z/OS

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

Version 1 Release 9

SA22-7997-07

z/OS

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

Version 1 Release 9

SA22-7997-07

Note!
Before using this information and the products it supports, be sure to read the general information under "Notices" on page 313.

Eighth Edition, June 2008

This is a major revision of SA22-7997-06.

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/servers/eserver/zseries/zos/integtst/

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

Opening remarks

A message from our team

Welcome to the *IBM System z Platform Test Report for z/OS and Linux Virtual Servers*. Our team focuses on 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 xx.

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

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

Important—Currency of the softcopy edition

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

Because we produce our test reports twice a year, June and December, we cannot meet the production deadline for the softcopy collections that coincide with the product's GA release and the softcopy collection refresh date six months later. Therefore, there is normally a one-edition lag between the release of our latest test report edition and the softcopy collection in which it is included. That is, the test report that appears in any given softcopy collection is normally one edition behind the most current edition available on the Web.

If you obtained this document from a softcopy collection on CD-ROM or DVD, you can get the most current edition from the IBM® System z Platform Test Report Web site at www.ibm.com/servers/eserver/zseries/zos/integtst/.

Contents

Figures

Tables

About this document

This document is a test report written from the perspective of a system programmer. The IBM 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 269
- "Our Parallel Sysplex software configuration" on page 275
- "Our networking configuration" on page 279
- "Appendix C. About our security environment" on page 287
- "Appendix D. About our test workloads" on page 293

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 z/OS Integration Test Web site at:

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

If you cannot get to our Web site for some reason, see "Appendix F. Availability of our test reports" on page 309 for other ways to access our test reports.

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
z/OS Parallel Sysplex Overview	SA22-7661
z/OS MVS Setting Up a Sysplex	SA22-7625
z/OS Parallel Sysplex Application Migration	SA22-7662
z/OS Planning for Installation	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/servers/eserver/zseries/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 form on our Web site at:

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

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

System z Platform Evaluation Test (zPET) focuses on 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 V1R9

This topic describes our migration to z/OS V1R9. Our migration experiences include:

- "z/OS V1R9 base migration experiences"
- "Other z/OS V1R9 migration experiences" on page 5
- "Coupling facility maintenance enhancements" on page 9
- "Testing greater than 32 CPU support" on page 10

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 V1R8 and z/OS.e V1R8	*zSeries® Platform Test Report for z/OS and Linux Virtual Servers, June 2007*
z/OS V1R7 and z/OS.e V1R7	*zSeries Platform Test Report for z/OS and Linux Virtual Servers, December 2005*

z/OS V1R9 base migration experiences

This topic describes our experiences with our base migration to z/OS V1R9, 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 V1R9 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 V1R8 to z/OS V1R9.

Table 2. Our high-level migration process for z/OS V1R9

Stage	Description
Updating PARMLIB for z/OS V1R9	We created SYS1.PETR19.PARMLIB to contain all the PARMLIB members that changed for z/OS V1R9 and we used our LOAD*xx* member for migrating our systems one at a time. (See our December 1997 test report for an example of how we use LOAD*xx* to migrate individual systems.)

Table 2. Our high-level migration process for z/OS V1R9 (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 *z/OS Planning for Installation* and make sure you install the fixes for any APARs that relate to your configuration before you migrate.
IPLing our first z/OS V1R9 image	We brought up z/OS V1R9 on our Z3 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 V1R9 image without first running the IRRMIN00 utility with PARM=UPDATE. As expected, the following message appeared: `ICH579E RACF TEMPLATES ON DATABASE ARE DOWNLEVEL` RACF initialization still completed successfully. We then ran IRRMIN00 with PARM=UPDATE to dynamically update the templates on all six RACF data sets without the need for an IPL. (See *z/OS Security Server RACF System Programmer's Guide*, SA22-7681 for details about RACF templates.)
IPLing additional z/OS V1R9 images	We continued to bring up additional z/OS V1R9 images across our sysplex, as follows: 1. Brought up z/OS V1R9 on our Z1 test system and ran with it for a week. 2. Migrated our last test system, Z2, and ran for a week. 3. Migrated some of our production systems, JA0, JE0, JC0 and J80, and ran with it for a couple of days. 4. At this point, we took two of our production V1R9 images, JC0 and JE0, back down to V1R8. This is part of our focus on migration testing and fallback. We ran for two full days and experienced no fallback issues. 5. Migrated three additional production systems, Z0, JB0, and TPN, and ran for a week. 6. Migrated the remaining production systems, JF0 and J90, to V1R9.

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 V1R8 and z/OS V1R9
- z/OS V1R8 JES2 and z/OS V1R9 JES2
- z/OS V1R8 JES3 and z/OS V1R9 JES3

Using concatenated PARMLIB

We continue to use concatenated PARMLIB support to add or update PARMLIB members for z/OS V1R9. 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 V1R9 in one place and makes it easier to migrate multiple systems. Rather than change many PARMLIB members each time we migrate another system to V1R9, we just add the PARMLIB statements at the appropriate places in SYS0.IPLPARM(LOAD*xx*) to allow that system to use SYS1.PETR19.PARMLIB.

Recompiling REXX EXECs for automation

We recompiled our IBM Tivoli® System Automation for z/OS REXX™ EXECs when we migrated to z/OS V1R9. We discuss the need to recompile these REXX EXECs in our our December 1997 test report.

Other z/OS V1R9 migration experiences

This topic highlights additional details about some of our migration experiences that are specific to z/OS V1R9.

z/OS V1R9 Unicode support enhancements

In z/OS V1R9, the LE C Run-Time Library **iconv()** family of functions is updated to use the Unicode Conversion Services. This change is generally transparent through the magic of z/OS!

One item in this area that we would like to point out, especially for those migrating from releases prior to z/OS V1R7, is that you may be able to remove any customized conversion image. In our case, we had created a customized image for DB2 support many years ago. The default conversion image that comes with z/OS V1R7 and above provides the most commonly used code page conversion tables needed to satisfy the DB2 code page conversion needs and eliminates the need to create or use a customized conversion image.

For further details, see the section “Remove CUNUNIxx parmlib members” in *z/OS Migration*.

System symbols documentation issue

During our migration to z/OS V1R9, we discovered an error in the V1R9 documentation regarding system symbols.

Starting in z/OS V1R9, the same ability to substring system symbols as described in *z/OS MVS Initialization and Tuning Reference* is now available for system symbols that are being used in JCL for started task procedures and TSO logon procedures. Previously, the ability to substring system symbols in JCL was not provided.

We discovered a problem when one of our started task procedures that offloads LOGREC data to a GDG based data set started failing. The following is an example of the procedure that failed:

```
//DUMPEREP PROC OUT=C,SYS=UNK,CAT=ABEND013,LOGDSN=ABEND013
//*************************************************************
//*  COPIES ALL THE RECORDS TO                                *
//*  'ICRWRTR.EREPXXX.G0000V00', WHERE XXX IS THE SYSTEM SMFID. *
//*************************************************************
```

```
//IEFPROC EXEC PGM=IFCEREP1,REGION=2M,
//            PARM='ACC=Y,ZERO=Y,PRINT=NO'
//OUTPUT1  OUTPUT FORMDEF=BJGR
//SERLOG    DD DSN=&LOGDSN,DISP=SHR
//EREPPT    DD SYSOUT=&OUT,OUTPUT=*.OUTPUT1,DCB=BLKSIZE=133
//TOURIST   DD SYSOUT=&OUT,OUTPUT=*.OUTPUT1,DCB=BLKSIZE=133
//DIRECTWK  DD UNIT=SYSDA,SPACE=(CYL,(10))
//ACCDEV    DD DSN=ICRWRTR.EREP&SYSNAME(0),DISP=MOD,
//             DCB=(ICRWRTR.EREP.MODEL.DSCB)
//ACCIN     DD DUMMY,DCB=BLKSIZE=133
//SYSIN     DD DUMMY,DCB=BLKSIZE=133,
//             VOL=SER=D83I80,UNIT=3390,DISP=OLD,DSN=&CAT&SYS
```

For instance, in this JCL, when the value of the &SYSNAME system symbol is Z1, the use of the &SYSNAME system symbol followed by "(0)" in the data set name ICRWRTR.EREP&SYSNAME(0) has always worked by generating a substitution value of ICRWRTR.EREPZ1(0). However, after migrating to z/OS V1R9, the substitution value that was generated was ICRWRTR.EREPZ, which caused incorrect results. The correction was to change the JCL to specify the data set name as either ICRWRTR.EREP&SYSNAME.(0) or ICRWRTR.EREP&SYSNAME(+0). The use of the + character in substrings is not allowed.

The information in *z/OS MVS JCL Reference* has been changed to document the ability to substring system symbols in started task procedures and TSO logon procedures.

Using the IBM Migration Checker for z/OS

From the Migration Checker documentation:

> The IBM Migration Checker for z/OS is a tool composed of several batch programs that check the applicability of certain migration actions on your currently running system. You can run each batch program independently (using separate jobs) or you can run them all "at once" (serially, using a single job). The IBM Migration Checker for z/OS was introduced for migrations from z/OS V1R7 to z/OS V1R8. However, other migration paths at the time were tolerated. The tool has been subsequently enhanced for migrations to z/OS V1R9. The program detects the z/OS release, and the output indicates whether any migration information can be provided for the release.

When we begin to evaluate a new release of z/OS, the Migration Checker is a useful tool for evaluation of base components and a verification of several basic configuration options.

The latest release of the IBM Migration Checker for z/OS is available from the z/OS downloads page at www.ibm.com/servers/eserver/zseries/zos/downloads/. Do the following to obtain the Migration Checker:

1. From the z/OS downloads page, download the three binary files, which are in TSO XMIT format, to your workstation.
2. Use FTP to transfer the binary files from your workstation to your z/OS host. The following example shows the FTP commands to do this:

```
FTP JC0EIP
<enter your user ID and password when prompted>
BIN
QUOTE SITE RECFM=FB LRECL=80 BLKSIZE=32720 TRACKS PRIM=100 SEC=100
```

```
PUT migrate.checker.clist.bin
PUT migrate.checker.jcl.bin
PUT migrate.checker.load.bin
QUIT
```

3. In a TSO session issue the RECEIVE command to receive each file into a PDS. Instructions are in the documentation provided with the application.

We wrote a REXX EXEC named MC that will take the output produced by the Migration Checker and write a sequential data set of HTML code that can be viewed with a Web browser. The REXX EXEC is invoked using the following JCL:

```
//AJNIMSS1 JOB 'SDSFTST','AL NIMS',REGION=4M,
//  CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=AJNIMS
//*
//MC       EXEC PGM=IKJEFT01,PARM='%MC J80'
//SYSPROC  DD DISP=SHR,DSN=AJNIMS.CLIST
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//MIGIN    DD  DISP=SHR,DSN=AJNIMS.ALPHA.MIGRATE.CHECKER.J80.OUTPUT
//HTMLOUT  DD  DISP=OLD,DSN=AJNIMS.MIGRATE.CHECK.HTML(J80)
//*
//*
```

The data set identified by the MIGIN ddname contains the output produced by the Migration Checker. The data set identified by the HTMLOUT ddname will contain the HTML formatted output. The output consists of sequential data, so it can be placed either in a PS or PO data set.

Our MC EXEC uses IBM SmartBatch for OS/390 BatchPipeWorks. The following is the REXX code for our MC EXEC:

```
/*** REXX *** */

Arg MCSys

Address "TSO"

If MCSys = '' Then Do
  Say "No System Specified"
  Exit
End

/*********************************************************************/
/*                                                                   */
/* DDNames Used: MIGIN <- Output Dataset from the IBM Migration      */
/*                        Checker for z/OS Run.                      */
/*               HTMLOUT <- HTML Output File Destination.            */
/*                          Output is a single Sequental             */
/*                          Data Set Output.                         */
/*                                                                   */
/*********************************************************************/

DDInfo = LISTDSI('MIGIN' 'FILE')

If DDInfo > 4  Then Do
  Say '-------------------------------------------------------------'
  Say 'DDNAME: MIGIN was not allocated.'
  Say 'DDNAME(MIGIN) Should be allocated to the'
  Say ' IBM Migration Checker for z/OS Output Dataset.'
  Say '-------------------------------------------------------------'
  Exit 20
End
```

```
If SYSDSORG ^= 'PO' Then Do
  Say '-------------------------------------------------------'
  Say 'The Dataset' SYSDSNAME
  Say 'Is not a Partitioned Dataset and not proper input for this'
  Say 'program.  DSORG Found:' SYSDSORG
  Say '-------------------------------------------------------'
  Exit 20
End

"PIPE LISTISPF '"||SYSDSNAME||"'",
    "| CHOP 8",
    "| STRIP BOTH",
    "| STEM MCMin."

If MCMin.1 ^= '$MIGALL' Then Do
  Say '-------------------------------------------------------'
  Say 'The Dataset' SYSDSNAME
  Say 'The First Member in the Data Set is not $MIGALL'
  Say '-------------------------------------------------------'
  Exit 20
End

/*****************************************************************/
/*                                                               */
/* Load in the $MIGALL member, the INDEX Member.                 */
/*                                                               */
/*****************************************************************/
"PIPE MEMBERS '"||SYSDSNAME||"' $MIGALL",
    "| STEM MigAll."

Queue '<HTML><HEAD> '
Queue '<title>Migrate Check of '||MCSys||'</title>'
Queue '<STYLE type="text/css"> '
Queue ' body '
Queue ' { Background-color: White ; '
Queue '   Color: Blue } '
Queue '</STYLE></HEAD> '
Queue '<BODY>'
Queue ,
    '<center><h1>IBM Migration Checker for z/OS<br>',
         'System Checked:' MCSys,
         '</h1></center>'
Queue '<P><A Name="TOP"></A></P> '

Queue 'Index'
Queue '<br><PRE>'

/*****************************************************************/
/*                                                               */
/* Go through the INDEX creating "Anchor" refrences to the       */
/* other members of the output dataset.                          */
/*                                                               */
/*****************************************************************/
Do i = 1 to MigAll.0

  If MigAll.i = "" Then Iterate

  If Substr(MigAll.i,1,2) = '/*' Then ,
    Queue MigAll.i
```

```
  Else ,
    Queue "<A HREF=#"||Word(MigAll.i,1)||">"||,
           Left(Word(MigAll.i,1),8)||"</A>" Subword(MigAll.i,2)
End

Queue "<br>"

/*********************************************************************/
/*                                                                   */
/* Skip the first entry in the list of member names, $MIGALL,        */
/* and queue up each member into the stream.                         */
/*                                                                   */
/* At begining of the member, create the "Anchor" point that is      */
/* refrenced in the INDEX at the begining.                           */
/*                                                                   */
/*********************************************************************/
Do i = 2 to MCMin.0

  Queue '<A NAME="'||MCMin.i||'">' ,
       '<FONT Color="RED"><b>' ,
        MCMIn.i||"</b></FONT></A>"
  "PIPE MEMBERS '"||SYSDSNAME||"' "||MCMin.i,
      "| STEM MIn."
  Do j = 1 to MIn.0
    Queue Min.j
  End
End

Queue '</PRE>'
Queue '<center><A HREF=#TOP><FONT color=Blue>Top</A></font></center>'
Queue '</BODY></HTML>'
Queue

'PIPE STACK',
    "| > DDName=HTMLOUT"

Return
```

Coupling facility maintenance enhancements

With z/OS V1R9, you can place a coupling facility (CF) into maintenance mode, which can simplify the process of removing all of the structures from a CF. Once a CF is placed in maintenance mode, XCF will not allocate any new structures on that CF. You can then remove the existing structures on the CF without being concerned that new allocations on the CF will occur.

See the topic "Sample procedure for coupling facility maintenance" in *z/OS MVS Setting Up a Sysplex*, SA22-7625 for the commands to use to place a CF into maintenance mode. However, note that in the example shown, the keyword MAINMODE is a typographical error; the correct keyword is MAINTMODE.

Toleration support for coupling facility maintenance enhancements: If you have a sysplex with mixed levels of z/OS, you should apply the fix for APAR OA17685. This will enable z/OS systems at z/OS V1R6 and higher to recognize that a CF is in maintenance mode and prevent those systems from allocating structures on the CF. You must use a z/OS V1R9 system to place a CF into or remove a CF from maintenance mode, but you do not have to wait for the entire sysplex to be migrated to z/OS V1R9 to begin using this function.

Our observations with coupling facility maintenance enhancements: When you place a CF into maintenance mode, the CF remains connected and the CF link paths remain online. If you have previously used the CF drain function of Tivoli System Automation for z/OS, you are used to seeing the CF link paths to the drained CF be taken offline from each z/OS image to prevent further structure allocations on the drained CF. This is no longer necessary with the new CF maintenance mode.

Testing greater than 32 CPU support

With the combination of z/OS V1R9 and System z9™ Enterprise Class (EC), you now can define a single z/OS LPAR image with up to 54 CPUs, which includes System z Application Assist Processors (zAAPs) and System z9 Integrated Information Processors (zIIPs). This function provides flexibility in choosing how to grow: horizontally, with Parallel Sysplex, or vertically, using the greater than 32 CPU support.

Our testing of this new support occurred in two phases:

- **Phase 1: One LPAR with dedicated general purpose CPUs**

  On a System z9 EC, we defined only one z/OS LPAR with 50 dedicated general purpose CPUs, two zAAPs, and two zIIPs. On this image, we ran a high stress level of our IMS, CICS, DB2, WebSphere MQ, z/OS UNIX®, and WebSphere Application Server workloads with a constant number of transactions. We started with 24 general purposes CPUs online and then varied CPUs online—in groups of eight, twice, and a final group of two—to achieve a total of 50 general purpose CPUs, while maintaining the same level of transactions. We preserved the percent of CPU utilization value and found that it scaled as expected.
- **Phase 2: Two LPARs with shared general purpose CPUs**

  We defined two z/OS LPARs, each with 50 shared general purpose CPUs, two shared zAAPs, and two shared zIIPs. On one z/OS LPAR (J80) we ran the high stress IMS, CICS, DB2, and WebSphere Application Server workloads. On the other LPAR (J90) we ran low priority workloads (WebSphere MQ, batch, and z/OS UNIX). On the high stress LPAR, we did a staging run where we gradually increased the number of transactions that the workloads were running. We started monitoring once the workloads started at low levels until they reached stress levels and the CPU utilization for the LPAR was more than 80%. We did this in order to see how WLM and IRD would manage processor resources. When the high stress LPAR reach 90% CPU utilization, we observed that processors were taken away from the low priority LPAR and the weights of both LPARs were adjusted accordingly.

The following are some examples of the RMF™ TM Monitor III screens that show the number of processors:

```
           HARDCOPY     RMF V1R9   CPC Capacity                  Line 1 of 14
Command ===>
Samples: 32      System: J80   Date: 07/26/07  Time: 12.30.00  Range: 60    Sec
Partition:   J80        2094 Model 750
CPC Capacity:    2295   Weight % of Max: 10.0   4h Avg:   32   Group:   N/A
Image Capacity:  2295   WLM Capping %:    0.0   4h Max:  591   Limit:   N/A
Partition  --- MSU ---  Cap  Proc    Logical Util %   - Physical Util % -
             Def   Act  Def   Num    Effect   Total   LPAR  Effect  Total
*CP                         96.0                       0.4    62.9   63.3
J80            0   980  NO  48.0       44.3    44.5    0.2    42.5   42.7
J90            0   468  NO  48.0       21.2    21.2    0.0    20.4   20.4
PHYSICAL                                               0.2             0.2
```

```
*AAP                      4.0                    0.3    91.2   91.4
J80                   NO  2.0       68.9    69.0    0.1    68.9   69.0
J90                   NO  2.0       22.3    22.3    0.1    22.3   22.3
PHYSICAL                                           0.2           0.2

*IIP                      4.0                    0.2    10.4   10.6
J80                   NO  2.0       10.4    10.5    0.0    10.4   10.5
J90                   NO  2.0        0.0     0.0    0.0     0.0    0.0
PHYSICAL                                           0.2           0.2
```

CPU utilization exceeded more than 80% on system J80:

```
            HARDCOPY     RMF V1R9   CPC Capacity                 Line 1 of 14
Command ===>
Samples: 47      System: J80   Date: 07/26/07  Time: 13.31.00  Range: 60    Sec
Partition:   J80        2094 Model 750
CPC Capacity:    2295   Weight % of Max: 10.0   4h Avg:  372   Group:   N/A
Image Capacity:  2295   WLM Capping %:    0.0   4h Max: 1879   Limit:   N/A
Partition  --- MSU ---  Cap  Proc    Logical Util %   - Physical Util % -
             Def   Act  Def   Num    Effect   Total   LPAR  Effect  Total
*CP                      73.0                        0.2    96.4   96.6
J80            0  2029  NO  45.0       98.1    98.2    0.1    88.3   88.4
J90            0   187  NO  28.0       14.5    14.5    0.0     8.1    8.1
PHYSICAL                                           0.1           0.1

*AAP                      4.0                    0.1    94.4   94.5
J80                   NO  2.0       93.4    93.4    0.0    93.4   93.4
J90                   NO  2.0        1.0     1.0    0.0     1.0    1.0
PHYSICAL                                           0.1           0.1

*IIP                      4.0                    0.2    16.0   16.2
J80                   NO  2.0       16.0    16.0    0.0    16.0   16.0
J90                   NO  2.0        0.0     0.0    0.0     0.0    0.0
PHYSICAL                                           0.2           0.2
```

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 14

Using HiperDispatch

We have implemented HiperDispatch on all of the z/OS V1R9 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 IEAOPT*xx* 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.

  In the following example, you can see 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
                                                                                                    PAGE    1
            z/OS V1R9               SYSTEM ID Z2              START 04/16/2008-15.00.00  INTERVAL 000.30.00
                                    RPT VERSION V1R9 RMF      END   04/16/2008-15.30.00  CYCLE 0.100 SECONDS
-CPU  2097   MODEL  742   H/W MODEL  E56  SEQUENCE CODE 00000000000699FF    HIPERDISPATCH=YES
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
```

- **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.

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 1 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.

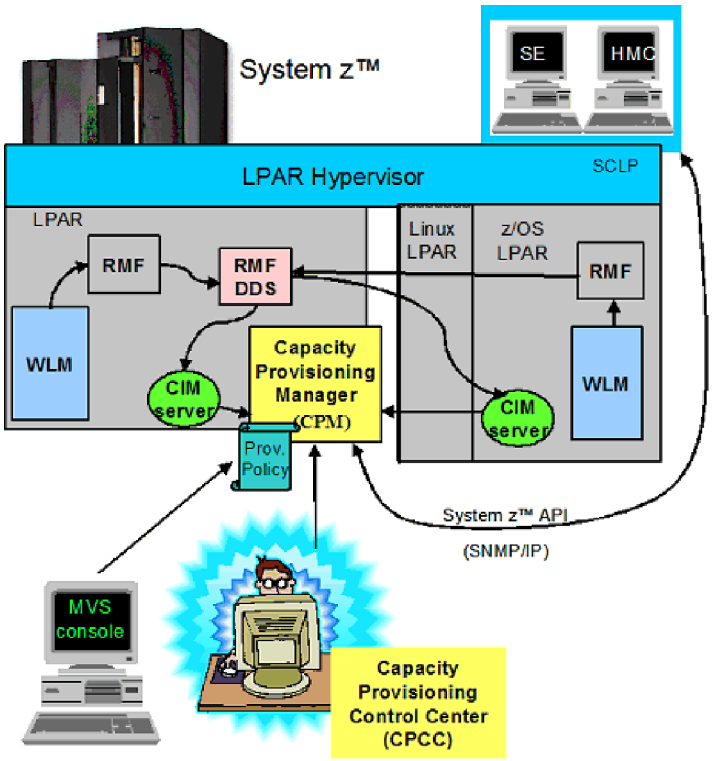


Figure 1. Capacity Provisioning components

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 OMPROUTE profiles across the sysplex to support a new application activated dynamic VIPA (DVIPA).
3. Configure the CIMSERVER 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 GPMSERVE 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             CPO4108I 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) CPO3030I Command completed successfully for CPC H91

CPM removed 1 zAAP:
 19.12.50 S0086030  BPXM023I (CPOSRV)  086
    086             CPO4109I 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) CPO3032I Command completed successfully for CPC H91

CPM added 3 zAAPs:
 23.12.50 S0086030  BPXM023I (CPOSRV)  524
    524             CPO4108I 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) CPO3030I Command completed successfully for CPC H91
 23.20.46 S0086030  BPXM023I (CPOSRV)  369
    369             CPO4108I 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) CPO3030I Command completed successfully for CPC H91
 23.28.43 S0086030  BPXM023I (CPOSRV)  903
    903             CPO4108I 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) CPO3030I 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             CPO4108I 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) CPO3030I 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             CPO4108I 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) CPO3030I 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             CPO4108I 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) CPO3030I 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             CPO4108I 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) CPO3030I Command completed successfully for CPC H91
```

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
- 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 MAN*x* data sets. In addition, since system logger manages data flow and available storage, there is no concern over buffer overrun due to MAN*x* 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 IFASMFDL dump job. Simply specify the appropriate log stream name or names along with the dates and times in which you are interested. The IFASMFDL 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 eligile 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
- SMFPRM*xx* 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 19, 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/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 is 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

We had two goals in mind when we were setting up our offload data sets:

1. We wanted to avoid frequent data set switches during offload processing.
2. We wanted to make the size small enough so that SMS did not have trouble allocating it when we needed an offload data set.

We decided to make our offload data sets 1000 cylinders each to begin with. You can use the LS_SIZE parameter to tell system logger what size offload data sets to use for a log stream, in 4K byte blocks. To convert from cylinders to 4K byte blocks, one cylinder equals 180 blocks; so, we defined our LS_SIZE as 180000.

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.

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 28) 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).

SMFPRM*xx* member definition

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

```
RECORDING(LOGSTREAM),
DEFAULTLSNAME(IFASMF.SMFDFLT.&SYSNAME),
LSNAME(IFASMF.SMF0TO29.&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 DSNAME statement for our MAN*x* data sets in the parmlib member. By doing this, the MAN*x* data sets are available if we ever need to dynamically switch back to SMF data set recording.

```
DSNAME(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 MAN*x* 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 28 for an example of the IFASMFDL 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 IFASMFDL 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('Days')
jdayyest = jday - 1
parse value date(standard) with jyear 5
jdate = jyear || jdayyest

Queue "        LSNAME(IFASMF.SMF70T79."||SysID||",OPTIONS(DUMP))"
Queue "        LSNAME(IFASMF.SMF0TO29."||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=&&SMFCTL,
//          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=IFASMFDL,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 25.
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.SMF0TO29.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.SMF0TO29.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=OZ.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.SMF0TO29.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 27, 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: XXXXXXX.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. 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

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 2 on page 37, 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 2 on page 37 provides a before-and-after illustration of both the initial Sysplex Timer topology and the planned STP timing topology.

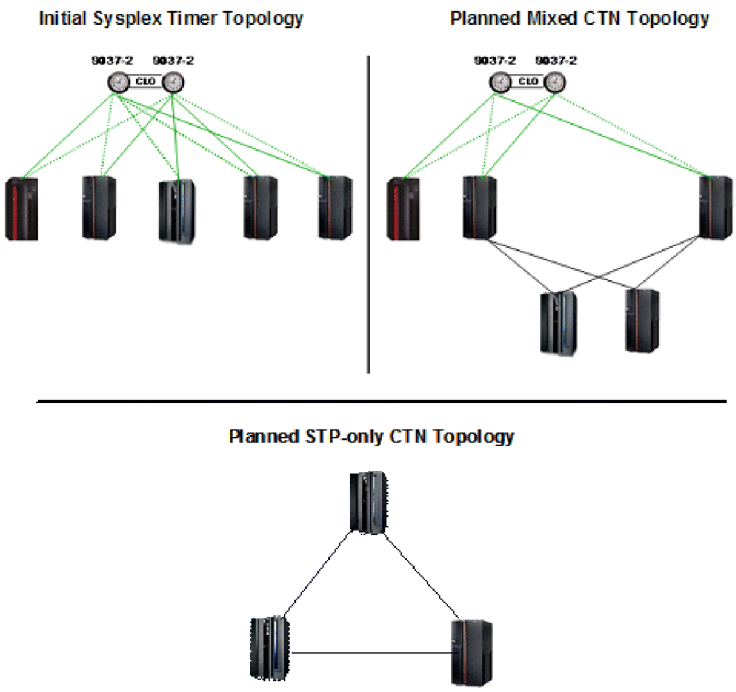


Figure 2. zPET initial Sysplex Timer topology, planned mixed CTN topology, and planned STP-only CTN topology

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	z9 EC, z9 BC, z990, and z890	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 *enabled* 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	From the upgraded HMC, select the Sysplex Timer task for the server where the STP facility was enabled in step 6. Look for new panels. See Figure 4 on page 42 for an example.
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.	This support is needed in order to achieve stated migration objective. Schedule this maintenance installation to coincide with the weekly service window.
10	Install z/OS STP enablement SPE.	The z/OS STP enablement support was delivered as a ++APAR at the time of this writing.	The z/OS V1R8 version of the enablement APAR is the only one that is needed. 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.
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(CLOCK*xx*).	Optional	Leave existing CLOCK*xx* member as is and allowed all the new parameters to use their default values.
13	IPL z/OS.	An IPL is requiredafter the z/OS STP enablement APAR is installed.	1. Schedule this step to coincide with the weekly service window IPLs. 2. 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	This step can be performed anytime prior to performing the IODF activation in step 16. Run two peer links, one ICB and one ISC-3, between K25 and G74.

