm
Preparing... ##### [100%]
1:idsldap-cltjava61 ##### [100%]
Java tar extracted.

```

At this stage, we proceeded to upgrade the various Tivoli Security Utilities:

```

littam01:/AM61/v6.1 # rpm -Uvh TivSecUtl-TivSec-6.1.0-0.s390.rpm
Preparing... ##### [100%]
1:TivSecUtl-TivSec ##### [100%]

```

We upgraded the Tivoli Access Manager license:

```

littam01:/AM61/v6.1 # rpm -Uvh PDlic-PD-6.1.0-0.s390.rpm
Preparing... ##### [100%]
usermod: `root' is primary group name.
usermod: `root' is primary group name.
usermod: `ivmgr' is primary group name.
1:PDlic-PD ##### [100%]

```

We upgraded the Tivoli Access Manager Runtime, which contains runtime libraries and supporting files that applications can use to access Tivoli Access Manager servers:

```

littam01:/AM61/v6.1 # rpm -Uvh PDRTE-PD-6.1.0-0.s390.rpm
Preparing... ##### [100%]
1:PDRTE-PD ##### [100%]

```

Next, we upgraded the Tivoli Access Manager Policy Server:

```

littam01:/AM61/v6.1 # rpm -Uvh PDMgr-PD-6.1.0-0.s390.rpm
Preparing... ##### [100%]
1:PDMgr-PD ##### [100%]

```

We upgraded the Tivoli Access Manager Authorization Server. The authorization server provides access to the authorization service for third-party applications that use the Tivoli Access Manager authorization API in remote cache mode. The authorization server also acts as a logging and auditing collection server to store records of server activity.

```

littam01:/AM61/v6.1 # rpm -Uvh PDAclD-PD-6.1.0-0.s390.rpm
Preparing... ##### [100%]
1:PDAclD-PD ##### [100%]

```

We updated the Tivoli Access Manager schema definitions on the LDAP server:

```

littam01:/AM61/v6.1 # /opt/PolicyDirector/sbin/ivrgy_tool -h ldap01 -p 389 -D "cn=root" -w linux390 -d schema
ivrgy_tool: Attempting to add schema.
ivrgy_tool: IRA interface reports result (x'0'):
Request was successful.

```

With the updated definitions in place, we started the policy server and checked the status after the start procedure was complete:

```

littam01:/AM61/v6.1 # pd_start start
Starting the: Access Manager policy server.
Starting the: Access Manager authorization server.
littam01:/AM61/v6.1 #
littam01:/AM61/v6.1 # pd_start status

```

```

Tivoli Access Manager servers

Server Enabled Running

pdmgrd yes yes
pdacld yes yes
pdmgrproxyd no no
littam01:/AM61/v6.1 #

```

With this last step completed, our Policy server upgrade procedure was complete. We can test communication to the policy server using its command line interface, **pdadmin**. We first logged into the **pdadmin** interface using the Tivoli Access Manager admin ID and issued a policy list command to see if it returns values as shown below:

```

littam01:/AM61/v6.1 # pdadmin -a sec_master -p password
pdadmin sec_master> acl list
_WebAppServer_deployedResources_Roles_iscadmins_admin-Authz_ACL
default-webseal
_WebAppServer_deployedResources_Roles_deployer_admin-Authz_ACL
default-management-proxy
_WebAppServer_deployedResources_Roles_operator_admin-Authz_ACL
default-management
_WebAppServer_deployedResources_Roles_CosNamingWrite_naming-Authz_ACL
_WebAppServer_deployedResources_Roles_adminsecuritymanager_admin-Authz_ACL
default-root
_WebAppServer_deployedResources_Roles_administrator_admin-Authz_ACL
default-gso
_WebAppServer_deployedResources_Roles_CosNamingDelete_naming-Authz_ACL
_WebAppServer_deployedResources_Roles_CosNamingCreate_naming-Authz_ACL
_WebAppServer_deployedResources_Roles_monitor_admin-Authz_ACL
default-policy
_WebAppServer_deployedResources_Roles_CosNamingRead_naming-Authz_ACL
default-config
default-domain
_WebAppServer_deployedResources_Roles_configurator_admin-Authz_ACL
default-replica

```

The upgrade process was straightforward using the upgrade guide. We encountered no problems.

---

## Upgrading the WebSEAL server instances

The names of our WebSEAL server instances are LITSTAM2 and LITSTAM3.

Now that we updated the Tivoli Access Manager policy server to v6.1, we wanted to verify that the current WebSEAL v6.0 servers can contact the upgraded policy server. From each of the WebSEAL instances, we logged on to **pdadmin** and issued the following command:

```

litstam2:~ # pdadmin -a sec_master -p password
pdadmin sec_master> acl list
_WebAppServer_deployedResources_Roles_iscadmins_admin-Authz_ACL
default-webseal
_WebAppServer_deployedResources_Roles_deployer_admin-Authz_ACL
default-management-proxy
_WebAppServer_deployedResources_Roles_operator_admin-Authz_ACL
default-management

```

```

| _WebAppServer_deployedResources_Roles_CosNamingWrite_naming-Authz_ACL
| _WebAppServer_deployedResources_Roles_adminsecuritymanager_admin-Authz_ACL
| default-root
| _WebAppServer_deployedResources_Roles_administrator_admin-Authz_ACL
| default-gso
| _WebAppServer_deployedResources_Roles_CosNamingDelete_naming-Authz_ACL
| _WebAppServer_deployedResources_Roles_CosNamingCreate_naming-Authz_ACL
| _WebAppServer_deployedResources_Roles_monitor_admin-Authz_ACL
| default-policy
| _WebAppServer_deployedResources_Roles_CosNamingRead_naming-Authz_ACL
| default-config
| default-domain
| _WebAppServer_deployedResources_Roles_configurator_admin-Authz_ACL
| default-replica

```

```

| litstam3:~ # pdadmin -a sec_master -p password
| pdadmin sec_master> acl list
| _WebAppServer_deployedResources_Roles_iscadmins_admin-Authz_ACL
| default-webseal
| _WebAppServer_deployedResources_Roles_deployer_admin-Authz_ACL
| default-management-proxy
| _WebAppServer_deployedResources_Roles_operator_admin-Authz_ACL
| default-management
| _WebAppServer_deployedResources_Roles_CosNamingWrite_naming-Authz_ACL
| _WebAppServer_deployedResources_Roles_adminsecuritymanager_admin-Authz_ACL
| default-root
| _WebAppServer_deployedResources_Roles_administrator_admin-Authz_ACL
| default-gso
| _WebAppServer_deployedResources_Roles_CosNamingDelete_naming-Authz_ACL
| _WebAppServer_deployedResources_Roles_CosNamingCreate_naming-Authz_ACL
| _WebAppServer_deployedResources_Roles_monitor_admin-Authz_ACL
| default-policy
| _WebAppServer_deployedResources_Roles_CosNamingRead_naming-Authz_ACL
| default-config
| default-domain
| _WebAppServer_deployedResources_Roles_configurator_admin-Authz_ACL
| default-replica

```

We then stopped the WebSEAL servers and any Tivoli Access Manager servers running on the system. It is best to do the backup while the system is stopped.

```
litstam2:~ # pdweb stop
Stopping the: webseald-litstam2-WebSeal
```

```
litstam3:~ # pdweb stop
Stopping the: webseald-litstam3-WebSeal
```

We used the **pdbackup** utility on each WebSEAL server instance sequentially to backup our WebSEAL information.

First on LITSTAM2:

```
litstam2:/tmp # pdbackup -action backup -list /opt/PolicyDirector/etc/pdbackup.lst -p /tmp -file pdbackup.LITSTAM2
```

The output of this command is written to /tmp/msg\_\_pdbackup.log.

Then, on LITSTAM3:

```
litstam3:~ # pdbackup -action backup -list /opt/PolicyDirector/etc/pdbackup.lst -p /tmp -file pdbackup.LITSTAM3
```

The output of this command is written to /tmp/msg\_\_pdbackup.log.

Note that even though we specified the file names on the **pdbackup** commands as **pdbackup.LITSTAMx**, we actually get files in **/tmp** with the specified name plus a **.tar** extension because they are, in fact, in the tar format:

```
pdbackup.LITSTAM2.tar
pdbackup.LITSTAM3.tar
```

Next, we copied the pd migration backup template from the install media (the IBM Tivoli Access Manager Web Security for Linux on System z CD-ROM) to **/tmp**:

```
litstam2:/AM61/mnt/linux_s390/migrate # cp mig60to61instanceweb.lst.template /tmp/mig60to61instanceweb.lst
litstam3:/AM61/mnt/linux_s390/migrate # cp mig60to61instanceweb.lst.template /tmp/mig60to61instanceweb.lst
```

Note that, although the files in **/tmp** that were created by the previous **cp** commands contain mostly correct data, we had to first alter their permissions so that we could edit them for personalization (match the Tivoli Access Manager instance to the host on which it resides), as required:

```
litstam2:/tmp # chmod 755 mig60to61instanceweb.lst
litstam3:/tmp # chown 755 mig60to61instanceweb.lst
```

To perform the actual edit, a simple pattern replace was used in **vi**. This command must be done on each of the Tivoli Access Manager hosts. The command, as shown below, simply changed each instance of the string *instance* to *<your TAM instance name>*. On our systems, the instance names were **litstam2-WebSEAL** and **litstam3-WebSEAL**, respective to the host name of the server.

The search and replace command was issued on **/tmp/mig60to61instanceweb.lst** on each system:

```
:%s/instance/<your TAM instance name>/g
```

The WebSEAL instructions now had us run another **pdbackup** procedure referencing the modified pd migration backup template. It is not clear why they have us issue the backup again. Our best assumption is that they want the **pdbackup** to be run using a v6.1 template.

```
litstam2:/tmp # pdbackup -action backup -list /tmp/mig60to61instanceweb.lst -path /tmp/ -file pdbackup.LITSTAM2.tar
litstam3:/tmp # pdbackup -action backup -list /tmp/mig60to61instanceweb.lst -path /tmp/ -file pdbackup.LITSTAM3.tar
```

The output from these commands was written to **/tmp/msg\_\_pdbackup.log** on each system.

For upgrading WebSEAL, we used the installation wizard, **install\_amweb**, located on the IBM Tivoli Access Manager Web Security for Linux on System z CD-ROM.

As a requirement for the installation wizard, you must ensure that IBM Java Runtime (JRE) 1.5 SR5 is installed before running the installation wizard. This involved installing and then configuring the system to recognize the new installation. We performed the actual installation of the JRE as follows:

```
litstam2:/AM61/mnt/linux_s390 # rpm -ivh ibm-java2-s390-sdk-5.0-5.0.s390.rpm
Preparing... ##### [100%]
 1:ibm-java2-s390-sdk ##### [100%]

litstam3:/AM61/mnt/linux_s390 # rpm -ivh ibm-java2-s390-sdk-5.0-5.0.s390.rpm
Preparing... ##### [100%]
 1:ibm-java2-s390-sdk ##### [100%]
```

Before proceeding, we ensured that the JRE was accessible through the PATH environment variable. Note that, in many instances, you might need to add the path to your JRE installation to your PATH environment variable, as well as export a JAVA\_HOME variable, as demonstrated in the following commands:

```
$> export PATH=/opt/ibm/java2-s390-50/jre/bin/:$PATH
$> export JAVA_HOME=/opt/ibm/java2-s390-50/jre/
```

At this point, we were able to run the install wizard GUI:

```
$> ./install_amweb
InstallShield Wizard

Initializing InstallShield Wizard...

Searching for Java(tm) Virtual Machine...
.....
```

We were then presented with the wizard in a graphical window. Figure 84 through Figure 89 on page 283 illustrate the procedure as seen on our screen during the upgrade process.



Figure 84. IBM Tivoli Access Manager Installation dialog: Language selection panel



Figure 85. IBM Tivoli Access Manager Installation dialog: Welcome panel

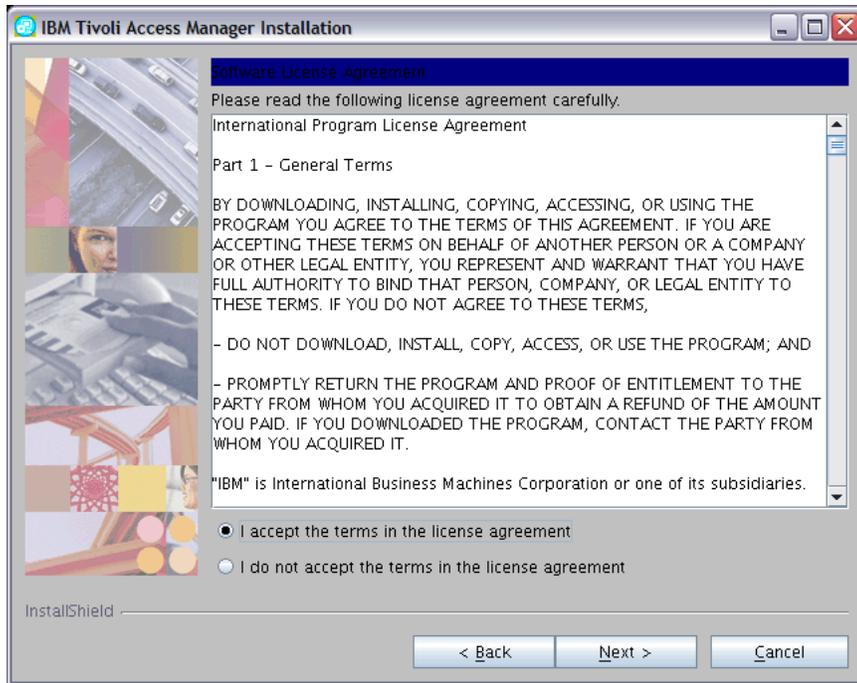


Figure 86. IBM Tivoli Access Manager Installation dialog: License agreement acceptance panel

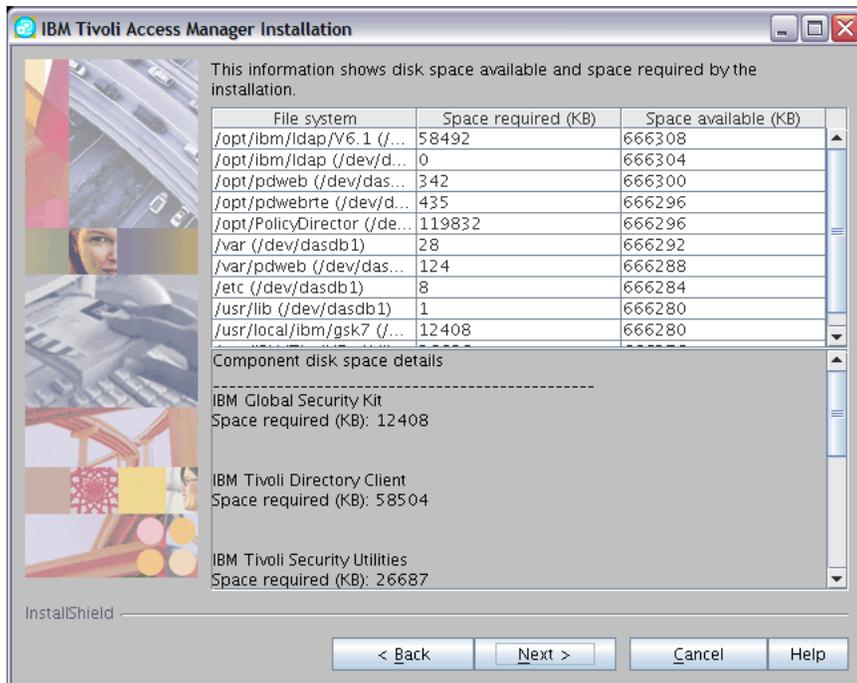


Figure 87. IBM Tivoli Access Manager Installation dialog: Disk space required and available panel

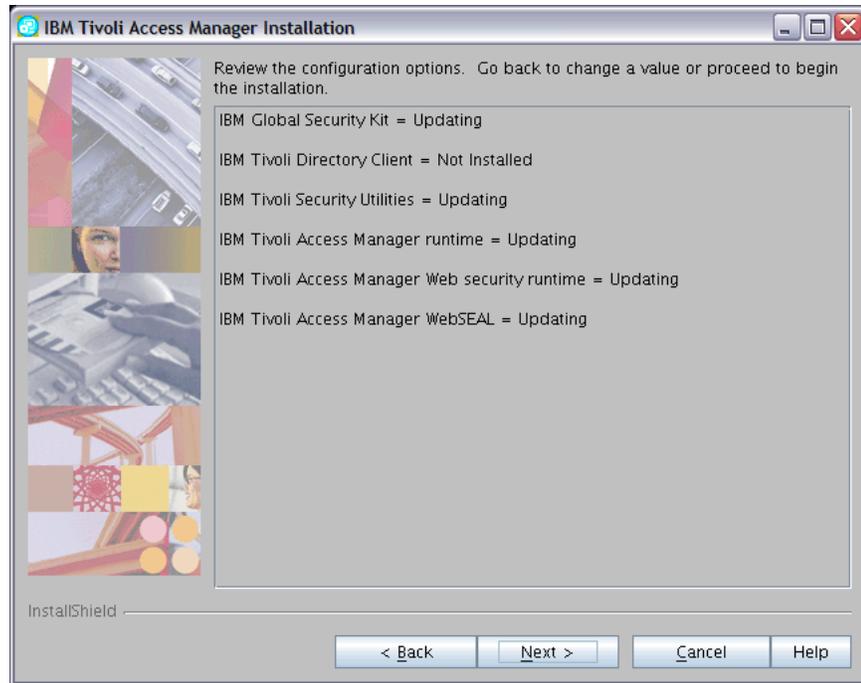


Figure 88. IBM Tivoli Access Manager Installation dialog: Review configuration options panel

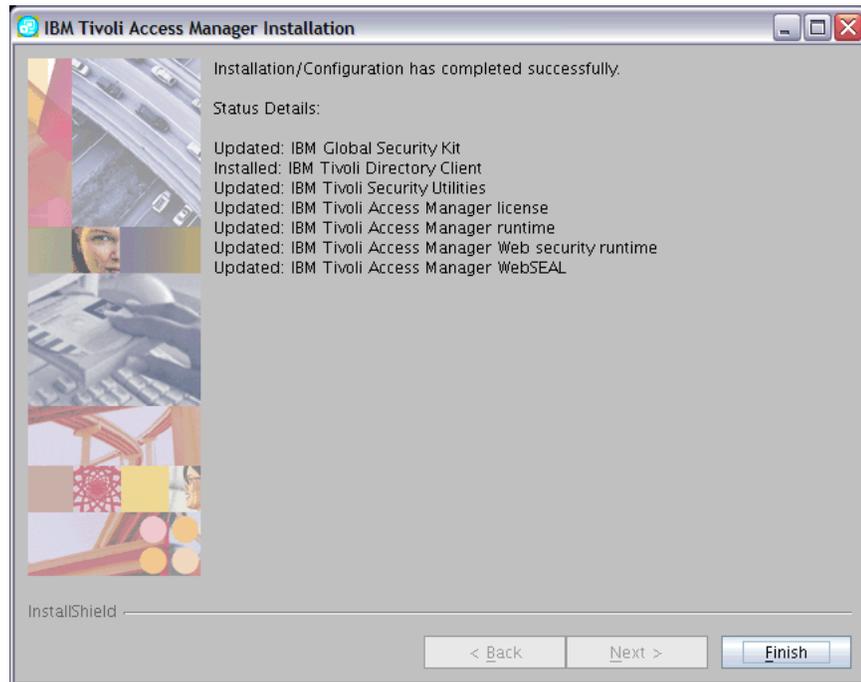


Figure 89. IBM Tivoli Access Manager Installation dialog: Installation completion panel

With the successful completion of the wizard, we started the WebSEAL daemon (webseald) manually in the foreground. This is done to force WebSEAL to migrate the various configuration files that it needs. This procedure must be done on each TAM node.

First, on LITSTAM2:

```
litstam2:/AM61/mnt # /opt/pdweb/bin/webseald -config etc/webseald-litstam2-WebSeal.conf -foreground
Access Manager WebSEAL Version 6.1.0.0 (Build 080319a)
Copyright (C) IBM Corporation 1994-2005. All Rights Reserved.
DPWAD0608I A backup copy of the configuration file has been saved using the file name '/opt/pdweb/etc/webseald-litstam2-WebSeal.conf.bak'.
DPWAD0610I WebSEAL is migrating the configuration file '/opt/pdweb/etc/webseald-litstam2-WebSeal.conf' from version 600 to version 610.
DPWAD0613I Configuration file migration was successful.
2008-11-17-12:28:26.506-05:00I----- 0x38CF0156 webseald WARNING wwa server config.cpp 1828 0x7688b6b0
DPWAA0342W The configuration data for this WebSEAL instance has been logged in '/var/pdweb/log/config_data__litstam2-WebSeal-webseald-litstam2.ltic.pok.ibm.com.log'
2008-11-17-12:28:30.446-05:00I----- 0x38AD54BA webseald WARNING wiv ssl WsSslListener.cpp 1022 0x71a98bb0
```

Then, we repeated the same process on LITSTAM3:

```
litstam3:/AM61/mnt # /opt/pdweb/bin/webseald -config etc/webseald-litstam3-WebSeal.conf -foreground
Access Manager WebSEAL Version 6.1.0.0 (Build 080319a)
Copyright (C) IBM Corporation 1994-2005. All Rights Reserved.
DPWAD0608I A backup copy of the configuration file has been saved using the file name '/opt/pdweb/etc/webseald-litstam3-WebSeal.conf.bak'.
DPWAD0610I WebSEAL is migrating the configuration file '/opt/pdweb/etc/webseald-litstam3-WebSeal.conf' from version 600 to version 610.
DPWAD0613I Configuration file migration was successful.
2008-11-17-12:31:52.513-05:00I----- 0x38CF0156 webseald WARNING wwa server config.cpp 1828 0x768886b0
DPWAA0342W The configuration data for this WebSEAL instance has been logged in '/var/pdweb/log/config_data__litstam3-WebSeal-webseald-litstam3.ltic.pok.ibm.com.log'
```

In order to confirm that we can still receive pages from the new WebSEAL daemon instance, we performed a **wget** test.

Note that the last option on the following **wget** command specifies to send a Web request to litstam2:

```
[root@lithub ~]$ wget --proxy=no --http-user=sec_master --http-passwd=password --timeout=10 --connect-timeout=30 https://litstam2/ --no-check-certificate
--13:33:01-- https://litstam2/
Resolving litstam2... 192.168.74.112
Connecting to litstam2|192.168.74.112|:443... connected.
WARNING: cannot verify litstam2's certificate, issued by `/C=US/O=IBM/OU=Tivoli Systems/CN=Test-Only':
Self-signed certificate encountered.
WARNING: certificate common name `Test-Only' doesn't match requested host name `litstam2'.
HTTP request sent, awaiting response... 200 OK
Length: 535 [text/html]
Saving to: `index.html'

100%[=====] 535 --.-K/s in 0s

13:33:01 (170 MB/s) - `index.html' saved [535/535]
```

The output of the index.html file should be inspected to ensure it is correct (a simple **cat** is usually sufficient).

We then checked for litstam3 in the same way:

```
[phil@lithub ~]$ wget --proxy=no --http-user=sec_master --http-passwd=password --timeout=10 --connect-timeout=30 https://litstam3/ --no-check-certificate
--13:34:34-- https://litstam3/
Resolving litstam3... 192.168.74.113
Connecting to litstam3|192.168.74.113|:443... connected.
WARNING: cannot verify litstam3's certificate, issued by `/C=US/O=IBM/OU=Tivoli Systems/CN=Test-Only':
Self-signed certificate encountered.
WARNING: certificate common name `Test-Only' doesn't match requested host name `litstam3'.
HTTP request sent, awaiting response... 200 OK
Length: 535 [text/html]
Saving to: `index.html.1'

100%[=====] 535 --.-K/s in 0s

13:34:34 (170 MB/s) - `index.html.1' saved [535/535]

13:31:22 (255 MB/s) - `index.html' saved [535/535]
```

Again, the output of the index.html file should be inspected to ensure it is correct.

Once we confirmed that the WebSEAL servers work as expected, we shut them down (using a CTRL -C) and restart them normally so that they run in the background as a daemon, as follows:

```
litstam2:/AM61/mnt # pdweb start
Starting the: webseald-litstam2-WebSeal
```

```
litstam3:/AM61/mnt # pdweb start
Starting the: webseald-litstam3-WebSeal
```

Our Tivoli Access Manager WebSEAL server upgrade completed successfully. We found the upgrade instructions relatively straightforward, as well. The only confusing area that we found in the documentation, as we noted earlier, was having to run a second `pdbbackup`.

---

## Migrating IBM Tivoli Directory Server from Version 6.0 to Version 6.1

IBM Tivoli Directory Server Version 6.0 for Linux on System z runs in 31-bit mode and is packaged with a 31-bit DB2 ESE V8.1, which is only supported on 31-bit systems. Consequently, our Tivoli Directory Server 6.0 servers and their corresponding DB2 instances are located on SUSE Linux Enterprise Server 9 31-bit images.

IBM Tivoli Directory Server Version 6.1 for Linux on System z runs in 64-bit mode and is packaged with DB2 ESE V9.1 FP2, which also runs in 64-bit mode. Our plan was to go from two peer-replica Tivoli Directory Server 6.0 servers on SUSE Linux Enterprise 9 to two peer-replica Tivoli Directory Server 6.1 servers on SUSE Linux Enterprise 10, as follows:

1. Install two new Tivoli Directory Server 6.1 server instances on SUSE Linux Enterprise 10.
2. Copy the LDAP database from one of the Tivoli Directory Server 6.0 servers to one of the Tivoli Directory Server 6.1 servers.
3. Configure peer replication between the two Tivoli Directory Server 6.1 servers.
4. Point the Tivoli Access Manager servers and WebSphere servers to the new Tivoli Directory Server 6.1 servers.

For step 1, we used the Tivoli Directory Server 6.1 CD-ROM images that are packaged with Tivoli Access Manager 6.1. These CD-ROM images contain an installation wizard called `install_ldap_server`. This is a GUI installer which prompts for information in a series of panels, then performs the installation of Tivoli Directory Server 6.1 and the prerequisite products, GSKit and DB2. The installer also configures DB2 and Tivoli Directory Server instances, configures the LDAP database, adds the Tivoli Access Manager default suffix (`secAuthority=Default`) and a user-defined suffix, and configures LDAP to use a key database for SSL.

The result is an Tivoli Directory Server instance that is ready for use with Tivoli Access Manager.

For details about performing a Tivoli Directory Server 6.1 installation using the CD-ROM images supplied with Tivoli Access Manager, see [publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/am61\\_install.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/am61_install.htm).

Figure 90 on page 286 shows the Tivoli Directory Server 6.1 configuration summary panel for our installation.

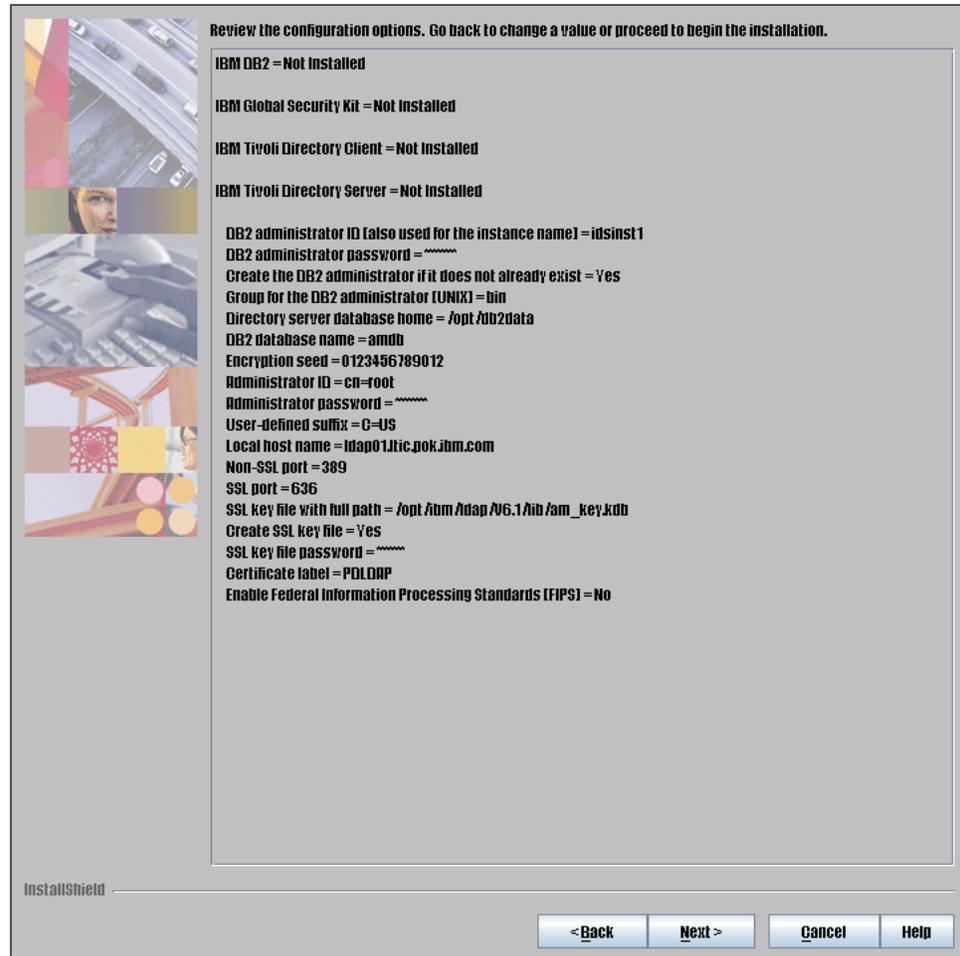


Figure 90. IBM Tivoli Directory Server 6.1 installation dialog: Configuration summary panel

For step 2, above, we used the **db2ldif** and **ldif2db** utilities to copy the contents of a Tivoli Directory Server 6.0 LDAP database to a Tivoli Directory Server 6.1 database. We noted the following information in *IBM Tivoli Directory Server Version 6.1 Command Reference* under the **db2ldif** utility:

When the source server (the server you are exporting data from) and the destination server (the server into which you are importing the data) are using non-matching directory key stash files, and you specify the encryption seed and salt values of the destination server, any AES-encrypted data is decrypted using the source server's AES keys, then re-encrypted using the destination server's encryption seed and salt values. This encrypted data is stored in the LDIF file.

We determined the seed and salt values of the destination server to use when running the **db2ldif** command on the source server. The seed value was specified during the installation of the Tivoli Directory Server 6.1 server and was entered into the installation wizard. Its value is 0123456789012. The salt value is a randomly generated value used to generate AES encryption keys. We found its value on the destination server by looking at the `cn=crypto,cn=localhost` entry:

```
dn: cn=crypto,cn=localhost
cn: crypto
objectclass: ibm-cryptoConfig
objectclass: ibm-slapedConfigEntry
```

```

objectclass: top
ibm-slapdCryptoSync: aaDNO/uKBRgNncrV/w==
ibm-slapdCryptoSalt: HY4#!!>c-!
ibm-entryuuid: 4f0379c0-a029-102c-9c87-ac6dfe0640f5

```

The non-alphanumeric characters in the salt value must be preceded by a backslash in the **db2ldif** command. The **db2ldif** command used on our source (Tivoli Directory Server 6.0) server was:

```
db2ldif -o litsldp2.080804.ldif -k 0123456789012 -t HY4#\!|\!|\>c-\{\!
```

GLPD2L011I 6848 entries have been successfully exported from the directory.

For details about using the **db2ldif** and **ldif2db** commands, see *IBM Tivoli Directory Server Command Reference* at [publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDS.doc/commandref.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDS.doc/commandref.htm).

Before loading the ldif data into one of the Tivoli Directory Server 6.1 servers, we first had to remove the entries dealing with the replication between the two Tivoli Directory Server 6.0 servers. Failure to do this would lead to the 6.1 server attempting to replicate, resulting in errors. We removed entries with `objectclass: ibm-replicationagreement` from the ldif file. For example:

```

dn: cn=litsldap:389,cn=litsldp2:389,ibm-replicaGroup=default,CN=IBMPOLICIES
control: 1.3.18.0.2.10.19 false:: MIQAAAD3MIQAAA3CgEAMIQAAAuBAXjcmVhdG9yc05hbW
UxhAAAABoEGENOPVRJT0xEQVAsRE
M9SUJNLERDPUNPTTCEAAAAOaOBADCEAAAAALwQPY3J1YXR1VG1tZXN0YW1wMYQAAAAYBBYyMDA2MT
AxMTE4MDk0OS4wMDAwMDBaMIQAAA4CgEAMIQAAAuBA1tb2RpZm1lcnNOYW11MYQAAAABbHDTj
1USU9MREFQLERDPUICTSxEQz1DT00whAAAADgKAQAwAAAAAC8ED21vZG1meVrpbWVzdGFtcDGEAA
AAGAQMjAwNzAzMjExODExMzYuMDAwMDAwWg==
cn: litsldap:389
objectclass: ibm-replicationagreement
objectclass: top
description: litsldap (peer master)
ibm-replicaconsumerid: 28882ec0-1122-1028-9ef0-bb1478738e3e
ibm-replicacredentialsdn: cn=litsldapbinddn,ibm-replicaGroup=default,CN=IBMPOL
ICIES
ibm-replicamethod: 1
ibm-replicaurl: ldap://litsldap:389
ibm-replicationonhold: FALSE
ibm-entryuuid: 6d8715c0-ed9f-102a-968e-a3586f20db83

```

The **ldif2db** command used on the Tivoli Directory Server 6.1 system to load the data:

```
ldif2db -i litsldp2.080804.norep.ldif -W ldif2db.output.080804
```

```

GLPRPL137I Restricted Access to the replication topology is set to false.
GLPRDB050E Attribute fn was not found in the schema definition.
GLPRDB053E Entry cn=wasadmin,DC=IBM,DC=COM violates the schema definition.
GLPRDB050E Attribute fn was not found in the schema definition.
GLPRDB053E Entry cn=tioldap,DC=IBM,DC=COM violates the schema definition.
GLPRDB052E Entry CN=IBMPOLICIES already exists.
GLPRDB052E Entry globalGroupName=GlobalAdminGroup,cn=ibmpolicies already exists.
GLPRDB052E Entry ibm-replicagroup=default,cn=ibmpolicies already exists.
GLPRDB052E Entry cn=replication,cn=IBMpolicies already exists.
GLPL2D003I ldif2db: 100 entries have been processed.
GLPRDB052E Entry dc=fit,dc=dcx already exists.
GLPL2D003I ldif2db: 200 entries have been processed.
.....
GLPRDB002W ldif2db: 6816 entries have been successfully added out of 6823 attempted.

```

**Note:** Because our LDAP servers are primarily used by Tivoli Access Manager, and because the Tivoli Directory Server 6.1 server comes with the schema

required by Tivoli Access Manager 6.0, which we are currently running, and Tivoli Access Manager 6.1, to which we will be migrating, there was no need to update the schema on our 6.1 servers. Obviously, if we had made schema updates to the 6.0 servers that needed to be carried forward to the 6.1 servers, we would have had to make those updates.

At this point, we switched our Tivoli Access Manager servers over to use the new Tivoli Directory Server 6.1 server to verify that it would work, intending to switch back to the 6.0 servers until peer replication was set up between two 6.1 servers.

Because we use SSL between our Tivoli Access Manager servers and our LDAP servers (`ibm-slappSslAuth:serverauth` is specified in the `ibmslapd.conf` file), prior to switching over to test the 6.1 server, we had to extract the signer of the server certificate used by the LDAP server and add it as a trusted certificate authority in the key database used by each Tivoli Access Manager server. In our case, the server certificate used by LDAP is self-signed, so we extracted that certificate and added it as a signer in each Tivoli Access Manager server's key database.

To switch the policy server over to using the Tivoli Directory Server 6.1 server, we made the following updates:

1. In file `/opt/PolicyDirector/etc/ldap.conf`, in the `[ldap]` stanza we changed:

```
host = litsldap.ltic.pok.ibm.com
```

to:

```
host = ldap01.ltic.pok.ibm.com
```

2. We further changed:

```
replica = litsldap2,636,readwrite,5
replica = litsldap,636,readwrite,9
```

to:

```
replica = ldap01.ltic.pok.ibm.com,636,readwrite,5
```

3. In file `/opt/PolicyDirector/etc/pd.conf`, in the `[pdrt]` stanza we changed:

```
user-reg-server = litsldap.ltic.pok.ibm.com
user-reg-host = litsldap.ltic.pok.ibm.com
```

to:

```
user-reg-server = ldap01.ltic.pok.ibm.com
user-reg-host = ldap01.ltic.pok.ibm.com
```

We also have a Tivoli Access Manager authorization server (`pdacld`) running on the policy server system, which we also switched to the new Tivoli Directory Server 6.1 server. We updated the `ivacl.d.conf` file, and in particular we changed the `[ldap]` stanza:

```
host = litsldap.ltic.pok.ibm.com
```

to:

```
host = ldap01.ltic.pok.ibm.com
```

We started the policy server and authorization server, verified that they started successfully, and verified that **pdadmin** commands worked.

To test that our WebSEAL servers could successfully switch to the new Tivoli Directory Server 6.1 server, we did the following five steps:

1. Stopped the WebSEAL server

2. Added the server certificate used by the LDAP server as a trusted certificate authority in the key database used by each WebSEAL server to communicate with LDAP using SSL

3. Updated the `/opt/PolicyDirector/etc/ldap.conf` file on each WebSEAL system  
In the `[ldap]` stanza we changed the host directive to point to the new Tivoli Directory Server 6.1 server:

```
[ldap]
enabled = yes
host = ldap01.ltic.pok.ibm.com
```

We also changed the replica statements to only use the new Tivoli Directory Server 6.1 server:

```
replica = ldap01.ltic.pok.ibm.com,636,readwrite,5
```

4. Updated the WebSEAL conf file on each WebSEAL system

In the `[ldap]` stanza, we changed the host directive to the following:

```
host = ldap01.ltic.pok.ibm.com
```

5. Started the WebSEAL server

Then, a `curl` command to WebSEAL containing a basic authentication header quickly verified that WebSEAL could successfully authenticate users with the new Tivoli Directory Server 6.1 server:

```
curl -k -v -u testuser:linux390 https://litstam3
```

```
* Server auth using Basic with user 'testuser'
```

```
> GET / HTTP/1.1
```

```
> Authorization: Basic dGVzdHVzZXI6bGludXgzOTA=
```

```
> User-Agent: curl/7.15.1 (s390x-ibm-linux) libcurl/7.15.1 OpenSSL/0.9.8a zlib/1.2.3 libidn/0.6.0
```

```
> Host: litstam2
```

```
> Accept: */*
```

```
>
```

```
< HTTP/1.1 200 OK
```

---

## Configuring LDAP replication

To prepare to configure LDAP replication, we installed the Tivoli Directory Server 6.1 Web Administration Tool into an existing WebSphere cell. We obtained the `IDSWebApp.war` file from the Tivoli Directory Server CD-ROM image #2, which is included with the Tivoli Access Manager 6.1 for Linux on System z CD-ROMs. The `.war` file is in the `/linux_s390/itds_tools` directory on the CD image.

The installation of the `IDSWebApp.war` file is straightforward using the WebSphere Administration Console. Details can be found in the appendix of *Tivoli Directory Server Version 6.1 Installation and Configuration Guide* at [publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDS.doc/install.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDS.doc/install.htm).

Our plan was to install a second Tivoli Directory Server 6.1 server (`ldap02`) and configure peer replication with our initial Tivoli Directory Server 6.1 server (`ldap01`).

We referred to the chapter about replication in *Tivoli Directory Server Version 6.1 Administration Guide* at [publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDS.doc/admin\\_gd.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDS.doc/admin_gd.htm). For reference, we also used a presentation about configuring replication from the IBM Education Assistant for Tivoli software products at [publib.boulder.ibm.com/infocenter/ieduasst/tivv1r0/index.jsp](http://publib.boulder.ibm.com/infocenter/ieduasst/tivv1r0/index.jsp). This

presentation was done using Tivoli Directory Server 6.0, but applies, with only slight variations, to Tivoli Directory Server 6.1 as well. It goes step-by-step (with audio) through the preparation for replication, configuring the replication agreements using the Web Administration Tool, copying the directory contents using **db2ldif** and **ldif2db** or **bulkload**, and starting replication.

Here are the steps we followed to enable replication:

1. Install the second directory server instance (ldap02).
2. Cryptographically synchronize the two server instances using the **idsgendirksf** utility.

The instructions in the appendix of *Tivoli Directory Server Version 6.1 Administration Guide*, under the topic “Synchronizing two-way cryptography between server instances,” say to use the **idsgendirksf** utility before starting the second directory server instance or adding any entries to the second instance. We used the **install\_ldap\_server** wizard included with Tivoli Access Manager to install our directory servers, which starts the server and adds an entry for a user-defined suffix.

To ensure that our ldap02 server was in the required state before running the **idsgendirksf** utility, we performed the following steps:

- a. Stop the ldap02 server instance.
  - b. Drop the ldap02 database using the **idsxinst** utility.
  - c. Create the ldap02 database using the **idsxinst** utility.
  - d. Run the **idsgendirksf** utility, using the salt and seed values from the ldap01 server instance, to create the **ibmslapddir.ksf** file for ldap02.
3. Add our suffixes to the ldap02 server.
  4. Start the ldap02 server.
  5. Add the ldap01 and ldap02 servers to the Web Administration Tool.
  6. Verify network communications between the ldap01 system and the ldap02 server, and vice-versa. Also, verify network communications from the WebSphere Application Server running the Web Administration Tool, and the ldap01 and ldap02 servers.

**Note:** If using SSL communication, this step will include extracting the certificate used by each directory server, and adding it as a trusted signer to the other directory server’s key database. Also, the Tivoli Directory Server certificates must be added to the trusted key store used by the application server running the Web Administration Tool.

7. Using the Web Administration Tool, connect to the ldap01 server and create a replication agreement for each suffix (or subtree, as it is called in the administration tool). The steps to create the replication agreement are done from the **Replication management > Manage topology** panels:
  - a. Select the subtree to be replicated and add it using the **Add Subtree** button.
  - b. Show the topology of the added subtree, then click **Add Master** to add ldap02 as a peer master for the subtree.
  - c. Create a credential object for the replication agreement. We chose to locate the credential object in each subtree being replicated.
  - d. Specify the bind DN and password to be used with the credential object
  - e. Use the **Additional** tab on the Add Master panel to add credential information on consumer. This adds the credential information to the ldap02 server.

- f. When prompted to select the credential to use for ldap02 (as the supplier) to connect to ldap01 (as the consumer), we selected the same credential and location as we used for the replication connection from ldap01 to ldap02.
8. Repeat step 7 for each subtree in our directory.
9. Quiesce each subtree on ldap01 so no updates occur while running **db2ldif** in the next step.
10. Run **db2ldif** on ldap01.
11. Unquiesce each subtree on ldap01 to allow updates.
12. Transfer the file created by running **db2ldif** to the ldap02 system.
13. Stop the ldap02 server.
14. Run **ldif2db** on ldap02.
15. Restart the ldap01 server and start the ldap02 server.  
The ldap01 server has to be restarted because replication credential information is stored in the `ibmslapd.conf` file, requiring a restart to become effective.
16. When created, the replication agreement for each subtree is set to the suspended state. To begin replication, using the Web Administration tool:
  - a. Log on to each server.
  - b. Go to **Replication management > Manage queues**.
  - c. Select each subtree and click **Queue details**, then click **Pending changes**.
    - If there are any pending changes, click **Skip all**.
    - If there are no changes pending, click **Cancel**.

In our case, there were no pending changes on the ldap01 server, but on the ldap02 server, there were several thousand pending changes. It seemed as if the ldap02 server was attempting to replicate each entry added by running **ldif2db**, back to ldap01. If we did not cancel these pending changes, then upon resuming replication for the subtree, we received many GLPRDB074E Replication conflict messages in the `ibmslapd.log` on ldap01, such as:

```
GLPRDB074E Replication conflict adding cn=pdwas-admin,dc=fit,dc=dcx occurred. The entry was replaced.
```

These errors did not seem to cause a problem. The queue size quickly dropped to zero, and the state changed from Active to Ready.

---

## Configuring WebSEAL for LDAP load balancing and failover

With replication enabled, we switched one of our WebSEAL servers to use ldap02 as its primary LDAP server, with ldap01 as a backup.

In the webSEAL conf file, in the `[ldap]` stanza, we changed the host parameter to point to ldap02:

```
[ldap]
##host = ldap01.ltic.pok.ibm.com
host = ldap02.ltic.pok.ibm.com
```

In the `/opt/PolicyDirector/etc/ldap.conf` file on that WebSEAL system, in the `[ldap]` stanza, we changed the host parameter to point to ldap02:

```
[ldap]
enabled = yes
##host = ldap01.ltic.pok.ibm.com
host = ldap02.ltic.pok.ibm.com
port = 389
ssl-port = 636
```

And, in the same stanza, added the following replica statements:

```
replica = ldap02.ltic.pok.ibm.com,636,readwrite,5
replica = ldap01.ltic.pok.ibm.com,636,readwrite,1
```

With these statements, WebSEAL will always attempt to use the ldap02 directory server first because it has the higher weight. If the ldap02 server is unavailable, then WebSEAL will use the ldap01 server.

Because WebSEAL connects to our directory servers using SSL/TLS, we extracted the certificate used by the ldap02 server and added it to the key databases used by each of our WebSEAL servers to establish the SSL connection with the directory servers. (This had already been done for the certificate used by the ldap01 server).

We enabled tracing on the ldap02 directory server, then restarted the WebSEAL server that we had just modified. By observing the LDAP trace, we could see that each time a user logged into WebSEAL, the authentication request was sent by WebSEAL to ldap02.

We then ran a few tests to verify that replication was working for our Tivoli Access Manager environment. Using the **pdadmin** utility on the policy server system, we changed a user's password:

```
pdadmin sec_master> user modify testuser password newpass
```

The policy server is still using ldap01 as its directory server so, by making an HTTPS request to the WebSEAL server using the ldap02 directory server and logging in with user ID *testuser* and password *newpass*, we could see that the password update to ldap01 had been replicated to ldap02.

We also created a new Access Manager user from the policy server system and verified that we could log in to the WebSEAL server using the ldap02 directory server, with that new user, verifying that the new user was replicated to the ldap02 directory server.

To test the LDAP failover scenario, we stopped the ldap02 server and verified that users could continue to log in to WebSEAL, with WebSEAL now sending the authentication requests to ldap01, the lower weighted LDAP server specified in replica statements in the ldap.conf file:

```
replica = ldap02.ltic.pok.ibm.com,636,readwrite,5
replica = ldap01.ltic.pok.ibm.com,636,readwrite,1
```

Having updated one of our WebSEAL servers to use ldap02 as its primary directory server and ldap01 as its backup, we updated the configuration on our second WebSEAL server to use ldap01 as its primary directory server and ldap02 as its backup.

In the WebSEAL conf file for the second WebSEAL server, in the [ldap] stanza, we point to ldap01 as the LDAP host:

```
[ldap]

host = ldap01.ltic.pok.ibm.com
```

In the /opt/PolicyDirector/etc/ldap.conf file on the second WebSEAL system, we point to ldap01 as the LDAP host, and, in the replica statements, we weighted ldap01 higher than ldap02:

```
[ldap]
enabled = yes
host = ldap01.1tic.pok.ibm.com
port = 389
ssl-port = 636

replica = ldap01.1tic.pok.ibm.com,636,readwrite,5
replica = ldap02.1tic.pok.ibm.com,636,readwrite,1
```

The result is that, as requests come into our Edge servers and get distributed between the two WebSEAL servers in our WebSEAL cluster, the authentication requests are also getting distributed between our two directory servers; we get a load balancing of the LDAP authentication workload.

If one of our directory servers fails, WebSEAL will automatically send authentication requests to the remaining server. In addition, recovery from an LDAP failover scenario is automatic. That is, the WebSEAL servers will periodically poll the down directory server. When that server becomes available, it will be added back to the pool of available LDAP servers with its assigned weight. If the server that has recovered has the highest weight among the LDAP replicas, then WebSEAL will begin directing LDAP requests to that server.

Our policy server is configured to use ldap01 as its primary directory server. We enabled the policy server to use ldap02 as a backup, should ldap01 be unavailable, by updating the `/opt/PolicyDirector/etc/ldap.conf` file on the policy server system with the following replica statements:

```
replica = ldap01.1tic.pok.ibm.com,636,readwrite,5
replica = ldap02.1tic.pok.ibm.com,636,readwrite,1
```

Note that the certificate used by the ldap02 server must be added to the key database used by the policy server for establishing SSL communications with LDAP.

The vast majority of updates to our LDAP directory is done from the policy manager system. With both directory servers available, updates will be made to the ldap01 server and replicated to the ldap02 server. Should the ldap01 server be unavailable, then updates would be made to the ldap02 server and replicated to ldap01 when it becomes available. Our testing confirmed this.

---

## Installing WebSphere Edge Components V7.0

We installed, configured, and tested WebSphere Edge Components: Load Balancer for IPv6 V7.0.

### Installing Load Balancer for IPv6

When updating the load balancer, there is no actual *update*. The procedure is to back up the configuration, uninstall the old load balancer rpms, and install the new ones. There is no migration path.

We opted to use this installation as an opportunity to switch from NAT to MAC now that we have implemented Layer 2 VSWITCH and OSA. Layer 2 provides us with a unique MAC address for each Linux system, which we did not have with Layer 3.

We began by creating two new SUSE Linux Enterprise Server 10 SP1 Linux systems on our z/VM instance:

- LITSLBN1 - 192.168.74.199
- LITSLBN2 - 192.168.74.235

**Note:** Our configuration involves a pair of load balancers that front a pair of WebSeal instances: LITSTAM2 (192.168.74.112) and LITSTAM3 (192.168.74.113).

Our active/passive cluster deployment uses litslbn1 as the active node and litslbn2 as the passive node.

**Note:** In your configuration, your load balancers might be sending work to some other appropriate type of Web server. The methodologies for configuration and testing will be similar.

For completeness, the following was our kernel IP routing table:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.71.0	192.168.74.251	255.255.255.0	UG	0	0	0	eth1
192.168.70.0	192.168.74.251	255.255.255.0	UG	0	0	0	eth1
192.168.74.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
169.254.0.0	0.0.0.0	255.255.0.0	U	0	0	0	eth1
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	192.168.74.100	0.0.0.0	UG	0	0	0	eth1

The initial step of the installation was to copy the installation source package, C1I7VML.tar.gz, to both LITSLBN1 and LITSLBN2 to a location for unpacking. We chose to create a new temporary directory /EDGEv7 for this purpose.

The second step was the untar and run the installation exec. For the untar process we ran the following command from within the /EDGEv7/ directory:

```
tar -zxvf C1I7VML.tar.gz
```

Note that you must have X-Window forwarding working properly before you can run the installation command, which will begin the installation process. We issued the following command:

```
litslbn1:/EDGEv7 # ./install
```

Figure 91 on page 295 through Figure 93 on page 296 illustrate the first three panels of the installation process and required no deviation from what would be expected. We include them here for completeness.

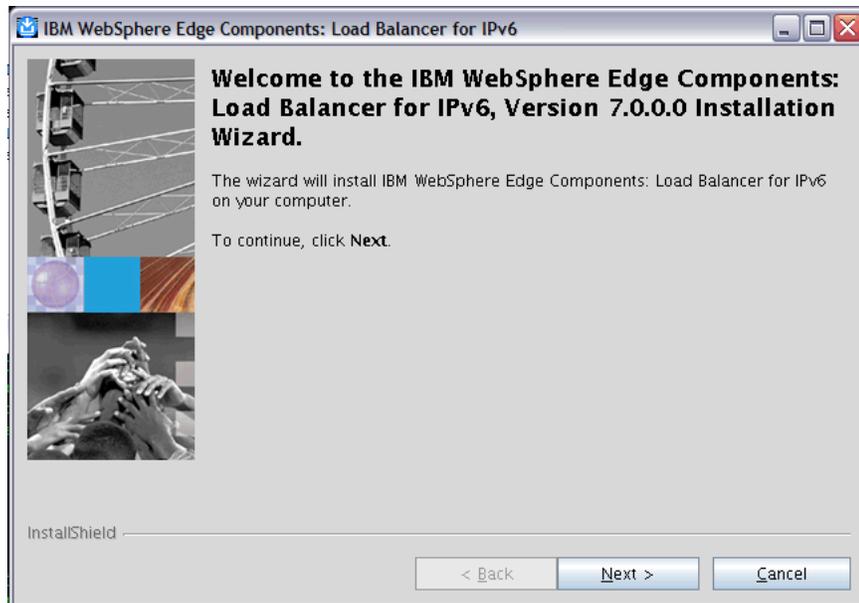


Figure 91. WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Welcome panel

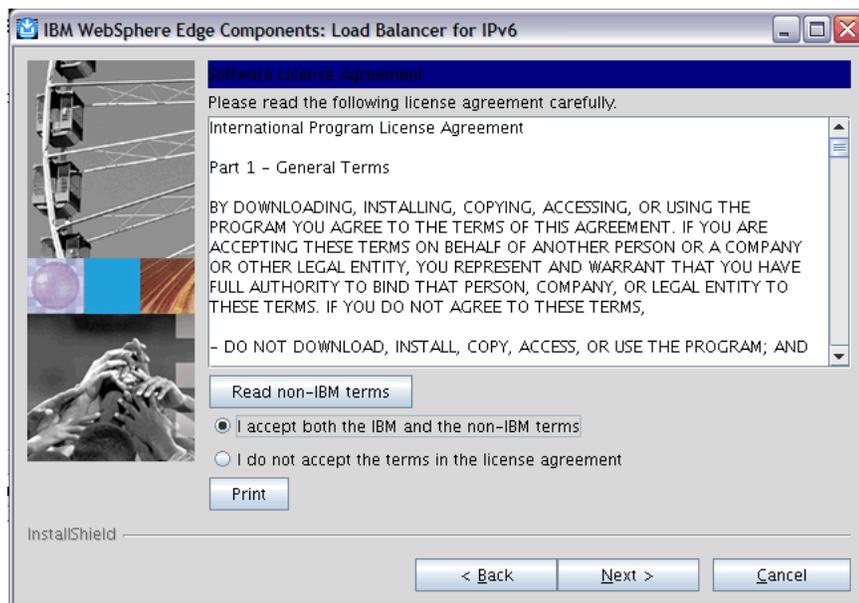


Figure 92. WebSphere Edge Components: Load Balancer for IPv6 installation dialog: License agreement acceptance panel



Figure 93. WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Language selection panel

We chose to perform a custom installation, as shown in Figure 94.

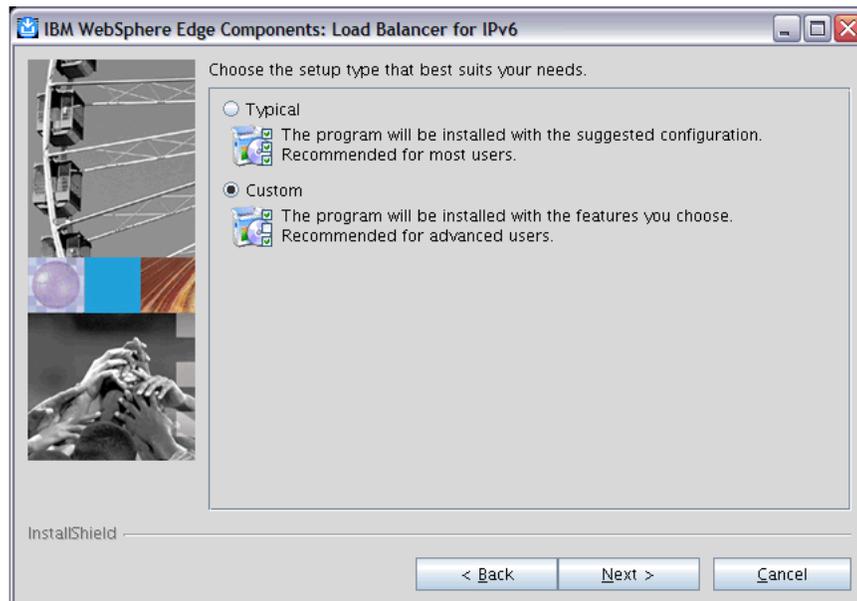


Figure 94. WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Setup type selection panel

We chose to install all of the features: License, Base, Dispatcher, and Metric Server, as shown in Figure 95 on page 297.

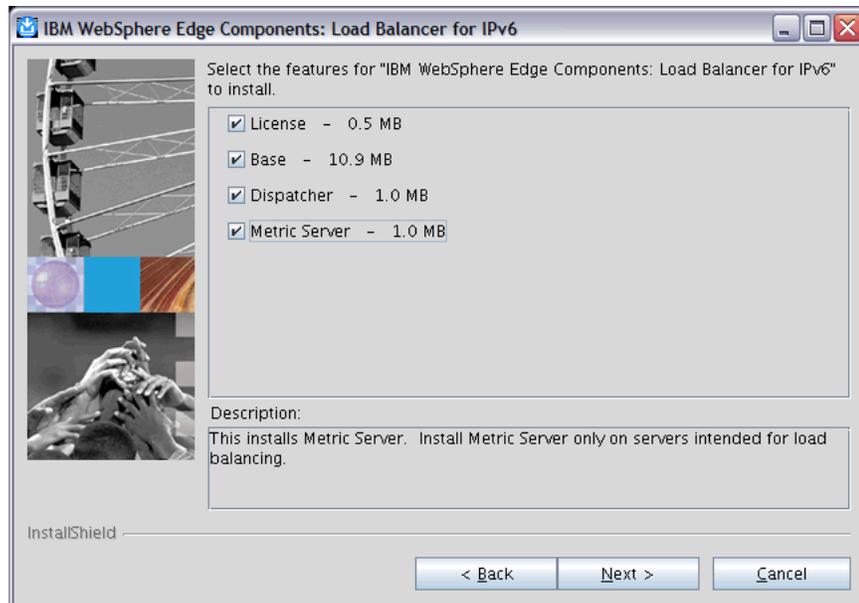


Figure 95. WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Feature selection panel

We were prompted with information about the additional disk capacity requirements necessary to comply with our package selection, as shown in Figure 96.

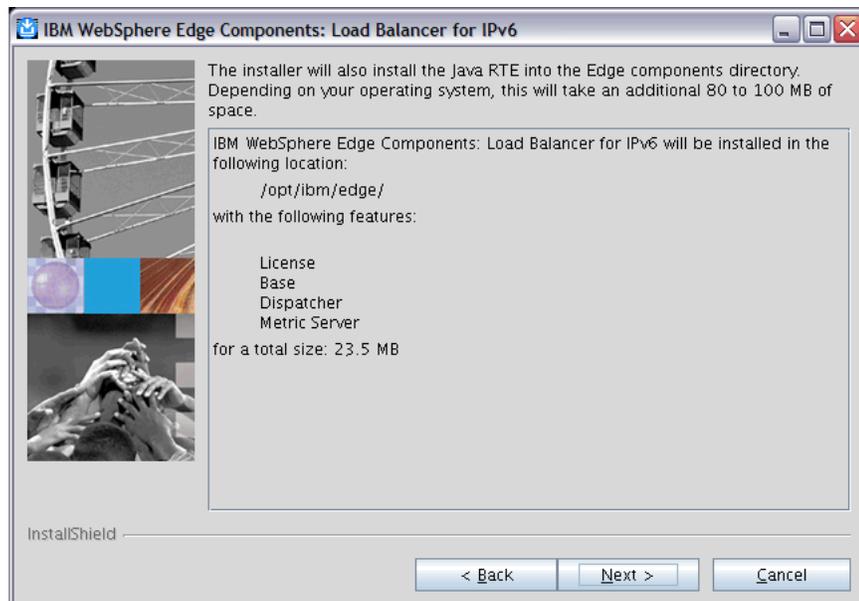


Figure 96. WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Installation options summary panel

Our installation completed successfully, as shown in Figure 97 on page 298.

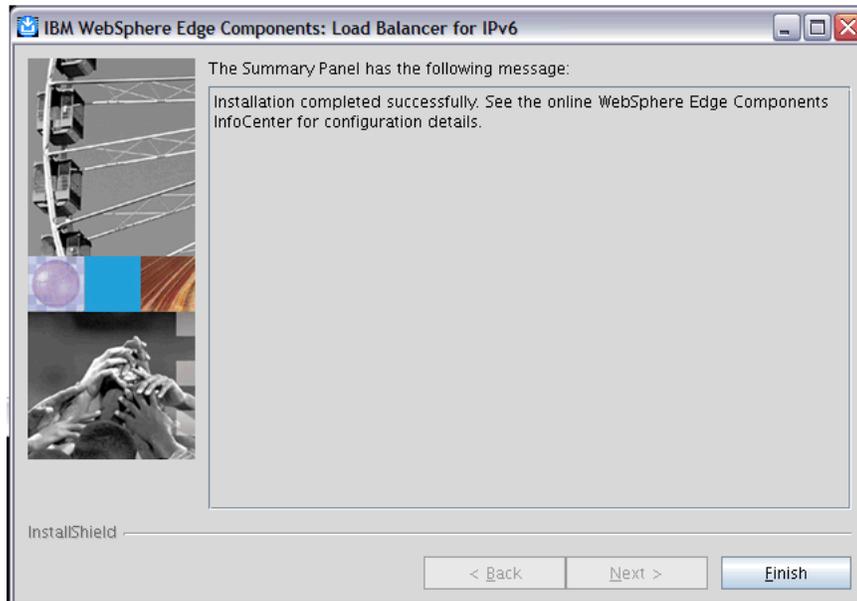


Figure 97. WebSphere Edge Components: Load Balancer for IPv6 installation dialog: Installation summary panel

## Configuring Load Balancer for IPv6

We followed the instructions at [publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.edge.doc/welcome.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.edge.doc/welcome.html) to configure Load Balancer for IPv6 V7.0.

We noticed a couple of major configuration changes between Load Balancer V7.0 and the old Load Balancer V6.1.0-0 scripts on LITSLB01 and LITSLB02:

- You might be familiar with the previous command to set up Load Balancer (**configure** argument), such as this example:  
`litslbn1:/opt/ibm/edge/ulb/servers # dscontrol executor configure 192.168.74.150`

The **configure** argument is no longer valid with version 7.0. The new valid arguments are:

EXECUTOR COMMAND ARGUMENTS:

```

report -Show executor current statistics
set <key> <value> [<k2> <v2>...] -Set fields of executor

 --- KEY --- ----- VALUE -----
 nfa <address> -Set nonforwarding address
 hatimeout <hatimeout> -Set high availability timeout
start -Start the executor
status -Show executor configurable settings
stop -Stop the executor
```

- Neither is the `set clientgateway`

- We are now using MAC forwarding, so we no longer needed to set a client gateway for the executor as we had done previously when using the NAT forwarding method on our version 6.0 installation:

```
litslbn1:/opt/ibm/edge/ulb/servers # dscontrol executor set clientgateway 192.168.74.100
Error: 'clientgateway' is not a valid executor set command.
```

```

Usage:
 set <key> <value> [<k2> <v2>...] -Set fields of executor
 --- KEY --- ----- VALUE -----
 nfa <address> -Set nonforwarding address
 hatimeout <hatimeout> -Set high availability timeout

```

We used the Dispatcher GUI to do most of the setup by issuing the command:

```
$> lbadmin
```

This GUI creates the configuration files shown below, which we opted to save in /opt/ibm/edge/ulb/servers/configurations/dispatcher/zLVS\_LB.cfg on each of our nodes. We have included the full contents of the configuration as generated and tested in our environment.

### Primary node: LITSLBN1 configuration file

```
litslbn1:/opt/ibm/edge/ulb/servers/configurations/dispatcher # cat zLVS_LB.cfg
```

```

dscontrol set loglevel 3
dscontrol executor start
dscontrol executor set hatimeout 3

dscontrol cluster add zLVS_LB_Cluster address 192.168.74.150
dscontrol cluster set zLVS_LB_Cluster proportions 49 50 1 0

dscontrol port add zLVS_LB_Cluster@80 selectionalgorithm connection
dscontrol port set zLVS_LB_Cluster@80 staletimeout 6400

dscontrol server add zLVS_LB_Cluster@80@LITSTAM3 address 192.168.74.113

dscontrol server add zLVS_LB_Cluster@80@LITSTAM2 address 192.168.74.112

dscontrol port add zLVS_LB_Cluster@443 selectionalgorithm connection
dscontrol port set zLVS_LB_Cluster@443 staletimeout 6400

dscontrol server add zLVS_LB_Cluster@443@LITSTAM3 address 192.168.74.113

dscontrol server add zLVS_LB_Cluster@443@LITSTAM2 address 192.168.74.112

dscontrol highavailability heartbeat add 192.168.74.199 192.168.74.235
dscontrol highavailability backup add primary=192.168.74.199 auto 9123
dscontrol highavailability reach add 192.168.74.100

dscontrol manager start manager.log 10004

dscontrol advisor start Http 80 HTTP_80.log

dscontrol advisor start Ssl 443 SSL_443.log
dscontrol advisor connecttimeout Ssl 443 9
dscontrol advisor receivetimeout Ssl 443 9

```

### Standby node: LITSLBN2 configuration file

```
litslbn2:/opt/ibm/edge/ulb/servers/configurations/dispatcher # cat zLVS_LB.cfg
```

```

dscontrol set loglevel 3
dscontrol executor start
dscontrol executor set hatimeout 3

dscontrol cluster add zLVS_LB_Cluster address 192.168.74.150
dscontrol cluster set zLVS_LB_Cluster proportions 49 50 1 0

```

```

dscontrol port add zLVS_LB_Cluster@80 selectionalgorithm connection
dscontrol port set zLVS_LB_Cluster@80 staletimeout 6400

dscontrol server add zLVS_LB_Cluster@80@LITSTAM3 address 192.168.74.113

dscontrol server add zLVS_LB_Cluster@80@LITSTAM2 address 192.168.74.112

dscontrol port add zLVS_LB_Cluster@443 selectionalgorithm connection
dscontrol port set zLVS_LB_Cluster@443 staletimeout 6400

dscontrol server add zLVS_LB_Cluster@443@LITSTAM3 address 192.168.74.113

dscontrol server add zLVS_LB_Cluster@443@LITSTAM2 address 192.168.74.112

dscontrol highavailability heartbeat add 192.168.74.235 192.168.74.199
dscontrol highavailability backup add backup auto 9123
dscontrol highavailability reach add 192.168.74.100

dscontrol manager start manager.log 10004

dscontrol advisor start Http 80 HTTP_80.log

dscontrol advisor start Ssl 443 SSL_443.log
dscontrol advisor connecttimeout Ssl 443 9
dscontrol advisor receivetimeout Ssl 443 9

```

## Testing load balancing and high availability

In order to test that our load balancer was working correctly, we attempted to do a **wget** from the 192.168.75.xxx LAN from lithub. which is an external client machine that we use as a test workload simulator. As a prerequisite to your installation, ensure you have appropriate network connectivity in the same way. Should you experience any difficulty with this initial test while trying to connect to the various backend servers, ensure that your routing tables are correct on each load balancer node by using the **route** command. This was imperative to us as we had used the previous routing tables which were configured for NAT, not the MAC approach we were currently trying to configure.

The following is an example of the **wget** command:

```

$> wget --proxy=no --http-user=sec_master --http-passwd=password --timeout=10 --connect-timeout=30 https://192.168.74.150/ --no-check-certificate
--14:31:26-- https://192.168.74.150/
Connecting to 192.168.74.150:443... connected.
WARNING: cannot verify 192.168.74.150's certificate, issued by `C=US/O=IBM/OU=Tivoli Systems/CN=Test-Only':
 Self-signed certificate encountered.
WARNING: certificate common name `Test-Only' doesn't match requested host name `192.168.74.150'.
HTTP request sent, awaiting response... 200 OK
Length: 535 [text/html]
Saving to: `index.html'

100%[=====] 535 --.-K/s in 0s

14:31:34 (170 MB/s) - `index.html' saved [535/535]

```

**Tip:** We suggest changing the default Web page for each of the Web servers or backend WebSEAL instances behind the load balancers to self-identify because it will make your testing easier. In our case, our load balancers front WebSEAL instances, so we changed: /opt/pdweb/www-litstam2-WebSeal/docs/index.html

When executing multiple sequential Web requests from our external machine to the load balancer cluster address, they returned alternating responses from each of the backend WebSEAL servers. This shows that our load balancing is working properly.

Next, in order to test out the high availability aspect of our load balancer installation, we took the active node, LITSLBN1, down and observed that the requests from our client were still load-balanced to the backend WebSEAL instances.

Next, we double-checked the status of the secondary load balancer, using the **dscontrol** command. The status for LITSLBN2 has changed to active, showing that it picked up the workload from LITSLBN1. This is evident in the example output below. Note the State ..... Active line, which would have normally indicated State ..... Standby if the primary node was functioning as expected in an active capacity.

```
litslbn2:/etc/init.d # dscontrol highavailability status
```

```
High Availability Status:
```

```

```

```
Role Backup
Recovery strategy Auto
State Active
Port 9123
Preferred target 192.168.74.199
```

```
Heartbeat Status:
```

```

```

```
Count 1
Source/destination ... 192.168.74.235/192.168.74.199
```

```
Reachability Status:
```

```

```

```
Count 1
Address 192.168.74.100 unreachable
```

## Creating a system init script for automated startup

We decided to create a startup script so that the load balancer starts up automatically upon bootup. The following steps describe how we did this.

1. We created a file called `ibm_load_balancer` in the `/etc/init.d/` directory, containing the following information:

```
#!/bin/bash
Copyright (c) 2008 IBM Corporation, US.
#
Authors: Phil & Eli
#
/etc/init.d/Load_Balancer
#
BEGIN INIT INFO
Provides: lb_start
Required-Start: $network
Required-Stop:
Default-Start: 3 5
Default-Stop:
Description: Starting WebSphere Network Deployment Load Balancer
END INIT INFO

. /etc/rc.status

rc_reset

case "$1" in
 start)
```

```

 echo -n "Starting and configuring the Dispatcher Server"
 /opt/ibm/edge/ulb/bin/dsserver
 /root/zLVS_LB.cfg
 exit 0
 ;;
stop)
 echo -n "Stopping and unconfiguring the Dispatcher Server"
 /opt/ibm/edge/ulb/bin/dscontrol executor stop
 /opt/ibm/edge/ulb/bin/dsserver stop
 exit 0
 ;;
restart)
 echo -n "Restarting Dispatcher Server"
 /opt/ibm/edge/ulb/bin/dscontrol executor stop
 /opt/ibm/edge/ulb/bin/dsserver stop
 /opt/ibm/edge/ulb/bin/dsserver
 /root/zLVS_LB.cfg
 exit 0
 ;;
*)
 echo "Usage: $0 {start|stop|restart}"
 exit 1
esac

rc_exit

```

Note that you must use full path names in the script `/root/zLVS_LB.cfg` because they are not established at boot time. We modified `zLVS_LB.cfg` to look like this:

```

#!/bin/sh

export LB_BIN=/opt/ibm/edge/ulb/bin

$LB_BIN/dscontrol set loglevel 3
$LB_BIN/dscontrol executor start
$LB_BIN/dscontrol executor set hatimeout 3

$LB_BIN/dscontrol cluster add zLVS_LB_Cluster address 192.168.74.150
$LB_BIN/dscontrol cluster set zLVS_LB_Cluster proportions 49 50 1 0

$LB_BIN/dscontrol port add zLVS_LB_Cluster@80 selectionalgorithm connection
$LB_BIN/dscontrol port set zLVS_LB_Cluster@80 staletimeout 6400

$LB_BIN/dscontrol server add zLVS_LB_Cluster@80@LITSTAM3 address 192.168.74.113
$LB_BIN/dscontrol server add zLVS_LB_Cluster@80@LITSTAM2 address 192.168.74.112

$LB_BIN/dscontrol port add zLVS_LB_Cluster@443 selectionalgorithm connection
$LB_BIN/dscontrol port set zLVS_LB_Cluster@443 staletimeout 6400

$LB_BIN/dscontrol server add zLVS_LB_Cluster@443@LITSTAM3 address 192.168.74.113
$LB_BIN/dscontrol server add zLVS_LB_Cluster@443@LITSTAM2 address 192.168.74.112

$LB_BIN/dscontrol highavailability heartbeat add 192.168.74.199 192.168.74.235
$LB_BIN/dscontrol highavailability backup add primary=192.168.74.199 auto 9123
$LB_BIN/dscontrol highavailability reach add 192.168.74.100

$LB_BIN/dscontrol manager start manager.log 10004

$LB_BIN/dscontrol advisor start Http 80 HTTP_80.log

```

```
$LB_BIN/dscontrol advisor start Ssl 443 SSL_443.log
$LB_BIN/dscontrol advisor connecttimeout Ssl 443 9
$LB_BIN/dscontrol advisor receivetimeout Ssl 443 9
```

2. We made the script executable by the system by using the **chmod** command:

```
chmod 755 ibm_load_balancer
```

3. Now we can run a variant of the **chkconfig** command to see if our new script shows up in the list of system services.

```
chkconfig --list
```

```
...
```

```
ibm_load_balancer 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

4. To enable our new service to run by default, we used another variant invocation of the **chkconfig** script which actually adds the service:

```
litslbn2:/etc/init.d # chkconfig --add ibm_load_balancer
```

```
ibm_load_balancer 0:off 1:off 2:off 3:on 4:off 5:on 6:off
```

This will automatically create symbolic links from rc3.d and rc5.d to /etc/init.d/ibm\_load\_balancer as confirmed with a number of individual file list (ls) commands:

```
litslbn1:/etc/init.d/rc3.d # ls -l S06ibm_load_balancer
```

```
lrwxrwxrwx 1 root root 20 Nov 4 18:35 S06ibm_load_balancer -> ../ibm_load_balancer
```

```
litslbn1:/etc/init.d/rc3.d # ls -l K15ibm_load_balancer
```

```
lrwxrwxrwx 1 root root 20 Nov 4 18:35 K15ibm_load_balancer -> ../ibm_load_balancer
```

```
litslbn1:/etc/init.d/rc5.d # ls -l S06ibm_load_balancer
```

```
lrwxrwxrwx 1 root root 20 Nov 4 18:35 S06ibm_load_balancer -> ../ibm_load_balancer
```

```
litslbn1:/etc/init.d/rc5.d # ls -l K15ibm_load_balancer
```

```
lrwxrwxrwx 1 root root 20 Nov 4 18:35 K15ibm_load_balancer -> ../ibm_load_balancer
```

5. We verified that the service **ibm\_load\_balancer** options (start, stop, restart) will work as expected with the service command, which is shorthand for the longer verbose service starts that include the /etc/init.d/ paths:

```
service ibm_load_balancer start
```

```
service ibm_load_balancer stop
```

```
service ibm_load_balancer restart
```

We rebooted the Linux virtual server image and monitored the z/VM console for the following output, which indicates success:

*The following messages result from stopping the server:*

```
Stopping the DispatcherSF2-INext-DROP-DEFLT IN=eth1 OUT= MAC=02:09:00:00:00:2f:
02:06:00:00:00:37:08:00 SRC=192.168.74.235 DST=192.168.74.199 LEN=68 TOS=0x00 PR
EC=0x00 TTL=1 ID=26941 PROTO=UDP SPT=9123 DPT=9123 LEN=48
```

```
Advisor 'Http' stopped on port 80.
```

```
Advisor 'Ssl' stopped on port 443.
```

```
The manager has been stopped.
```

```
Executor stopped at your request.
```

```
Server stopping at your request.
```

*The following messages result from starting the server:*

```
Starting and configuring the Dispatcher Server
```

```
Log level successfully set to 3.
```

```
Executor started successfully.
```

```
Executor field(s) successfully set.
```

```
Cluster zLVS_LB_Cluster has been added.
```

```
Cluster field(s) successfully set.
```

```
Port 80 successfully added to cluster zLVS_LB_Cluster.
```

```

Port field(s) successfully set.
Server LITSTAM3 was added to port 80 of cluster zLVS_LB_Cluster.
Server LITSTAM2 was added to port 80 of cluster zLVS_LB_Cluster.
Port 443 successfully added to cluster zLVS_LB_Cluster.
Port field(s) successfully set.
Server LITSTAM3 was added to port 443 of cluster zLVS_LB_Cluster.
Server LITSTAM2 was added to port 443 of cluster zLVS_LB_Cluster.
Heartbeat '192.168.74.199' to '192.168.74.235' successfully added.
Backup information successfully added.
Reach information successfully added.
The manager has been started.
Advisor 'http' has been started on port 80.
Advisor 'ssl' has been started on port 443.
The connect timeout for the advisor was set to 9.
The receive timeout for the advisor was set to 9.

```

Note that, at this stage, if you only reboot the primary load balancer (LITSLBN1, in our case), you can again verify that the peer (LITSLBN2) has temporarily taken over and changed to the active state:

```
litslbn2:/etc/init.d # dscontrol highavailability status
```

```

High Availability Status:

Role Backup
Recovery strategy Auto
State Standby
Port 9123
Preferred target 192.168.74.199

Heartbeat Status:

Count 1
Source/destination ... 192.168.74.235/192.168.74.199

Reachability Status:

Count 1
Address 192.168.74.100 unreachable

```

```
litslbn2:/etc/init.d # dscontrol highavailability status
```

```

High Availability Status:

Role Backup
Recovery strategy Auto
State Active

```

```

Port 9123
Preferred target 192.168.74.199

Heartbeat Status:

Count 1
Source/destination ... 192.168.74.235/192.168.74.199

Reachability Status:

Count 1
Address 192.168.74.100 unreachable

```

After LITSLBN1 has started up again, it shows that it is in the active state:

```
litslbn1:~ # dscontrol highavailability status
```

```

High Availability Status:

Role Primary
Recovery strategy Auto
State Active
Port 9123
Preferred target 192.168.74.235

Heartbeat Status:

Count 1
Source/destination ... 192.168.74.199/192.168.74.235

Reachability Status:

Count 1
Address 192.168.74.100 unreachable

```

Likewise, LITSLBN2 has gone back to standby status:

```

High Availability Status:

Role Backup
Recovery strategy Auto
State Standby
Port 9123
Preferred target 192.168.74.199

Heartbeat Status:

Count 1
Source/destination ... 192.168.74.235/192.168.74.199

Reachability Status:

Count 1
Address 192.168.74.100 unreachable

```

Overall, we found the installation relatively easy to perform, though the installation/creation of a system init script was something we did not find obvious from the documentation, and have chosen to remedy that here.

## Creating a custom DirMaint usermod for integration with RACF

After turning on the DirMaint to RACF integration to allow DirMaint to issue all the necessary RACF commands to define a new guest, an IBM Director z/VM Center component (the z/VM Management Access Point) could not mount the new clone's boot disk to add the personalization file. The z/VM MAP must be able to mount a newly cloned Linux guest's boot disk so that it can add a file that will be used to personalize the clone's hostname and IP addresses.

To solve this problem, we created a VMSES usermod for the DirMaint to RACF integration so that DirMaint will automatically grant the MAP guest write access to the new clone's boot disk. The following steps describe how we did this:

1. Log on to MAINT and link to all the necessary DirMaint disks:

```
VMFSETUP 5VMDIR40 DIRMSFS (LINK
```

```
VMFSET2760I VMFSETUP processing started for 5VMDIR40 DIRMSFS
VMFSET2204I Linking 5VMDIR40 492 as 492 with the link mode MR
RPIMGR031E RESOURCE 5VMDIR40.492 SPECIFIED BY LINK COMMAND NOT FOUND
VMFSET2204I Linking 5VMDIR40 41F as 41F with the link mode MR
RPIMGR031E RESOURCE 5VMDIR40.41F SPECIFIED BY LINK COMMAND NOT FOUND
VMFUTL2205I Minidisk|Directory Assignments:
 String Mode Stat Vdev Label/Directory
VMFUTL2205I LOCALMOD E R/W DIR VMSYS:5VMDIR40.DIRM.LOCALMOD
VMFUTL2205I LOCALSAM F R/W DIR VMSYS:5VMDIR40.DIRM.SAMPLE
VMFUTL2205I APPLY G R/W DIR VMSYS:5VMDIR40.DIRM.APPLYALT
VMFUTL2205I H R/W DIR VMSYS:5VMDIR40.DIRM.APPLYPROD
VMFUTL2205I DELTA I R/W DIR VMSYS:5VMDIR40.DIRM.DELTA
VMFUTL2205I BUILDO J R/W DIR VMSYS:5VMDIR40.DIRM.MAINT19E
VMFUTL2205I BUILD1 K R/W 492 DRM492
VMFUTL2205I BUILD3 L R/W 41F DRM41F
VMFUTL2205I BUILD6 M R/W DIR VMSYS:5VMDIR40.DIRM.HELP
VMFUTL2205I BILD6U N R/W DIR VMSYS:5VMDIR40.DIRM.HELPU
VMFUTL2205I BASE O R/W DIR VMSYS:5VMDIR40.DIRM.OBJECT
VMFUTL2205I BASE1 P R/W DIR VMSYS:5VMDIR40.DIRM.SOURCE
VMFUTL2205I ----- A R/W F91 USR191
VMFUTL2205I ----- B R/W 5E5 MNT5E5
VMFUTL2205I ----- C R/W 191 MNT191
VMFUTL2205I ----- D R/W 51D MNT51D
VMFUTL2205I ----- S R/O 190 MNT190
VMFUTL2205I ----- Y/S R/O 19E MNT19E
VMFSET2760I VMFSETUP processing completed successfully
Ready; T=0.04/0.04 17:26:29
```

2. Create a new RACF group named MAPGROUP and add all the MAP guests to it:

```
RAC ADDGROUP (MAPGROUP)
RAC CONNECT (MAPVM1 MAPVM2 MAPVM4 MAPVM5 MAPVM6 MAPVM9) GROUP(MAPGROUP)
```

3. Use the LOCALMOD tool to create a modification to the DVHXDN EXEC:

```
LOCALMOD 5VMDIR40%DIRMSFS DVHXDN $EXEC
```

- a. The LOCALMOD tool does some setup work and puts you into an XEDIT session showing the actual REXX code. Add the highlighted section shown below in the indicated place:

```
When Cmd = 'AMDISK'| Cmd = 'ADD'| Cmd = 'CLONEDISK'
Then Do
 'EXEC RAC RDEFINE VMMDISK 'Resource_Name' OWNER('Targetid') ,
 'Other'
Call Racf Rc
'EXEC RAC PERMIT 'Resource_Name' CLASS(VMMDISK) ID('Whoami'),
```

```

 DELETE'
 Call RAcf_Rc
 If OwnerAccess <> '' then
 Do
 'EXEC RAC PERMIT 'Resource_Name' CLASS(VMMDISK)' ,
 'ID('Targetid') 'OwnerAccess
 Call RAcf_Rc
 End
 ▶ If Vaddr = '201' then
 ▶ Do
 ▶ 'EXEC RAC PERMIT 'Resource_Name' CLASS(VMMDISK)' ,
 ▶ 'ID(MAPGROUP) ACC(ALTER)'
 ▶ Call RAcf_Rc
 ▶ End
 End

```

- b. Enter the **file** command on the XEDIT command line to save the changes and exit.

```

VMFEXU2760I VMFEXUPD processing started
DMSUPD178I Updating DVHXDN $EXEC A1
DMSUPD178I Applying DVHXDN UPL0001 E1
VMFEXU2507I DVHXDN TEMP$ created on your E-disk for use in a VMSES/E environment
VMFEXU2760I VMFEXUPD processing completed successfully
VMFLMD1966W The command, REXXC, completed with return code 4 while operating
upon file DVHXDN TEMP$
VMFREP2760I VMFREPL processing started
VMFREP2509I The version vector table 5VMDIR40 VVTLCL E will be updated for the
part DVHXDN CEX
VMFREP2760I VMFREPL processing completed successfully
VMFLMD2760I LOCALMOD processing completed with warnings
Ready(00004); T=0.85/0.92 17:54:18

```

4. Build the newly altered part:

```
VMFINS BUILD PPF 5VMDIR40 DIRMSFS (SERVICED
```

5. (*Optional*) If you want to generate a plain text (uncompiled) version of the exec to use for testing:

```
VMFEXUPD DVHXDN EXEC 5VMDIR40 DIRMSFS (OUTMODE LOCALMOD
```

6. Run this command to use the automatic service process to copy the changed part into production:

```
PUT2PROD DIRMSFS
```

7. Force and restart DirMaint to pick up the changes:

```
FORCE DIRMAINT
XAUTOLOG DIRMAINT
```



---

## Chapter 36. Future Linux on System z projects

We hope that this report has provided you with some valuable insight into the tools and strategies we have discussed. Our focus for the next six months will be the planned move of our laboratory (along with our z/OS PET sister team) to a newer facility. Our topic area will continue to be systems management with a special emphasis on IBM Director provisioning and the continued maintenance stream from our enterprise Linux vendors. In addition, we will evaluate some new types of workloads which have scope outside of the e-commerce (Bookstore and Stock trading) arena. As always, we encourage you to send us your comments, suggestions, questions, or proposed areas of investigation.