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. • 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 3 on page 41.

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.

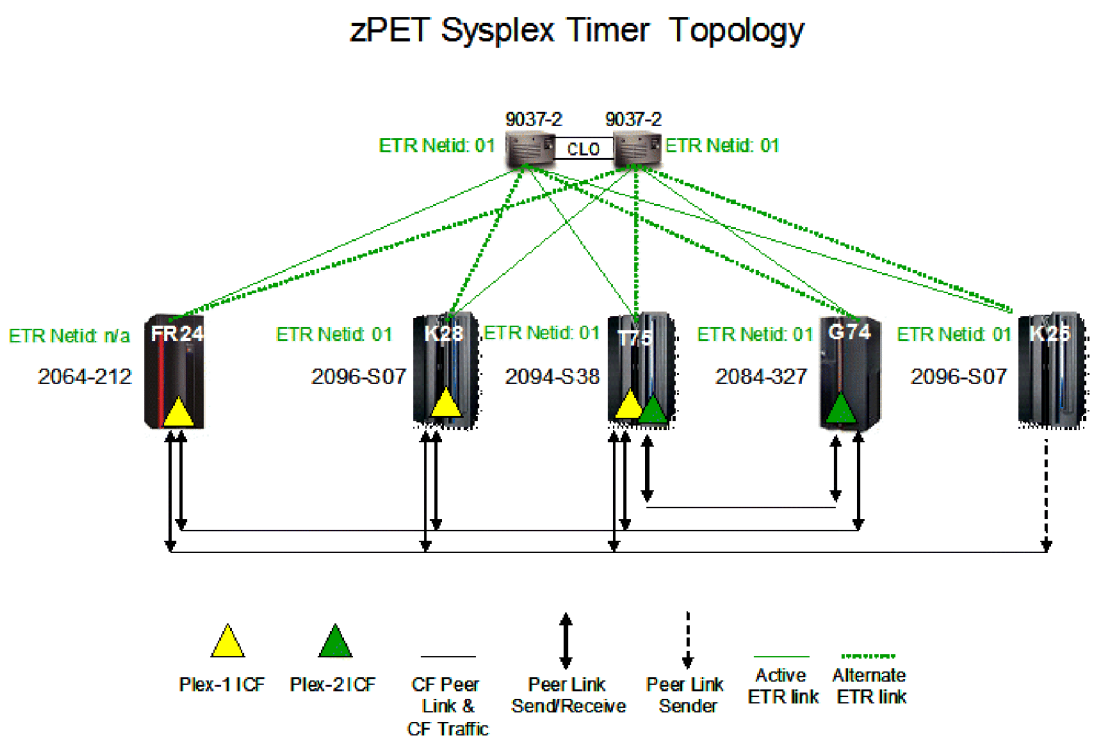


Figure 3. zPET Sysplex Timer topology

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 4 on page 42. 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.

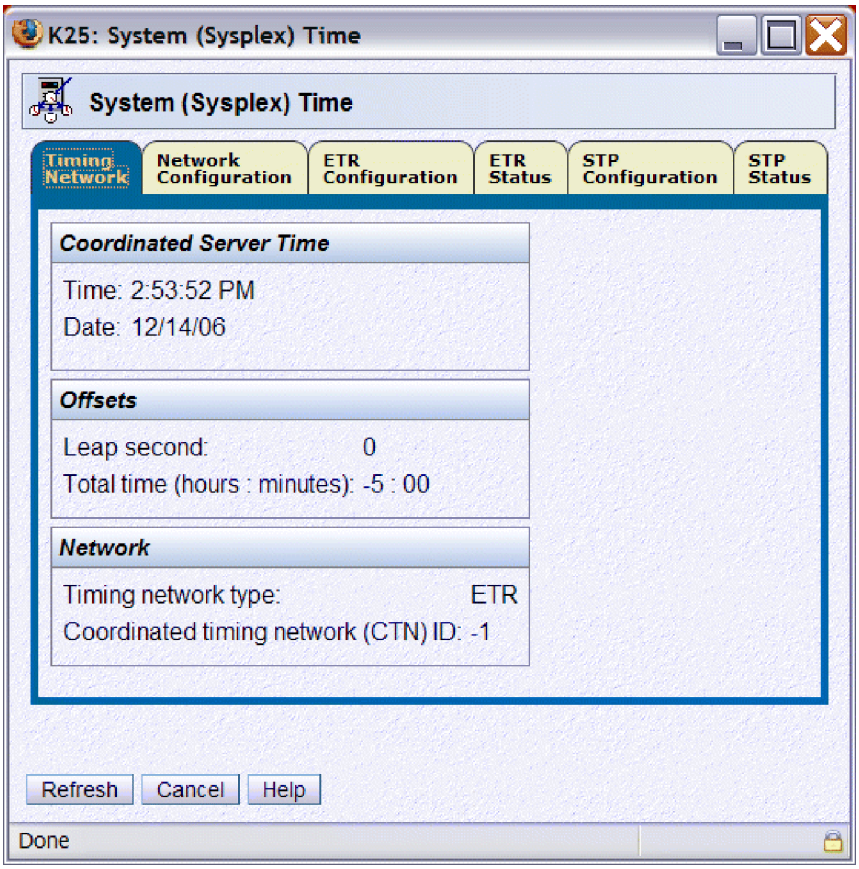


Figure 4. System (Sysplex) Time panels, viewed from the Support Element (SE)

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 34.

Adding STP timing-only links

As mentioned in "Recovery considerations" on page 38, 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 5 on page 45.

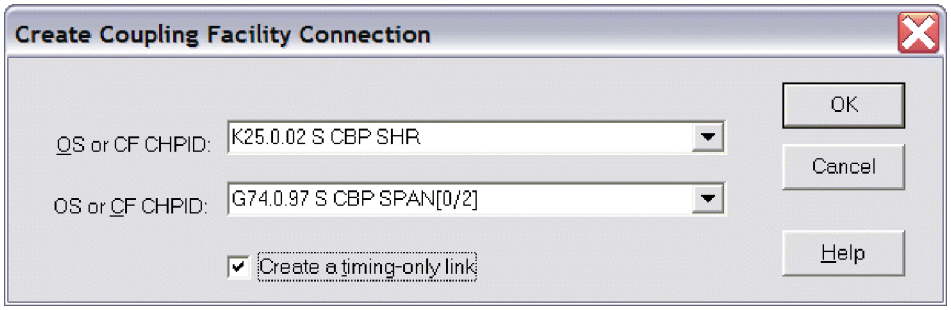


Figure 5. HCM Create Coupling Facility Link Connection dialog for defining a STP timing-only link

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
    DEVICE      SUBCHANNEL      STATUS
     FEBB          3280        OPERATIONAL
     FEBC          3281        OPERATIONAL