---

## Appendix A. About our Parallel Sysplex environment

Here we describe our Parallel Sysplex computing environment, including information about our hardware and software configurations.

**Note:** In our test reports, 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 run two Parallel Sysplexes, one with nine members and the other with four members that consist of the following:

- Five *central processor complexes* (CPCs) running z/OS in 13 logical partitions (LPARs).

The CPCs consist of the following machine types:

- One IBM eServer zSeries 990 (z990) CPC
- One IBM System z9 Enterprise Class (z9 EC) CPC
- One IBM System z10 Enterprise Class (z10 EC) CPC
- Two IBM System z10 Business Class (z10 BC) CPCs

The z/OS images consist of the following:

- Eight production z/OS systems
- Four test z/OS systems
- One z/OS system to run TPNS (Our December 1998 test report explains why we run TPNS on a non-production system.)

- Seven *coupling facilities* (CFs):

- Two failure-independent coupling facility that runs in a LPAR on a standalone CPC
- Five non-failure-independent coupling facilities that run in LPARs on three 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.

“Our Parallel Sysplex hardware configuration” describes all of the above in more detail.

Outside of the Parallel Sysplex itself, we also have ten LPARs in which we run the following:

- Two native Linux images
- Eight z/VM images that host multiple Linux guest images running in virtual machines

---

### Our Parallel Sysplex hardware configuration

This topic 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 98 on page 312 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.

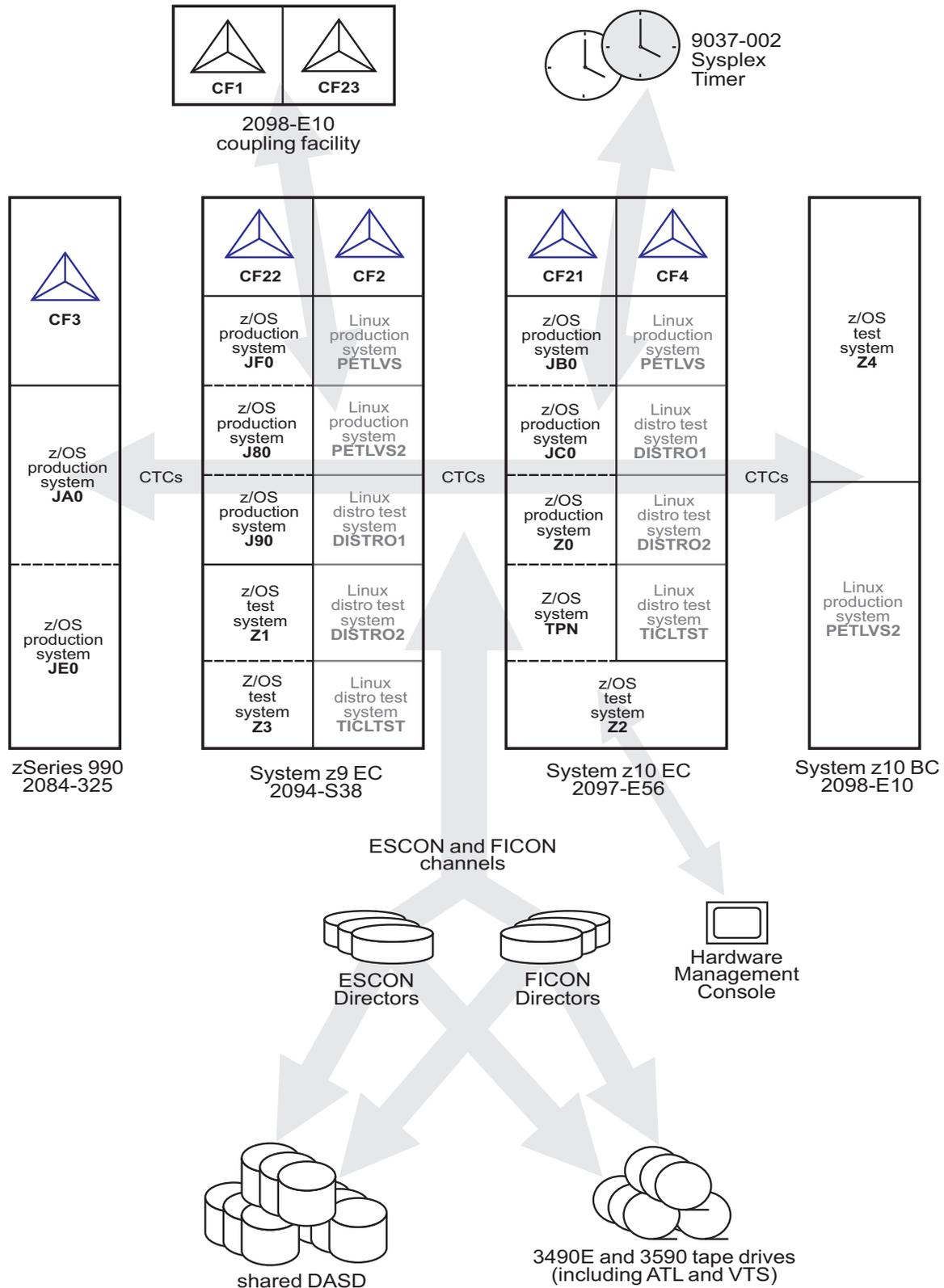


Figure 98. Our sysplex hardware configuration

## 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 98 on page 312.

### Mainframe server details

Table 11 provides information about the mainframe servers in our sysplex environment.

Table 11. Our mainframe servers

Server model (Machine type-model)	CPCs CPs	Mode LPARs	HSA	Storage	LCSS	System name, usage Sysplex membership CPs, zIIPs, zAAPs Initial LPAR weight
IBM eServer zSeries 990 Model 325 (2084-325)				30720M	2	<b>JA0</b> , z/OS production system Plex 1 20 shared CPs 2 shared zAAPs
				22528M	2	<b>JE0</b> , z/OS production system Plex 1 20 shared CPs 2 shared zAAPs
IBM System z9 EC Model S54 (2094-S54)	1 CPC 54 CPs	LPAR 9 LPARs	2176M	112640M	0	<b>J80</b> , z/OS production system Plex 1 42 shared CPs 2 shared zIIPs 4 shared zAAPs
				112640M	0	<b>J90</b> , z/OS production system Plex 1 41 shared CPs 2 shared zIIPs 4 shared zAAPs
				15360M	0	<b>JF0</b> , z/OS production system Plex 1 16 shared CPs 2 shared zIIPs 4 shared zAAPs
				30720M	0	<b>Z1</b> , z/OS test system Plex 2 8 shared CPs 2 shared zIIPs 4 shared zAAPs
				30720M	0	<b>Z3</b> , z/OS test system Plex 2 8 shared CPs 2 shared zIIPs 4 shared zAAPs
				256M	1	<b>PETLVS</b> , Linux production system 1 shared CP weight of 10
				4096M	1	<b>PETLVS2</b> , Linux production system 4 shared CPs weight of 10
				8192M	1	<b>DISTR01</b> , Linux distribution test system 2 shared IFLs weight of 10
				2048M	1	<b>DISTR02</b> , Linux distribution test system 2 shared IFLs weight of 10
				1024M	1	<b>TICLTST</b> , Linux distribution test 1 shared IFL weight of 10

Table 11. Our mainframe servers (continued)

Server model (Machine type-model)	CPCs CPs	Mode LPARs	HSA	Storage	LCSS	System name, usage Sysplex membership CPs, zIIPs, zAAPs Initial LPAR weight
IBM System z10 BC Model E10 (2098-E10)	1 CPC	LPAR mode	8192M	30720M	0	<b>Z4</b> , z/OS test system Plex 2 5 shared CPs 2 zIIPs 3 zAAPs
	10 CPs			10240M	1	<b>PETLVS</b> , Linux distribution test system 3 shared CPs weight of 10
IBM System z10 EC Model E56 (2097-E56)	1 CPC	LPAR mode 4 LPARs (1 LP is a coupling facility)	256M	9216M		<b>Z0</b> , z/OS production system Plex 1 8 shared CPs 2 shared zIIPs 4 shared zAAPs
	45 CPs			30720M	0	<b>JB0</b> , z/OS production system Plex 1 37 shared CPs 2 shared zIIPs 4 shared zAAPs
	5 ICFs			22528M	0	<b>JC0</b> , z/OS production system Plex 1 37 shared CPs 2 shared zIIPs 4 shared zAAPs
				6144M	0	<b>TPN</b> , z/OS system for TPNS Plex 1 12 shared CPs 2 shared zIIPs 4 shared zAAPs
				10752M	0	<b>Z2</b> , z/OS test system Plex 2 8 shared CPs 2 shared zIIPs 4 shared zAAPs
				4096M	1	<b>PETLVS</b> , Linux production system 4 shared CPs weight of 10
				3072M	1	<b>DISTRO1</b> , Linux distribution test system 2 shared IFLs weight of 10
				2048M	1	<b>DISTRO2</b> , Linux distribution test system 2 shared IFLs weight of 10
				1024M	1	<b>TICLTST</b> , Linux distribution test 1 shared IFL weight of 10

## Coupling facility details

Table 12 provides information about the coupling facilities in our sysplex. Figure 98 on page 312 further illustrates the coupling facility channel distribution as described in Table 12.

Table 12. Our coupling facilities

Coupling facility name	Model description CPCs and CPs CFLEVEL (CFCC level) Controlled by	Storage
CF1 (Plex 1)	IBM System z10 BC Model 2098-E10 standalone coupling facility 1 CPC with 4 CPs CFLEVEL=16 (CFCC Release 16.00, Service Level 00.26) Controlled by the HMC	31G
CF2 (Plex 1)	Coupling facility LPAR on a System z9 EC Model S38 (2094-S54) 3 dedicated ICF CPs CFLEVEL=15 (CFCC Release 15.00, Service Level 02.05) Controlled by the HMC	30G
CF3 (Plex 1)	Coupling facility LPAR on a zSeries 990 Model 325 (2084-325) 3 dedicated ICF CPs CFLEVEL=14 (CFCC Release 14.00, Service Level 00.28) Controlled by the HMC	24G
CF4 (Plex 1)	Coupling facility LPAR on a System z10 EC Model E56 (2097-E56) 3 dedicated ICF CPs CFLEVEL=16 (CFCC Release 16.00, Service Level 00.26) Controlled by the HMC	31G
CF21 (Plex 2)	Coupling facility LPAR on a System z10 EC Model E56 (2097-E56) 1 dedicated ICF CP CFLEVEL=16 (CFCC Release 16.00, Service Level 00.26) Controlled by the HMC	6G
CF22 (Plex 2)	Coupling facility LP on a System z9 EC Model S38 (2094-S54) 1 dedicated ICF CP CFLEVEL=15 (CFCC Release 15.00, Service Level 02.05) Controlled by the HMC	6G
CF23 (Plex 2)	Coupling facility on a System z10 BC Model E10 (2098-E10) 1 dedicated ICF CP CFLEVEL=16 (CFCC Release 16.00, Service Level 00.26) Controlled by the HMC	6G

Table 13 illustrates our coupling facility channel configuration on Plex 1.

Table 13. Coupling facility channel configuration on Plex 1

Machine type z/OS images CF images	Coupling facility (CF) images			
	2098-E10 CF1	2094-S54 CF2	2084-325 CF3	2097-E56 CF4
2084-325 JA0, JE0 CF3	1 CBP 3 CFP	1 CBP 3 CFP	4 ICP	4 CFP
2097-E56 Z0, JB0, JC0, TPN CF4	4 CFP 4 CIB *	4 CFP 1 CBP	4 CFP	4 ICP
2094-S54 J80, J90, JF0 CF2	3 CFP	8 ICP	1 CBP 3 CFP	4 CFP 1 CBP

\* = Same links

Table 14 illustrates our coupling facility channel configuration on Plex 2.

*Table 14. Coupling facility channel configuration on Plex 2*

Machine type z/OS images CF images	Coupling facility (CF) Images		
	2097-E56 CF21	2094-S54 CF22	2098-E10 CF23
2097-E56 Z2, Z4CF21 CF21	2 ICP	2 CFP 1 CBP	2 CFP 4 CIB *
2094-S54 Z1, Z3 CF22	2 CFP 1 CBP	2 ICP	2 CFP
2098-E10 Z4 CF23	3 CFP 4 CIB *	3 CFP	2 CFP 4 CIB *

\* = These CIB CF connections share physical PSIFB fiber connections. For example, each of the two PSIFB fibers connecting our 2097-E56 to our 2098-E10 coupling facility carries the traffic for two CIB connections for Plex 1 and two CF connections for Plex 2.

### Other sysplex hardware details

Table 15 highlights information about the other hardware components in our sysplex.

*Table 15. 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 sometimes 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.  We also test in various STP configurations, including mixed CTN (with the ETR providing timing for some CPCs) and STP mode (no ETR).

Table 15. 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 both FICON and ESCON <sup>®</sup> CTC channels on all of our CPCs. <b>Note:</b> 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. Under some circumstances, XCF signaling using CTCs is faster than using structures. See <i>S/390 Parallel Sysplex Performance</i> for a comparison.
	Coupling facility channels	We use a combination of ISC, ICB, IC, and CIB 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 <i>FICON Native Implementation and Reference Guide</i> , 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 <i>FICON CTC Implementation</i> available on the IBM Redbooks Web site.
DASD	Enterprise Storage Server(R) (ESS, 2105-F20, 800) IBM System Storage (DS6000™, DS8000)	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. <b>Note:</b> Do not run 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)	3494 Model L10 with 16 Escon and Ficon attached 3590 tape drives and 8 3592 (Encryption capable) tape drives	All tape drives are accessible from all systems.
Virtual Tape Server (VTS)	3494 Model L10 with 32 virtual 3490E 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) V3R2
- IMS V9 (and its associated IRLM)
- DB2 UDB for z/OS and OS/390 V8 (and its associated IRLM)
- DB2 UDB for z/OS and OS/390 V9.1 (and its associated IRLM)

- WebSphere Application Server for z/OS V6.0.2
- WebSphere Application Server for z/OS V6.1
- WebSphere Message Broker V6
- WebSphere MQ for z/OS V6
- WebSphere Process Server V6.1
- WebSphere Service Registry and Repository V6.1

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 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 Planning for Installation* and *z/OS MVS Product Management* for more information about dynamic enablement.

## Overview of our software configuration

Figure 99 shows a high-level view of our sysplex software configuration.

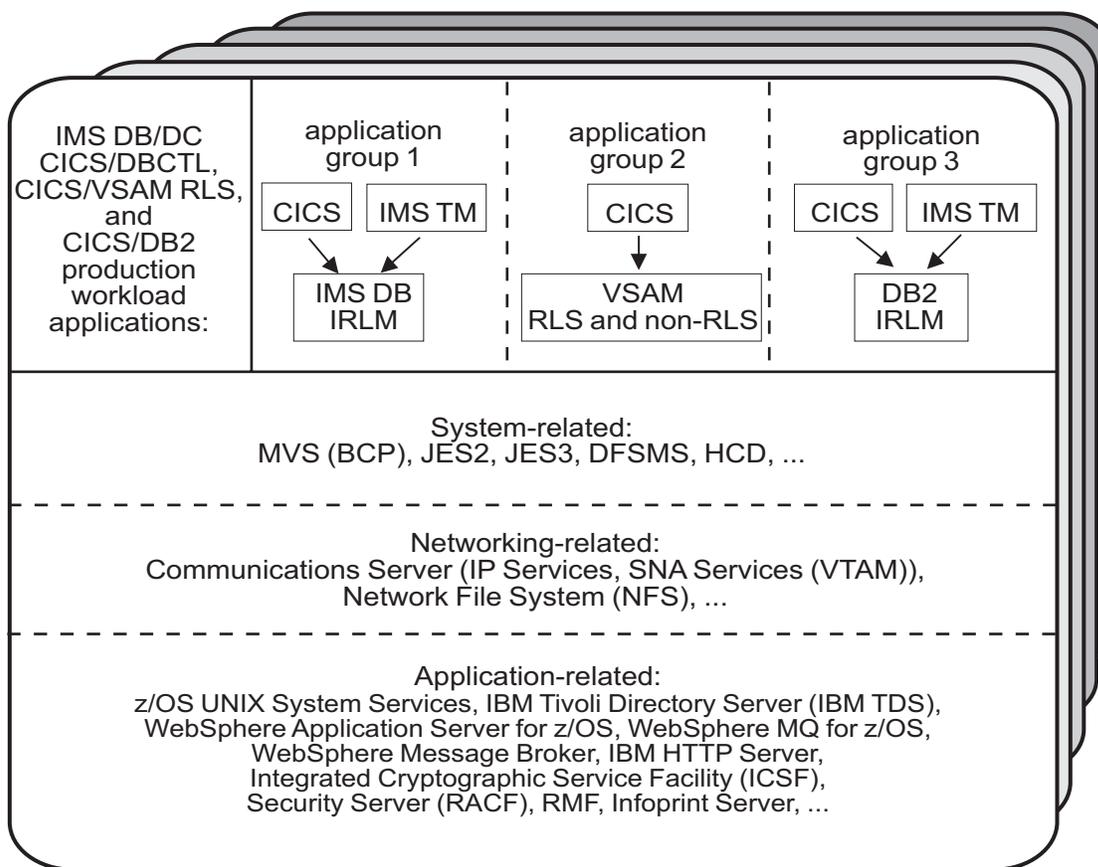


Figure 99. Our sysplex software configuration

We run three separate application groups in one sysplex and each application group spans multiple systems in the sysplex. Table 16 on page 319 provides an overview of the types of transaction management, data management, and

serialization management that each application group uses.

Table 16. 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 test report 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 341.

## 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, and so on
- *i* represents the instance of the region and can be A, B, C, and so on (we may have 3 AORs in our application group per 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.



## Appendix B. About our networking environment

This topic describes our networking environment, including a high-level overview of our TCP/IP, network file systems, and VTAM configuration.

### Our networking configuration

Figure 100 provides a logical view of our networking configuration.

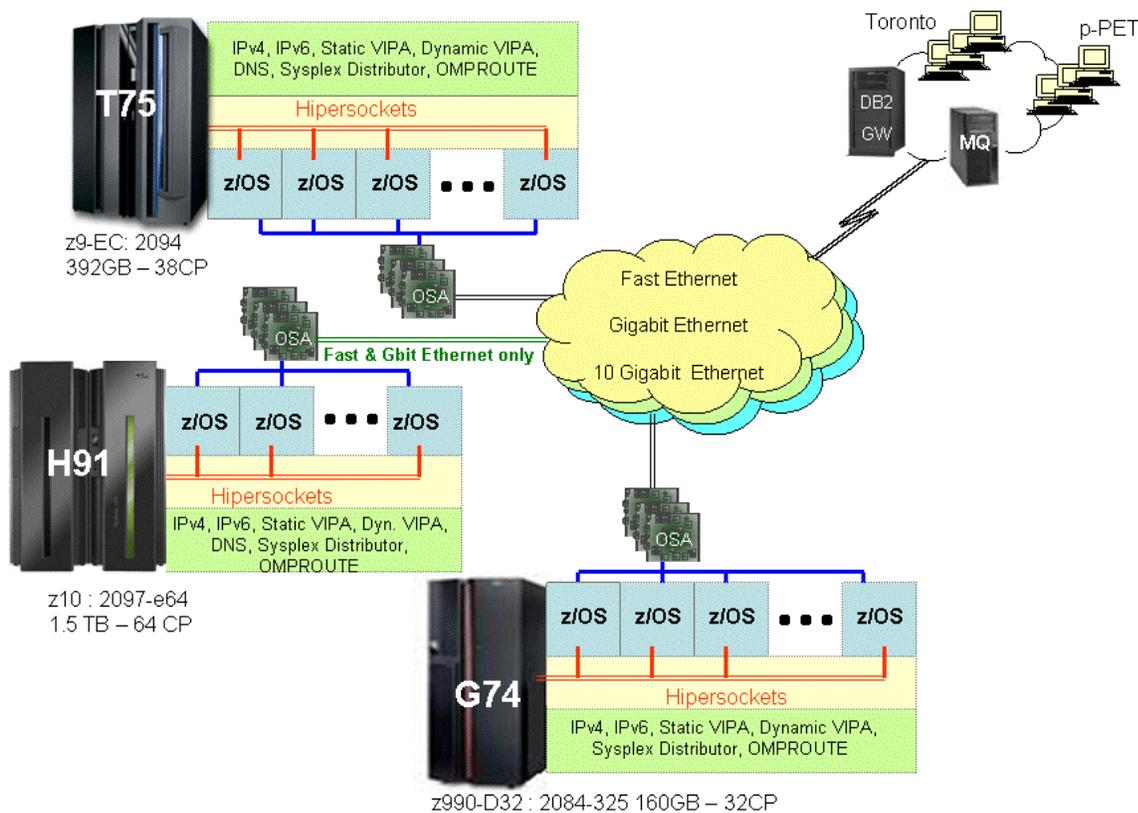


Figure 100. Our networking topology

### Configuration overview

Our networking environment is entirely Ethernet. Currently we have Fast Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet, each running on separate networks. This setup provides a robust environment for our z/OS testing. Across these networks we run workloads that exercise many z/OS components and IBM products.

Figure 100 illustrates the following points:

- We have OSA Fast Ethernet, OSA Gigabit Ethernet, and OSA 10 Gigabit Ethernet configured on three of our 4 CECs. Since our fourth CEC, the z900, does not

support the 10 Gigabit OSA feature, we only have the OSA Fast Ethernet and OSA Gigabit Ethernet features configured.

- We use OMPROUTE on each z/OS image to provide dynamic OSPF routing support across our data center.
- We have a DNS setup using master and slave all on z/OS.
- We have dynamic XCF configured so that we can use hypersockets on the CEC's where there is more than one image for communications between those images.
- All of the networks are VLAN Tagged.
- We have a fully implemented IPv6 environment.
- We run many sysplex distributors for workload balancing with a variety of distribution methods using IPv4 and IPv6.

## Our IPv6 environment configuration

We currently run a fully implemented IPv6 environment utilizing IPv6 OMPROUTE and DVIPA/Sysplex distributor, This is used to support WebSphere MQ V6 and DB2 V9.1 implementations.

## 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 both 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 TCP/IP stack that supports IPv4 and IPv6.

We added the following two NETWORK statements to have a TCP/IP stack that supports IPv4 and IPv6:

```
FILESYSTYPE TYPE(CINET) ENTRYPOINT(BPXCINT)
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:

```

VIPADefine V6Z2FTP 2003:0DB3:1::2
VIPADistribute SysplexPorts V6Z2FTP PORT 20 21
DESTIP FEC0:0:0:1:0:168:49:37

```

**Note:** V6Z2FTP is the INTERFACE name for this VIPA.

### 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 separate 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.

```

## 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 allows 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 Guide and Reference, SC26-7417*. In addition, you can read about WebNFS support in our December 1999 test report at *OS/390 Parallel Sysplex Test Report*, and the use of the LAN Server NFS in our December 2004 edition at *zSeries Platform Test Report*. All of our editions can be found at:

<http://www.ibm.com/servers/eserver/zseries/zos/integtst/library.html>

## Our VTAM configuration

Figure 101 illustrates our VTAM® configuration.

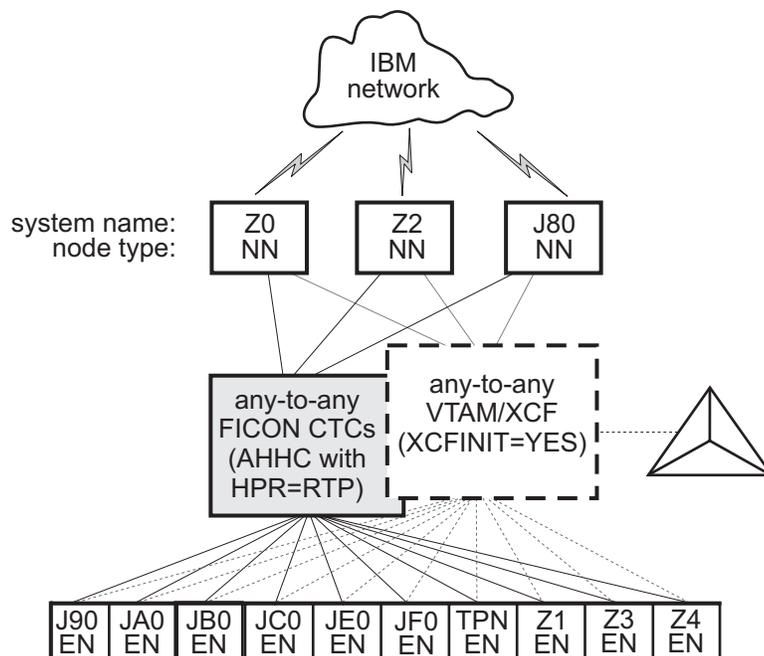


Figure 101. Our VTAM configuration

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

Our VTAM configuration is a pure any-to-any AHHC. Systems 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 signaling, 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 signaling for test purposes.

---

## Testing our networking environment

We have implemented several workloads to stress and test our networking environment. For information about our these workloads, see “Appendix D. About our test workloads” on page 335.

---

## 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 102 on page 326: 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 102 on page 326 : 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.