:
```

Figure 6 on page 46 shows our existing Sysplex Timer topology after being updated with the STP timing-only links.

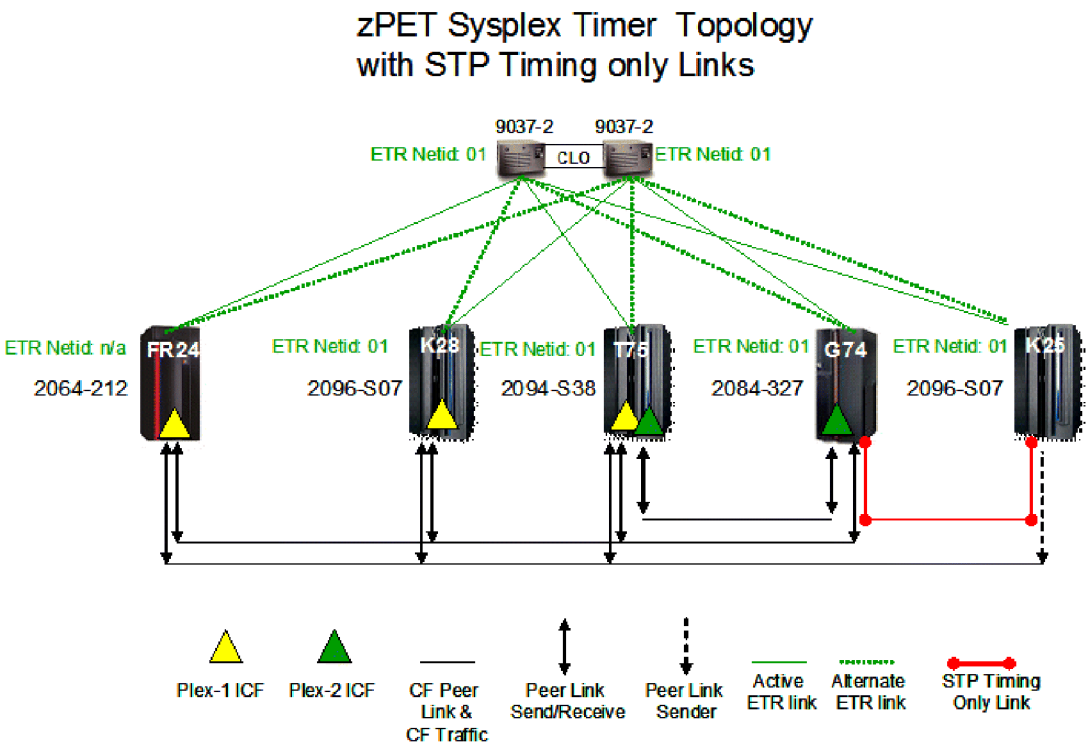


Figure 6. zPET Sysplex Timer topology with STP timing-only links

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 7 on page 47 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.

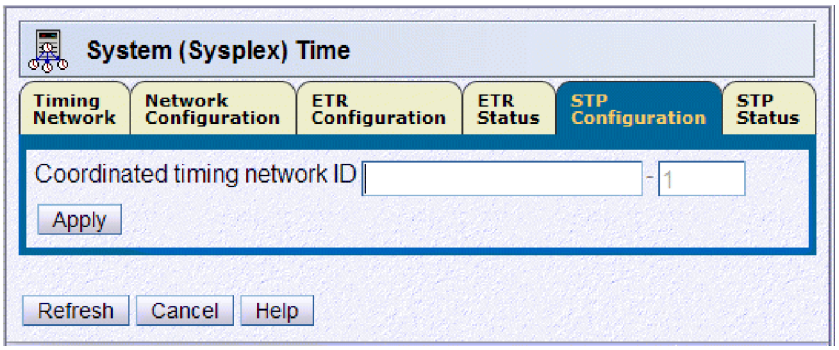


Figure 7. System (Sysplex) Time: STP Configuration panel

Once we entered the STP ID field, we clicked **Apply** to configure a CTN ID of `PETCTN-01` on the server. Figure 8, Figure 9, and Figure 10 on page 48 show the sequence of STP configuration panels involved with configuring the CTN ID on one STP-enabled server.

Figure 8 shows where the STP ID portion of the CTN ID was entered.

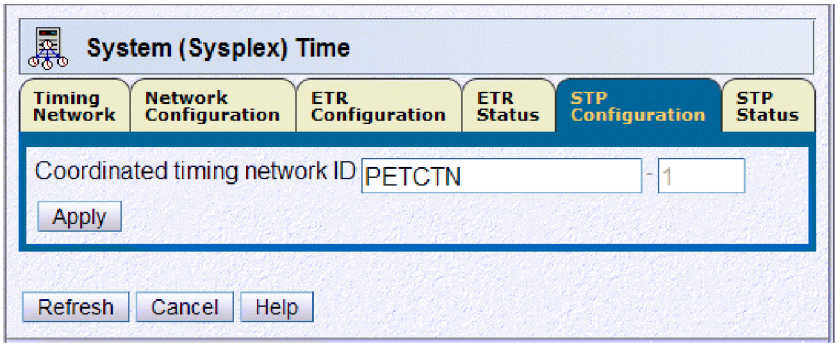


Figure 8. STP Configuration panel with STP ID value entered

Figure 9 shows the confirmation panel that appears when a CTN ID change is detected.

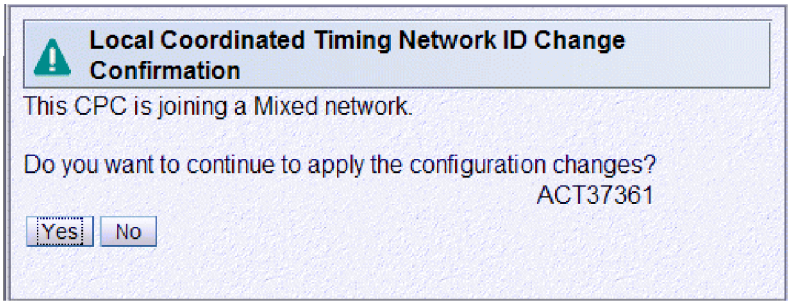


Figure 9. STP configuration confirmation panel

Figure 10 on page 48 shows the panel that acknowledges that the CTN ID was successfully changed.

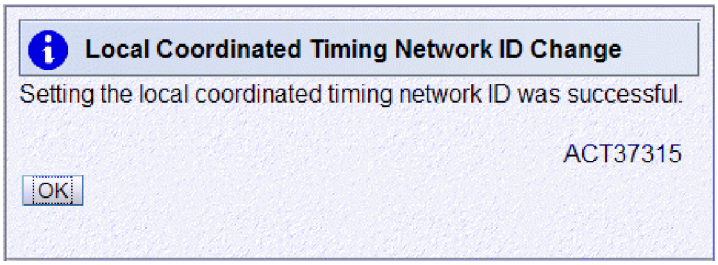


Figure 10. STP CTN ID change completion panel

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 CLOCK*xx* 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 11 on page 49 shows STP Status tab of the System (Sysplex) Time task, which displays the timing configuration from the server's perspective.

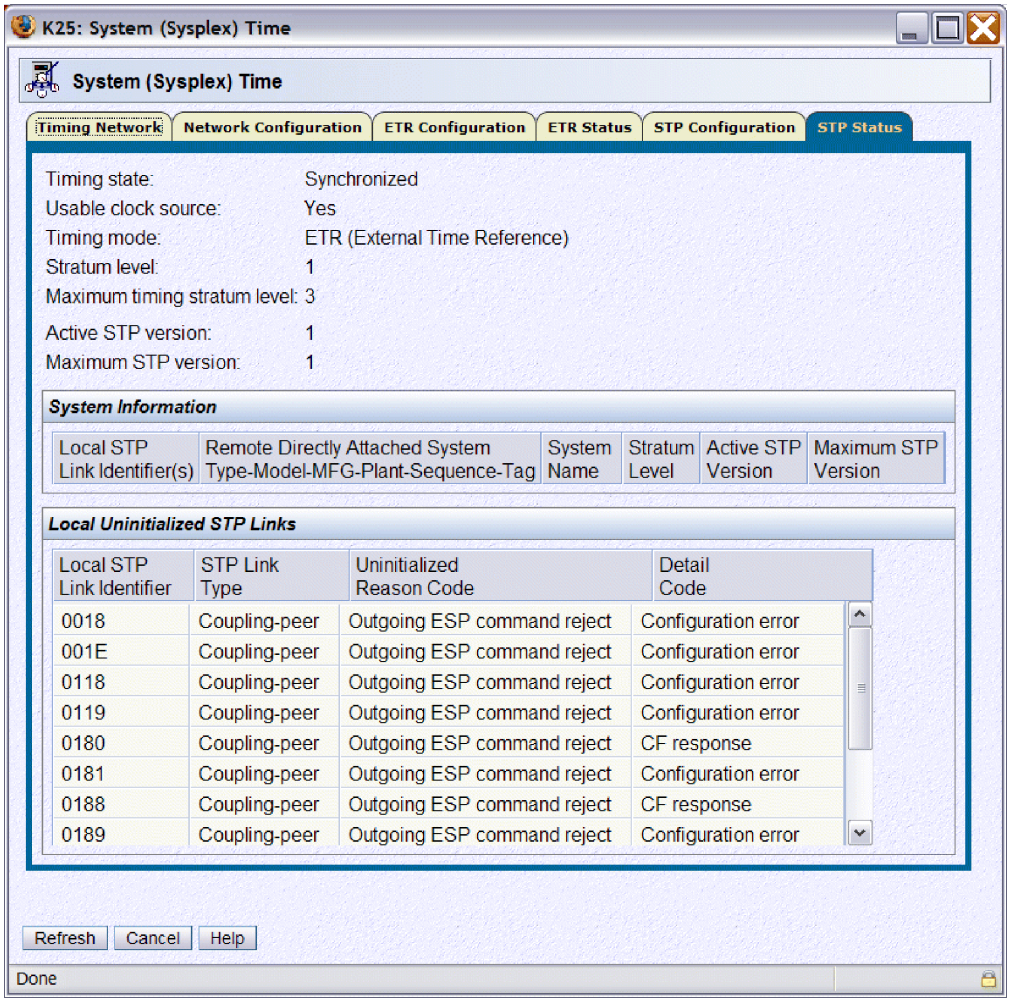


Figure 11. System (Sysplex) Time: STP Status panel, viewed from the SE on K25

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 11 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 12 on page 50 contains the results of this configuration action.

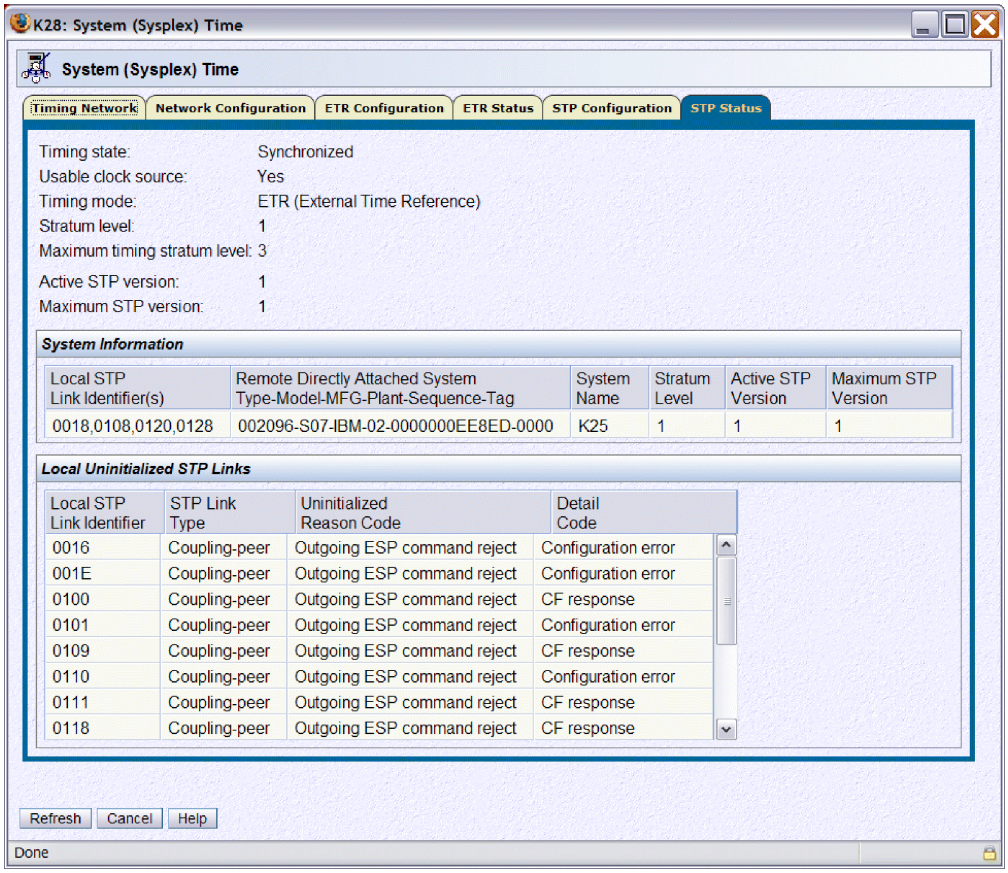


Figure 12. System (Sysplex) Time: STP Status panel on K28, after configuring the CTN ID on K28

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 13 on page 51 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.

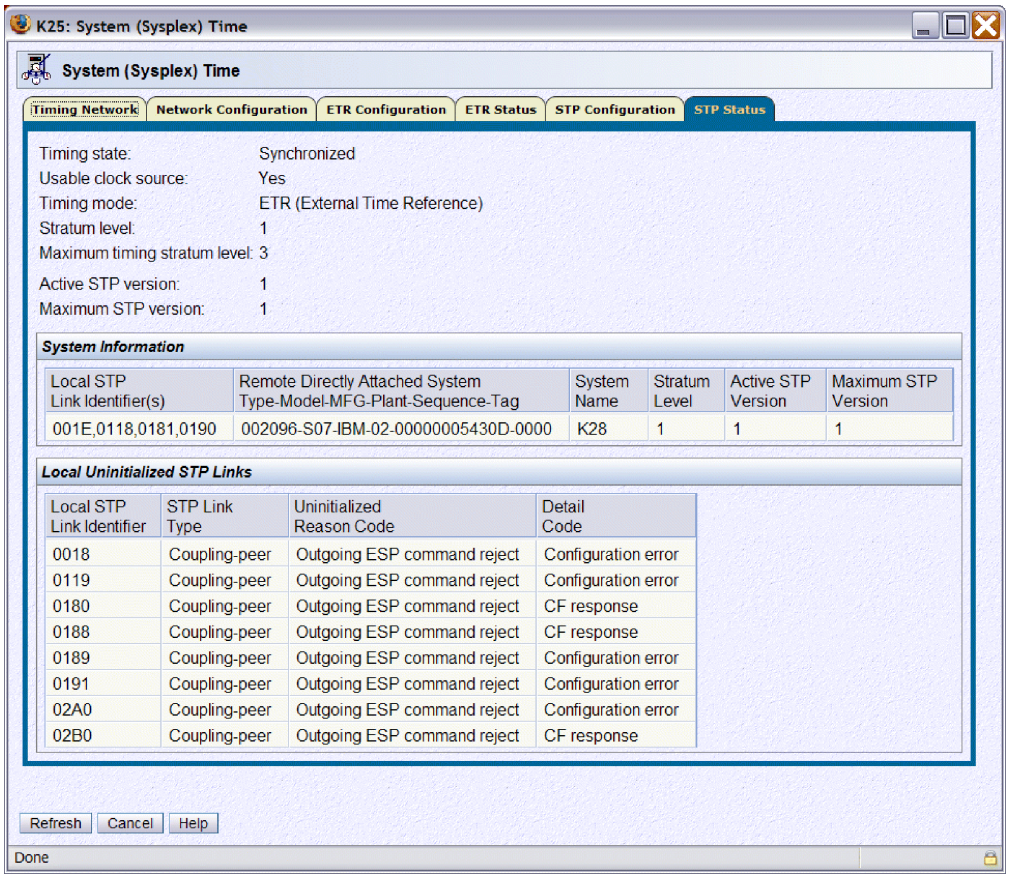


Figure 13. System (Sysplex) Time: STP Status panel on K25

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 13 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 14 through Figure 17 on page 55 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.

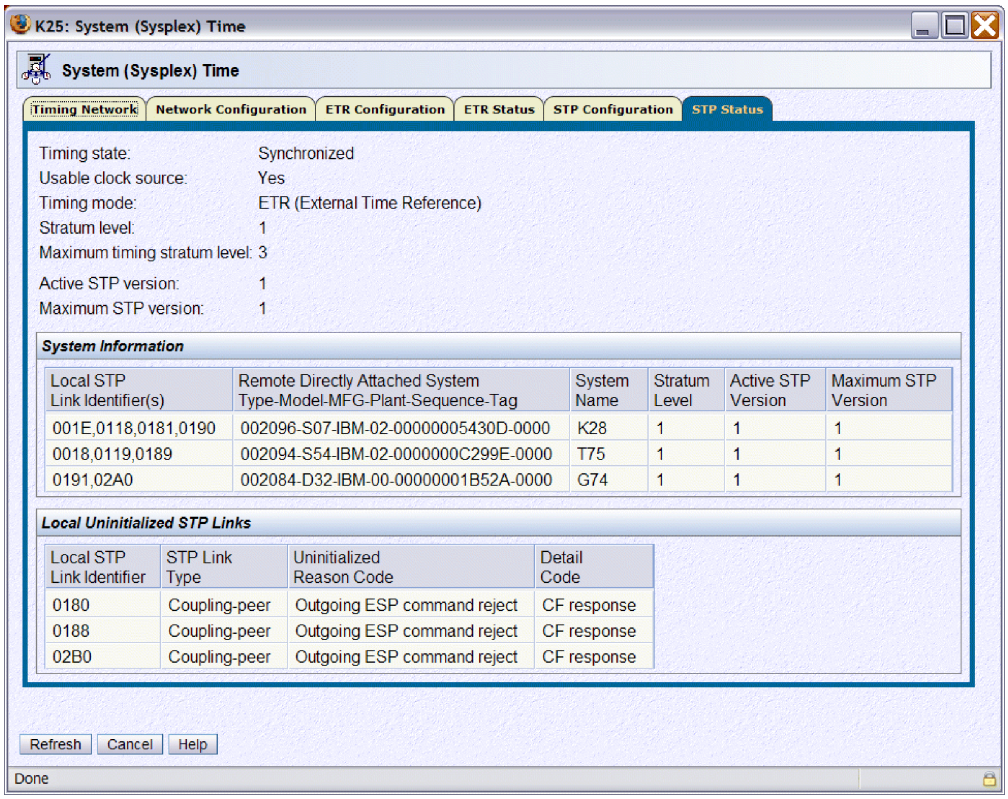


Figure 14. System (Sysplex) Time: STP Status panel on K25, showing connectivity to the other three servers

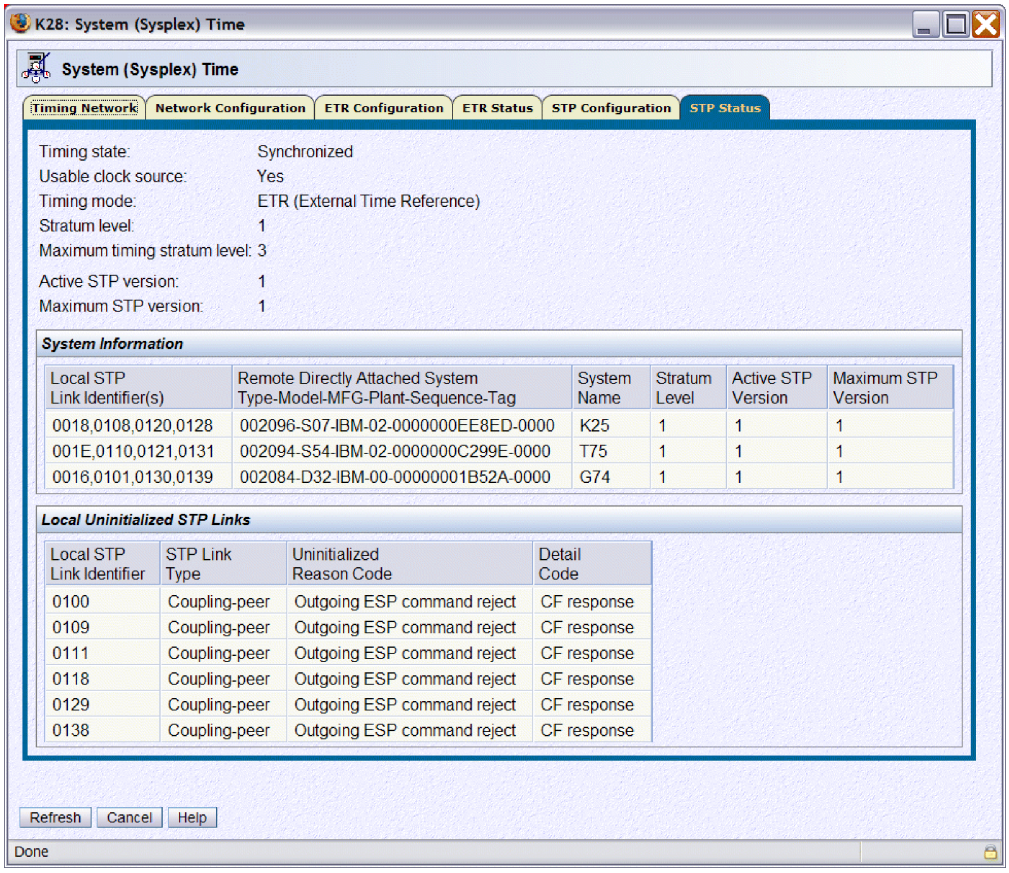


Figure 15. System (Sysplex) Time: STP Status panel on K28, showing connectivity to the other three servers

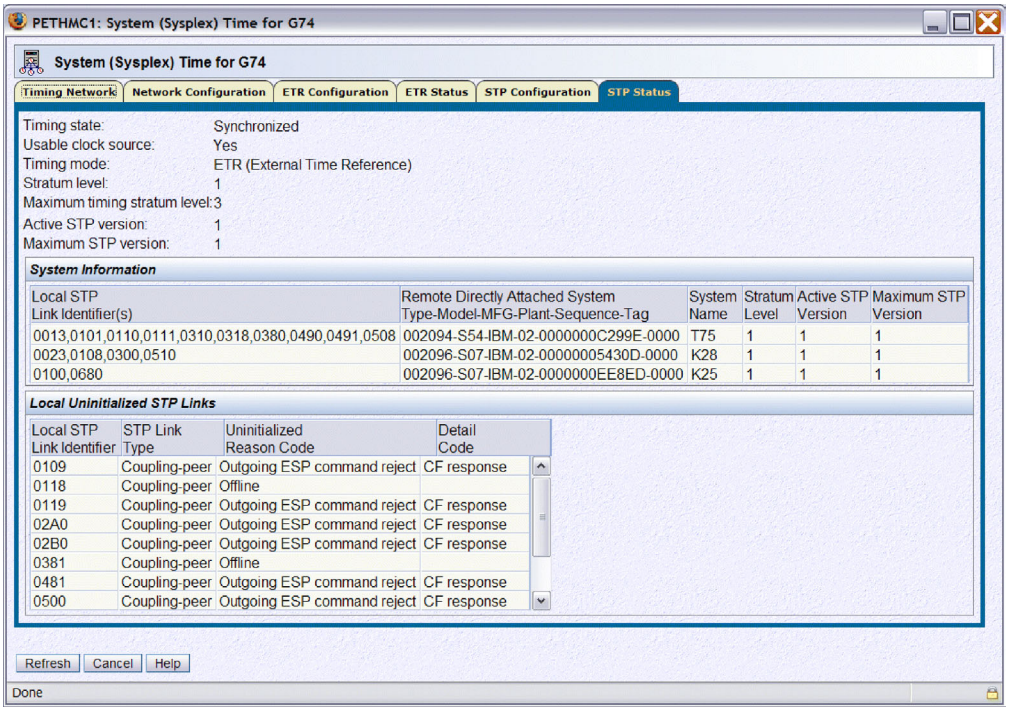


Figure 16. System (Sysplex) Time: STP Status panel on G74, showing connectivity to the other three servers

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 **Unintialized 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 16 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.

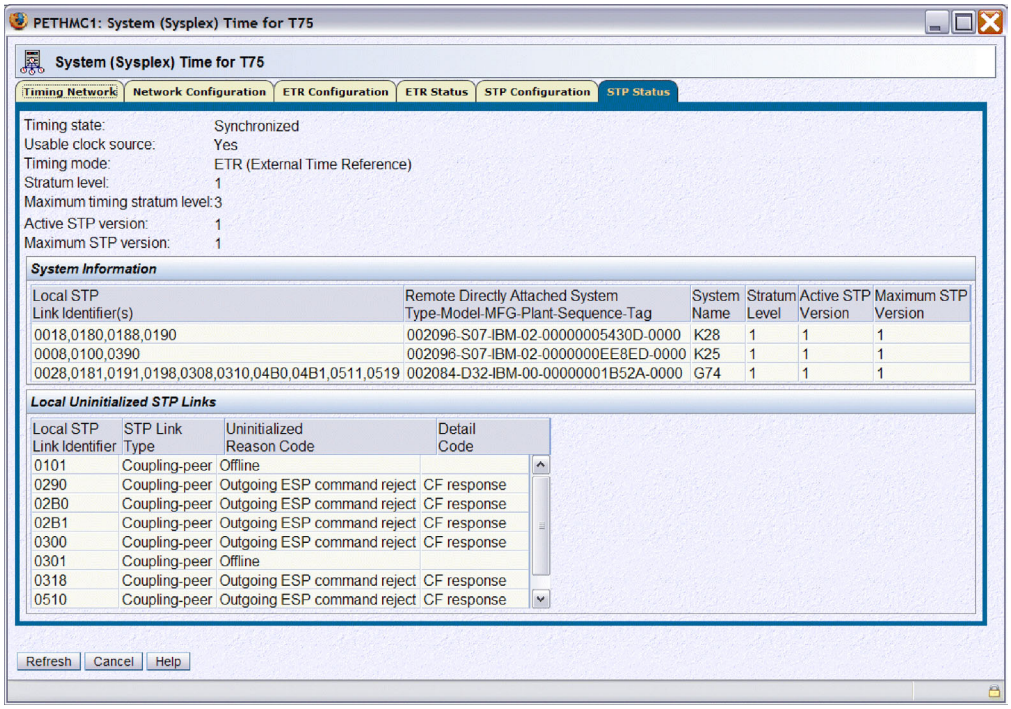


Figure 17. System (Sysplex) Time: STP Status panel on T75 showing connectivity to the other three servers

Figure 18 on page 56 illustrates the timing topology within our data center up to this point in the migration effort.

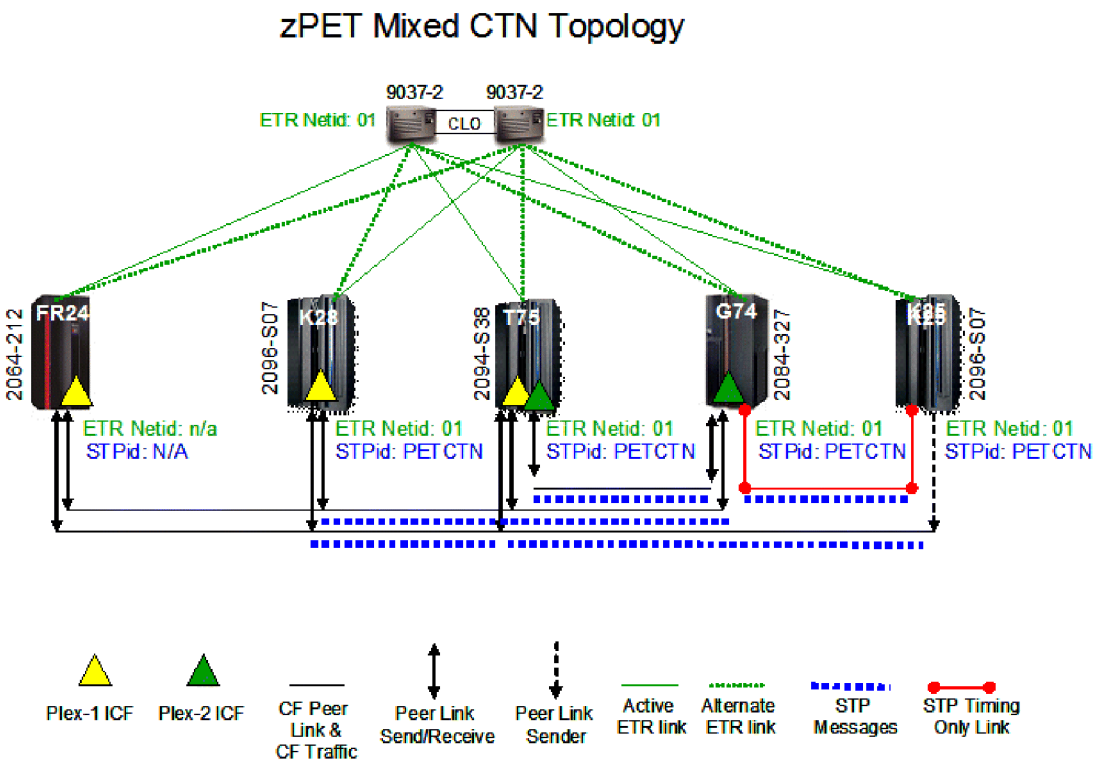


Figure 18. zPET mixed CTN topology

Note the following points about Figure 18:

- 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 19 shows the initial panel that appears when the System (Sysplex) Time task is selected on T75.

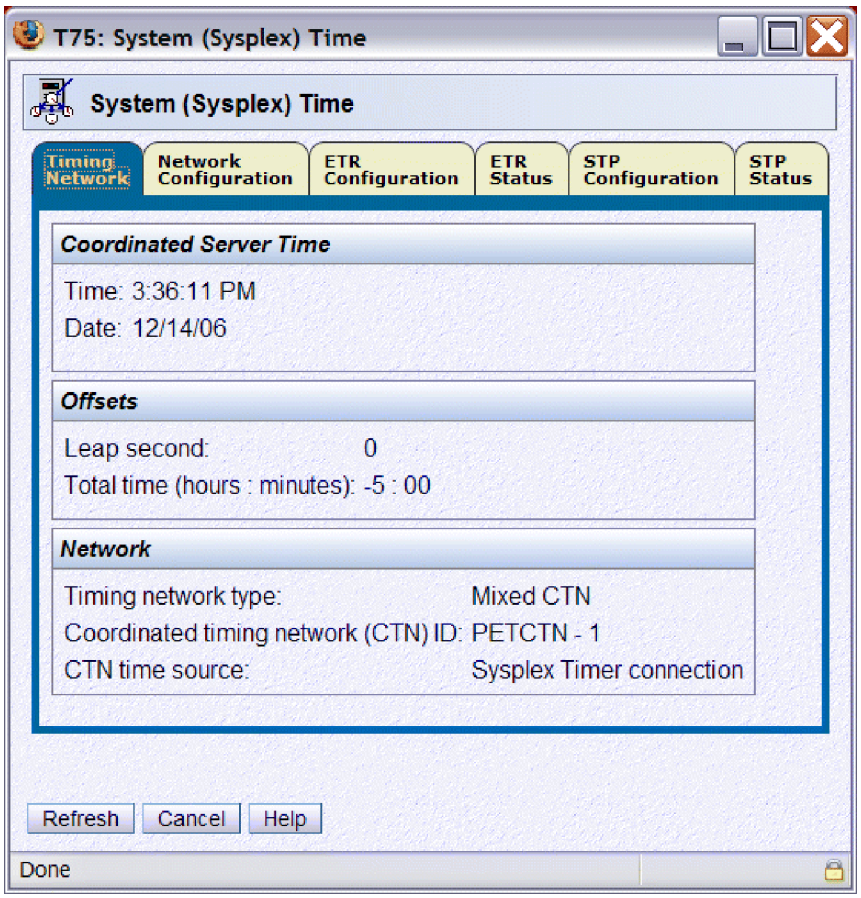


Figure 19. System (Sysplex) Time: Timing Network panel before moving T75 to stratum 2

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 20 on page 58.

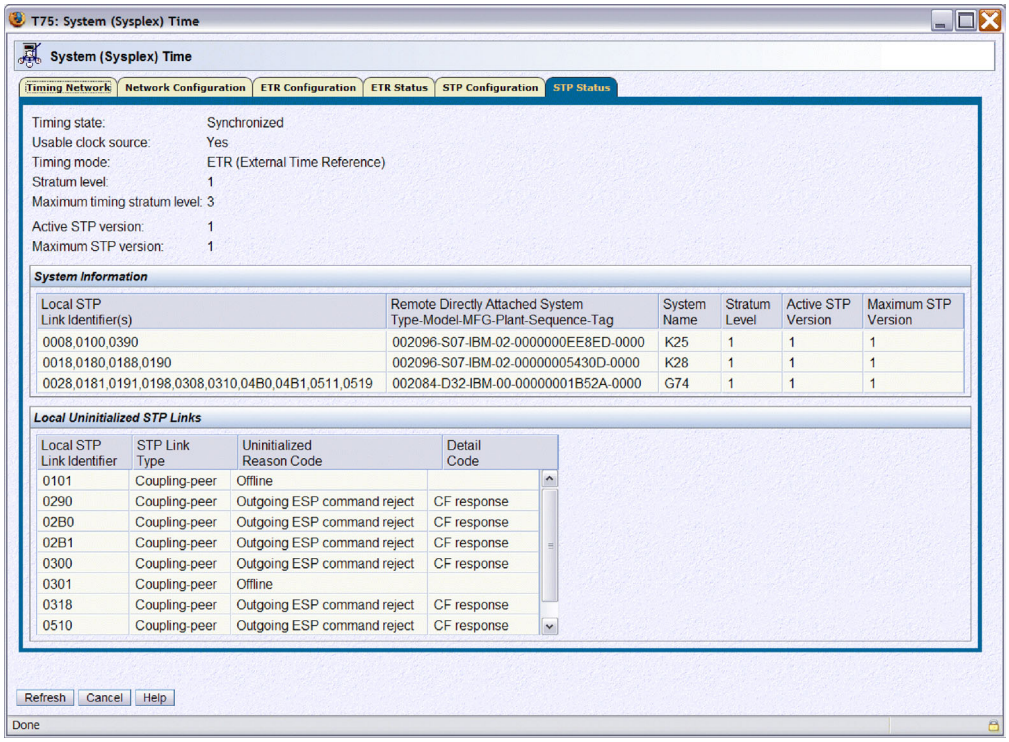


Figure 20. System (Sysplex) Time: STP Status panel before moving T75 to stratum 2

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 21 on page 59 shows how we selected the **Disabled** buttons for both ETR ports.

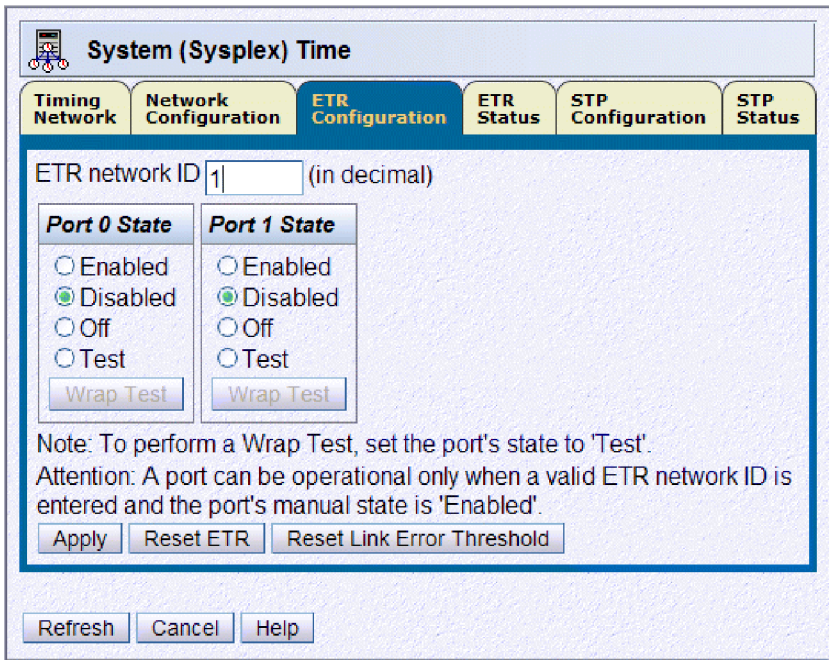


Figure 21. System (Sysplex) Time: ETR Configuration panel with ETR ports disabled

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 22.

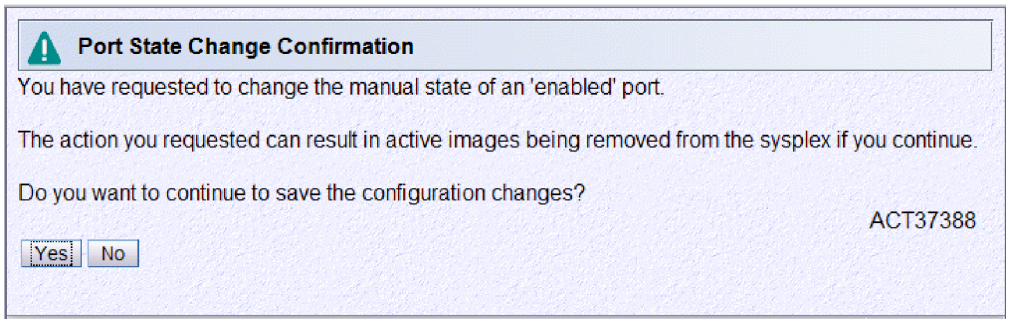


Figure 22. System (Sysplex) Time: ETR Port State Change Confirmation panel

Because we were comfortable with the configuration change, we proceeded by clicking the **Yes** button. Figure 23 on page 60 shows the results of this ETR configuration change.

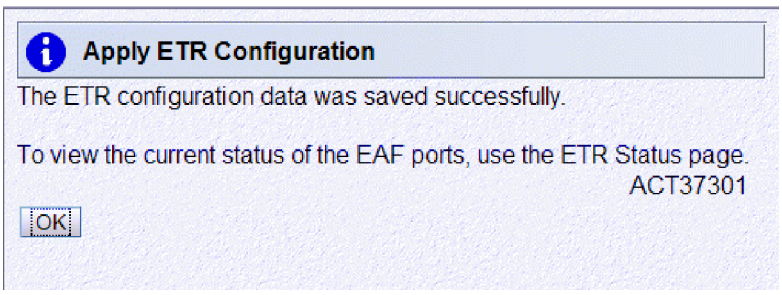


Figure 23. System (Sysplex) Time: Apply ETR Configuration panel, indicating a successful configuration change

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 our 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 24 on page 61.

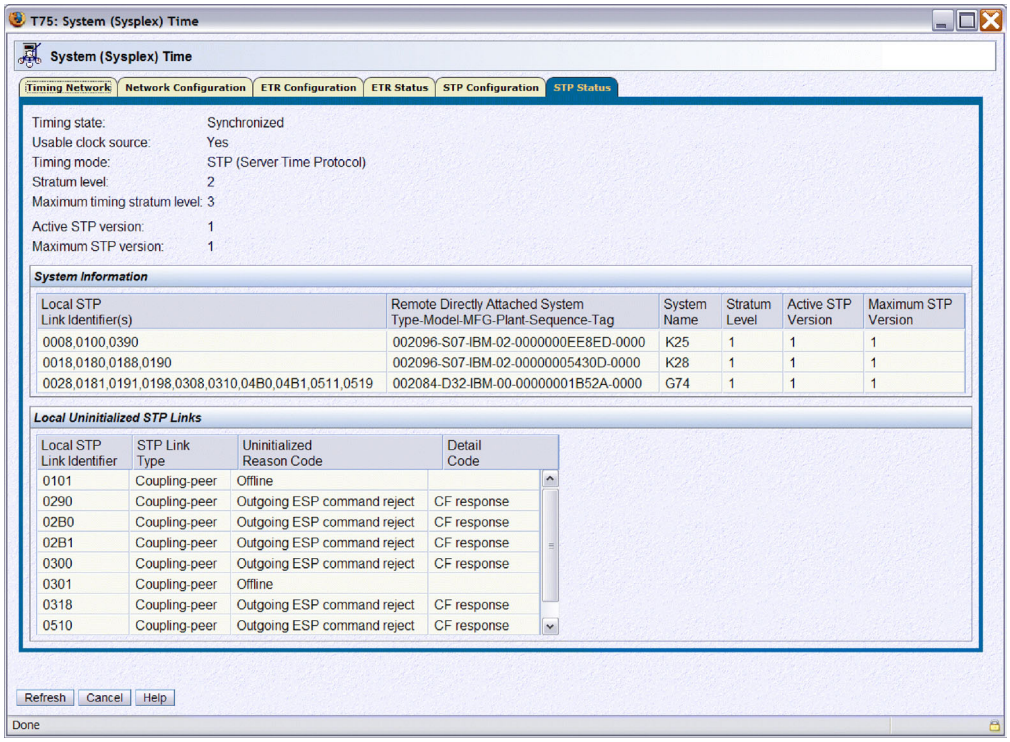


Figure 24. System (Sysplex) Time: STP Status panel with T75 at stratum 2

Note the following about Figure 24:

- 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 25 on page 62.

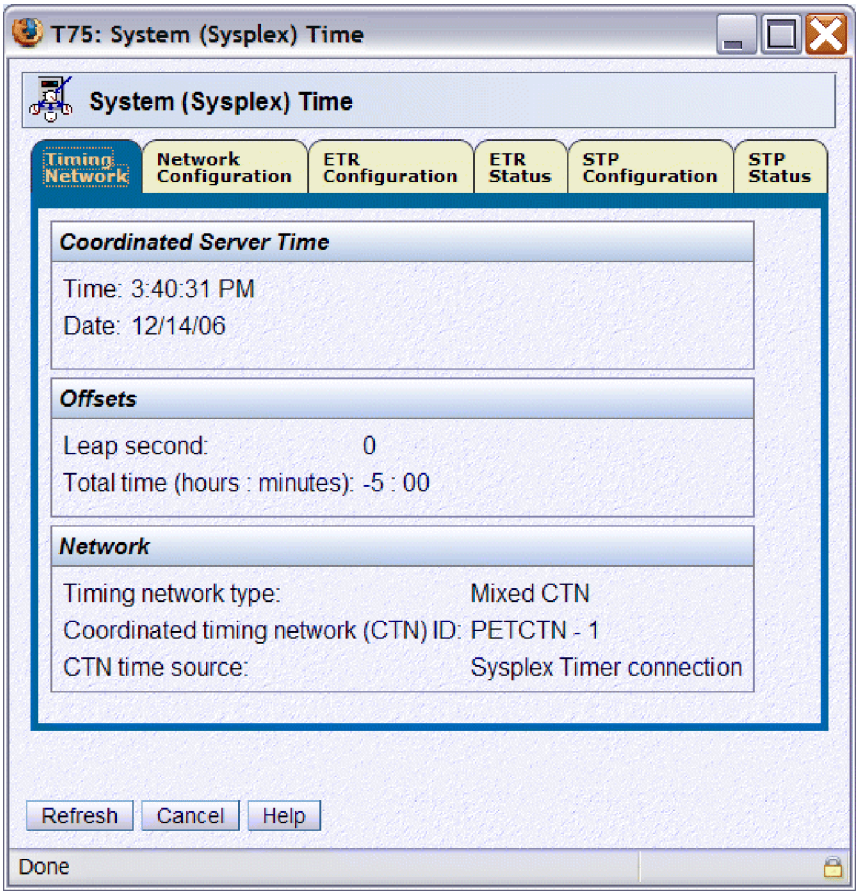


Figure 25. System (Sysplex) Time: Timing Network panel with T75 at stratum 2

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

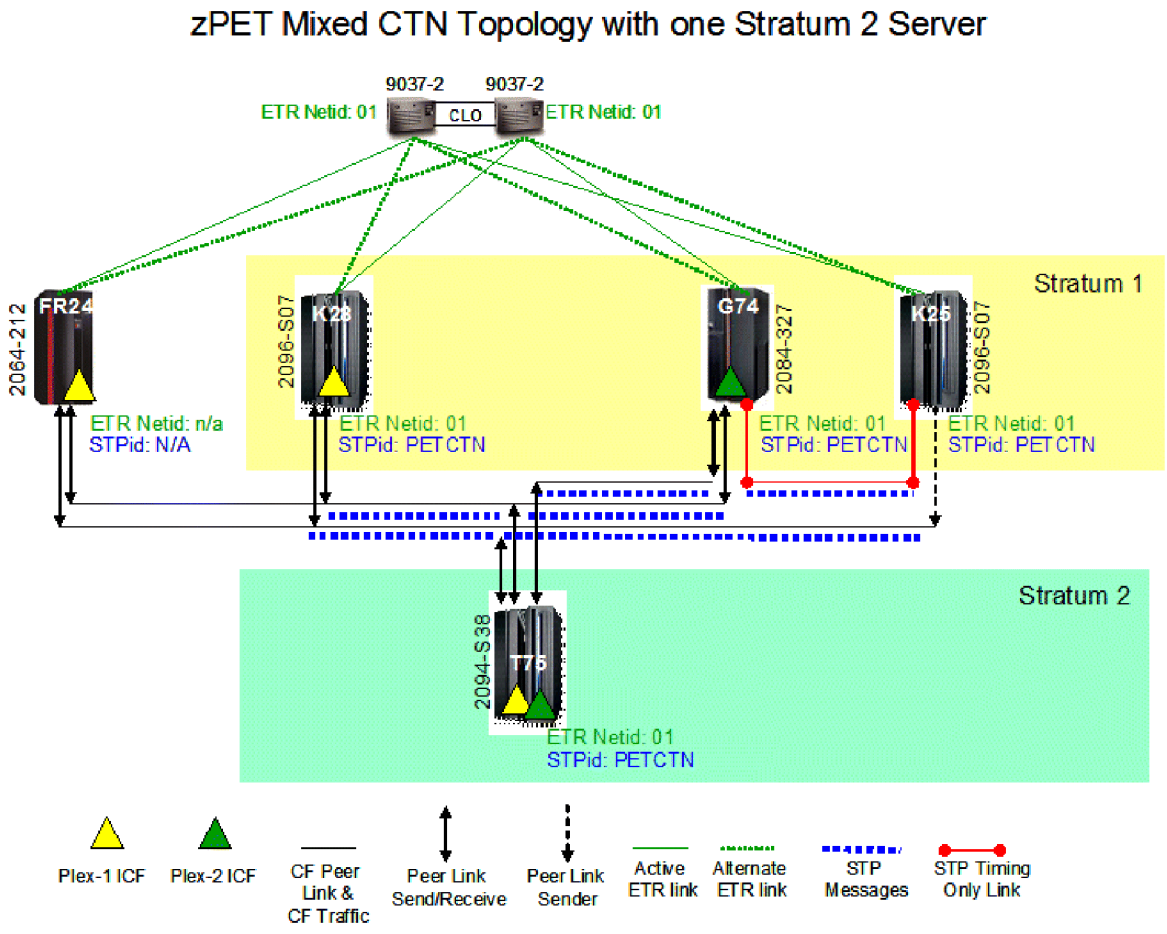


Figure 26. zPET mixed CTN with T75 at stratum 2

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 56) 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 27, 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

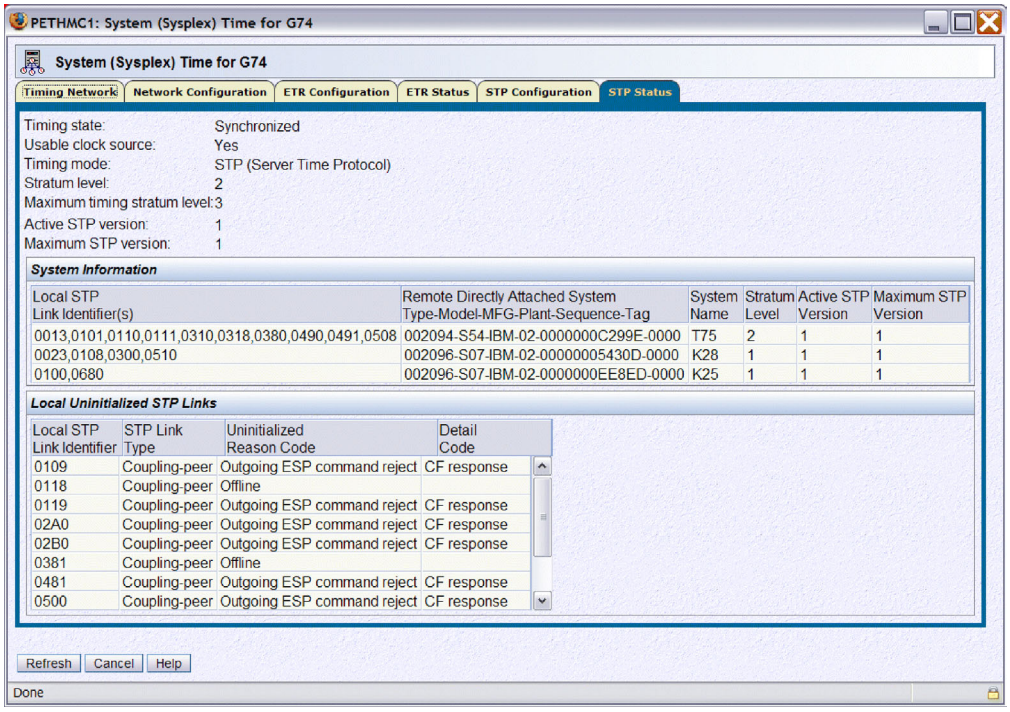


Figure 27. System (Sysplex) Time: STP Status panel for G74 with G74 and T75 at stratum 2

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 28 on page 66 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.

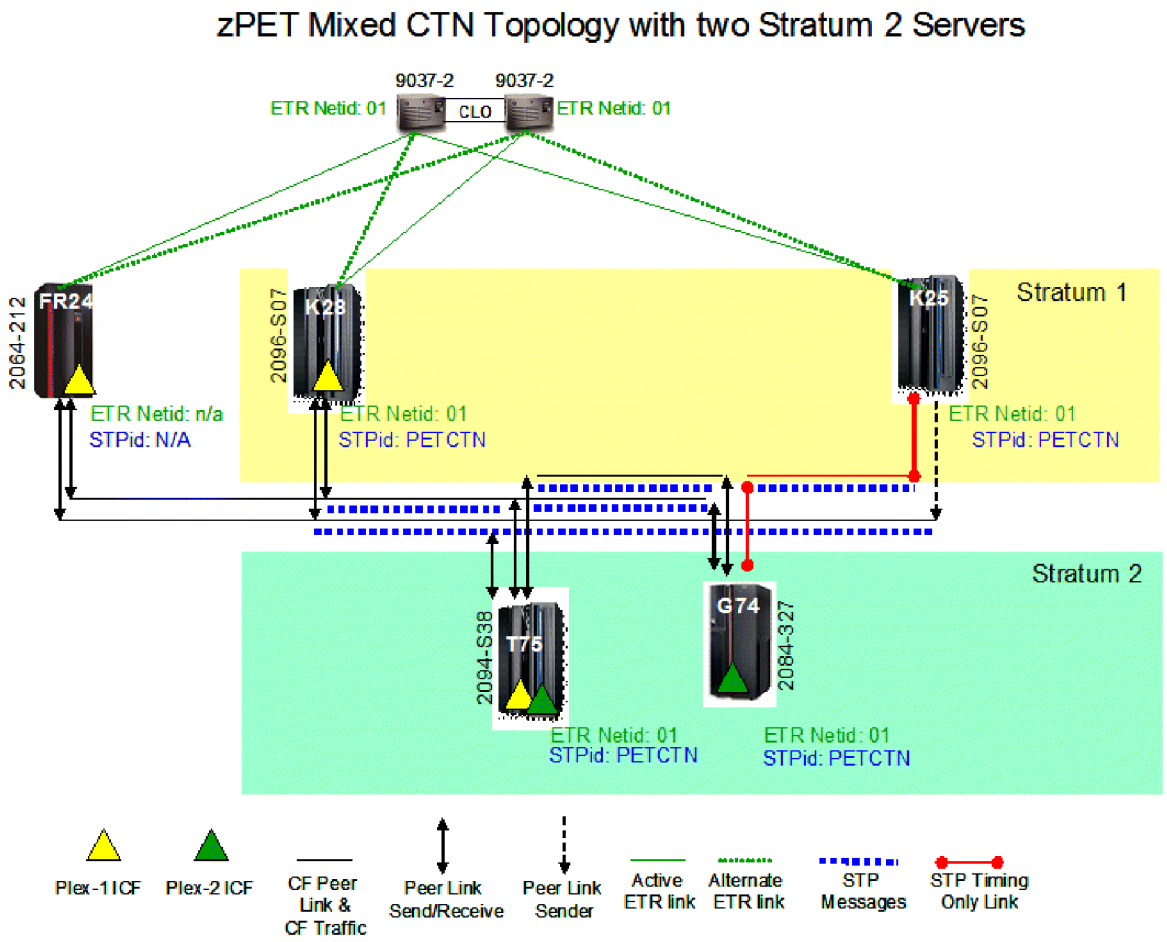


Figure 28. zPET mixed CTN with two stratum 2 nodes: T75 and G74

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 28 to the original topology shown in Figure 3 on page 41.

The first step is to move all stratum 2 servers to the stratum 1 level by enabling their respective ETR ports. Figure 29 on page 67, Figure 30 on page 67, and Figure 31 on page 68 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 29 on page 67. 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.

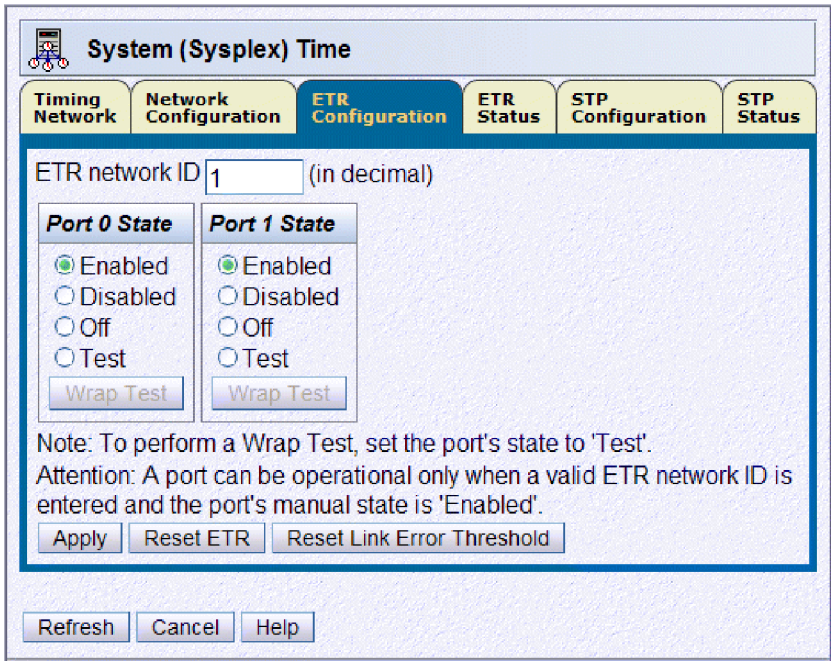


Figure 29. System (Sysplex) Time: ETR Configuration panel for port enablement

A confirmation panel appears to indicate that the ETR ports were successfully enabled, as shown in Figure 30.

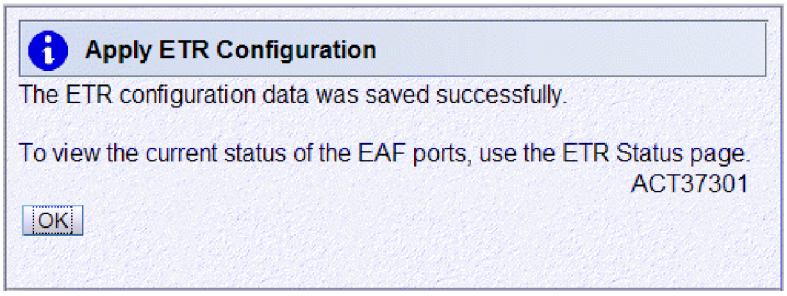


Figure 30. ETR Configuration confirmation panel

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

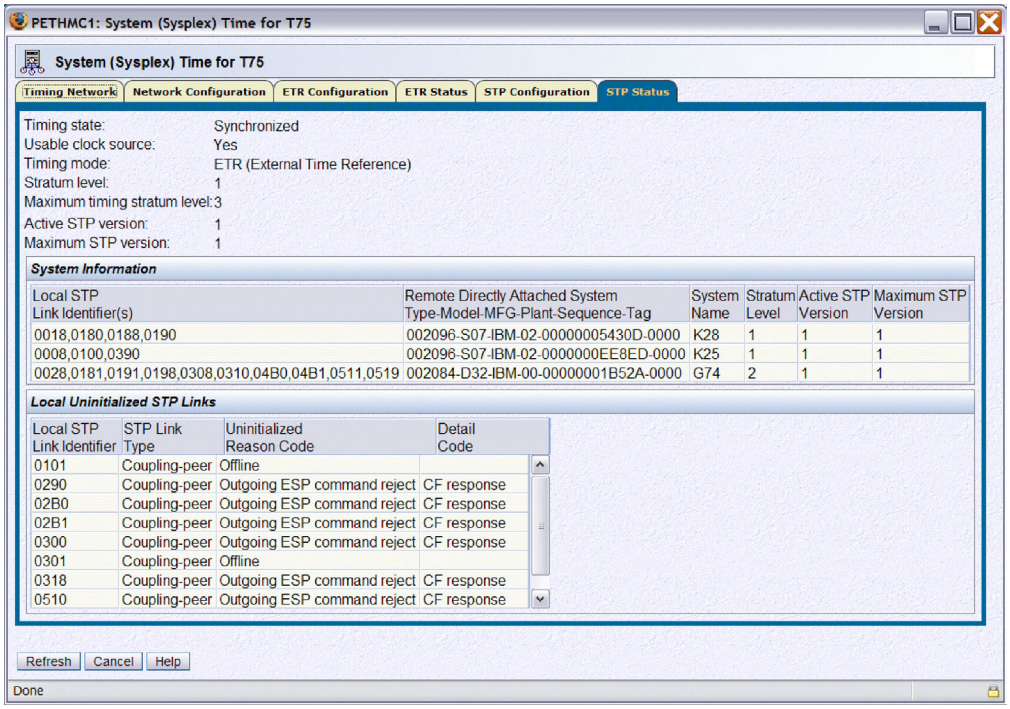


Figure 31. System (Sysplex) Time: STP Status panel, showing T75 back at stratum 1

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 32 through Figure 34 on page 70 show the sequence of panels associated with removing the STP ID from the CTN ID. Specifically, Figure 32 shows the removal of the STP ID.

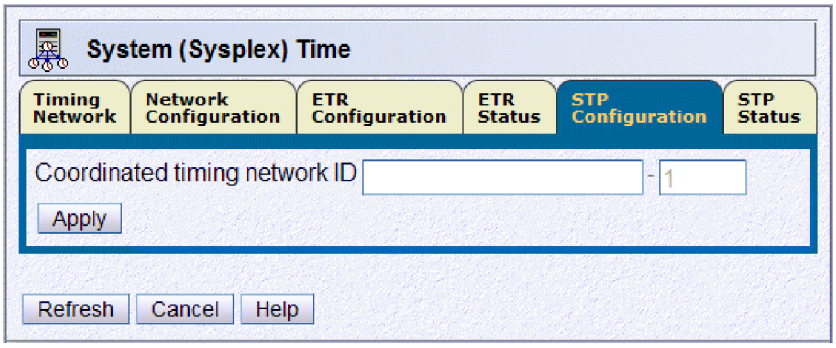


Figure 32. STP Configuration: STP ID removal

After clicking the **Apply** button on the STP Configuration panel, a confirmation panel appeared, as shown in Figure 33 on page 70.

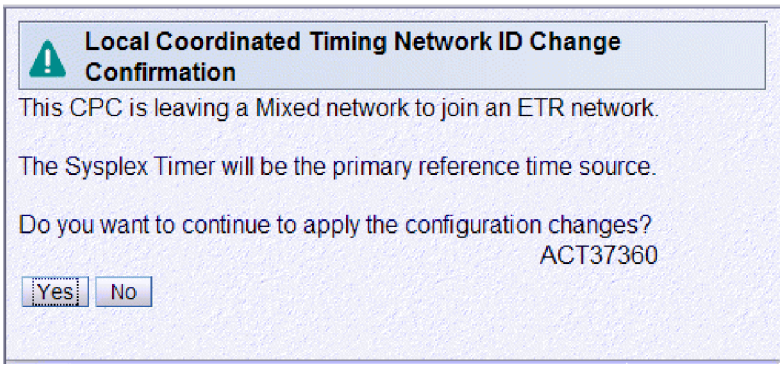


Figure 33. STP Configuration: CTN Network ID Change Confirmation panel

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 34.

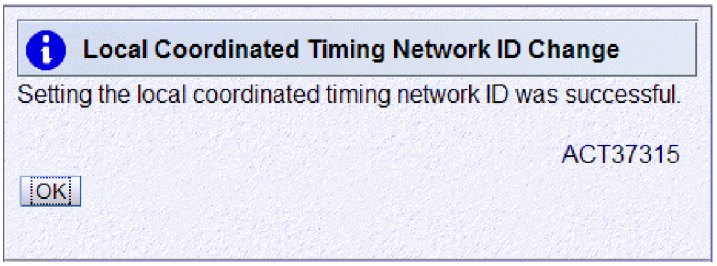


Figure 34. STP Configuration: CTN Network ID Change completion

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 35 on page 71, we were able to confirm that G74 no longer had any servers listed in the System Information section.

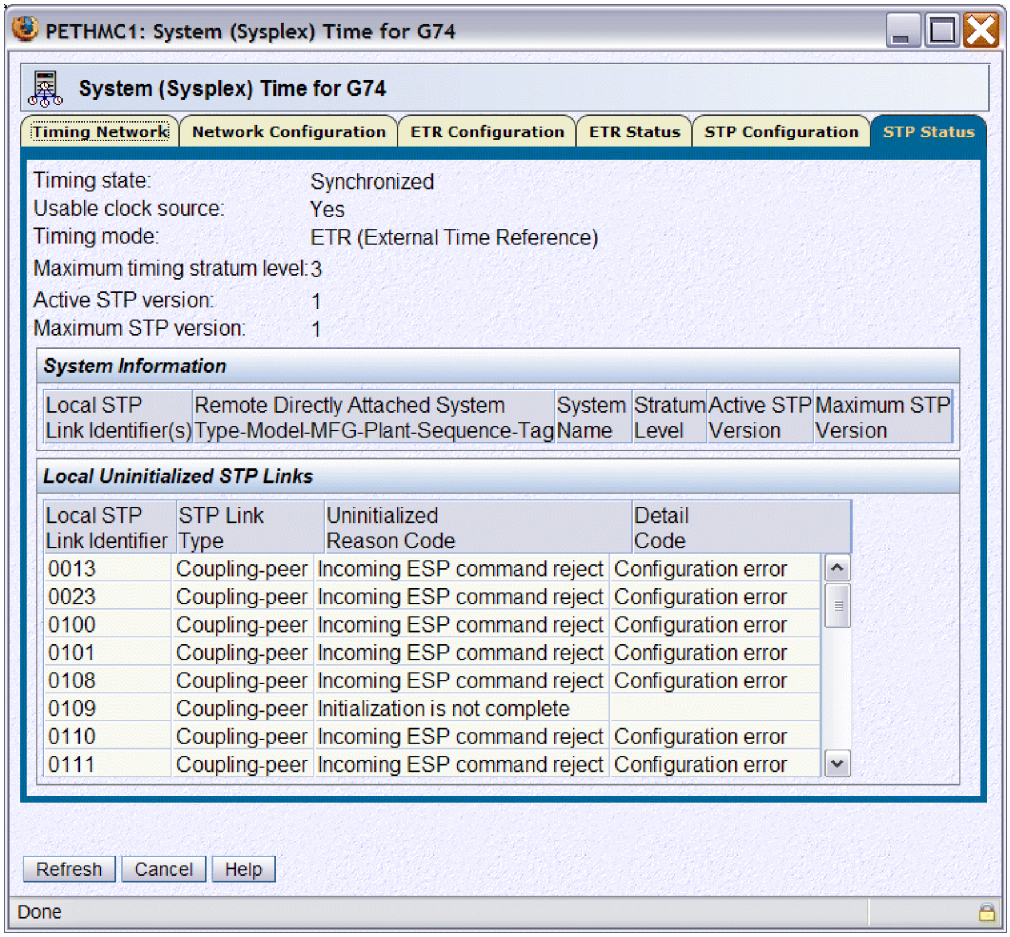


Figure 35. System (Sysplex) Time: STP Status panel, showing that G74 had returned to an ETR timing network

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 3 on page 41.

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 18 on page 56.

As we discussed in the planning considerations in “Considerations for migrating from a mixed CTN to an STP-only CTN” on page 37, 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 269, 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 36.

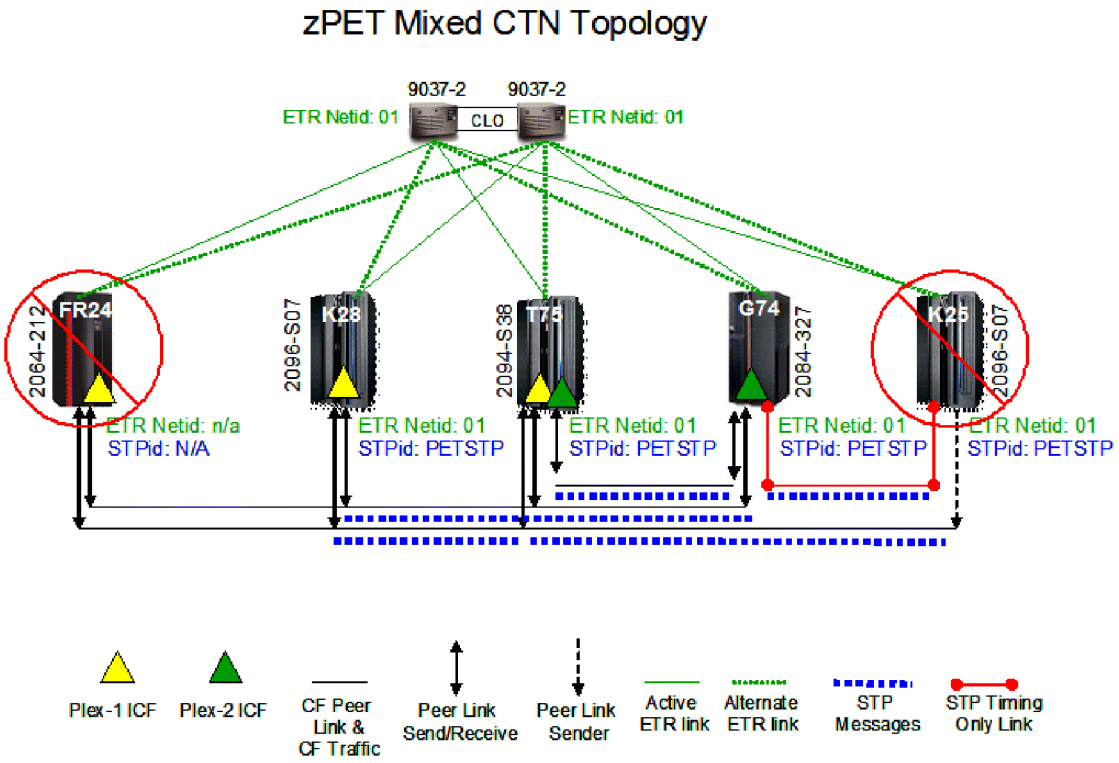


Figure 36. zPET mixed CTN with FR24 and K25 removed

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

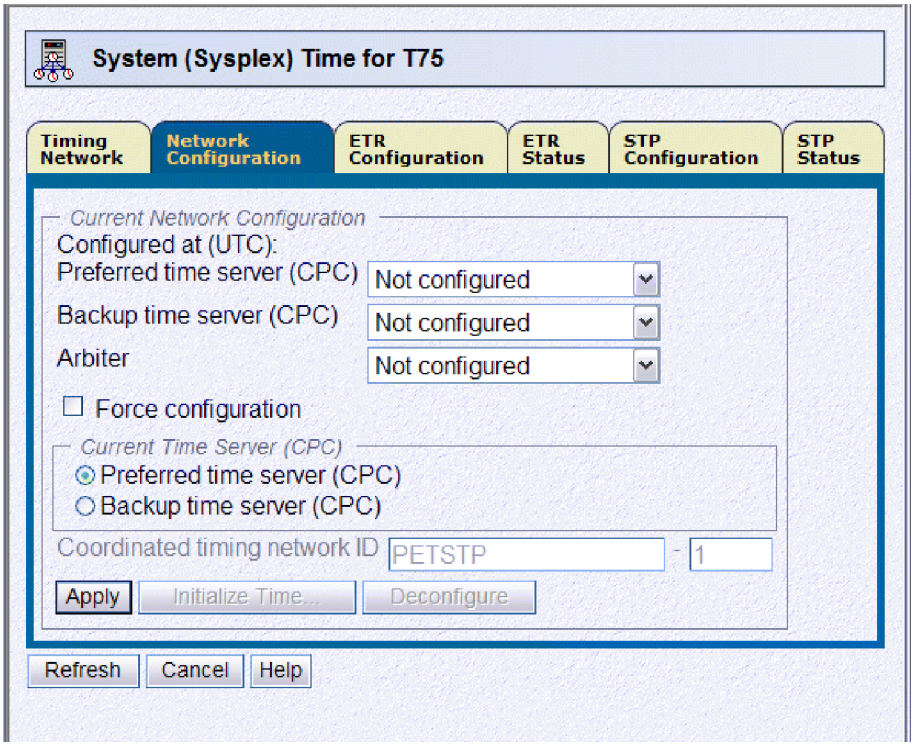


Figure 37. Initial view of the System (Sysplex) Time – Network Configuration panel on T75

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 34, 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 38 on page 75 shows this configuration in our environment.

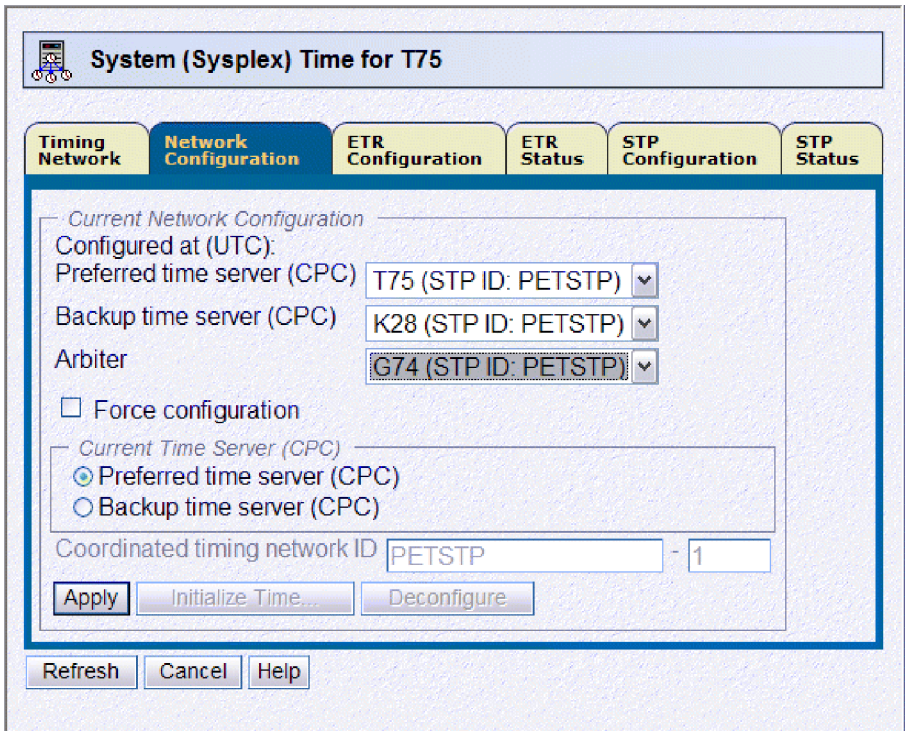


Figure 38. System (Sysplex) Time task – Network Configuration panel with all server roles assigned

Clicking the **Apply** button on the **Network Configuration** tab causes the mixed CTN to STP-only CTN migration to begin. Figure 39 shows the Global Timing Network ID Change Confirmation dialog that appears.

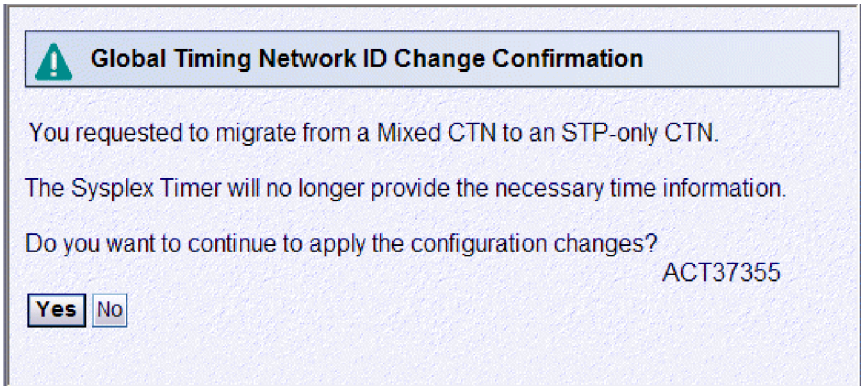


Figure 39. Global Timing Network ID Change Confirmation

Clicking **Yes** in Figure 39 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 40.

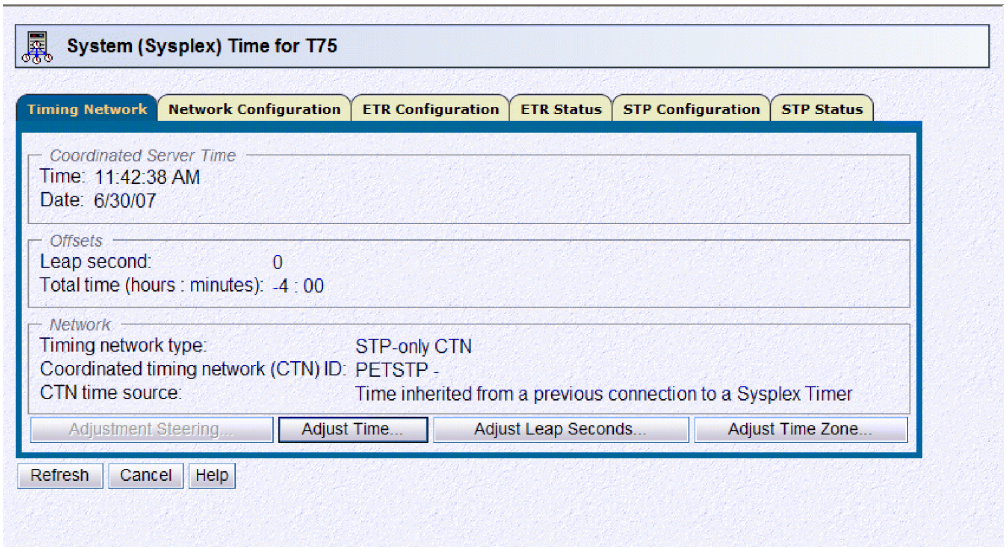


Figure 40. System (Sysplex) Time: Timing Network panel on T75 – STP-only CTN

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 41 on page 78.

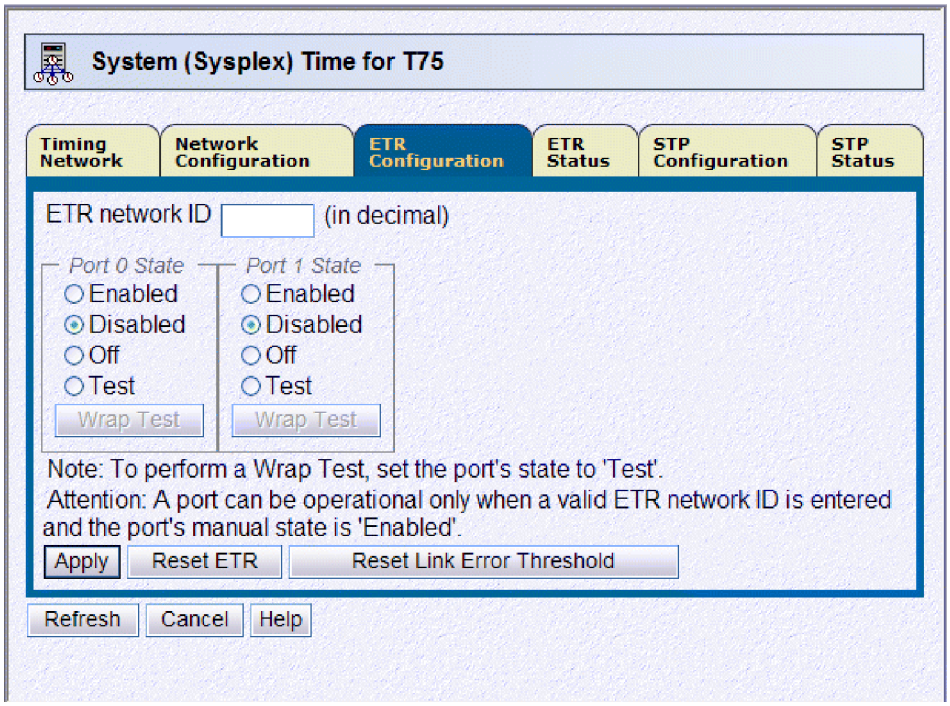


Figure 41. System (Sysplex) Time: ETR Configuration panel – STP-only CTN

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 42 on page 79 illustrates these points.

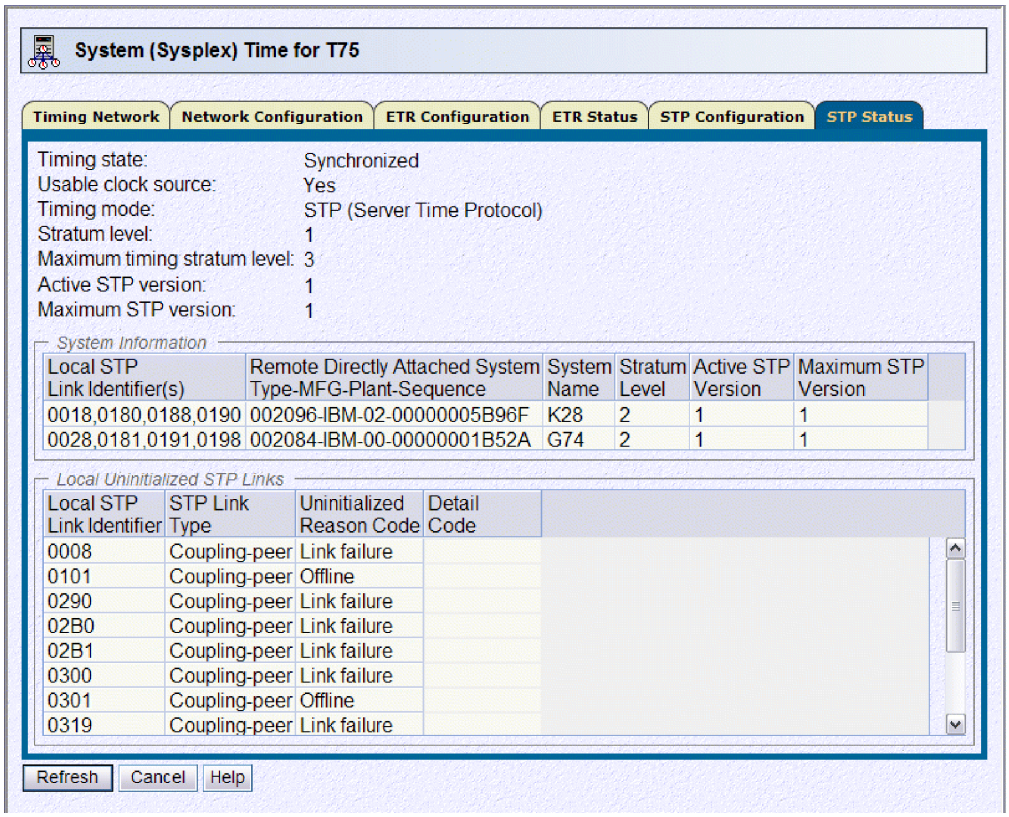


Figure 42. System (Sysplex) Time: STP Status panel – STP-only CTN

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 43 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.

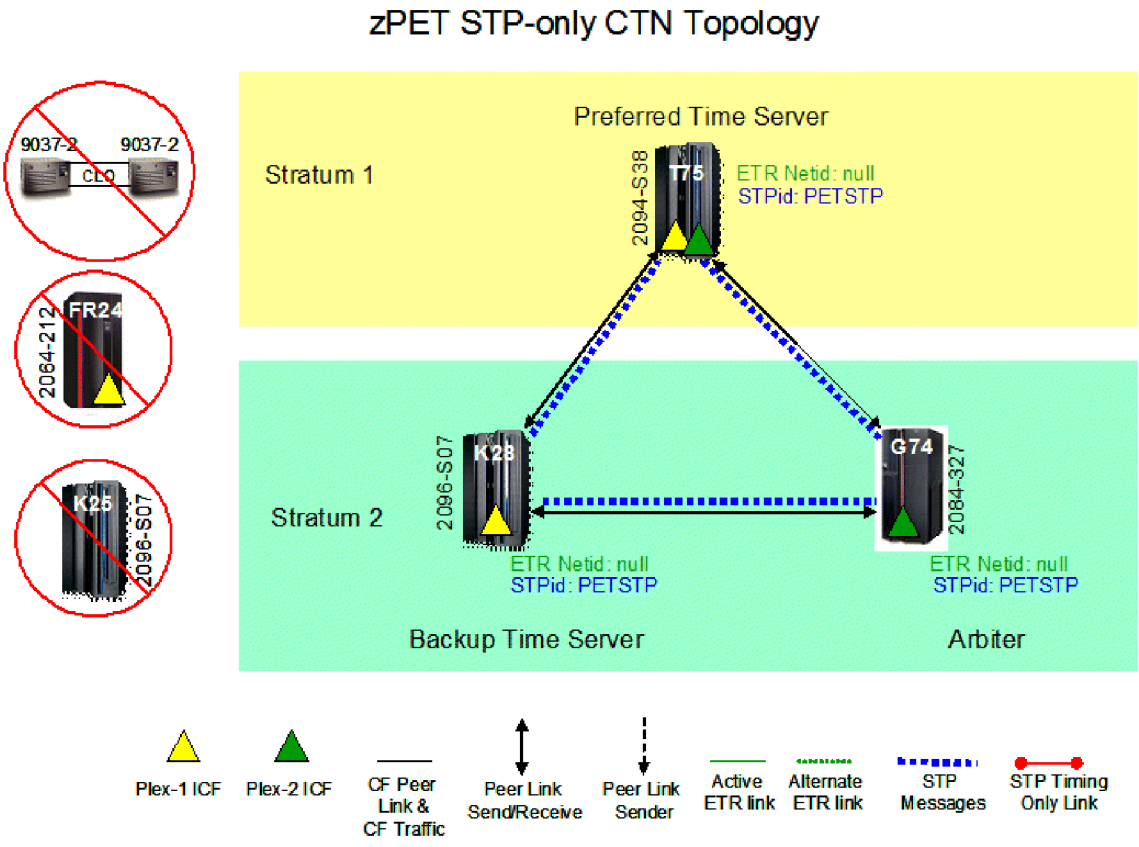


Figure 43. zPET STP-only CTN

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 72, 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 44.

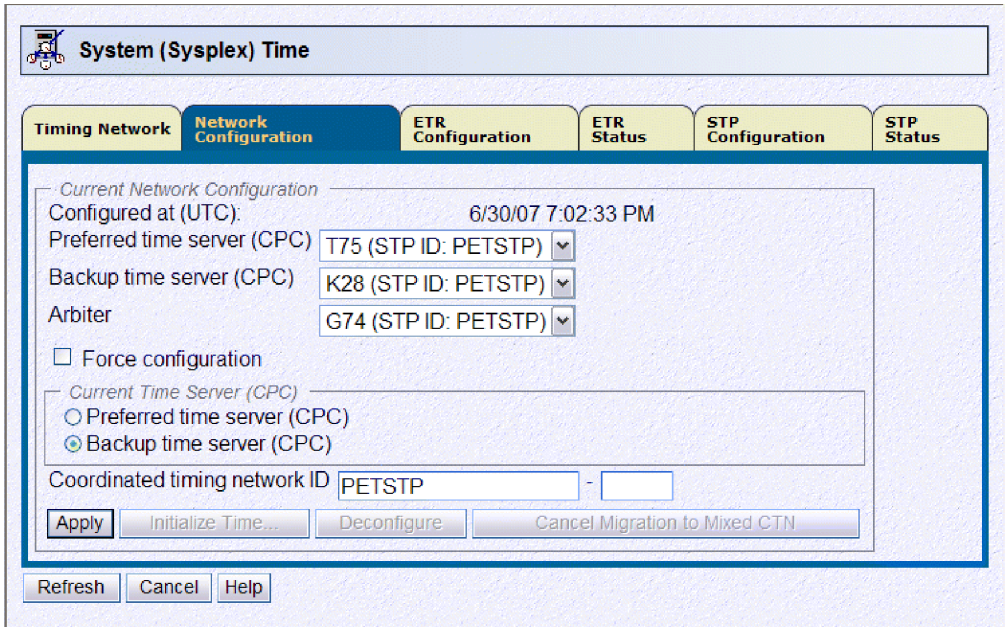


Figure 44. System (Sysplex) Time: Network Configuration panel – assigning K28 as current time server

After clicking the Apply button, we were prompted to confirm the change, as shown in Figure 45.

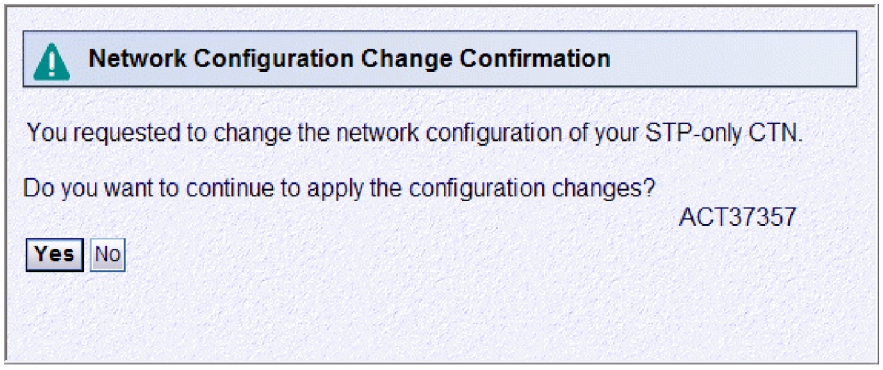


Figure 45. Network Configuration Change Confirmation panel – apply CTN role change