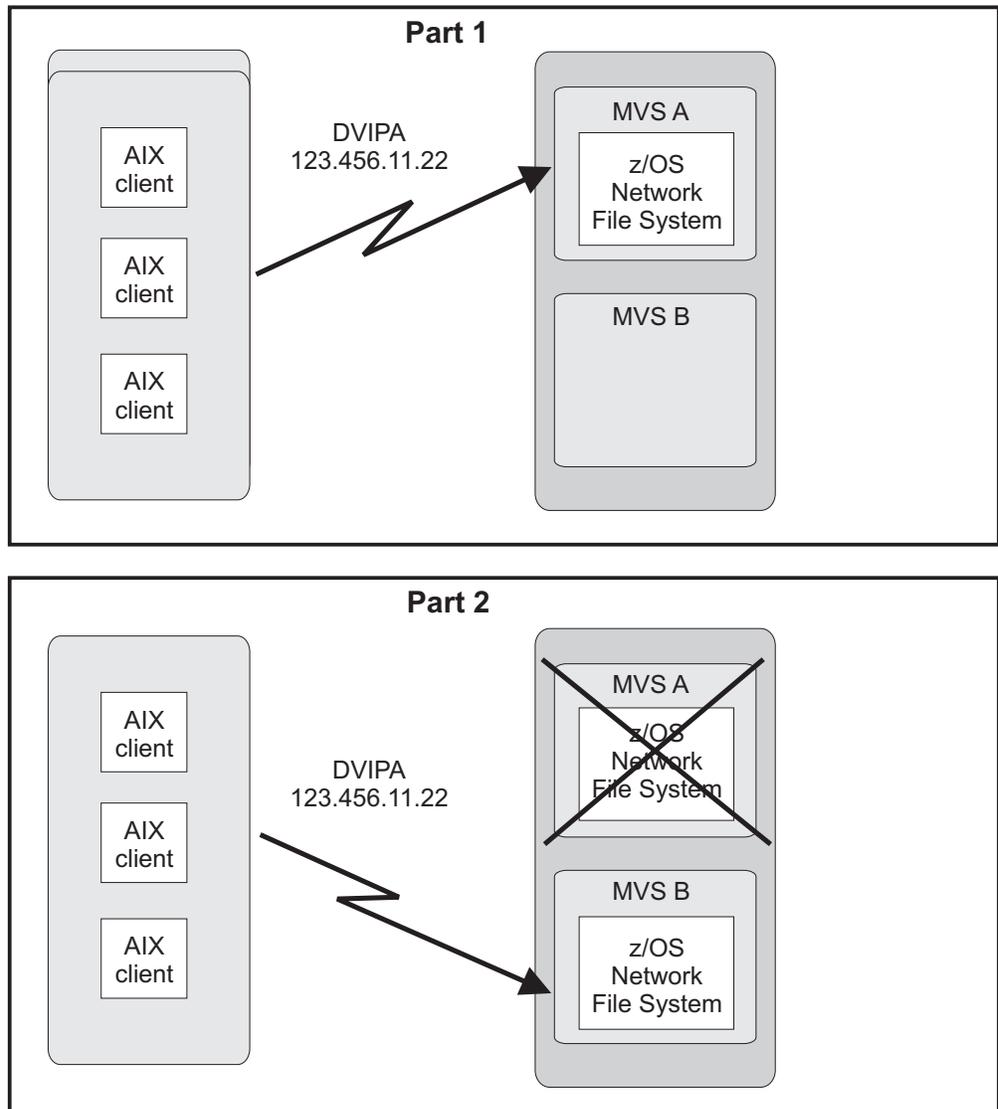


Figure 102. NFS configuration

**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,',
// 'ELEMENTYPE=SYSTCPIP')
/* ----- *
/* DELETE VIPA FOR NFS SERVER *
/* ----- *
//DELVIPA EXEC PGM=EZBXFDVP,
// PARM='POSIX(ON) ALL31(ON) /-p TCP/IP -d &VIPA'
//SYSPRINT DD SYSOUT=*
/* ----- *
/* ACQUIRE VIPA FOR NFS SERVER *
/* ----- *
//DEFVIPA EXEC PGM=EZBXFDVP,
// PARM='POSIX(ON) ALL31(ON) /-p TCP/IP -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 TCP/IP -d &VIPA'
//SYSPRINT DD SYSOUT=*
/*****
/*FOR NORMAL TERMINATION,DEREGISTER FROM ARM
/*FOR NORMAL TERMINATION,DEREGISTER FROM ARM
/*****
//ARMDREG EXEC PGM=ARMWRAP,
// PARM=('REQUEST=DEREGISTER')

```



---

## Appendix C. About our security environment

Information about our security computing environment includes:

- “Our Integrated Cryptographic Service Facility (ICSF) configuration”
- “Network Authentication Service configuration” on page 330
- “Our LDAP configuration” on page 331

---

### Our Integrated Cryptographic Service Facility (ICSF) configuration

z/OS Integrated Cryptographic Service Facility (ICSF) is a software element of z/OS that works with the hardware cryptographic features and the Security Server (RACF) to provide secure, high-speed cryptographic services in the z/OS environment. ICSF provides the application programming interfaces by which applications request the cryptographic services. The cryptographic feature is secure, high-speed hardware that performs the actual cryptographic functions.

We currently have ICSF HCR7750 installed across both sysplexes. This became generally available in November, 2007. See our testing experiences with this level in Chapter 14, “Migrating to and using ICSF HCR7750,” on page 143.

The available cryptographic hardware features are dependent on the server. Because we have many types of servers in our environment, we run with various cryptographic hardware features. Following is a list of cryptographic hardware features we currently have in our environment:

- Crypto Express2 Accelerator (CEX2A)
- Crypto Express2 Coprocessor (CEX2C)
- PCI Cryptographic Accelerator (PCICA)
- PCI X Cryptographic Coprocessor (PCIXCC)
- CP Assist for Cryptographic Functions (CPACF)
- CP Assist for Cryptographic Functions DES/TDES Enablement (CPACF, feature 3863)
- PCI Cryptographic Coprocessor (PCICC)
- Cryptographic Coprocessor Facility (CCF)

On each sysplex within our environment, we share the CKDS, PKDS, and TKDS data sets among all systems.

Since our goal is to run a customer-like environment, we have various workloads and jobs which take advantage of the products that interface with ICSF (which interfaces with the cryptographic hardware). These products include the following:

- SSL (through WebSphere Application Server, FTP, HTTP, LDAP and CICS)
- Enterprise Key Manager Offering for Tape Encryption
- Encryption Facility for z/OS V1 R1
- Encryption Facility for z/OS V1 R2
- Network Authentication Service (Kerberos) (through LDAP, EIM, and FTP)

We also have an ICSF specific workload that runs daily which exercises the cryptographic services available through the ICSF Callable Services.

**Note:** For additional information on the Enterprise Key Manager Offering for Tape Encryption and the Encryption Facility for z/OS V1 R2, see our June 2007 test report. For Encryption Facility for z/OS V1R1, see our December 2006 test report.

## Network Authentication Service configuration

Figure 103 shows an overview of our Network Authentication Service (NAS) configuration.

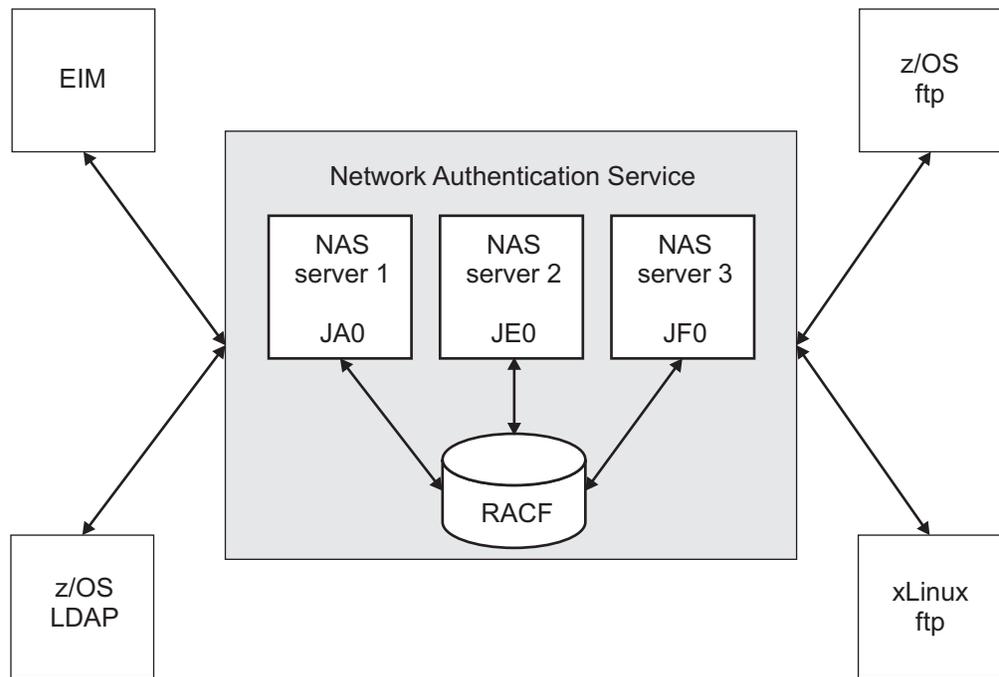


Figure 103. Overview of our Network Authentication Service configuration

We currently have three NAS servers configured within one sysplex. All of the servers use RACF as the registry database.

The EIM, z/OS LDAP, z/OS ftp and xLinux ftp clients have all been enabled to exploit NAS, as discussed in previous editions of our test report:

- For information about our enablement of EIM with NAS, see our September 2004 test report, *zSeries Platform Test Report Version 1 Release 6*, SA22-7997-00.
- For information about our enablement of z/OS LDAP with NAS, see our December 2002 test report, *Parallel Sysplex Test Report Version 1 Release 3 & Version 1 Release 4*, SA22-7663-07.
- For information about our enablement of z/OS ftp and xLinux ftp with NAS, see our December 2005 test report, *zSeries Platform Test Report for z/OS and Linux Virtual Servers Version 1 Release 7*, SA22-7997-02.

### skrb5.conf file

Our skrb5.conf file follows the example provided in `/usr/lpp/skrb/examples/skrb5.conf`, except that we have configured for all encryption levels.

### envar file

Our envar file follows the example provided in `/usr/lpp/skrb.examples/skrbkdc.envar`, except that we have configured for all encryption levels.

To validate our configuration, a **kinit** command is first issued to obtain our Kerberos credentials. Then, a transaction using each of the four clients is issued using those credentials.

## Our LDAP configuration

We have a multiplatform LDAP configuration for both the Integrated Security Services (ISS) LDAP environment and the IBM Tivoli Directory Server (IBM TDS) environment. The following figures illustrate both environments followed by a listing of exploiters of each environment.

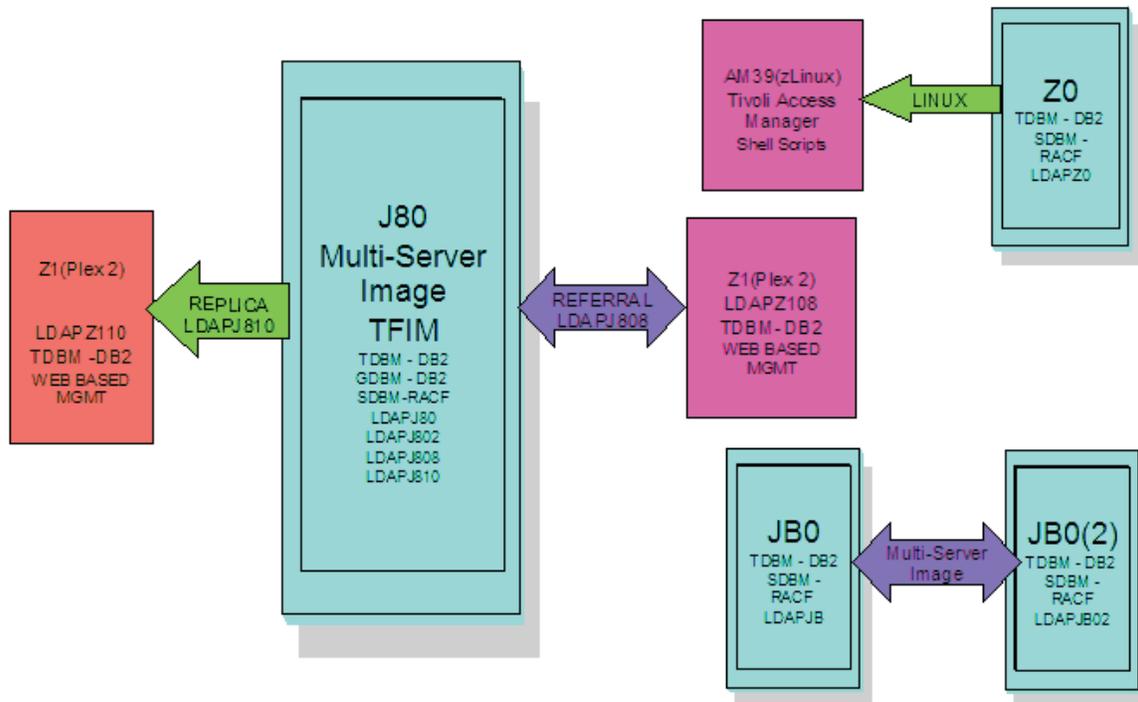


Figure 104. Integrated Security Services (ISS) LDAP environment

**DB2** has a TDBM backend, connecting LDAP to the DB2 Database Directory. It also has a GDBM backend. The GDBM backend is used to store Change Log entries created as a result of RACF modifications. These environments are exploited using scripts run from Windows agents that allow stress to be placed on the z/OS LDAP Servers using DB2.

**RACF** has a SDBM backend, connecting to the RACF directory found on our plex.

## Integrated Security Services (ISS) LDAP exploitation

### LDAP Referral

This configuration is set up between a LDAP Server on Plex 2 (LDAPZ108) and a LDAP Server on Plex 1 (LDAPJ808) both using TDBM backends. The Plex 2 LDAP Server (LDAPZ108) has a general referral found in its configuration file that points to a master LDAP server on z/OS (LDAPJ808). This allows the user to run the **ldapsearch** command from the LDAP server on Plex 2 for an entry that is not found in that directory, but may be found in the LDAPJ808 master server's directory. The command will return all entries found that match from both directories.

### **Replication (Master/Slave)**

We run our ISS replication transactions between LDAPJ810 and LDAPZ110. Replication functions quite like the Stress operation above but with one important difference. The master receives the new entry and its modifications and eventual deletion. However the slave, which has been initialized like the master, is checked twice, the first time after the entry is added to insure it has been replicated on the slave and also after the deletion to insure it has indeed been deleted from the slave through the replication process. The checking process is repeated until it is either found (during the add) or not found (during the deletion) or the server reaches a specified search count (which causes a failure).

### **Replication (Peer/Peer)**

We run our ISS replication transactions between LDAPJ810 and LDAPZ110. Peer to peer replication functions similarly to master/slave except that each server takes turns at being the "master", that is having its entries manipulated by the program while the other server is checked for entry availability. When the program is run in a loop, the "master" and "slave" switch places on each new loop cycle.

### **Persistent Search**

We run our ISS persistent search transactions between LDAPJ810 and LDAPZ110. The persistent search function detects the revisions that have been made to a server's entries and prints out the results, the detail depending on the display level setting. The program is initiated with the entry filter and operation monitor parameters set and it will listen to the designated server until a specified entry type operation is encountered for reporting. This repeats until the program is terminated. Of course for this function to operate, there must be some activity on the server being monitored. That is one use for the Stress function. An instance of it can be run to stimulate the desired server. Also, the persistent search could be directed against one of the replication servers if desired. For another workload scenario, several instances of the persistent search can be run, with each detecting a different change type (or combination thereof).

### **Tivoli Access Manager on zLinux**

We have set up Tivoli Access Manager (TAM) on our zLinux SUSE 8 machine to enable cross platform testing between Linux and z/OS. TAM uses z/OS LDAP as a backend to store userid information that will either allow or deny user access to TAM. Testing is done using Shell scripts run on Linux that allow stress to be placed on the z/OS LDAP Server on Z0.

### **Tivoli Federated Identity Manager on zLinux**

We have setup Tivoli Federated Identity Manager (TFIM) on our zLinux machine to enable cross platform testing between Linux and z/OS. TFIM uses z/OS LDAP as a backend to store userid information in a similar capacity to TAM. However, our TFIM setup requires the use of two LDAP Servers; LDAPJ80 and LDAPJ802. This environment is exploited using Shell scripts run from Windows agents that allow stress to be placed on the z/OS LDAP Servers on J80.

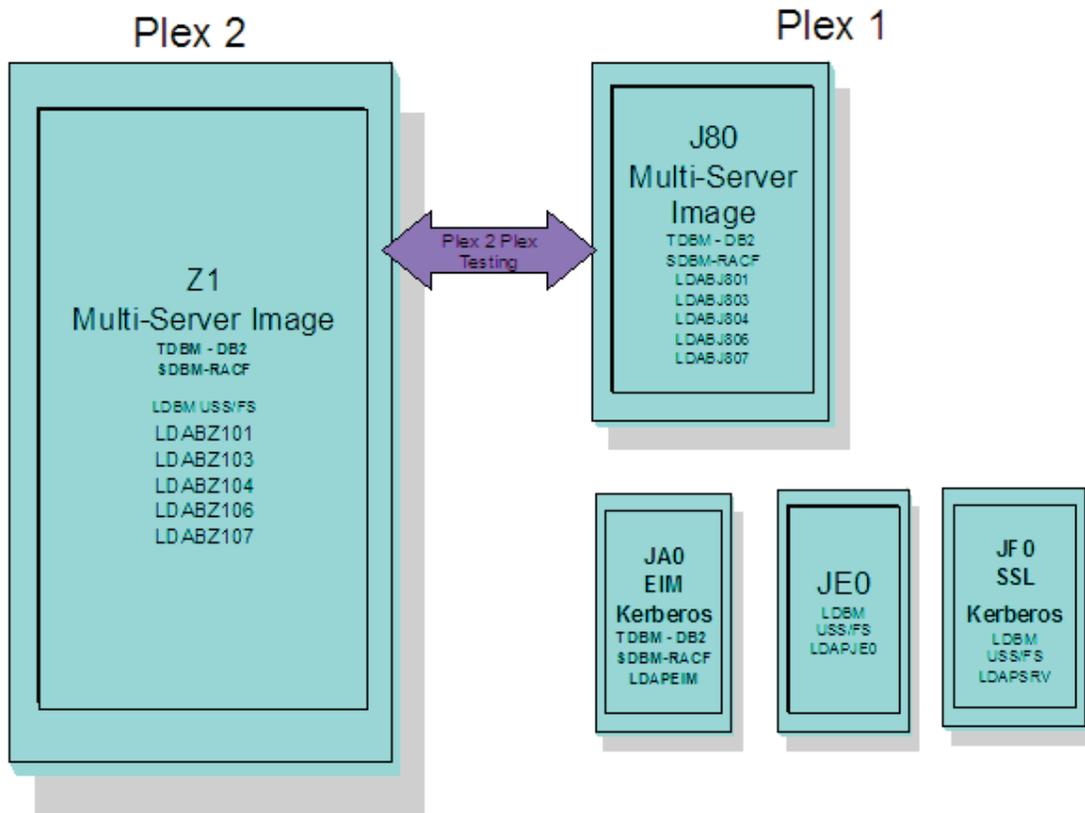


Figure 105. IBM Tivoli Directory Server (IBM TDS) environment

- DB2** has a TDBM backend, connecting LDAP to the DB2 Database Directory. This environment is exploited using scripts run from Windows agents that allow stress to be placed on the z/OS LDAP Servers using DB2.
- RACF** has a SDBM backend, connecting to the RACF directory found on our plex.
- Unix System Services file system** has an LDBM backend, connecting to a Unix System Services file system on our plex. This environment is exploited in two ways. The first is with tso http servers. The IBM HTTP Server powered by Domino running on one of our z/OS images and Apache running on an xLinux box. Both of these http servers access the LDAPJEO IBM TDS for authentication to access http resources. The second is to drive Kerberos transactions using shell scripts run from within our USS environment. This workload accesses the LDAPJF0 IBM TDS.

## IBM Tivoli Directory Server (IBM TDS) exploitation

### Kerberos

We currently have two LDAP servers on our plex that are setup for Kerberos transactions. They are LDAPSRV on JF0 and LDAPEIM on JA0.

**EIM** We currently have one LDAP server on our plex that is setup for EIM transactions. It is LDAPEIM on JA0.

### LDAP Referral

This configuration is set up between a LDAP Servers on Plex 2 and a LDAP Servers on Plex 1 (LDABJ804/LDABJ806) using the TDBM and

LDBM backends. The Plex 2 LDAP Servers (LDABZ104/LDABZ106) have a general referral found in its configuration file that points to master LDAP servers on z/OS (LDABJ804/LDABJ806). This allows the user to run the **ldapsearch** command from the LDAP servers on Plex 2 for an entry that is not found in that directory, but may be found in the master servers' directories. The command will return all entries found that match from both directories.

#### **Replication (Master/Slave)**

We run our IBM TDS master/slave replication transactions between LDABJ801 and LDABZ101. Replication functions quite like the Stress operation above but with one important difference. The master receives the new entry and its modifications and eventual deletion, but the slave, which has been initialized like the master, is checked twice, the first time after the entry is added to insure it has been replicated on the slave and also after the deletion to insure it has indeed been deleted from the slave through the replication process. The checking process is repeated until it is either found (during the add) or not found (during the deletion) or the server reaches a specified search count (which causes a failure).

#### **Replication (Peer/Peer)**

We run our IBM TDS peer/peer replication transactions between LDABJ803 and LDABZ103. Peer to peer replication functions similarly to master/slave except that each server takes turns at being the "master", that is having its entries manipulated by the program while the other server is checked for entry availability. When the program is run in a loop, the "master" and "slave" switch places on each new loop cycle.

#### **Persistent Search**

We run our IBM TDS persistent search transactions between LDABJ804 and LDABZ104. The persistent search function detects the revisions that have been made to a servers entries and prints out the results, the detail depending on the display level setting. The program is initiated with the entry filter and operation monitor parameters set and it will listen to the designated server until a specified entry type operation is encountered for reporting. This repeats until the program is killed. Of course for this function to operate there must be some activity on the server being monitored. That is one use for the Stress function. An instance of it can be run to stimulate the desired server. Also, the persistent search could be directed against one of the replication servers if desired. For another workload scenario, several instances of the persistent search can be run, with each detecting a different change type (or combination thereof).

## Appendix D. About our test 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 17 summarizes the workloads we run during our prime shift and off shift. We describe each workload in more detail below.

Table 17. Summary of our workloads

Shift	Base system workloads	Application enablement workloads	Networking workloads	Database product workloads
<b>Prime shift</b>	<ul style="list-style-type: none"> <li>Automatic tape switching</li> <li>Batch pipes</li> <li>JES2/JES3 printer simulators</li> </ul>	<ul style="list-style-type: none"> <li>Enterprise Identity Mapping (EIM)</li> <li>IBM HTTP Server</li> <li>LDAP Server</li> <li>Kerberos Server</li> <li>z/OS UNIX Shelltest (rlogin/telnet)</li> <li>z/OS UNIX Shelltest (TSO)</li> <li>WebSphere Application Server for z/OS</li> <li>WebSphere MQ for z/OS</li> <li>WebSphere Message Broker</li> </ul>	<ul style="list-style-type: none"> <li>AutoWEB</li> <li>FTP workloads</li> <li>MMFACTS for NFS</li> <li>NFSWL</li> <li>Silk Test NFS video stream</li> <li>TCP/IP CICS sockets</li> <li>TN3270</li> </ul>	<ul style="list-style-type: none"> <li>CICS DBCTL</li> <li>CICS/DB2</li> <li>CICS/QMF online queries</li> <li>CICS/RLS batch</li> <li>CICS/RLS online</li> <li>CICS/NRLS batch</li> <li>CICS/NRLS online</li> <li>DB2 Connect</li> <li>DB2 online reorganization</li> <li>DB2/RRS stored procedure</li> <li>IMS AJS</li> <li>IMS/DB2</li> <li>IMS full function</li> <li>IMS Java</li> <li>IMS SMQ fast path</li> <li>QMF™ batch queries</li> </ul>
<b>Off shift</b>	<ul style="list-style-type: none"> <li>Random batch</li> <li>Automatic tape switching</li> <li>JES2/JES3 printer simulators</li> </ul>	<ul style="list-style-type: none"> <li>Enterprise Identity Mapping (EIM)</li> <li>IBM HTTP Server</li> <li>LDAP Server</li> <li>Kerberos Server</li> <li>z/OS UNIX Shelltest (rlogin/telnet)</li> <li>z/OS UNIX Shelltest (TSO)</li> <li>WebSphere Application Server for z/OS</li> <li>WebSphere MQ for z/OS</li> <li>WebSphere Message Broker</li> </ul>	<ul style="list-style-type: none"> <li>FTP workloads</li> <li>Silk Test NFS video stream</li> <li>MMFACTS for NFS</li> </ul>	<ul style="list-style-type: none"> <li>CICS /DBCTL</li> <li>CICS/DB2</li> <li>CICS/RLS batch</li> <li>CICS RLS online</li> <li>CICS/NRLS batch</li> <li>CICS/NRLS online</li> <li>DB2 DDF</li> <li>DB2 utility</li> <li>IMS/DB2</li> <li>IMS Java</li> <li>IMS utility</li> <li>MQ/DB2 bookstore application</li> <li>QMF online queries</li> </ul>

### 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 test report, 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 Tivoli Workload Scheduler (TWS, formerly called 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.

#### **Tivoli Workload Scheduler (TWS) EXIT 51 tip**

Due to changes in JES2 for z/OS V1R7, TWS has made a new EXIT called EXIT51. TWS will only support TWS 8.1 or higher for z/OS V1R7 users. If you have z/OS V1R7 and use TWS 8.1 or higher you will need to:

- compile and linkedit your usual JES2/TWS EXITS
- compile and linkedit the new EXIT51.

EQXIT51 is provided in the SEQQSAMP Lib. You will also need to add the following to both your JES2 PARM and existing OPCAXIT7 statement:

```
LOAD(TWSXIT51)
EXIT(51) ROUTINES=TWSENT51,STATUS=ENABLED
```

Once EXIT51 was installed and enabled we found no problems with our normal use of TWS 8.1.

**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.

### **HFS/zFS file system recursive copy/delete**

This TPNS driven workload copies over 700 directories from one large filesystem to another. It then deletes all directories in the copy with multiple remove (rm) commands.

### **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:

- MVS datasets
- FastCGI programs
- Counters
- Static html pages
- Static pages through an SSL connection
- REXX Exec through GWAPI
- Protection through RACF userid
- Sysplex Distributor
- Standalone http server
- Scalable http server

## ICSF

This workload runs on MVS. It is run by submitting a job through TSO. This one job kicks off 200+ other jobs. These jobs are set up to use ICSF services to access the crypto hardware available on the system. The goal is to keep these jobs running 24/7.

## LDAP Server

LDAP Server consists of the following workloads:

- Segue Silk Performer - is setup on a remote Windows machine. The workload is setup to run a Performer Script for 20 users. The script is designed to issue several LDAP commands (ldapsearch, ldapadd, ldapdelete) issued to the z/OS LDAP server. At the start of the workload simulation, each virtual user is setup to have a 15 second delay between executing the script, thus making the simulation more "customer like". This workload simulation is then executed on a 24/7 basis.
- Tivoli Access Manager - Tivoli Access Manager uses z/OS LDAP to store user information. The workload that is executed is a shell script that consists of several TAM user admin commands that places stress on the TAM/LDAP environment.
- Mindcraft Workload Simulator - The DirectoryMark benchmark is designed to measure the performance of server products that use LDAP. We have this product installed on a Windows server machine. Scripts generated by DirectoryMark are run against z/OS LDAP on a 24/7 basis.
- Authentication - This workload is driven from an AIX/RISC workstation. It runs against the IBM HTTP Server on z/OS and Apache on Linux to provide LDAP authentication when accessing protected resources.
- Swiss Army Knife (SAK) - A multipurpose C language program that utilizes the LDAP APIs to check various operational characteristics of the server code. Test specifics are controlled by configuring various parameters when calling the test program. This program exploits aliasing, referral, persistent search, and replication.

## Network Authentication Service (Kerberos)

This workload runs from the shell as a shell script. It uses the z/OS LDAP, z/OS EIM, z/OS ftp, and xLinux clients to bind through Kerberos with LDAP, EIM, and ftp.

## **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 Application Server 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
  - Access WebSphere MQ using Java Message Service (JMS)
  - Access WebSphere MQ and the Websphere Message Broker
- 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
  - Use persistent messaging
  - Use RACF or LDAP for Local OS security
  - Use WebSphere Network Deployment (ND) configuration(s)
  - Utilize Sysplex Distributor
  - Use HTTP Server / J2EE Server clustering
  - Use DB2 Legacy RRS / DB2 UDB JCC driver(s)

## **WebSphere MQ for z/OS workloads**

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

Our workloads test the following WebSphere MQ features:

- CICS Bridge
- IMS Bridge

- Distributed queuing with SSL and TCP/IP channels
- Large messages
- Shared queues
- Clustering
- Transaction coordination with RRS
- CICS Adapter

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 through z/OS UNIX shell scripts

Some of the 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.

***DQM and DQMssl:*** This workload tests the communication between z/OS queue managers using SSL TCPIP 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 local queues that have persistent messages. We also have a non-Web version of MQCICS that uses shared cluster queues with persistent messages. We defined a separate coupling facility structure for this application. Another version of the workload uses the MQ CICS adapter to process transactions. All three queues (request, reply, and initiation) are shared. All members of our queue sharing group have a CICS region monitoring the queue.

***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 Message Broker workloads

Our WebSphere Message Broker environment consists of six message brokers: three on test systems and three on production systems. All are running WebSphere Message Broker v6.0. We will refer to this broker version as WMB. 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 through z/OS UNIX shell scripts

The Web applications consist of HTML pages, Java servlets, and message flows to process the messages. These Java-based workloads have recently been converted to use WebSphere Application Server 5.1 instead of the IBM HTTP Server with the WebSphere V4.0 plugin.

**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 also have two batch-driven workloads:

**Sniffer:** This workload tests basic MQ and broker 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>)

**Football:** This workload tests basic broker publish/subscribe functionality. Using the Subscribe portion of the workload, a subscription is registered with the broker. The Publish portion publishes messages to the broker, which then routes them to the matching subscribers. Like the Sniffer workload, this workload is based on SupportPac IP13.

We have one TPNS workload that uses WMB:

**Retail\_TPNS:** This workload is another version of Retail\_IMS, but rather than being driven by WebSphere Studio Workload Simulator, it is driven by TPNS through 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 through 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.

**MMEFACTS 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

Our database product workloads include online transaction processing (OLTP) workloads, batch workloads, and our WebSphere MQ / DB2 bookstore application.

### 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 in record level sharing (RLS), local shared resource (LSR), and non-shared resource (NSR) modes
- 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<sup>®</sup> enabled (that is, they invoke Language Environment support).

*IMS data sharing workloads*: In application group one, we run the following IMS data sharing workloads:

- CICS/DBCTL
- IMS EMHQ Fast Path
- IMS Java
- 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 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 access VSAM files in RLS mode.

**VSAM/NRLS workload:** Also in application group 2, we additional workloads. One uses transactions similar to our VSAM/RLS workload but accessing non-RLS VSAM files (using CICS FORs). The other is an 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 exercises 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 TWS control and use WLM-managed initiators. Our implementation of WLM batch management is described in our December 1997 test report.

*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 TWS, 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 test report for details on how we set up this application.



## Appendix E. Some of our RMF reports

We provide the following examples of some of our RMF reports:

- "RMF Monitor I Post Processor Summary"
- "RMF Monitor III Online Sysplex Summary"
- "RMF Workload Activity in WLM goal mode" on page 347

### RMF Monitor I Post Processor Summary

The following 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.

```

RMF SUMMARY REPORT
 PAGE 001
z/OS VIR10 SYSTEM ID J80 START 08/21/2008-09.45.00 INTERVAL 00.14.59
 RPT VERSION VIR10 RMF END 08/21/2008-10.30.00 CYCLE 0.100 SECONDS

NUMBER OF INTERVALS 3

DATE TIME INT CPU DASD DASD TAPE JOB JOB TSO TSO STC STC ASCH ASCH OMVS OMVS SWAP DEMAND
MM/DD HH.MM.SS MM.SS BUSY RESP RATE RATE MAX AVE MAX AVE MAX AVE MAX AVE MAX AVE RATE PAGING
08/21 09.45.00 14.59 73.2 0.8 3177 0.0 188 186 10 8 527 525 1 0 29 20 0.00 0.01
08/21 10.00.00 14.59 84.0 0.8 3815 455.6 191 187 10 10 528 524 1 0 26 20 0.00 0.06
08/21 10.15.00 15.00 79.5 1.0 3922 557.4 188 186 11 10 530 525 0 0 28 17 0.00 0.00
1

```

```

RMF SUMMARY REPORT
 PAGE 001
z/OS VIR10 SYSTEM ID JA0 START 08/21/2008-09.45.00 INTERVAL 00.14.59
 RPT VERSION VIR10 RMF END 08/21/2008-10.30.00 CYCLE 0.100 SECONDS

NUMBER OF INTERVALS 3

DATE TIME INT CPU DASD DASD TAPE JOB JOB TSO TSO STC STC ASCH ASCH OMVS OMVS SWAP DEMAND
MM/DD HH.MM.SS MM.SS BUSY RESP RATE RATE MAX AVE MAX AVE MAX AVE MAX AVE MAX AVE RATE PAGING
08/21 09.45.00 15.00 89.6 0.8 21381 0.0 32 30 30 30 419 407 2 0 72 61 0.00 0.01
08/21 10.00.00 15.00 85.2 0.9 11054 0.0 31 16 30 30 436 417 2 0 70 61 0.00 0.14
08/21 10.15.00 14.59 83.4 1.5 3684 0.0 9 8 29 29 426 409 2 0 73 58 0.00 0.00

```

### RMF Monitor III Online Sysplex Summary

The following contains information from the *RMF Monitor III Online Sysplex Summary*. 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 1870 CICS transactions/second.