Figure 46 indicates successful completion of the CTN configuration change.

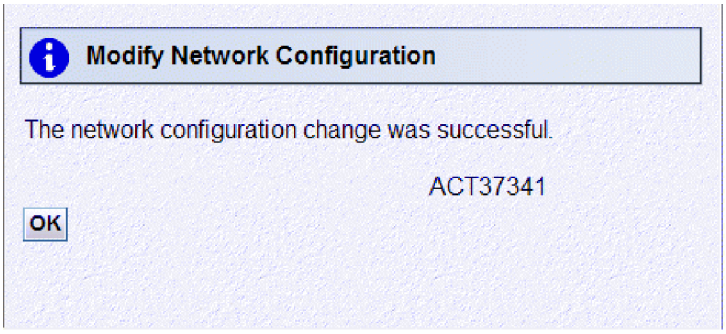


Figure 46. Modify Network Configuration panel – successful CTN role change

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 47.
Figure 47 confirms that K28 is now the stratum 1 server and the System

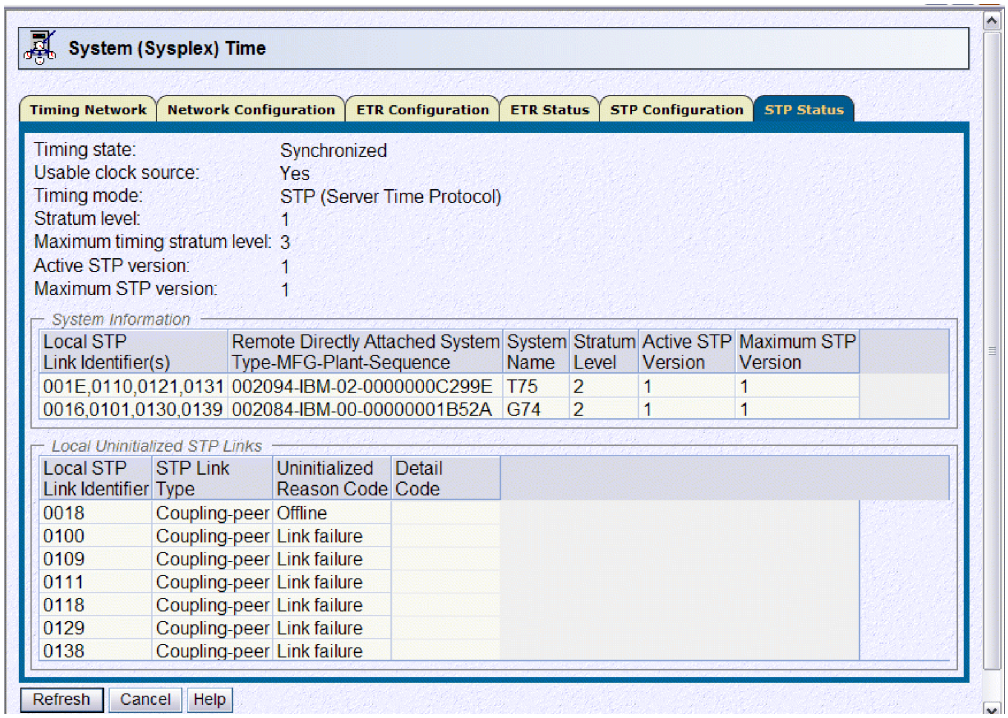


Figure 47. System (Sysplex) Time: STP Status panel with K28 as current time server

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 48 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).

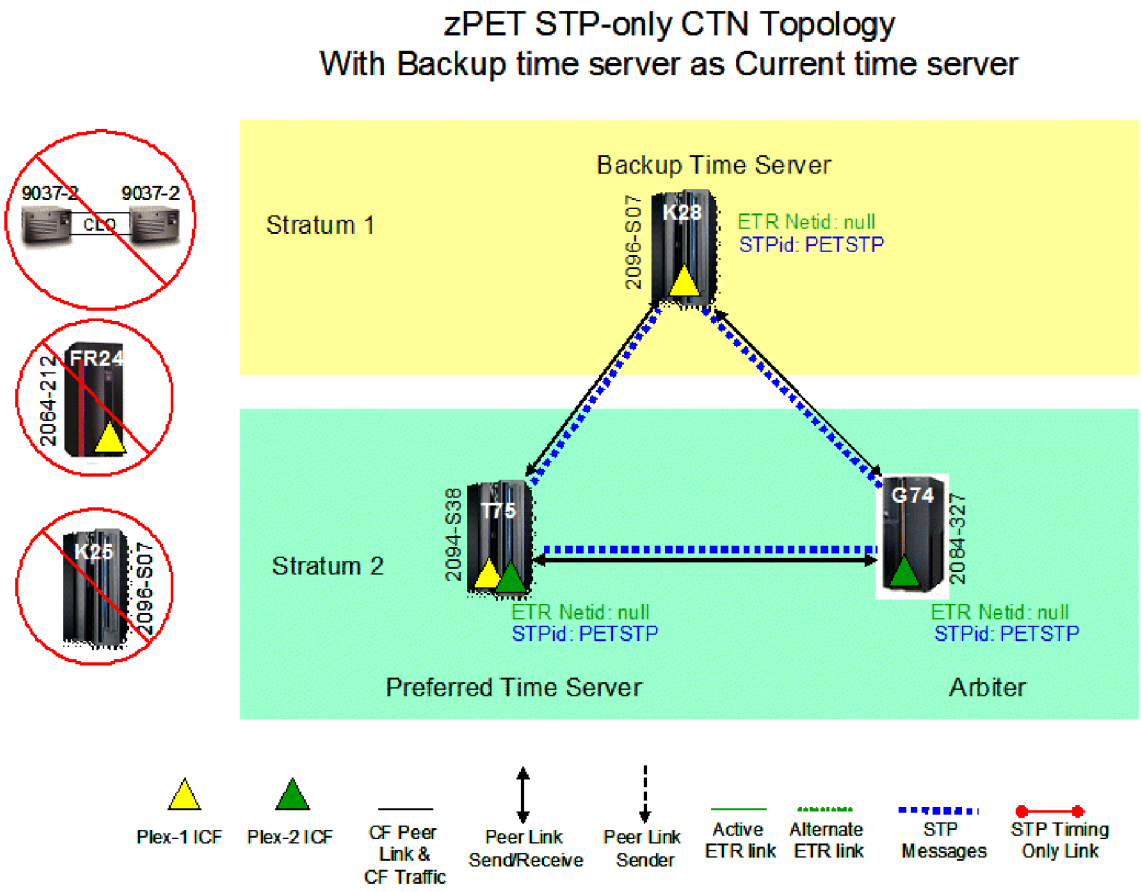


Figure 48. zPET STP-only CTN with backup time server as current time server

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 48 to the topology depicted in Figure 36 on page 73.

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 49 shows the starting point for this reverse migration. In "Changing server roles in an STP-only CTN" on page 80, we made the backup time server (K28) the current time server, so we must start with the **Network Configuration** tab for K28.

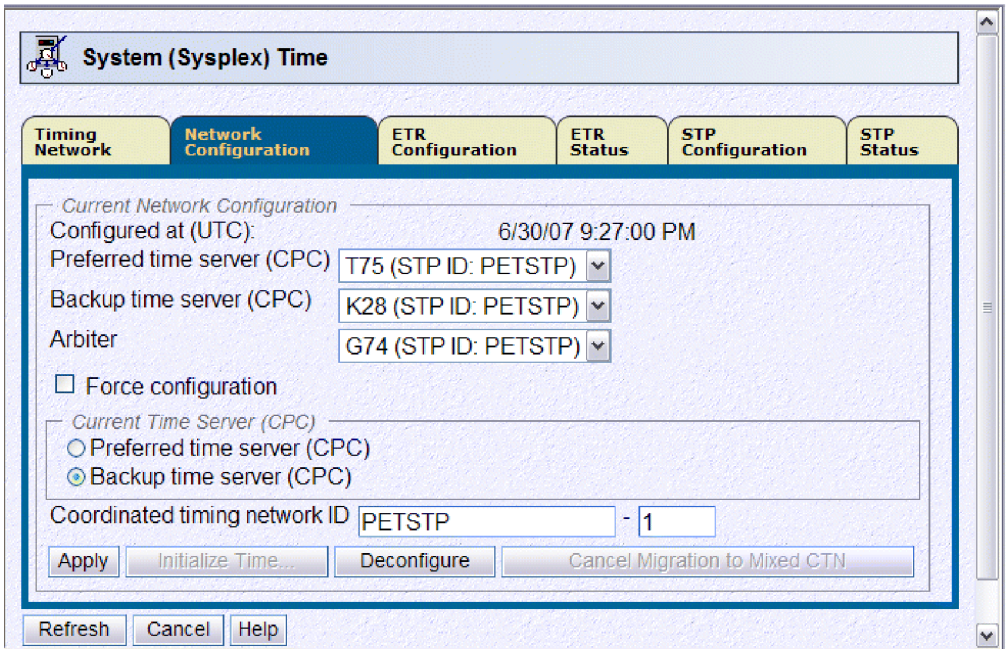


Figure 49. System (Sysplex) Time: Network Configuration panel – K28 starting reverse migration to mixed CTN

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 3 on page 41). When we clicked the **Apply** button, we were presented with the cautionary confirmation message shown in Figure 50 on page 85.

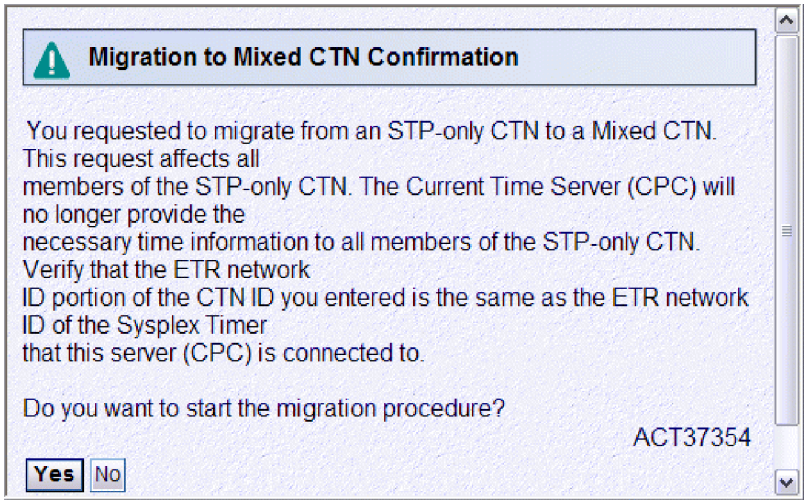


Figure 50. Migration to Mixed CTN Confirmation: Confirm migration from STP-only to mixed CTN (step 1)

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 50. We were then presented with an additional warning before final confirmation, as shown in Figure 51. 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.

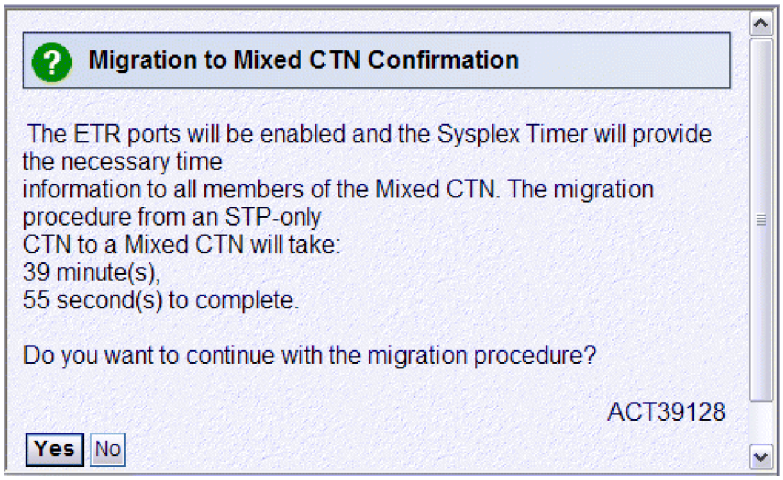


Figure 51. Migration to Mixed CTN Confirmation: Confirm migration from STP-only to mixed CTN (step 2)

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 52 on page 86.

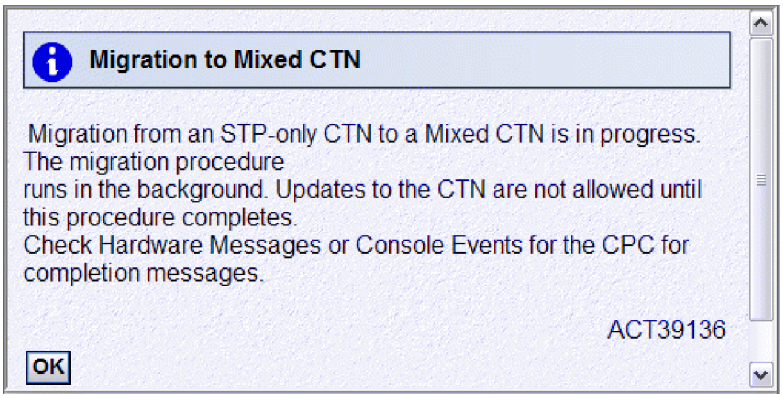


Figure 52. Migration to Mixed CTN: STP-only to mixed CTN migration in progress

Figure 52 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 53 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.

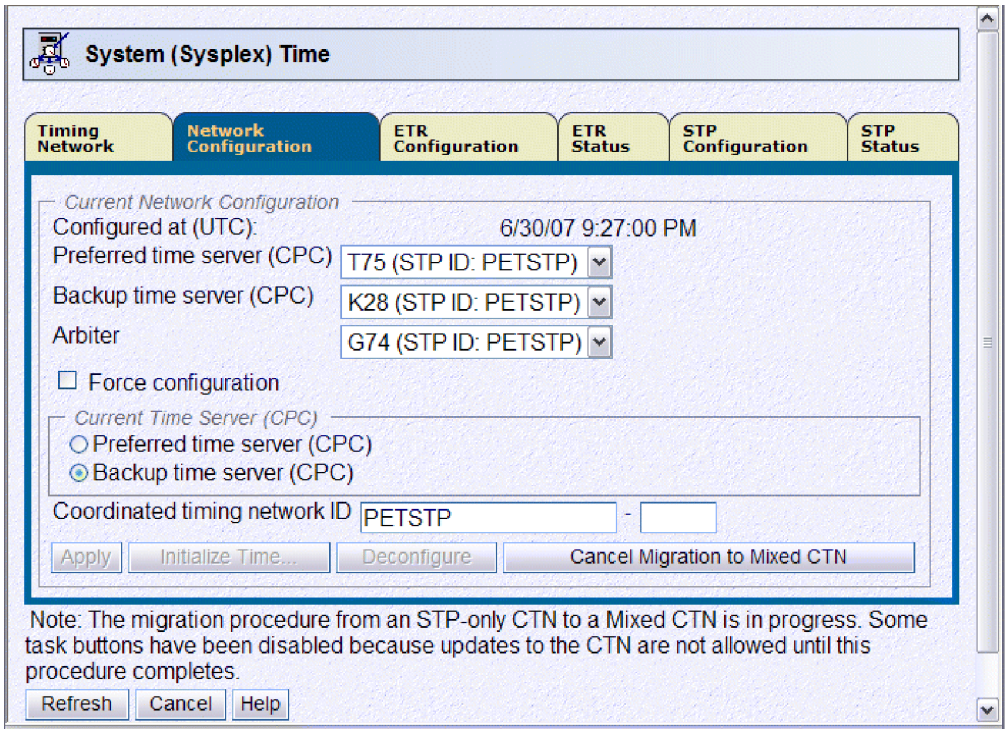


Figure 53. System (Sysplex) Time: Network Configuration panel – Migration to mixed CTN in progress

Figure 52 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 72, 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: J80  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: J80  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 54 on page 88.

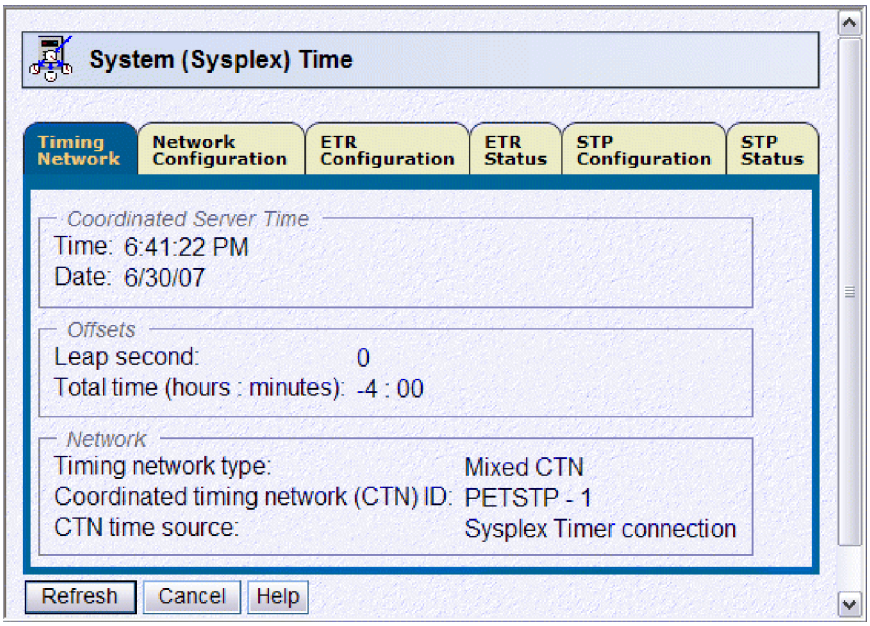


Figure 54. System (Sysplex) Time: Timing Network panel – K28 back in mixed CTN

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 55.

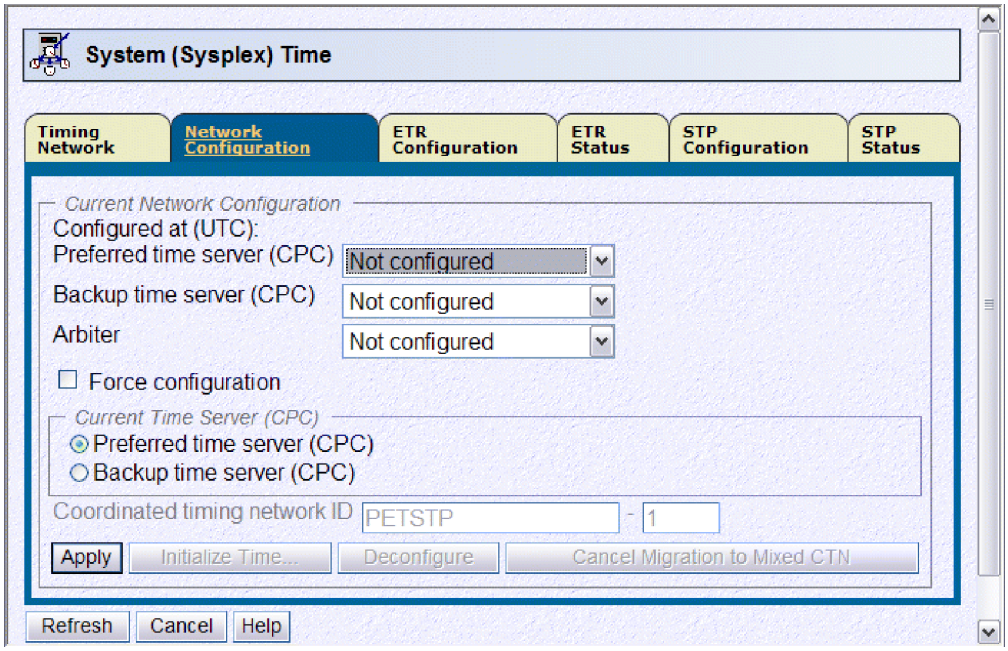


Figure 55. System (Sysplex) Time: Network Configuration panel –K28 back in mixed CTN

Next, the **ETR Configuration** tab for K28 confirmed that its ETR ports were enabled for ETR network ID 01, as shown in Figure 56 on page 89.

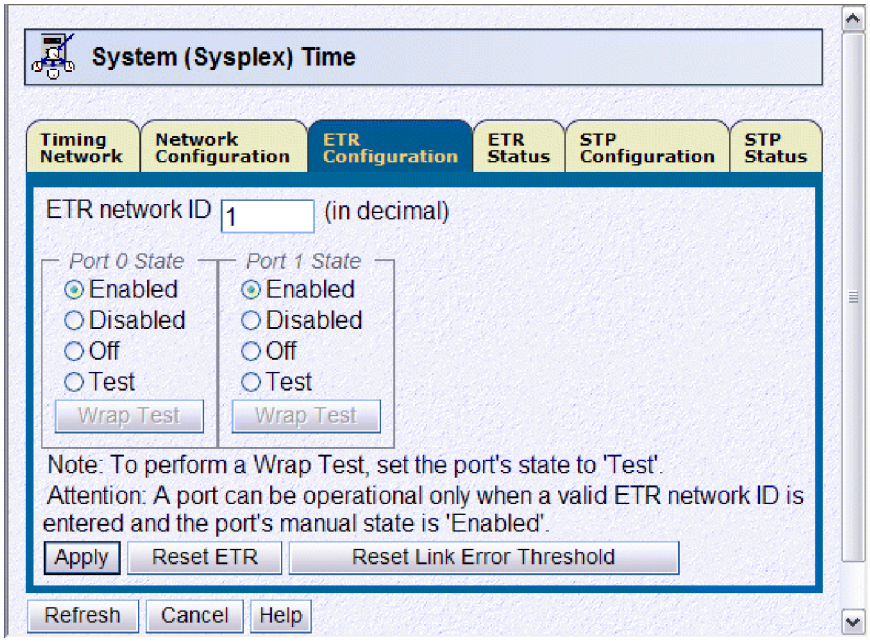


Figure 56. System (Sysplex) Time: ETR Configuration panel – K28 back in mixed CTN

Finally, we used the **STP Status** tab to verify the CTN configuration from K28's point of view, as shown in Figure 57. Figure 57 showed that K28 was at stratum 1 and synchronized in ETR timing mode, as was T75, while G74 remained at Stratum 2.

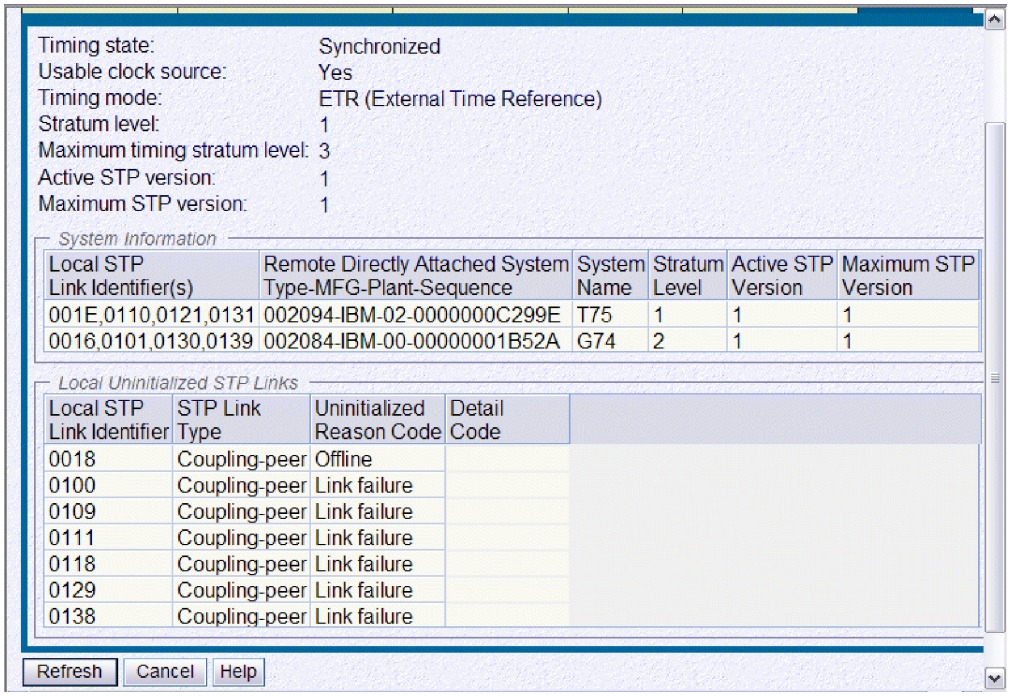


Figure 57. System (Sysplex) Time: STP Status panel – K28 back in mixed CTN

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 13, and decided to leave our timing network topology in the mixed CTN configuration shown in Figure 36 on page 73. 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 5. 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 58 shows an example of the image profile for our J80 z/OS image with 2 zIIPs defined:

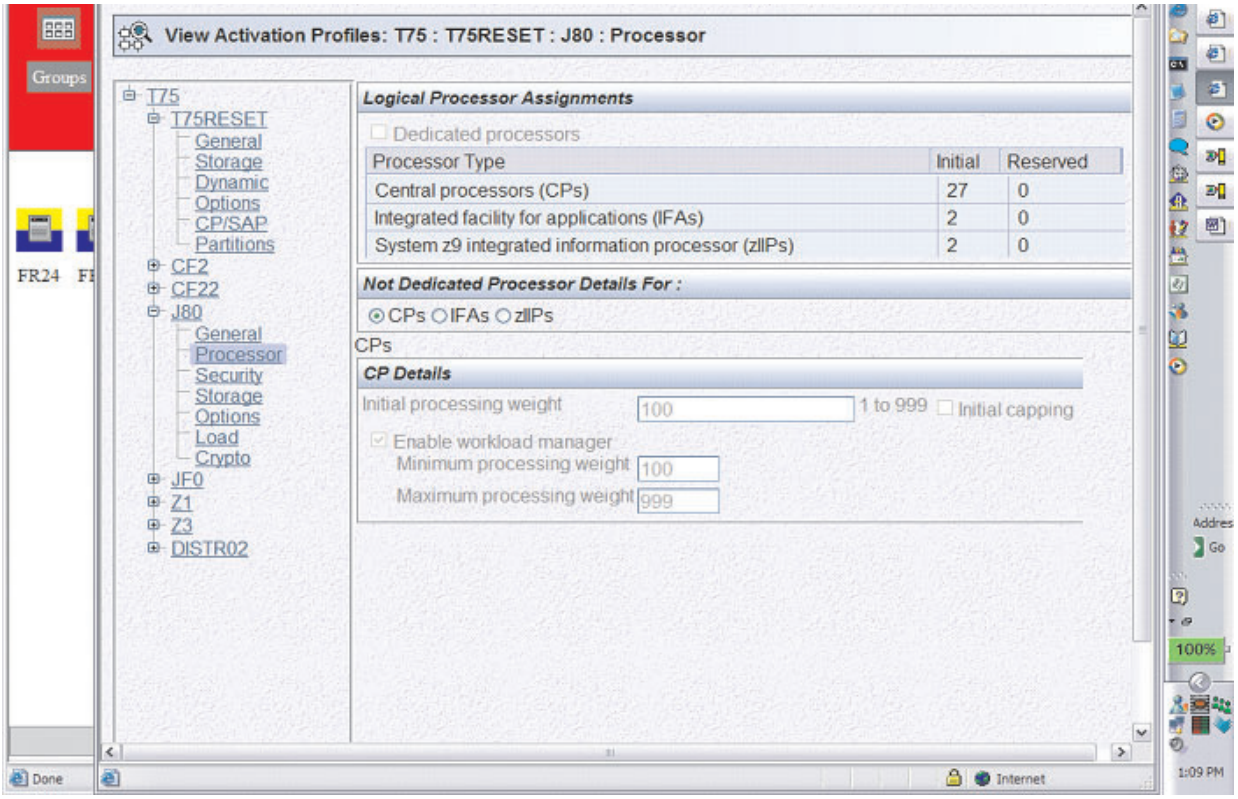


Figure 58. Image profile for our J80 z/OS image with 2 zIIPs defined

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.02.00000000000C299E
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                 Line 1 of 35
Command ===>                                                  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                        91.0                       1.5    41.8   43.3
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                                              1.0            1.0
*AAP                       12.0                       0.5    82.1   82.6
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                                              0.4            0.4

*ICF                       7.0                        0.0     100    100
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                                              0.0            0.0

*IIP                      12.0                        0.2     0.1    0.3
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:

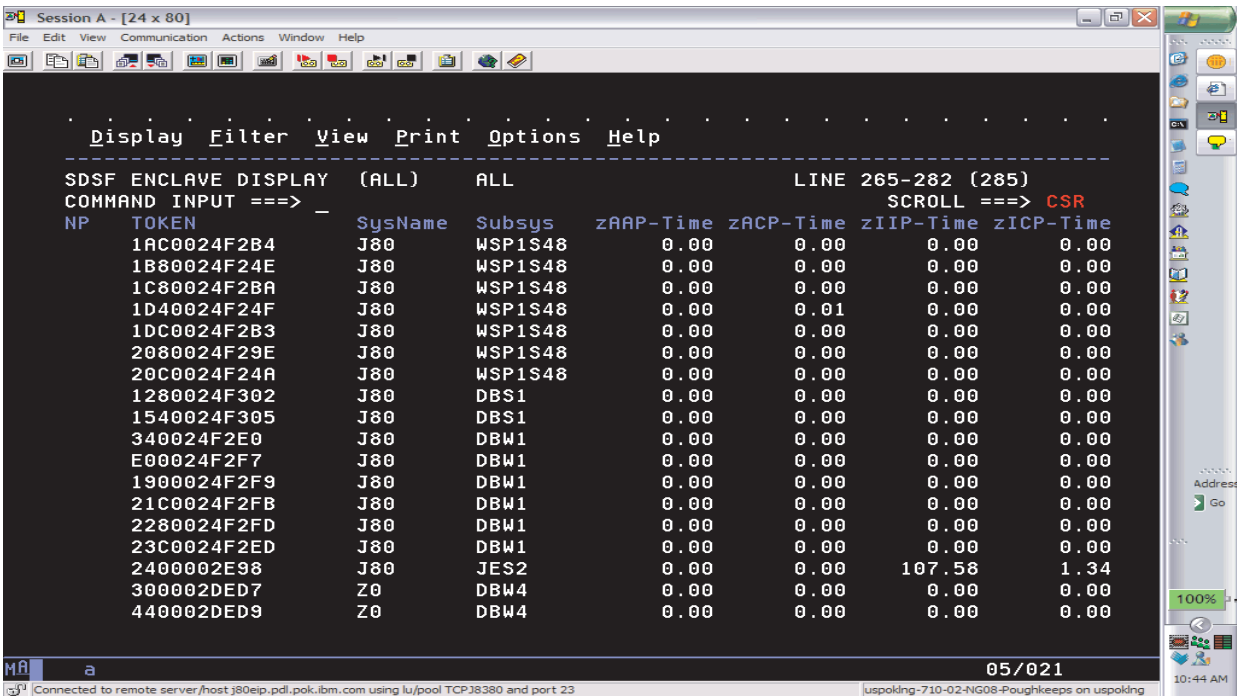


Figure 59. SDSF display showing zIIP utilization

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:

```
OB550:  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 60. zIIP is represented by **IIP**.

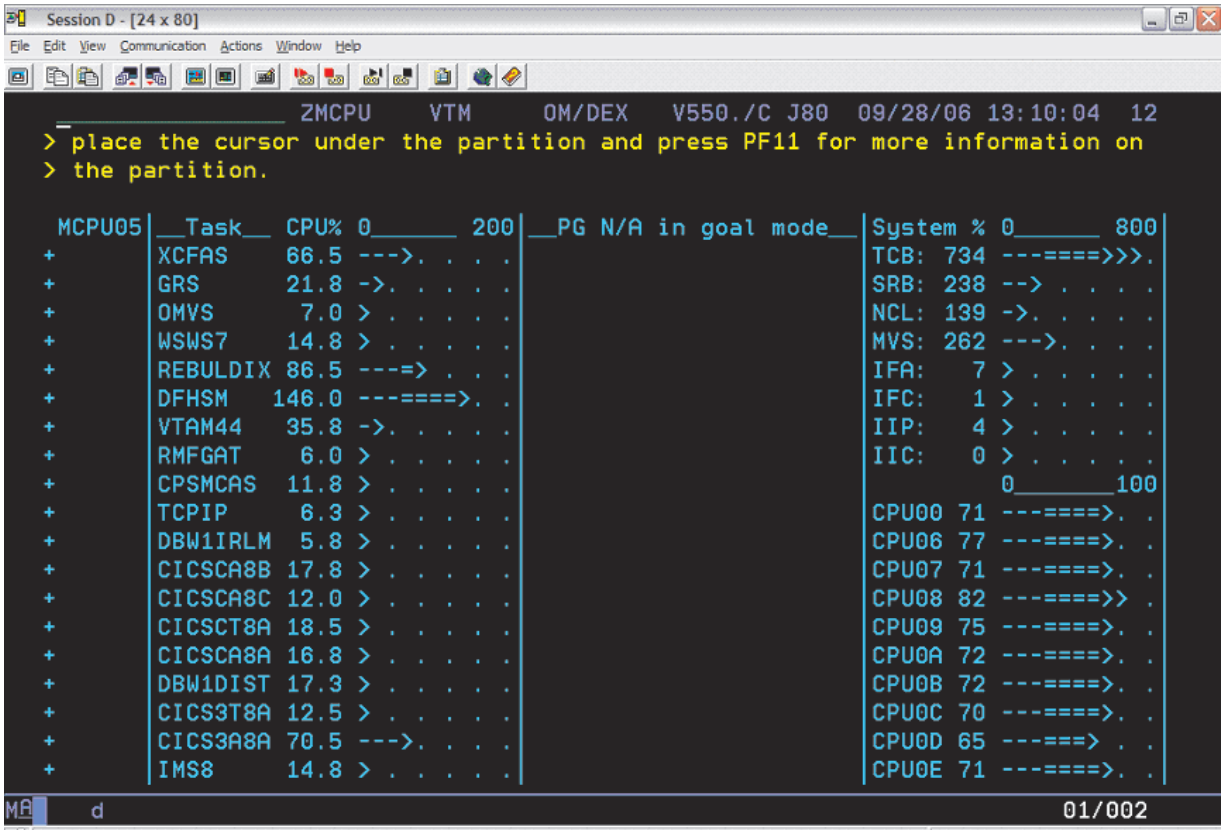


Figure 60. OMEGAMON ZMCPU screen

From your TEP server, the OMEGAMON XE for z/OS zIIP can be found in the predefined workspace System CPU Utilization. Figure 61 on page 99 and Figure 62 on page 100 show the TEP OMEGAMON XE for z/OS System CPU Utilization workspace:

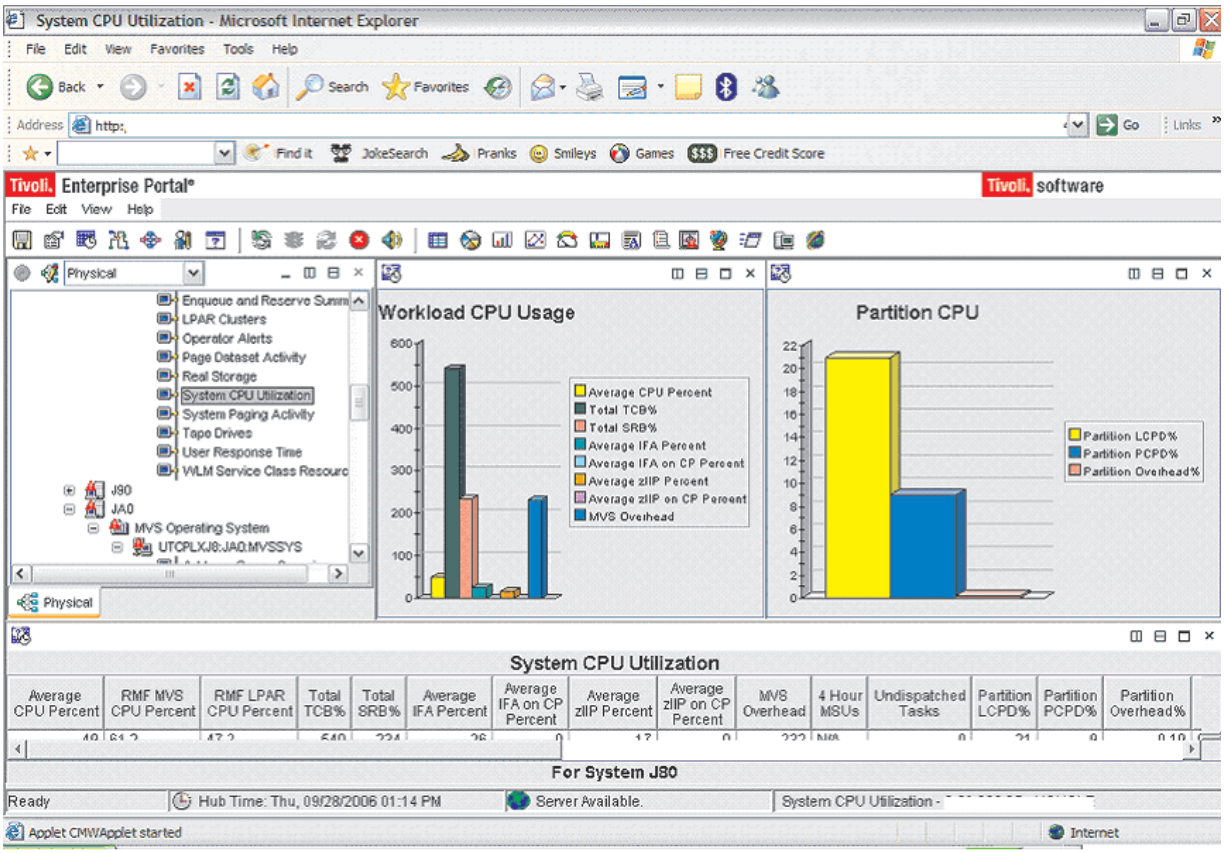


Figure 61. OMEGAMON System CPU Utilization 1

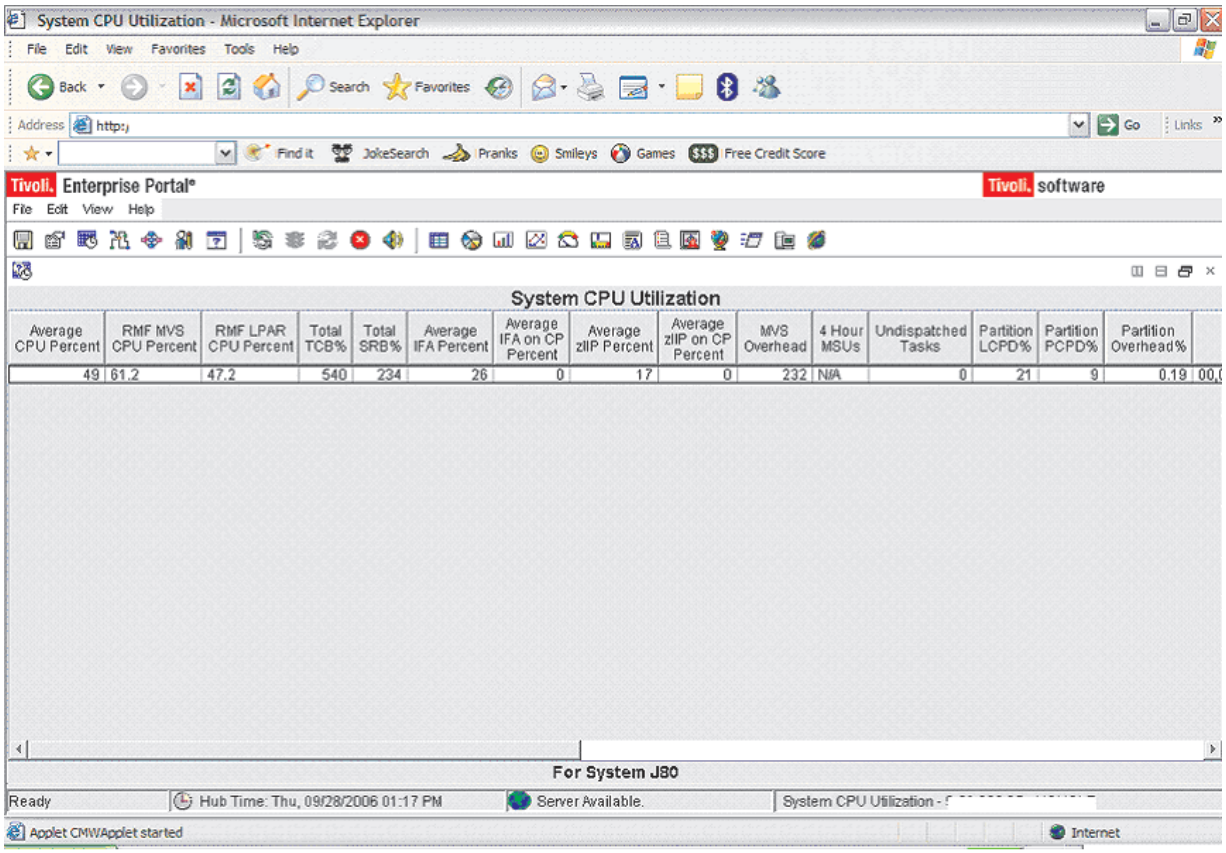


Figure 62. OMEGAMON System CPU Utilization 2

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 63 on page 101.

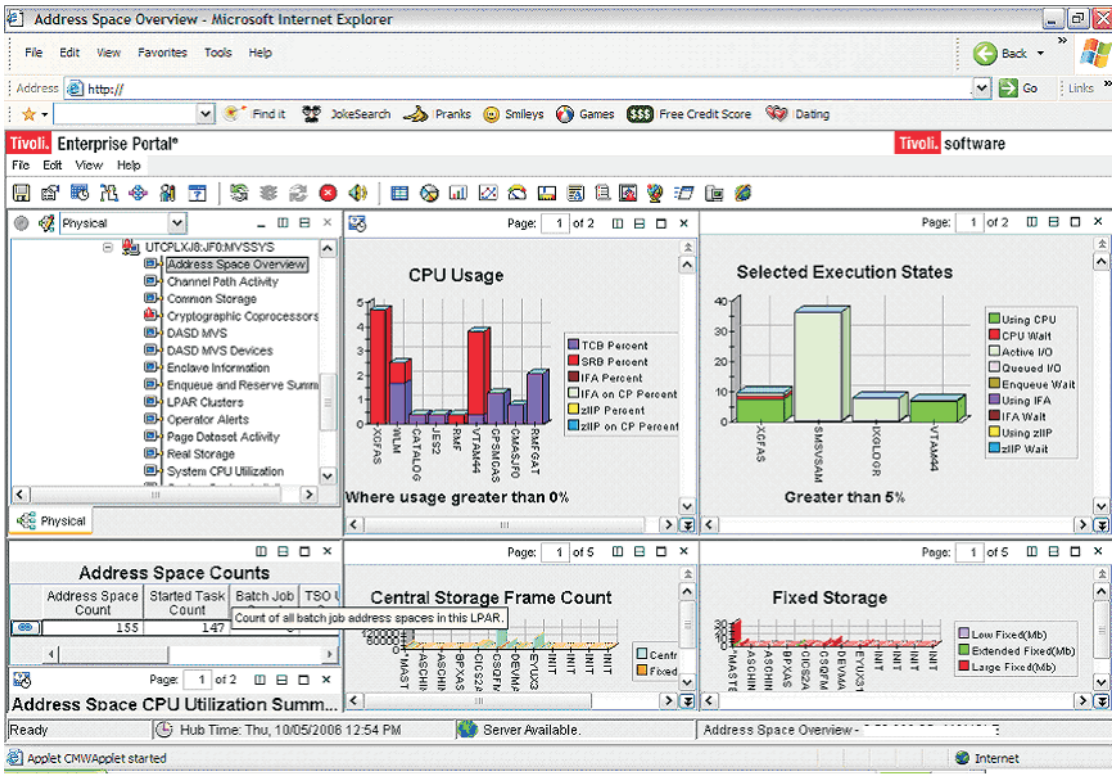


Figure 63. OMEGAMON Address Space Overview

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 IEAOPT*xx* 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/atsmastr.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 10 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 (ANTAS0*nn* only)

When zIIPEnable is set to YES, the ANTAS000, ANTAS0*nn*, and ANTCL0*nn* 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 ANTAS0*nn* 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 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 ANTAS0*xx* address spaces. Then we started the CXRC session with zIIP processors offline and monitored the XRC address spaces (ANTAS0*xx*) via RMF. We noticed that all the SDM address spaces were using the general purpose CPUs for the processing, as expected. Figure 64 and Figure 65 show how RMF reported the amount of zIIP eligible work by ANTAS0*xx* address spaces.

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

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

            Service    --- Time on CP % ---   ----- EAppl % -----
Jobname  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
DBW1DBM1 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
TCPIP    SO SYSSTC       6.5    0.0    0.0      6.5    0.0    0.0
```

Figure 64. RMF Monitor III Processor Usage screen showing amount of zIIP eligible work by ANTAS0xx address spaces (scenario 1)

Figure 65 shows the RMF Workload Activity Report showing zIIP eligible workload by the ANTAS0*xx* 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
        ABSRPTN    54K                                  SHARED        0.0
        TRX SERV   54K  PROMOTED    0.000                HSP           0.0
```

Figure 65. 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 66 on page 105 and Figure 67 on page 105 show the zIIP utilization by ANTAS0*xx* 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 % ---   ----- EAppl % -----
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
CSQ8MSTR 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 66. RMF Monitor III Processor Usage screen showing amount of work processed by zIIP processors for ANTAS0xx address spaces (scenario 2)

Figure 67 shows the workload that was processed by zIIP processors for the ANTAS0*xx* 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 67. RMF Workload Activity Report showing work processed by zIIP processors for the ANTAS0xx address spaces (scenario 2)

Chapter 6. Using TPC-R V3.3 in our zPET environment

We have added the IBM TotalStorage Productivity Center for Replication (TPC-R) for System z to our lineup of products in our z/OS integration test environment.

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

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

Overall, TPC-R simplified our configuration and we have been very happy with the product. Once up and running, we found the Web browser interface to be intuitive and very easy to use. TPC-R improved our operations support by providing different features such as overwrite protection and improved monitoring and messages. It provided simple operational control of copy services tasks, which includes starting, suspending, and resuming sessions. Our only real trouble area was in the installation and setup (see "Installing and setting up TPC-R" for more about this).

TPC-R environment in zPET

We used the following software and hardware during our testing:

- z/OS V1.8 and V1R9
- TPC-R V3.3
- DB2 Version 9.1 (sharing between 3 systems)
- Websphere Application Server V6.1
- DFSMSsdm w/APAR OA18953
- IBM System Storage DS8000 (2107), microcode level 2.4 or higher
- IBM TotalStorage Enterprise Storage Server (ESS) Model 800 (2105-800), LIC level 2.4.4.45 or higher

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.

We performed periodic failover/failback scenarios to ensure that application is consistent on the secondary volumes. We also tested disaster recovery support with failover/failback capability for the DS8000s.

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.rm33.doc/welcome.html.

TPC-R V3.3 will not install into a WebSphere Application Server network deployment (ND) configuration; it will only install properly into a base server setup. Also, there should be only a single copy of the application running within a single sysplex.

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.
- Re-ran 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 was enabled.

We checked the output of the customization jobs carefully, including the z/OS UNIX files created (install_RM.log and install_RM_err.log) for errors.

TPC-R product documentation

See the TPC-R home page at www-306.ibm.com/software/tivoli/products/totalstorage-replication/ for full information about the product. The home page contains links to detailed information, in particular, the TPC-R Information Center and support.

Chapter 7. 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 8. 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,ICSFMIG7731_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 9. Migrating to and using Enterprise Key Manager 2.1

We upgraded our existing Enterprise Key Manager (EKM) to the latest release, 2.1. We previously had been using Release 1 of the product. One noticeable feature in the new release is that the password fields in the configuration file are no longer visible in the clear once EKM has been started. The passwords that you have coded for keystores will be obfuscated, such as in the following example, once EKM server has started.

```
config.keystore.password.obfuscated = EE08856F909094848772
```

To perform the upgrade, you must obtain both the new EKM code, which is contained in **IBMKeyManagementServer.jar** and also the latest JZOS code, which is in **jzosekm.jar**.

The latest level of EKM and JZOS is available for downloading from the IBM Support & downloads site at www.ibm.com/support/docview.wss?&uid=ssg1S4000504.

Be sure to save a backup copy of the two prior .jar files before writing over them with the new versions. You might need them if you run into problems and decide to back out from the upgrade. The new **jzosekm.jar** is not backward compatible and will not work with EKM version 1, so the upgrade needs both new jar files.

The upgrade from version 1 to version 2.1 also requires a parameter addition to the configuration file or the server will fail to initialize. You must first stop the active EKM before making updates to the configuration file, as any updates made while EKM is running will only be wiped out when it is stopped. This is because EKM writes back to the configuration file its current active settings when it is stopped. The parameter to add once you have stopped EKM is:

```
Audit.metadata.file.name = metadata/EKMData.xm
```

If you forget to code this parm in the configuration file, you will get the following error when EKM attempts to start the server:

```
JVMJZBL2999T JvmExitHook entered with exitCode=-3, javaMainReturnedOrThrewException=0
```

After you download the two new jar files, copy them to your JDK directory (/J1.4/lib/ext) and overwrite the prior release 1 .jar files. Once you have completed the copy and have added the new parameter to the configuration file, start EKM and your upgrade is complete.

Automated handling of EKM audit and debug logs

We recently implemented an automated process for managing the audit and debug logs that the EKM server creates. Since the audit log has a maximum size per file and then creates a new one when the active one is filled up, maintaining the logs was a manual process. EKM server needs space to write the logs; otherwise, the server will not. Thus, it makes good sense to have a process for managing the log data.

The process we implemented is to use the z/OS UNIX **cron** utility to run a shell script at a specified date and time. This customized script also deletes old files

based on the retention we specify in the script. In our case, we decided to only keep the last 10 backups, which is 10 weeks worth of past audit and debug data.

Once you have sized the file system according to the number of desired backups, you no longer need to be concerned about filling up the log directory. This process removes the need to do manual cleanup, which had to be performed in the past. Another benefit is when you need to refer back to the audit data and you know the date when the encryption was performed, the dated logs makes it easier to find the entry.

The directory from which our **cron** processes run in our shared HFS environment is /J80/spool/cron/crontabs. Each system in the sysplex has its own crontabs directory so we had to replicate this process and customize the shell script for each system.

We created a new member named TAPEKMS using the **crontabs** command and inserted the following two lines. These scripts will be run on day 1 (Monday) at 11:55 PM.

```
55 23 * * 1 /ekmlogs/J80_debug_log_prune.sh
55 23 * * 1 /ekmlogs/J80_audit_log_prune.sh
```

The contents of the J80_audit_log_prune.sh script is:

```
# prune_file=<filename specified in the logfile directive
#             in the EKM configuration file>
prune_file=/ekmlogs/J80/audit/kms_audit.log
#
# number of backup files to save
files_to_save=10
#
# backup the activity log
i=1
date_append=`date | awk '{print $2 $3 $6}'`
cp $prune_file $prune_file.$date_append
#
# clear out all activity log records
:>$prune_file
#
# capture all of the backup files in new to old sequence
file_list=`ls -lat $prune_file.* | awk '{print $9}'`
#
# delete files greater than the designated number to save
for n in $file_list
do
file_name=$n
 if [ "$i" -gt "$files_to_save" ] ; then
  rm $file_name
 fi
 i=`expr "$i" + 1`
done
```

For more information about the setup and usage of the **cron** utility, see *z/OS UNIX System Services Planning*.