```

HARDCOPY RMF VIR10 Sysplex Summary - UTCPLXJ8 Line 1 of 103
Command ==> Scroll ==> CSR
WLM Samples: 479 Systems: 9 Date: 08/21/08 Time: 09.42.00 Range: 120 Sec

```

```
>>>>>>>XXXXXXXXXXXXXXXXXXXX<<<<<<<<
```

```

Service Definition: CAPPV0V Installed at: 08/01/08, 14.29.46
Active Policy: WLMPL01 Activated at: 08/01/08, 14.30.05
----- Goals versus Actuals ----- Trans --Avg. Resp. Time-

```

		Exec Vel	--- Response Time ---		Perf	Ended	WAIT	EXECUT	ACTUAL	
Name	T I	Goal Act	---Goal---	--Actual--	Indx	Rate	Time	Time	Time	
BATCH	W	75				0.442	0.656	21.54	22.16	
BATDISCR	S D	15				0.442	0.656	21.54	22.16	
BATI1V90	S 1	90 84			1.08	0.000	0.000	0.000	0.000	
BATI2V30	S 2	30 55			0.55	0.000	0.000	0.000	0.000	
BATI2V50	S 2	50 82			0.61	0.000				
BATI3V90	S 3	90 93			0.96	0.000	0.000	0.000	0.000	
CICS	W	N/A				<b>1870</b>	0.000	0.152	0.496	
CI280%P6	S 2	N/A	0.600	80%	90%	0.50	1432	0.000	0.071	0.295
CI290%P5	S 2	N/A	0.500	90%	80%	****	64.91	0.000	3.457	1.923
CI350%10	S 3	N/A	10.00	50%	86%	0.70	2.550	0.000	6.995	6.995
CI390%01	S 3	N/A	1.000	90%	95%	0.50	370.4	0.000	0.657	0.975
ICSS	W	75				30.88	0.000	0.015	0.015	
ICI2V50	S 2	50 77			0.65	24.70	0.000	0.018	0.018	
ICI3V50	S 3	50 60			0.83	6.175	0.000	0.004	0.004	
IMS	W	N/A				128.7	0.000	0.175	0.403	
II290%P5	S 2	N/A	1.000	90%	100%	0.50	39.89	0.000	0.039	0.058
II390%P7	S 3	N/A	0.700	90%	82%	1.20	87.42	0.000	0.238	0.566
II490%01	S 4	N/A	1.000	90%	100%	0.50	1.400	0.000	0.056	0.059
STC	W	33				62.03	0.001	3.782	3.783	
STCDISCR	S D	0.9				0.008	0.230	0.101	0.332	
STCI1V40	S 1	40 97			0.41	0.375	0.002	0.274	0.276	
STCI2V30	S 2	30 27			1.12	0.017				
STCI2V40	S 2	40 66			0.61	0.133	0.001	9.947	9.948	
STCI2V50	S 2	50 55			0.91	0.008				
STCI2V60	S 2	60 64			0.94	0.000				
STCI2V70	S 2	70 52			1.34	0.000	0.000	0.000	0.000	
STCI3V50	S 3	50 78			0.64	0.000	0.000	0.000	0.000	
STCI5V05	S 5	5 24			0.21	6.692	0.000	0.010	0.010	
STCOMVS	S	15				54.79	0.001	2.834	2.835	
	1 1	30 75			0.40	16.65	0.000	1.987	1.987	
	2 2	20 64			0.31	7.908	0.001	3.505	3.505	
	3 3	10 14			0.70	30.23	0.002	3.126	3.127	
SYSTEM	W	84				0.042	0.821	1.670	2.491	
SYSSTC	S	N/A 80	N/A			0.042				
SYSTEM	S	N/A 89	N/A			0.000	0.000	0.000	0.000	
TSO	W	61				87.74	0.000	2.942	2.942	
TSO	S 2	61	2.000	AVG 2.942	AVG 1.47	87.74	0.000	2.942	2.942	
WAS	W	30				357.5	0.004	0.107	0.110	
WI1VEL30	S 1	30 87			0.34	0.000	0.000	0.000	0.000	
WI180%01	S 1	38	1.000	80%	98%	0.50	357.5	0.004	0.107	0.110
WI2VEL50	S 2	50 22			2.23	0.000				
BATCHHI	R	84				0.000	0.000	0.000	0.000	
BATCHLOW	R	93				0.000	0.000	0.000	0.000	
BATCHMED	R	82				0.000	0.000	0.000	0.000	
----- Goals versus Actuals -----						Trans	--Avg.	Resp.	Time-	
Name	T I	Goal Act	---Goal---	--Actual--	Indx	Rate	Time	Time	Time	
BATDISCR	R	14				0.442	0.656	21.54	22.16	
CICSCONV	R	N/A				2.550	0.000	6.995	6.995	
CICSCPSM	R	N/A				1.900	0.000	0.001	0.001	
CICSSLOW	R	N/A				181.4	0.000	0.657	1.897	
CICSMED	R	N/A				1497	0.000	0.115	0.366	
CICSMISC	R	N/A				189.0	0.000	0.090	0.090	
CICSSSTC	R	63				0.000	0.000	0.000	0.000	
CICSWEB	R	83				0.000	0.000	0.000	0.000	
CQSREP	R	57				0.000	0.000	0.000	0.000	
CSQCCHIN	R	50				0.000	0.000	0.000	0.000	
CSQCMSTR	R	75				0.000	0.000	0.000	0.000	

I	DBWIDIST	R	0.0	0.000	0.000	0.000	0.000
I	DB2IRLM	R	90	0.000	0.000	0.000	0.000
I	DB2REP	R	46	0.000	0.000	0.000	0.000
I	DB2WLM	R	32	0.000	0.000	0.000	0.000
I	DDF	R	25	6.692	0.000	0.010	0.010
I	FDBRREP	R	76	0.000	0.000	0.000	0.000
I	GRS	R	93	0.000	0.000	0.000	0.000
I	IMSLOW	R	N/A	88.82	0.000	0.235	0.558
I	IMSMED	R	N/A	39.89	0.000	0.039	0.058
I	IMSREG	R	82	0.000	0.000	0.000	0.000
I	IMSREP	R	85	0.000	0.000	0.000	0.000

## RMF Workload Activity in WLM goal mode

The following illustrates a couple of sections from our RMF *Workload Activity* report in WLM goal mode. This report is based on a 15-minute interval. Highlighted on the report you see 93.6% of our CICS transactions are completing in 0.5 seconds, and our CICS workload is processing 1898.89 transactions per second.

```

 W O R K L O A D A C T I V I T Y
 PAGE 12
z/OS V1R10 SYSPLEX UTCPLXJ8 START 08/21/2008-10.00.00 INTERVAL 000.15.00 MODE = GOAL
 RPT VERSION V1R10 RMF END 08/21/2008-10.15.00
 POLICY ACTIVATION DATE/TIME 08/01/2008 14.30.05

```

---TIME---	--NUMBER OF TRANSACTIONS--		-----RESPONSE TIME DISTRIBUTION-----													
	HH.MM.SS.TTT	CUM TOTAL	IN BUCKET	CUM TOTAL	IN BUCKET	0	10	20	30	40	50	60	70	80	90	100
< 00.00.00.500	320K	320K	93.6	93.6	93.6	..... ..... ..... ..... ..... ..... ..... ..... ..... .....										
<= 00.00.00.600	321K	964	93.8	0.3	>											
<= 00.00.00.700	322K	1096	94.2	0.3	>											
<= 00.00.00.800	323K	1217	94.5	0.4	>											
<= 00.00.00.900	324K	1251	94.9	0.4	>											
<= 00.00.01.000	326K	1493	95.3	0.4	>											
<= 00.00.01.100	329K	3751	96.4	1.1	>											
<= 00.00.01.200	331K	1061	96.7	0.3	>											
<= 00.00.01.300	331K	512	96.9	0.1	>											
<= 00.00.01.400	332K	639	97.1	0.2	>											
<= 00.00.01.500	332K	558	97.2	0.2	>											
<= 00.00.02.000	333K	1192	97.6	0.3	>											
<= 00.00.04.000	336K	2667	98.4	0.8	>											
> 00.00.04.000	342K	5625	100	1.6	>>											

===== WORKLOAD

```

REPORT BY: POLICY=WLMPOL01 WORKLOAD=CICS
 cics workload

```

-TRANSACTIONS-	TRANS-TIME	HHH.MM.SS.TTT	
AVG	0.00	ACTUAL	264
MPL	0.00	EXECUTION	94
ENDED	1708966	QUEUED	0
END/S	1898.89	R/S AFFIN	0
#SWAPS	0	INELIGIBLE	0
EXCTD	1413408	CONVERSION	0
AVG ENC	0.00	STD DEV	14.925
REM ENC	0.00		
MS ENC	0.00		



---

## 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 Vol I* 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/systems/z/os/zos/bkserv/](http://www.ibm.com/systems/z/os/zos/bkserv/)



---

## 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.

---

## Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (<sup>®</sup> or <sup>™</sup>), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Intel is a registered trademark of Intel Corporation in the United States and other countries.

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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



---

# Index

## A

accessibility 349  
alternate subchannel set, PPRC secondary devices in 133  
application enablement  
  configuration 321  
ARM enablement 326  
auxiliary storage shortage messages 37

## B

Basic HyperSwap 113, 114

## C

Capacity on Demand 8  
Capacity Provisioning 8  
channel subsystem  
  coupling facility channels 317  
  CTC channels 317  
  ESCON channels 317  
  FICON channels 317  
CIB channels 13  
CICS TG  
  *See* CICS Transaction Gateway  
CICS Transaction Gateway 195  
  daemon statistics 195  
  migrating to CICS TG V7 195  
  port requirements for statistics interface 195  
  reserving the statistics interface port 196  
CICS Transaction Server for z/OS 189  
  migrating CICSplex SM 191  
  migrating the CICSplex SM Web User Interface 193  
  migrating the CMASs 192  
  migrating the MASs 192  
  migrating to CICS TS 3.2 189  
  migration experiences 193  
CICSplex SM 191  
  Web User Interface 193  
configuration  
  hardware details 313  
  mainframe servers 313  
  hardware overview 311  
  ICSF 329  
  LDAP  
    IBM Tivoli Directory Server (IBM TDS) 331  
    Integrated Security Services (ISS) 331  
  Network Authentication Service 330  
  networking 321  
  Parallel Sysplex hardware 311  
  sysplex hardware details  
    coupling facilities 315  
    other sysplex hardware 316  
  sysplex software 317  
  VTAM 324

configuration (*continued*)  
  WebSphere Application Server for z/OS 217  
Coordinated timing network (CTN) 39  
coupling facility channels  
  CIB 13  
Cryptographic Services PKI Services 163

## D

database workloads 341  
DFSYDRU0 198  
DFSYPX0 198  
disability 349  
DSPSCIX0 198  
DVIPA 326  
dynamic enablement  
  relation to IFAPRDxx parmlib member 318

## E

EAV  
  *See* extended address volumes  
environment  
  networking enablement 321  
  Parallel Sysplex 311  
  security 329  
  WebSphere Application Server for z/OS 217  
  workloads 335  
ESCON channels 317  
extended address volumes 117  
  migrating to 119  
  requirements for 117  
    hardware 117  
    software 117  
  setting up 117  
    DASD 117  
    DFSMS 118  
    ICKDSF 118

## F

FICON channels 317  
  FICON native (FC) mode 317

## H

hardware  
  configuration details 313  
  mainframe servers 313  
  configuration overview 311  
  Parallel Sysplex configuration 311  
High Performance FICON for System z 129  
  command examples 129  
HiperDispatch 7

## I

IBM TotalStorage Productivity Center for Replication for System z  
  *See* TPC-R V3.4  
IBM zIIP  
  *See* zIIP, IBM  
ICSF configuration 329  
ICSF, HCR7750 143  
  exercising CPACF on the z10 EC 143  
  migrating to 143  
IFAPRDxx parmlib member  
  relation to dynamic enablement 318  
IMS Transaction Manager Resource Adapter V10.2  
  migrating to 203  
IMS Version 10  
  IMS exits 198  
  IMS Java migration 198  
  IMS syntax checker 199  
  IMS utilities 201  
  IRLM support 201  
  migrating to 197  
  migrating V9 RECON data sets to V10 199  
  migration and coexistence 197  
  DFSYDRU0 198  
  DFSYPX0 198  
  DSPSCIX0 198  
  our staged migration 197  
IRA210E, message 37  
IRA211I, message 37  
IRA420I, message 38  
IRA421D, message 38

## J

JZOS 205  
  batch launcher 205  
  installation and setup 205  
  JZOS Cookbook 206  
  MVS operations with 205

## K

keyboard 349

## L

LDAP  
  configuration overview 331  
LDAP Server  
  *See* Security Server LDAP Server  
Linux on zSeries 237  
  capacity management 269  
  DASD groups and automatic allocation 269  
  environment 237, 238  
  configuration 240  
  goals and priorities 237

- Linux on zSeries 237 *(continued)*
  - implementation 238
  - production systems and usages 242
  - test (MDAT) systems and usages 244
  - workloads 239
- middleware upgrades 251
  - upgrading application servers and deployment server 251
  - upgrading Tivoli Storage Manager 258
- operating system upgrades 245
  - Tivoli Storage Manager server operating system upgrade 245
  - upgrading WebSphere Application Server prior to operating system upgrade 251
  - z/VM 5.3 to z/VM 5.4 transition notes 251
- security management 275
  - custom DirMaint usermod for integration with RACF 306
  - installing WebSphere Edge Components Load Balancer V7.0 293
  - LDAP load balancing and failover 291
  - LDAP replication 289
  - migrating to IBM Tivoli Directory Server V6.1 285
  - upgrading the Tivoli Access Manager policy server 275
- software maintenance strategy and methodology 245
- systems management 259
  - changing default to disk by-path in FSTAB and kernel 266
  - installing open source VPN server 259
  - migrating from ReiserFS v3 to ext3 267

## M

- messages
  - IRA210E 37
  - IRA211I 37
  - IRA420I 38
  - IRA421D 38
- MQSeries
  - See* WebSphere Business Integration

## N

- naming conventions
  - CICS and IMS subsystem jobnames 319
- Network Authentication Service (Kerberos) 147
  - configuration 330
  - password phrase support 147
- networking
  - configuration 321
  - workloads 325
- networking workloads 340

- NFS
  - migrating to the OS/390 NFS 324
  - preparing for system outages 325
  - recovery 325
- NFS environment
  - acquiring DVIPA 326
  - setting up ARM 326
- Notices 351

## O

- OSA-Express3
  - multi-port support for IP 13

## P

- pageable storage shortage messages 37, 38
- Parallel Sysplex 311
  - hardware configuration 311
- Parallel Sysplex InfiniBand
  - CIB channels 13
  - coupling facility links 13
- performance
  - See also* RMF
  - RMF reports 345
- PPRC devices in alternate subchannel set 133
- PSIFB
  - See* Parallel Sysplex InfiniBand

## R

- RACF
  - See* z/OS Security Server RACF
- Recovery
  - preparing for with NFS 325
- RMF 345
  - Monitor I Post Processor Summary 345
  - Monitor III Online Sysplex Summary 345
  - Workload Activity in WLM goal mode 347

## S

- security
  - Cryptographic Services PKI Services
    - See* Cryptographic Services PKI Services
  - environment 329
  - ICSF configuration 329
  - LDAP Server
    - See* Security Server LDAP Server
  - Network Authentication Service (Kerberos)
    - See also* Network Authentication Service (Kerberos)
    - configuration 330
  - System SSL
    - See* System SSL
  - Security Server LDAP Server 151
  - IBM TDS plug-in support 151

- Security Server LDAP Server *(continued)*
  - LDAP server wait for DB2 startup 154
  - LDAP support for RACF custom fields 158
  - TDS differentiation, currency, and certification validation 153
  - TDS password phrase support 156
  - Using SHA and MD5 encrypted passwords 160
- Server Time Protocol (STP) 39
  - migration experiences 46
  - overview 39
  - planning 42
  - terminology 40
- Serve Oriented Architecture (SOA)
  - deploying a solution 231
- shortcut keys 349
- SOA
  - See* Serve Oriented Architecture (SOA)
- software
  - configuration overview 318
  - sysplex configuration 317
  - special secondary devices 133
  - storage shortage messages
    - auxiliary storage 37
    - pageable storage 37, 38
  - sysplex
    - See* Parallel Sysplex
  - sysplex root file system, migrating 184
- System SSL 165
  - 4096-bit hardware support 166
  - gskkyman enhancements 167
  - System SSL CPACF hardware support 165
- System z10 EC 7
  - Capacity Provisioning 8
  - HiperDispatch 7
- System z10 Integrated Information Processor
  - See* zIIP, IBM
- System z9 Integrated Information Processor
  - See* zIIP, IBM

## T

- TPC-R V3.4 113
  - documentation 115
  - our environment 113
  - setting up 114

## U

- UNIX
  - See* z/OS UNIX System Services
- USS\_CLIENT\_MOUNTS health check 182
- USS\_PARMLIB\_MOUNTS health check 179

## V

- VTAM
  - configuration 324

## W

- WebSphere Application Server for z/OS
  - changes and updates 221
    - CICS Transaction Gateway V7 222
    - DB2 client information 222
    - IMS Transaction Manager Resource Adapter V10.2 222
    - TPC-R V3.4 226
    - WebSphere Application Server for z/OS V6.1 221
  - configuration and workloads 218
    - configuration updates 218
    - naming conventions 220
    - test and production configurations 218
    - Web application workloads 219
  - information resources 226
  - our test environment 217
    - current software products and release levels 217
    - software products and release levels
      - workstation software products 217
      - z/OS software products 217
  - using 217
- WebSphere Business Integration 207
  - high availability for WebSphere MQ 212
  - MQCICS, WebSphere MQ-CICS adapter/bridge workload 214
  - shared queues and coupling facility structures
    - coupling facility structure configuration 208
  - using shared queues and coupling facility structures 207
    - queue sharing group configuration 207
    - recovery behavior with queue managers and coupling facility structures 208
  - WebSphere Message Broker 210
- WebSphere Message Broker workloads 339
- WebSphere MQ
  - See WebSphere Business Integration
- WebSphere MQ for z/OS workloads 338
- WebSphere Process Server for z/OS 227
  - installation and configuration 227
  - security 228
- WebSphere Service Registry and Repository for z/OS 229
  - installation and configuration 229
  - security 230
- workload
  - networking 325
- workloads 335
  - application enablement 336
  - automatic tape switching 336
  - base system functions 335
  - batch, database 342
  - bookstore application 343
  - database products 341
  - database, OLTP 341
  - DB2 batch 343
  - DB2 data sharing 342

workloads (*continued*)

- Enterprise Identity Mapping (EIM) 336
- file systems 336
- IBM HTTP Server 336
- ICSF 337
- IMS data sharing 342
- LDAP Server 337
- Network Authentication Service (Kerberos) 337
- networking 340
- VSAM/NRLS 342
- VSAM/RLS data sharing 342
- WebSphere Application Server for z/OS 338
- WebSphere Message Broker 339
- WebSphere MQ for z/OS 338
- z/OS UNIX shell (rlogin/telnet) 338
- z/OS UNIX shell (TSO) 338

## Z

- z/OS Security Server LDAP Server
  - See Security Server LDAP Server
- z/OS Security Server RACF
  - enhancements in z/OS V1R10 135
    - custom fields 140
    - password reset granularity 138
    - RACDCERT support for 4096-bit keys 142
  - reorganizing our RACF databases 135
- z/OS UNIX System Services 169
  - DIRLIST service 172
  - enhancements in z/OS V1R10 169
  - health checks
    - USS\_CLIENT\_MOUNTS 179
    - USS\_PARMLIB\_MOUNTS 179
- z/FS enhancements 183
  - migrating sysplex root file system 184
- z/OS V1R10
  - migration 3
    - base migration activities 4
    - base migration experiences 3
    - concatenated PARMLIB 5
    - high-level migration process 3
    - mixed product levels 4
    - other migration experiences 5
    - recompiling automation EXECs 5
- zHPF
  - See High Performance FICON for System z
- zIIP, IBM 97
  - configuring 98
  - DB2 workloads 102
  - monitoring utilization 100
  - OMEGAMON XE support for 103
  - prerequisites for 97
  - System Data Mover (SDM) 108
  - zIIP assisted IPSec 107



---

## Readers' Comments — We'd Like to Hear from You

z/OS  
System z Platform Test Report  
for z/OS and Linux Virtual Servers  
Version 1 Release 10

Publication No. SA22-7997-08

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product 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.

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. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

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



Fold and Tape

Please do not staple

Fold and Tape



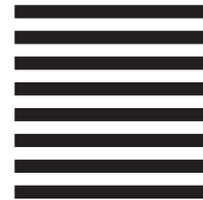
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# 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





Printed in USA

SA22-7997-08