What happens when this customized script is run by **cron** is that the active kms_audit.log and debug.log files are copied to a new file that has the current date. The contents of the two active files are then cleared. Examples of how the directories look after this process has run for a few weeks is shown below:

```
171:/ekmlogs/J80/audit $ ls -al
drwxr-xr-x   3 LORAIN0  sys1         672 Mar  3 23:55 ..
-rw-r--r--   1 LORAIN0  sys1       76938 Mar  5 03:38 kms_audit.log
-rw-r--r--   1 LORAIN0  sys1      212619 Dec 31 23:55 kms_audit.log.Dec312007
```

```
-rw-r--r--   1 LORAIN0  sys1      180630 Feb 11 23:55 kms_audit.log.Feb112008
-rw-r--r--   1 LORAIN0  sys1      162255 Feb 18 23:55 kms_audit.log.Feb182008
-rw-r--r--   1 LORAIN0  sys1      148923 Feb 25 23:55 kms_audit.log.Feb252008
-rw-r--r--   1 LORAIN0  sys1      268464 Feb  4 23:55 kms_audit.log.Feb42008
-rw-r--r--   1 LORAIN0  sys1      126503 Jan 14 23:55 kms_audit.log.Jan142008
-rw-r--r--   1 LORAIN0  sys1      206490 Jan 21 23:55 kms_audit.log.Jan212008
-rw-r--r--   1 LORAIN0  sys1      317805 Jan 28 23:55 kms_audit.log.Jan282008
-rw-r--r--   1 LORAIN0  sys1      120572 Jan  7 23:55 kms_audit.log.Jan72008
-rw-r--r--   1 LORAIN0  sys1      125129 Mar  3 23:55 kms_audit.log.Mar32008

173:/ekmlogs/J80 $ ls -al
drwxr-x---  12 LORAIN0  sys1        1824 Sep 12 11:03 ..
drwxr-xr-x   2 LORAIN0  sys1         992 Mar  3 23:55 audit
-rw-r--r--   1 LORAIN0  sys1     6496031 Mar  5 03:38 debug
-rw-r--r--   1 LORAIN0  sys1     9831723 Feb 11 23:55 debug.Feb112008
-rw-r--r--   1 LORAIN0  sys1     9014114 Feb 18 23:55 debug.Feb182008
-rw-r--r--   1 LORAIN0  sys1     8131919 Feb 25 23:55 debug.Feb252008
-rw-r--r--   1 LORAIN0  sys1     14622804 Feb  4 23:55 debug.Feb42008
-rw-r--r--   1 LORAIN0  sys1     6561571 Mar  3 23:55 debug.Mar32008
```

Chapter 10. 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.

Using AES encryption with Network Authentication Service

Network Authentication Service (NAS), has been enhanced in the z/OS V1R9 release to provide both AES128 and AES256 encryption levels. We'll talk about our experiences with enabling these encryption levels and then the exploitation of them.

We used the following documentation to help us with the enablement and exploitation:

- *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926
- *z/OS Integrated Security Services LDAP Client Programming*, SC24-5924
- *z/OS Security Server RACF Command Language Reference*, SA22-7687
- *z/OS Integrated Security Services EIM Guide and Reference*, SA22-7875
- *z/OS Communications Server: IP Configuration Guide*, SC31-8775

Enabling AES encryption with Network Authentication Service

The NAS administration documentation indicates the following:

> Do not enable DES3, AES128 or AES256 ticket support until the Kerberos runtimes for all systems in the realm support that encryption type. Otherwise, you can obtain tickets that cannot be processed on a given system. In addition, do not enable DES3, AES128 or AES256 encryption support for user data unless all systems in the realm support that encryption type for user data. Otherwise, you can obtain session keys that are unusable for exchanging encrypted data. This means that all systems sharing the database must be running z/OS Version 1 Release 2 or later for DES3 or z/OS Version 1 Release 9 or later for AES128 or AES256.

Updating the /etc/krb5.conf file

A backup copy of each krb5.conf file was made with the following naming convention: krb5.conf.*date*, using the following command:

```
cp /etc/krb5.conf /etc/krb5.conf.date
```

The krb5.conf file located in /usr/lpp/skrb/examples was used as a guide when making the updates.

The default_tkt_enctypes and default_tgs_enctypes variables were updated to add the aes256-cts-hmac-sha1-96 and aes128-cts-hmac-sha1-96 encryption types.

```
default_tkt_enctypes = aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1,
   des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
default_tgs_enctypes = aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1,
   des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
```

It is important to remember that the encryption type selected is chosen from left to right. So, in our example, the first encryption type attempted would be aes256-cts-hmac-sha1-96.

As of the time of our testing, the NAS administration documentation makes it sound like all three variables, SKDC_TKT_ENCTYPES, default_tkt_enctypes, and default_tgs_enctypes are located in the krb5.conf file. This is not true. SKDC_TKT_ENCTYPES is located in the envar file. This will be clarified in a future edition of the documentation. The prologue for the sample krb5.conf file in /usr/lpp/skrb/examples does indicate that the SKDC_TKT_ENCTYPES is located in the envar file.

Updating the /etc/skrb/home/kdc/envar file

A backup copy of each envar file was made with the following naming convention: envar.*date*, using the following command:

```
cp /etc/skrb/home/kdc/envar /etc/skrb/home/kdc/envar.date
```

The envar file located in /usr/lpp/skrb/examples/skrbkdc.envar was used as a guide when making the updates.

The SKDC_TKT_ENCTYPES variable was updated to add the aes256-cts-hmac-sha1-96 and aes128-cts-hmac-sha1-96 encryption types.

```
SKDC_TKT_ENCTYPES=aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1,
   des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
```

Enabling the NAS principals

In order to enable a principal for AES encryption, the password needs to be changed on a z/OS V1R9 image.

Tip: Be aware that the LISTUSER command displays the available encryption types. For example, consider the following command:

```
lu ldapeim noracf kerb
```

which returns the following results:

```
USER=LDAPEIM

KERB INFORMATION
----------------
KERBNAME= LDAP/ja0eip.pdl.pok.ibm.com
KEY VERSION= 006
KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256
```

Notice that AES128 and AES256 are listed for the key encryption type. This is misleading! Listing of the AES128 and AES256 encryption types only indicates that they are available; it does not mean that they are active with keys associated. Again, the password must be changed in order to create the keys for the AES128 and AES256 encryption types. There is currently no way to determine whether or not the keys are available for a given encryption type. Marketing requirement MR0816074757 has been opened against RACF to address this concern.

Verifying AES encryption with Network Authentication Service

We used traces on the NAS server and on the clients to verify the operation of AES encryption.

Tracing on the server and clients

To verify that AES encryption is being used turn on tracing for both the NAS server and the client. Tracing is only needed on the server from the image where the transactions are initiated. So, for example, if you are logged on to System A and doing a transaction to System B, only the System A server's log will show signs of encryption.

To turn on the tracing for the server, update the /etc/skrb/home/kdc/envar file. Ensure the following are active:

```
_EUV_SVC_DBG_MSG_LOGGING=1
_EUV_SVC_DBG=*.8
_EUV_SVC_DBG_FILENAME=/tmp/kerberos.%.out
```

The trace data will be written to the images /tmp directory in the kerberos.%.out file. The % is replaced with a random number.

To turn on tracing for the client, issue the following exports from the z/OS UNIX session where the commands will be issued:

```
export _EUV_SVC_DBG=*.8
export _EUV_SVC_DBG_FILENAME=krb_client.trc
export _EUV_SVC_DBG_MSG_LOGGING=1
```

The trace data will be written to the working directory in the krb_client.trc file.

Verifying the hardware

In the host system's trace file, /tmp/kerberos.%.out, there will be a section at the beginning of the trace that indicates the available hardware encryption levels.

The following output shows how the trace appeared for one of our systems. Notice that the AES128 crypto assist is available and AES256 is not currently available.

```
070810 19:06:40 (00000000) DBG1 KRB/KRB_GENERAL krb5_crypto_initialization(): Enabling strong software cryptographic support
070810 19:06:40 (00000000) DBG1 KRB/KRB_GENERAL krb5_crypto_initialization(): ICSF FMID is HCR7750
070810 19:06:40 (00000000) DBG1 KRB/KRB_GENERAL krb5_crypto_initialization(): DES crypto assist is available
070810 19:06:40 (00000000) DBG1 KRB/KRB_GENERAL krb5_crypto_initialization(): DES3 crypto assist is available
070810 19:06:40 (00000000) DBG1 KRB/KRB_GENERAL krb5_crypto_initialization(): AES128 crypto assist is available
070810 19:06:40 (00000000) DBG1 KRB/KRB_GENERAL krb5_crypto_initialization(): AES256 crypto assist is not available
```

Verifying the NAS server

In the host system's trace file, /tmp/kerberos.%.out, there will be statements indicating the enabled encryption levels. These are the default_tkt_enctypes and default_tgs_enctypes variables, updated in the krb5.conf file. Verify that both AES128 and AES256 are enabled.

The following output shows how the trace appeared for one of our systems:

```
070810 19:06:40 (00000000) DBG8 KRB/KRB_GENERAL parse_line(): Line: default_tkt_enctypes = aes256-cts-hmac-sha1-96,aes128-cts-hmac-s
ha1-96,des3-cbc-sha1,des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
070810 19:06:40 (00000000) DBG8 KRB/KRB_GENERAL parse_line(): Line: default_tgs_enctypes = aes256-cts-hmac-sha1-96,aes128-cts-hmac-s
ha1-96,des3-cbc-sha1,des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
```

Verifying the clients

We used four different clients to verify NAS's use of AES encryption: EIM, LDAP, z/OS FTP, and xLinux FTP. For information about how we enabled each of these clients for use with NAS, see our previous test reports as listed in "Network Authentication Service configuration" on page 288.

Prior to issuing each client transaction, a **kinit** transaction must be issued to obtain the Kerberos credentials for the principal to be used within the client transaction. With the client tracing set, here is what you should expect to see in the trace to confirm the use of the AES encryption:

```
23317  070814 15:34:29 (00000029) DBG8 KRB/KRB_CRYPTO k5_aes_encrypt(): Software AES256 encryption performed for 16 bytes
23318  070814 15:34:29 (00000029) DBG8 KRB/KRB_CRYPTO k5_aes_decrypt(): Software AES256 decryption performed for 142 bytes
```

Notice that you should see both encryption and decryption messages.

In these examples, software was also used for the encryption and decryption. This is because our hardware is currently not enabled for AES256. A test was done

using AES128 encryption. We do have a machine where the hardware is enable for AES128 encryption. Here are the type of messages you would expect to see in that case:

```
070823 11:29:00 (00000000) DBG8 KRB/KRB_CRYPTO k5_aes_encrypt(): Clear key AES128 encryption performed for 122 bytes
```

In this example the "Clear key" designation in the message is the indication that crypto hardware was used in place of software for the encryption and decryption.

Tip: If you are using Kerberos with FTP, be aware that there are two principals used within the transaction. The level of encryption used may vary depending upon the encryption types enabled for each principal. The two principals used in the transaction will be the initial principal obtained via the **kinit** command. Then the KDC uses the FTP service principal to communicate with the FTP server. When executing the initial FTP transactions, the trace files showed that two different encryption levels were being used. This was due to the FTP service principal not having its password changed within the z/OS V1R9 environment to enable the AES keys. Again, there is no current way to determine the encryption keys that are enabled for the various encryption types of a given principal. When in doubt, change the principals' password to enable the keys for the various encryption types.

KEYTAB file verification

During testing of the Network Authentication Service using FTP, we found a problem where the version of the FTP service principal listed in the KEYTAB file was out of sync with that of the FTP service principal's KERB segment in RACF.

Diagnosing the problem

While executing the FTP transaction, the following error message was displayed:

```
535-GSSAPI error major status code: d0000 - EUVF02016E Security mechanism detects error
535-GSSAPI error minor status code: 96c73ab5 - Key table entry is not found
535 Request to accept security context failed
```

We found the following information about the error minor status code, 96c73ab5, in *z/OS Integrated Security Services Network Authentication Service Administration*:

96C73AB5 Key table entry is not found.

Explanation: The requested key table entry was not found in the key table.

User response: No action is required.

From the z/OS UNIX shell, we issued the **keytab list** command to display the contents of the keytab file. The response is similar to the following:

```
Key table: /etc/skrb/krb5.keytab

Principal: ftp@IBM.COM
  Key version: 1
  Key type: 56-bit DES
  Entry timestamp: 2007/06/12-10:55:45

Principal: ftp@IBM.COM
  Key version: 1
  Key type: 56-bit DES using key derivation
  Entry timestamp: 2007/06/12-10:55:45

Principal: ftp@IBM.COM
  Key version: 1
  Key type: 168-bit DES using key derivation
  Entry timestamp: 2007/06/12-10:55:45
```

Notice that the value of the key version is 1.

We use RACF for the KDC. To display the associated FTP principal in the RACF KDC, we issued the following LISTUSER command at the TSO Ready prompt:

```
LU FTP NORACF KERB
```

The command response is similar to the following:

```
USER=FTP

KERB INFORMATION
----------------
KERBNAME= ftp
KEY VERSION= 004
KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256
```

Notice here that the value of the key version is `004`.

So, even though the same principal exists in both the keytab file and the RACF KDC, the **ftp** command using Kerberos will fail because the key versions are not the same. What was misleading for us was that the FTP principal did exist in the keytab file. It took awhile before we realized that the key version mismatch was the problem.

Another problem that we had was that when we added the principal to the keytab file, we did not specify the key version. The **keytab add** command does not require that the key version be specified either, as it does for the password. If the key version is not specified on the **keytab add** command, it defaults to a value of 1.

Resolving the problem

To correct this condition, the principal with the correct key version needs to be added to the keytab file. We also removed the existing principal from the keytab file. Here are the series of commands we issued. These commands are all issued from the z/OS UNIX shell. Note that `ftp` is used as the principal in these examples. Replace `ftp` with the principal you will be adding to the keytab file.

1. Remove the existing principal:

   ```
   keytab delete ftp
   ```

2. List the keytab contents just to make sure the FTP principal is not there:

   ```
   keytab list
   ```

3. Add the principal into the keytab file using the key version found in the KDC. In this example, we use a key version of `004` to match the earlier example:

   ```
   keytab add ftp -p password -v 004
   ```

Additional resolution actions

We took some additional actions to help you avoid the amount of time we spent on this problem.

The first was to update informational APAR II13471 and open a publication update request for the 96C73AB5 error minor status code. Currently, *z/OS Integrated Security Services Network Authentication Service Administration* indicates that no action is required as the user response for the error minor status code. The APAR and publication update state that the user response should be:

List the entries in the key table file and if there is no entry for the principal used by the application then you will need to add one with the correct version number. If there is an entry already there, you will need to verify that the version number in the key table entry matches the version number for the same principle in the KDC database. If the KDC database has more than one entry for the principle, you need the highest version number.

The second action was to submit a marketing requirement. The marketing requirement number is MR0803074720. It requests a function to verify the contents of the keytab file with an associated KDC. If an out of sync condition exists, the function will flag it.

FTP Kerberos single signon support

The z/OS Communications Server FTP has been enhanced in the z/OS V1R9 release to enable FTP to use Kerberos for single signon. We'll describe our experiences with enabling z/OS Communications Server FTP for Kerberos single signon and then its exploitation.

We used the following documentation to help us with the enablement and exploitation:

- *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926
- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS Communications Server: New Function Summary*, GC31-8771

The client will still be prompted for a user ID during the authorization phase of the FTP transaction. What this new function prevents is the prompting and requirement of the password for the submitted user ID if it matches the ID in the Kerberos credentials that are received.

Enabling FTP Kerberos single signon support

This discussion will be just for the enablement of z/OS Communication Server FTP for single signon using Kerberos. See the discussion in our December 2005 test report for our enablement of z/OS Communication Server FTP with Kerberos.

To eliminate the client password prompt, code the following statement in the server's FTP.DATA configuration file:

```
SECURE_PASSWORD_KERBEROS OPTIONAL
```

The default value of the SECURE_PASSWORD_KERBEROS statement is `REQUIRED`. If you want to use this function, you must update the statement. In our experience, we got caught by not updating this statement. We did have to enter a password regardless of whether or not the user ID we entered matched the ID in the Kerberos credentials. So we can safely say that if you want to always have a prompt for the password, coding the statement value as `REQUIRED` will do it for you.

Verifying FTP Kerberos single signon support

A **kinit** command was first issued to obtain our Kerberos credentials. Two **ftp** commands were then issued to verify the enablement. First, when prompted for the user ID, the ID in the Kerberos credentials was entered. A password was not required for this condition, as expected. For the second **ftp** command, at the

prompt for the user ID, the ID entered did not match the Kerberos credentials. A password was required for this condition, again as expected.

The following figures show how the two transactions looked.

Figure 68 shows an FTP transaction where the user ID matches the Kerberos credentials.

```
Using /u/smith/ftp.data for local site configuration parameters.
IBM FTP CS V1R9
FTP: using TCPIP
Connecting to: host_name host_ip port: 21.
220-FTPD1 IBM FTP CS V1R9 at host_name, 19:57:26 on 2007-06-29.
220 Connection will close if idle for more than 5 minutes.
>>> AUTH GSSAPI
334 Using authentication mechanism GSSAPI
>>> ADAT
235 ADAT=YGgGCSqGSIb3EgECAgIAb1kwV6ADAgEFoQMCAQ+iSzBJoAMCAQGiQgRA
Ol4LlQI557FV1w3g7DHnE7qQiyWOgdM3KLY9fXUIRYwPDzU8U3UQxxNcoVYBQyxHv
nwGWYn6ZtjNEG/cAxzM5g==
Authentication negotiation succeeded
NAME (je0eip.pdl.pok.ibm.com:SMITH):
smith
>>> USER smith
230-User SMITH is an authorized user
230 SMITH is logged on.  Working directory is "SMITH.".
Command:
quit
>>> QUIT
221 Quit command received. Goodbye.
```

Figure 68. FTP transaction in which the user ID matches the Kerberos credentials

Figure 69 shows an FTP transaction where the user ID does not match the Kerberos credentials.

```
Using /u/smith/ftp.data for local site configuration parameters.
IBM FTP CS V1R9
FTP: using TCPIP
Connecting to: host_name host_ip port: 21.
220-FTPD1 IBM FTP CS V1R9 at host_name, 20:03:04 on 2007-06-29.
220 Connection will close if idle for more than 5 minutes.
>>> AUTH GSSAPI
334 Using authentication mechanism GSSAPI
>>> ADAT
235 ADAT=YGgGCSqGSIb3EgECAgIAb1kwV6ADAgEFoQMCAQ+iSzBJoAMCAQGiQgRA
KKrEYfNHzX3dyHlf1AH1FZDUx4mJ4On/rD1W8hKPilU5DDkVPNYgiBs7iVJjxLm76
XMF6ZspcdjCDAgu/ZtgoQ==
Authentication negotiation succeeded
NAME (je0eip.pdl.pok.ibm.com:SMITH):
jones
>>> USER jones
331 Send password please.
PASSWORD:

>>> PASS
230 JONES is logged on.  Working directory is "JONES.".
Command:
quit
>>> QUIT
221 Quit command received. Goodbye.
```

Figure 69. FTP transaction in which the user ID does not match the Kerberos credentials

Chapter 11. 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:

Using AES encryption with IBM Tivoli Directory Server

IBM Tivoli Directory Server (IBM TDS) was enhanced in the z/OS V1R8 release to provide AES encryption. We'll talk about our experiences with enabling AES encryption and then the exploitation of it.

We used the information in *IBM Tivoli Directory Server Administration and Use for z/OS*, SC23-5191 to help us with the enablement and exploitation.

Enabling AES encryption with IBM Tivoli Directory Server

Although AES encryption was made available with the initial release of IBM TDS during the z/OS V1R8 time period, we are now reporting on its enablement in this our z/OS V1R9 test report.

As a word of caution, the Integrated Security Services (ISS) LDAP Server is not enabled for AES encryption. Special consideration should be taken if the IBM TDS server will be interacting with an ISS LDAP Server regarding the use of AES encryption. See *IBM Tivoli Directory Server Administration and Use for z/OS* for details.

The first step—and the one that was the most confusing for us—was to create the data set that would hold the AES key. It turns out that this is just a simple sequential data set and the ISPF editor is used to enter the key label and key parts. The documentation does explain this but we thought that there would be some kind of tool to create and maintain these keys. We didn't think it would be this simple but it is. We created a sequential data set, IBMTDS.LDAP.AESKEYS, and added the following for the AES key:

```
LDAPSRV  123456789ABCDEF0  23456789ABCDEF01  ABCDEF0123456789  F9E8D7C6B5A43210
```

The format for the key is:

```
key-label key-part-1 key-part-2 key-part-3 key-part-4
```

The next step is to add the LDAPKEYS DD statement to the servers startup procedure to point to the AES key data set that we just created:

```
//LDAPKEYS DD DSN=IBMTDS.LDAP.AESKEYS,DISP=SHR
```

The userPassword attribute values were used for the AES encryption verification. To enable this, we updated the configuration file with the following:

```
pwEncryption AES:LDAPSRV
```

Notice that LDAPSRV matches the key-label of the record in the IBMTDS.LDAP.KEYS data set for the AES key to use.

The secretKey or replicaCredentials attributes are also available for AES encryption. These attributes have not yet been enabled in our environment.

IBM TDS was then recycled to pick up the changes to the configuration file and the startup procedure.

Verifying AES encryption with IBM Tivoli Directory Server

After IBM TDS was recycled, the server's JES log contained the following statements which helped validate the AES encryption enablement:

- In the DSOUT:

  ```
  pwEncryption: AES:LDAPSRV
  ```

- In the JESMSGLG:

  ```
  IAT4401  LOCATE FOR STEP=GO       DD=LDAPKEYS DSN=IBMTDS.LDAP.AESKEYS
  ```

To AES encrypt the user password in the IBM TDS backend, the password must be changed. To do this, we used the **ldapmodify** command along with an ldif file which contained the password change. To verify that the actual encryption takes place, we turned on tracing using this console command:

```
MODIFY ldap_started_task,DEBUG ALL
```

Here is an example of the ldif file contents:

```
dn: cn=Eddie Catu, ou=In Flight Systems, ou=Home Town, o=Your Company
changetype: modify
replace:x
userpassword: catu
```

Notice the userpassword is set to change to `catu`.

Then, we issued the following **ldapmodify** command to make the change:

```
ldapmodify -h host_ip -D "cn=LDAP Administrator, o=Your Company" -w admin_password
  -f /directory/EddieCatu.ldif
```

We found the following in the trace after the **ldapmodify** command:

```
*c..f}.Acn=Eddie *
*Catu, ou=In Flig*
*ht Systems, ou=H*
*ome Town, o=Your*
* Company0806...0*
*1..userpassword1*
*!0...{AES:......*
*.}d...|z.....LTf*
*%...".%cn=LDAP A*
*dministrator, o=*
*Your Company..20*
*070731225431.432*
*914Z.........I  *
```

Notice the string {AES: which indicates the beginning of the AES encryption. The closing brace (}) indicates the end of the encryption.

We issued the following console command to turn off tracing:

```
MODIFY ldap_started_task,DEBUG 0
```

As a final verification of the AES encryption, the admin ID was used to display the Eddie Catu user password entry in the IBM TDS backend. Here is an example of the **ldapsearch** command used to display that data:

```
ldapsearch -h host_ip -D "cn=LDAP Administrator, o=Your Company" -w admin_password
  -b "cn=Eddie Catu,ou=In Flight Systems,ou=Home Town,o=Your Company"  "(objectclass=*)" userpassword
```

The following results were returned:

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

Notice that the userpassword returned was `catu`.

Initially, we expected the userpassword to be returned encrypted. However, *IBM Tivoli Directory Server Administration and Use for z/OS* does indicate that AES encryption is two-way encryption. This means that the data is stored encrypted and it is returned from a query unencrypted.

Using operations monitor

We implemented and tested a new IBM TDS feature for z/OS V1R9 which holds search data and recognizes search patterns. This feature allows IBM TDS to properly monitor and report possible spam or undesirable client traffic from a specific IP address and also provides a way to determine the most executed or longest running searches. The search data will be stored in searchStats or searchIPStats entries according to search patterns. These entries contain attribute values for search rates and other activity tracked for searches. The operations monitor entries will be displayed in LDAP URL format so they can be parsed easily. Operations monitor entries that have matching search characteristics can be grouped together, regardless of IP address, and tracked in searchStats entries, if active. Each matching pattern can also be separately tracked in parallel by client IP address. This allows LDAP users to determine both the most active client and most active search pattern.

Implementing operations monitor

To use this feature, we applied the operations monitor SPE (APAR OA23352) code on z/OS V1R9. We then added two configurations options into the LDAP configure file:

operationsMonitorSize
Specifies the maximum number of entries to store in the operations monitor cache

operationsMonitor
Specifies which type of entries operations monitor should monitor. There are three options:

ip Monitor IP specific search entries only

ipAny
Monitor general IP-ANY entries only

all Monitor both IP and IP-ANY search entries

The subentry **cn=operations,cn=monitor** under **cn=monitor** is used to hold search data. We used the following command to display this subentry:

```
ldapsearch –h host –p port –b "cn=monitor" "objectclass=*"
```

Note that the **/f ldapsrv,display monitor** command will *not* display content of this subentry.

Testing Operations Monitor

For our testing of this new function, we tried all of the **operationsMonitor** values (ip, ipAny, all) and we used various **operationsMonitorSize** values (such as 0, 1, 5, 10, 1000). The value `0` means disable this function. With the various configurations of IBM TDS server, we tested various search functions, such as basic search, referral search, SSL search, search with different attributes, null-base subtree search, basic LDAP searches, and alias searches. We also tested failed search operations monitor as well. (Note that persistent searches are not monitored.) We also set different values of **operationMonitor** and **operationMonitorSize** on our other (more than 10) IBM TDS servers on which we run LDAP workloads every day.

We also designed one special scenario to use this function. We launched several workloads on more than ten test machines to simulate different users doing add, search, and delete entries operations against one LDAP server at the same time. Then the LDAP server recorded all the searching operations from each client. It grouped the same search model as one record. We also distinguished different searching operations from each client by IP address as we configured it with **operationsMonitor all**. When the number of records exceeded the **cachesize** value, it trimmed the records according to record update time. In addition to this workload, we also ran an LDAP Searches workload that executed various searches throughout the LDAP directory while using the new operations monitor configuration options.

The operations monitor output looks like this:

```
searchStats=ldap:///CN=_v??sub?(objectclass=*)?success,numOps=6,avg=646,rate=0,maxRate=2
,maxRateTimeStamp=20080311054410.937269Z,createTimeStamp=20080311053525.815516Z
searchIPStats=ldap://9.12.20.157/O=_v,C=_v?cn?sub?(cn=_v*)?success,numOps=161,avg=10238,
rate=0,maxRate=28,maxRateTimeStamp=20080311054109.918992Z,createTimeStamp=20080311053938.088107Z
currenttimestamp=20080311054702.830981Z
resettimestamp=20080311033711.273132Z
resets=0
numtrimmed=0
entries=30
cachesize=100
```

Chapter 12. Using the Cryptographic Services PKI Services

For z/OS V1R9, there were a few updates for PKI Services that we'd like to highlight.

Automatic certificate renewal

This new PKI ability allows the user to set up certificates for automatic renewal, along with email notification of certificate renewal.

We did the following to enable automatic certificate renewal:

1. Set the ExpireWarningTime variable in the CertPolicy section of pkiserv.config:
   ```
   EXPIREWARNINGTIME = 1W
   ```
2. Copied and customized the renewed certificate notification form. See *z/OS Cryptographic Services PKI Services Guide and Reference* for detailed information on this step.
3. Updated the General Section of pkiserv.config to point to the newly configured form:
   ```
   RENEWCERTFORM=Z0/ETC/PKISERV/RENEWCERTFORM.FORM
   ```
4. We went back to the certificate template that we want to renew and enabled automatic renewal by adding the <AUTORENEW=Y> tag to the template. It is recommended to place this tag after the <NICKNAME> tag in the template:
   ```
   <NICKNAME=1YBSSL>
   <AUTORENEW=Y>
   ```
5. Enabled e-mail notification in the certificate template by changing the original setting for the NotifyEmail tag from:
   ```
   %%NOTIFYEMAIL (OPTIONAL)%%
   ```
   to:
   ```
   %%NOTIFYEMAIL%%
   ```

Automatic certificate renewal was successfully enabled.

RACF/SDBM distinguished name support

PKI Services now allows the configuration of RACF-style distinguished names with IBM Tivoli Directory Server. To verify this we configured PKI Services with the following changes:

From the old distinguished name:
```
CN=LDAP ADMINISTRATOR
```

To the new distinguished name:
```
RACFID=WEBADM, PROFILETYPE=SYSPLEX=UTCPLXJ8,C=US
```

The password field was also updated.

The configuration change was successful and PKI functioned as normal.

Chapter 13. Using System SSL

This topic describes our experiences testing the following System SSL functions while running on z/OS V1R9:

- "System SSL hardware to software notification"
- "Using System SSL CPACF hardware support"
- "Using System SSL 4096-bit hardware support" on page 132

System SSL hardware to software notification

For z/OS V1R9, System SSL has been enhanced to provide information about when an application has switched from using hardware encryption processing to software encryption processing. If a System SSL application encounters an error when using hardware services through ICSF, System SSL automatically switches to software services for that encryption work. In z/OS V1R9, System SSL added notification via messages to the console and system log that this switch has taken place.

For example, when we stopped ICSF while attempting an SSL connection through the IBM HTTP Server, the SSL connection completed successfully but when looking in the log, we saw the following messages:

```
GSK01051E IMWEBZ1/01FD Hardware encryption error. ICSF hardware encryption processing is unavailable.
GSK01052W IMWEBZ1/01FD Hardware encryption error. PKE encryption processing switched to software.
```

Using System SSL CPACF hardware support

We recently tested the exploitation of System SSL using the CP Assist for Cryptographic Function (CPACF). 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 recently tested the exploitation of 4096 bit keys through System SSL on our z/OS V1R9 system running on our z10 EC and z9 EC CPCs. In order to perform this function, you need the following:

- System SSL APAR OA22481
- 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.

Chapter 14. Implementing and using PKCS #11 support

In z/OS V1R9, ICSF provides support for the PKCS #11 standard. PKCS #11 is the standard for the cryptographic token interface. z/OS' implementation of the PKCS #11 tokens are virtual, very similar to RACF (SAF) key rings. z/OS PKCS #11 tokens can be created using system software, such as the gskkyman utility, RACF, or by applications using the C API. ICSF supports PKCS #11 tokens as follows:

- A token data set called a TKDS that serves as the repository for cryptographic keys and certificates used by PKCS #11 applications
- A C application programming interface (API) that supports a subset of the V2.20 level of the PKCS #11 specification
- Token management callable services that are used by the C API

Setting up PKCS #11 support

For setup of PKCS #11 support, we used *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*, SA23-2231. Using this documentation, we performed the following steps:

1. Created TKDS VSAM data set, SYS1.TKDS1, using the sample provided in SYS1.SAMPLIB(CSFTKDS) as the base.
2. Updated the ICSF options dataset to specify the necessary TKDS options. We specified the following to indicate the name of our TKDS data set:

   ```
   TKDSN(SYS1.TKDS1)
   ```

   And the following to indicate that we wanted to share our TKDS across all systems in our sysplex:

   ```
   SYSPLEXTKDS(YES,FAIL(YES))
   ```
3. The CRYPTOZ class is used to control access to the tokens. We chose to setup generic profiles so we activated the CRYPTOZ class as follows:

   ```
   SETROPTS CLASSACT(CRYPTOZ) GENERIC(CRYPTOZ) RACLIST(CRYPTOZ)
   ```
4. We created the USER.* and SO.* resources and gave CONTROL authority for both to the ID creating the tokens:

   ```
   RDEFINE CRYPTOZ SO.* UACC(NONE)
   RDEFINE CRYPTOZ USER.* UACC(NONE)
   PERMIT SO.* CLASS(CRYPTOZ) ID(TOKADMIN) ACC(CONTROL)
   PERMIT USER.* CLASS(CRYPTOZ) ID(TOKADMIN) ACC(CONTROL)
   SETROPTS RACLIST(CRYPTOZ) REFRESH
   ```

   Note that any ID whose applications attempt to access tokens will need READ access to the appropriate resources (SO and USER).
5. We tested to ensure our setup was correct by running the pre-compiled version of testpkcs11.

Using the PKCS #11 support

We tested the PKCS#11 support by using the available applications that support PKCS #11 tokens.

Using gskkyman and the new panels for tokens, we created a token the same way we would create a certificate/key. This was fairly straight forward. For more

information about how to create tokens using the gskkyman interface, see *z/OS Cryptographic Services System SSL Programming*.

We used this token in 2 ways:

- Created an SSL connection on the HTTP servers. In the config file for the HTTP server, we indicated the following for the KeyFile directive:

  ```
  KeyFile *TOKEN*/token.name SAF
  ```

  where *token.name* is the actual name given to the token.
- Created an SSL connection while using FTP. Much like the HTTP server above, we defined the Keyring directive in the FTP config file as follows:

  ```
  Keyring *TOKEN*/token.name SAF
  ```

  where *token.name* is the label given to token upon creation.

You need to ensure that the IDs that the FTP server and HTTP server are running under have READ access to the profiles (USER.* and SO.*) in the CRYPTOZ class; otherwise, things will not work. For example, when starting the HTTP server without the proper authority, the HTTP server will start but you will see the following messages in the log:

```
ICH408I USER(WEBSRV  ) GROUP(IMWEB   ) NAME(####################)  791
  USER.IMWEBZ1.CERTS.JUNE12 CL(CRYPTOZ )
  INSUFFICIENT ACCESS AUTHORITY
  FROM USER.* (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IMW6310E SSL support initialization failed, server will run only in non-secure mode
ICH408I USER(WEBSRV  ) GROUP(IMWEB   ) NAME(####################)  792
  SO.IMWEBZ1.CERTS.JUNE12 CL(CRYPTOZ )
  INSUFFICIENT ACCESS AUTHORITY
  FROM SO.* (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

We also tested that the C API interface worked correctly, using a home grown workload.

Chapter 15. Implementing and using the RACF Java API

z/OS V1R9 introduces a Java API that includes some basic administrative functions that can be used with RACF. This API does not include all RACF functionality nor does it add any new RACF functionality—it takes advantage of what is already possible with LDAP SDBM, an LDAP server with a RACF backend.

In order to test this API, we first needed to implement the infrastructure and then we created a J2EE Web application that would let us manage our RACF users and groups. For simplicity, we called our Web application zRacfAdmin. Figure 70 shows our setup.

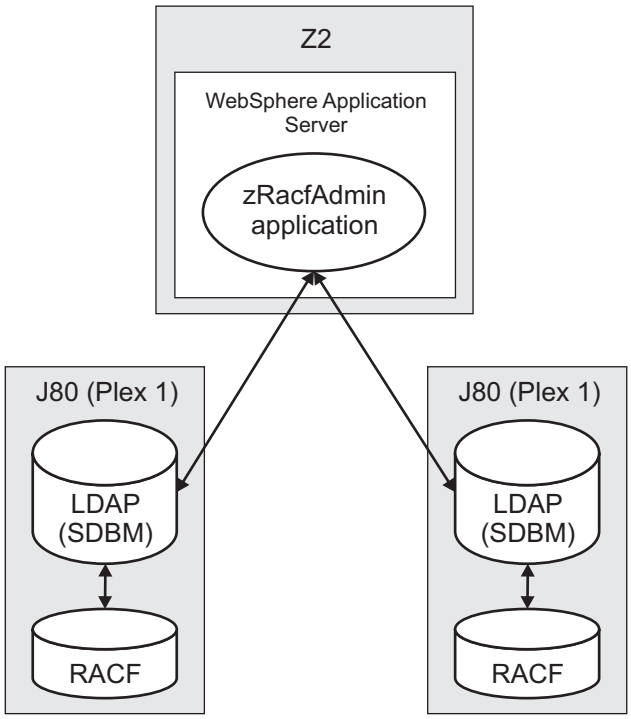


Figure 70. Our setup for testing the RACF Java API

As Figure 70 shows, we were able to manage two different RACF databases in separate sysplexes by writing one J2EE Web application that was able to connect to both.

The type of functionality supported by the Java API includes the following:

- Creating a connection to the LDAP server
- Creating/Editing/Deleting users and user attributes
- Creating/Editing/Deleting groups and group attributes
- Adding/Editing/Removing user-to-group connection attributes

In order to exploit the Java API, you need two jar files, both of which are located in /usr/include/java_classes. The two jar files are RACFuserregistry.jar and userregistry.jar.

The Java API is simple to use for the most part. For example, to retrieve a user object from the RACF database through LDAP, you issue a **getUser()** call. You can then use other function calls to retrieve the attributes associated with the user object.

Figure 71 shows a screen capture from our zRacfAdmin Web application. In this figure, we are viewing the attributes of the WASADM user. We can edit any of those attributes, as well as add the user to or remove the user from various groups.

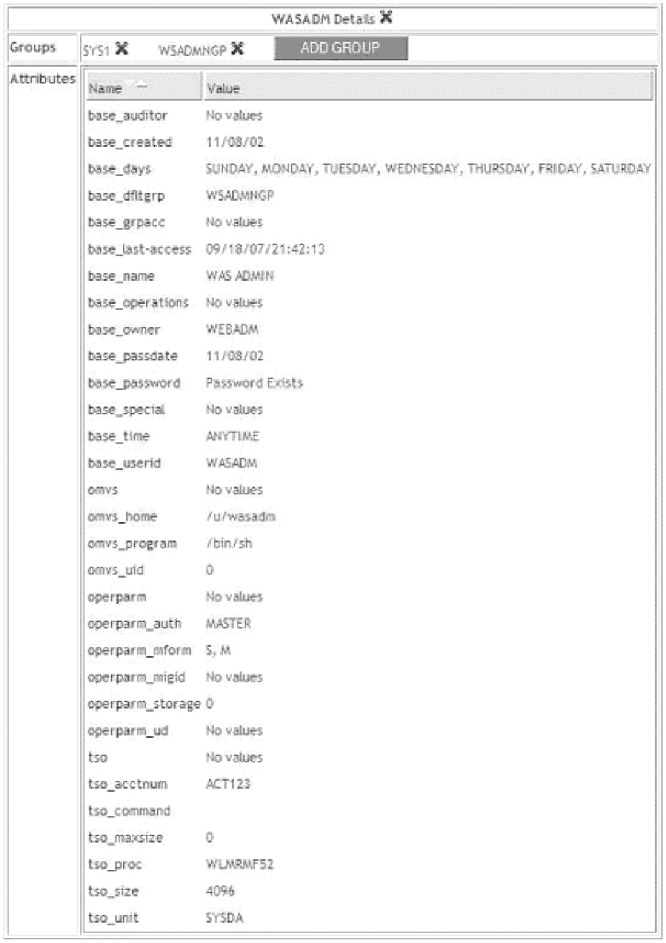


Figure 71. Our zRacfAdmin Web application, attribute display for user WASADM

The Java API does not include any searching functionality for users or groups. However, thanks to Anne Emerick (a colleague in z/OS RACF Development), we were able to use a Java-LDAP API to implement the search functionality into our Web application.

Overall, our experience with the Java API for RACF was a positive one and we also found the API useful for managing RACF databases from homegrown Web applications.

Chapter 16. CICS migration experiences

The following topics describe our CICS migration experiences:

- "Migrating to CICS Transaction Gateway V6.1"
- "CICS experiences with z/OS V1R9"

Migrating to CICS Transaction Gateway V6.1

We migrated our CICS Transaction Gateway (CICS TG) setups from V6.0 to V6.1. Our migration was very simple and straight-forward and our CICS TG continues to run solidly.

The basic setups for CICS TG V6.1 are much the same as for V6.0. See the "Migration to CICS Transaction Gateway V6.0" topic in our December 2005 test report for full details about our CICS TG V6 setups.

Migrate CICS TG daemon to V6.1 first

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 V6.1 before the client side code (such as WebSphere Application Server resource adapters).

Do not mix CICS TG in WebSphere Application Server

Both WebSphere Application Server V6.0 and V6.1 support the resource adapter provided by CICS TG V6.0 or V6.1, so there is greater flexibility here for when you choose to migrate your application server environment. However, you cannot have multiple levels of CICS TG resource adapters within your WebSphere Application Server nodes. Uninstall any existing version of the CICS TG resource adapter prior to installing version 6.1.

CICS TG references

See the following resources for more information about CICS Transaction Gateway:

- CICS Transaction Gateway home page at www.ibm.com/software/htp/cics/ctg/
- Documentation for CICS TG V6.1 at www.ibm.com/software/htp/cics/ctg/library/

CICS experiences with z/OS V1R9

In our test sysplex, we run CICS TS 3.2 on three systems; in our production sysplex, we run CICS TS 3.1 across nine systems.

Following are the experiences we encountered while running CICS on z/OS V1R9.

DFHDUMPX messages during remote dump processing

In the beginning, we first noticed CICS producing numerous DFH* type messages when we were executing our recovery tests. These DFH* type messages happened when local dump processing requested remote dump processing to be initiated. In other words, a single application requested a dump from the local system including dumps from all other systems in the sysplex. Although these DFH* messages are not new to z/OS or CICS, the frequency when it happened and the

numerous messages being produced from each system in the plex was greatly increased. Our investigation found when one system takes a dump the symptom description is produced, but some remote dumps process without the symptom description text. This text is known as the *primary symptom string* and, when missing from remote dumps, CICS reported these messages:

```
DFHDU0213 REMOTE SDUMPX REQUEST FAILED - NO PROBDESC
PARAMETERS SUPPLIED TO DFHDUMPX.
DFHDU0213 REMOTE SDUMPX REQUEST FAILED - NO PROBDESC
PARAMETERS SUPPLIED TO DFHDUMPX.
DFHDU0213 REMOTE SDUMPX REQUEST FAILED - NO PROBDESC
PARAMETERS SUPPLIED TO DFHDUMPX.
DFHDU0215 DFHDUMPX IS ABOUT TO SUPPRESS A REMOTE SDUMPX.
DFHDU0215 DFHDUMPX IS ABOUT TO SUPPRESS A REMOTE SDUMPX.
DFHDU0213 REMOTE SDUMPX REQUEST FAILED - NO PROBDESC
PARAMETERS SUPPLIED TO DFHDUMPX.
```

With so many of these messages appearing in the log, it looks as though CICS is having a serious problem. But the message is validly issued; CICS is at the mercy of any remote dump requests from other components or products that may leave out the optional remote problem description text.

The DFHDUMPX exit module gains control at the z/OS level rather than in the CICS environment, although the message is a bit alarming when it states that a failure occurred. For that reason, the CICS change team opened APAR PK62629. With this APAR the DFHDU0213 message is replaced by DFHDU0218, like this:

```
DFHDU0218 NO PROBDESC PARAMETERS SUPPLIED TO DFHDUMPX
DFHDU0215 DFHDUMPX IS ABOUT TO SUPPRESS A REMOTE SDUMPX.
```

Our testing of the usermod for the APAR showed the above messages are now being produced.

Storage overlay in EWLM exploitation code running VSAM RLS and CICS TS 3.2

We uncovered a storage overlay problem introduced in z/OS V1R9 EWLM exploitation code while running RLS and CICS TS 3.2 . We found the following problems:

Abend0F4 reason code C in IGWLSSLS

Abend0F4 RC0000000C RSN1F031236 IGWLSCCB +2CDE

Abend0F4 RC0000000C RSN1F011236 IGWLSLOP +48E4

Abend0F4 RC00000024 RSN66600259 IGWLNL10 +0B62

Abend0F4 RC00000024 RSN66F32059 IGWLNR10 +0958

We successfully tested z/OS V1R9 WLM APAR OA24021, PTF UA39607, and it resolved all of these problems.

Note: We will discuss our CICS TS 3.1 to CICS TS 3.2 migration experiences in the next edition of our test report.

New SMS diagnostic command for SMSVSAM latch hang conditions

Our RLS experiences found that, in z/OS V1R9, DFSMS introduced a new SMS diagnostic command to help debug SMSVSAM latch hang conditions. This command is retrofitted back to z/OS V1R7.

The command, DISPLAY SMS,SMSVSAM,DIAG(CONTENTION), is described in *z/OS DFSMSdfp Diagnosis* along with the circumstances when you would want to use it.

We decided to use this command as an aid in proactively monitoring our plex. In our environment, we issue this command hourly on every image.

SMSVSAM VSAM RLS sysplex-wide dumping

Another change in z/OS R9 is SMSVSAM VSAM RLS sysplex-wide dumping. When SMSVSAM takes an abend and dumps local storage, other systems in the sysplex that are running SMSVSAM will also take a dump. These dumps are important as they help capture the complete SMSVSAM environment.

We did notice that the remote dumps are missing their descriptive text.

Example: The following is the dump information for the local system:

```
SYS: NM=Z1 2094 02299E   DAT/TIM = 07/06/26 08:48:30
SYM: AB/S00F4 024 IGWLN000 IGWLNR10 IGWFCLRR REGS/0E012
TTL: COMPID=DF122,CSECT=IGWLNR10+0958,DATE=04/08/07,MAI
---> NTID= NONE   ,ABND=0F4,RC=00000024,RSN=66F32059
```

Example: The following is the dump information for the remote system, which is missing the symptom text:

```
SYS: NM=Z2 2064 221526   DAT/TIM = 07/06/26 08:48:46
SYM:
TTL: COMPID=DF122,CSECT=IGWLNL10+0B62,DATE=04/08/07,MAI
---> NTID= NONE   ,ABND=0F4,RC=00000024,RSN=66602059
```

This is not a problem as the symptom text is optional.

Sysplex-wide dumping is described in *z/OS Migration*.

Chapter 17. Migrating to DB2 Version 9.1

This chapter addresses the processes and experiences encountered during the migration of the Integration Test production 3 way DB2® data sharing group DBSG from DB2 Version 8 to Version 9.1 (composed of members DBS1, DBS2, DBS3).

We used the *DB2 Installation Guide, (GC18-9846-00)* for our migration. Whenever we refer to a **Migration Step** in **bold** in our migration discussion, we are referring to the same numbered migration step in the *DB2 Installation Guide*.

Migrating DB2 on z/OS requires common known administration skills on zSeries(z/OS) platform.

Our migration discussion is organized into the following topics:

Migration considerations

Before you migrate to DB2 Version 9, note the following points:

- Migrations to DB2 Version 9 are only supported from subsystems currently running DB2 Version 8; unpredictable results can occur if a migration is attempted from another release of DB2.
- **Migration Step 24** is an optional step that is used to verify the DB2 Version 9 subsystem after it is in compatibility mode. For this step, only the following selected Version 8 IVP jobs can be executed:
  1. Version 8 phase 2 IVP applications
     a. DSNTEJ2A - All steps except the first two
     b. DSNTEJ2C - Only step PH02CS04, statement RUN PROGRAM(DSN8BC3) PLAN(DSN8BH61), is to be executed
     c. DSNTEJ2D - Only step PH02DS03, statement RUN PROGRAM(DSN8BD3) PLAN(DSN8BD61), is to be executed
     d. DSNTEJ2E - Only step PH02ES04, statement RUN PROGRAM(DSN8BE3) PLAN(DSN8BE61), is to be executed
     e. DSNTEJ2F - Only step PH02FS03, statement RUN PROGRAM(DSN8BF3) PLAN(DSN8BF61), is to be executed
     f. DSNTEJ2P - Execute step PH02PS05
  2. Version 8 phase 3 IVP applications
     a. ISPF-CAF applications, with the exception of DSNTEJ3C and DSNTEJ3P.

Note: If you want to run these IVPs as part of the verification of DB2 Version 9 compatibility mode, they must first be run under Version 8 in their entirety before you start the Version 9 migration process and must remain available for use after you complete the migration to Version 9 compatibility mode.

- Examining "Migration Considerations" of the *DB2 Installation Guide, (GC18-9846-00)*, the following items are of particular interest:
  - Global temporary tables require a 32K buffer pool.
  - Declared global temporary tables and static scrollable cursor result tables require a table space with a 32-KB page size because 8-KB and 16-KB page sizes are not supported for table spaces that are created in the work file database.
  - Declared global temporary tables need a 32-KB table space in the work file database.
  - There are changes to the format of the BSDS. To support up to 10000 data sets per copy for archive logs and 93 data sets per copy for active logs, the BSDS must be converted using job DSNTIJUZ
  - The work file database is the only temporary database. The TEMP database is no longer used by DB2
  - If the application uses GROUP ATTACH, then the GROUP ATTACH process is randomized so that all members running on the same z/OS image have an equal chance of getting attach.
  - Changes in the BIND PACKAGES and BIND PLAN defaults changed from CURRENTDATA YES to NO.
- Functions that are no longer supported:
  - Java stored procedures no longer run in resettable JVMs.
  - DB2-established stored procedure address spaces are no longer supported. Stored procedures must be moved to a WLM environment.
  - JBDC/SQLJ Driver for OS/390 and z/OS is no longer supported. All procedures need to be modified to work with the IBM DB2 Driver for JDBC and SQLJ.
  - Simple table spaces are no longer supported. The default is segmented.
- During the migration of the first member of a data sharing group to DB2 Version 9, other members of the data sharing group can be active, although they can experience delays or time-outs when accessing catalog objects as these objects might be locked because of the migration process. Upon completion of the migration process for all data sharing group members, you must update TSO and CAF logon procedures to reference the DB2 Version 9 libraries exclusively.
- The Administrative Task Scheduler (ATS) as currently implemented in DB2 for z/OS is the first piece of tooling infrastructure for our next-gen Web-and Eclipse based tooling.

  The subsystem parameter ADMTPROC, in macro DSN6SPRM, saves the start procedure name of the Admin Scheduler that is associated with the DB2 subsystem. ADMTPROC cannot be updated online. Whenever DB2 starts up, it starts the Admin Scheduler that is specified in ADMTPROC, if it is not up yet. In addition, every time DB2 starts or stops, it posts an event to the Admin Scheduler so that the Admin Scheduler can execute tasks that depend on these events.

  Reference material:
  - *DB2 Installation Guide, (GC18-9846-00)*
  - *DB2 Version 9.1 for z/OS Administration Guide (SC18-9840-00)*

– *DB2 Version 9.1 for z/OS Application Programming and SQL Guide (SC18-9841-00)*

Premigration activities

Before migrating to DB2 Version 9, application of the fallback SPE to all members of the Version 8 data sharing group is necessary.

Also, ensure that the size of the work file database is sufficiently large enough to support the sorting of indexes when migration job DSNTIJTC is run.

After making a backup of the current logon procedure in use, we updated the procedure to reflect the following DB2 Version 9 concatenations before invoking the DB2 installation CLIST:

- DB2.DB2910.SDSNSPFM was concatenated to ISPMLIB.
- DB2.DB2910.SDSNSPFP was concatenated to ISPPLIB.
- DB2.DB2910.SDSNSPFS was concatenated to ISPSLIB.
- DB2.DB2910.SDSNSPFT was not concatenated to ISPTLIB, as DB2 online help was not installed.

After we logged on with the updated logon procedure, we invoked the installation CLIST DSNTINST from the ISPF Command Shell by entering the following command:

```
ex 'DB2.DB2910.SDSNCLST(DSNTINST)'
```

We filled in the first panel DSNTIPA1 as shown in Figure 72 on page 146.

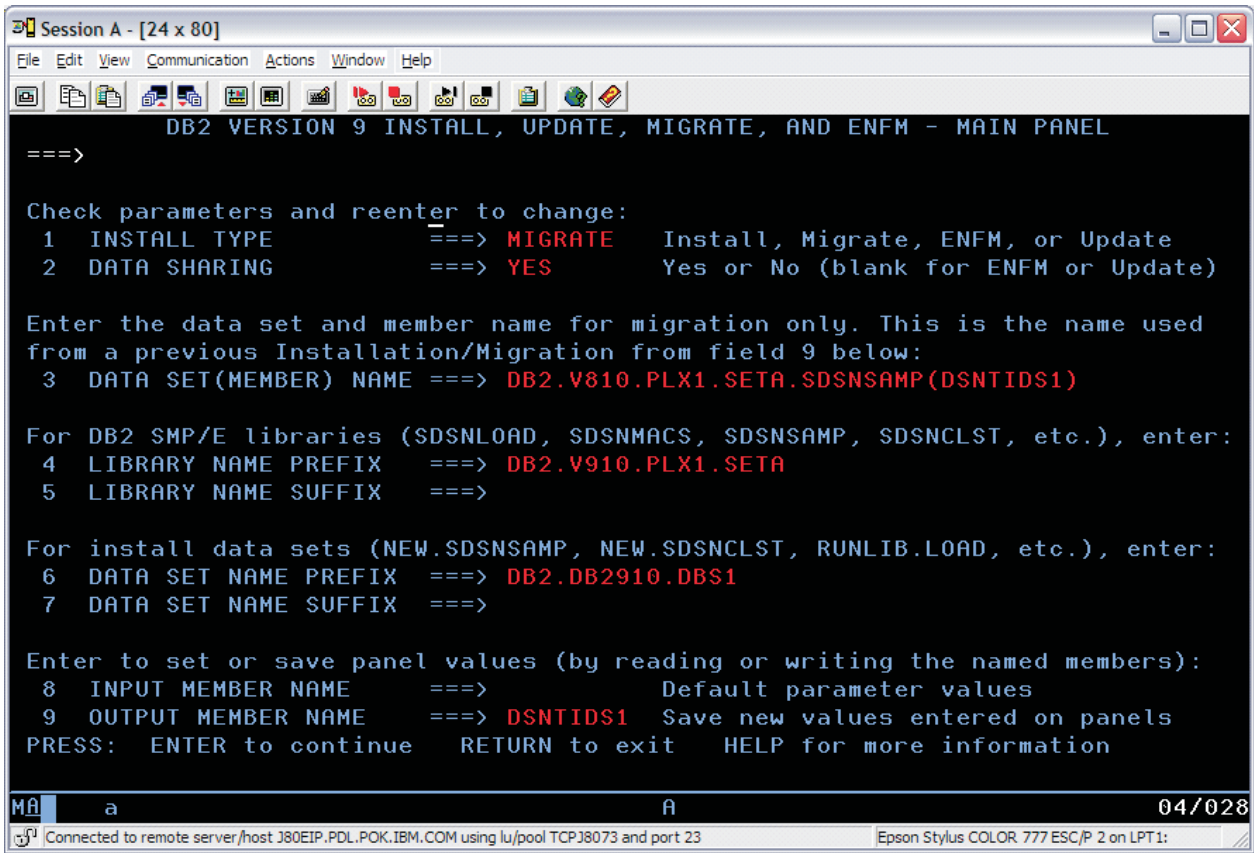


Figure 72. DSNTIPA1

When we pressed enter, the pop-up screen DSNTIPP2 appeared as shown in Figure 73 on page 147.

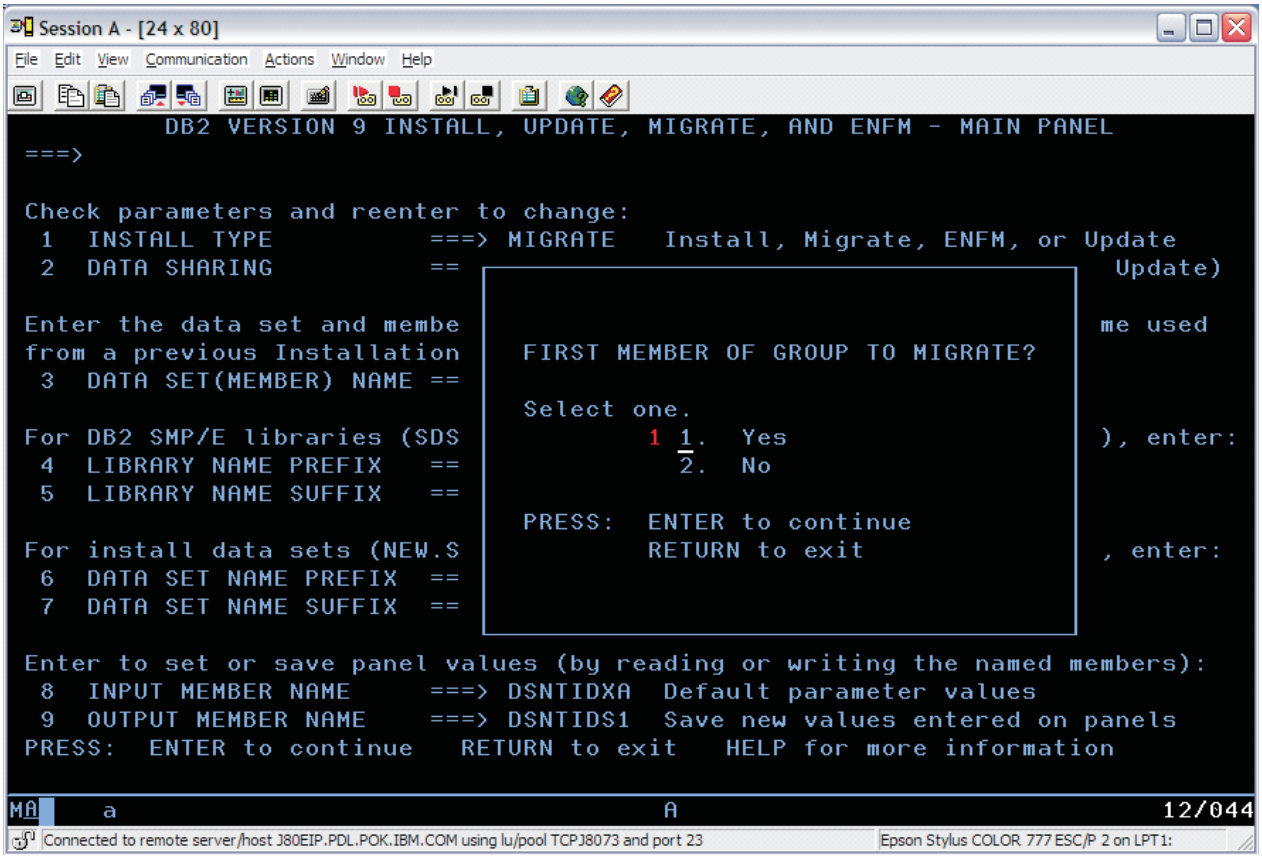


Figure 73. DSNTIPP2

We entered '**1**' to reflect that this was the first member of the data sharing group to be migrated to DB2 Version 9. From this point, we scrolled through the panels and accepted the existing values; upon completion, we placed the tailored JOBS in DB2.DB2910.DBS1.NEW.SDSNSAMP and PROCS in DB2.DB2910.DBS1.NEW.SDSNTEMP as shown in the following:

```
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJMV)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJIN)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJTC)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJTM)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJIC)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJVC)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJSG)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJOS)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJEX)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJGF)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJFT)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJPD)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNTEMP(DSNU)',     CLIST
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNTEMP(DSNH)',     CLIST
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNTEMP(DSNHC)',    CLIST
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNTEMP(DSNEMC01)', CLIST
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJCX)', MIGRATE JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJRI)', INSTALL JCL
IKJ52338I DATA SET 'DB2.V910.PLX1.SETB.SDSNSAMP(DSNTIJRI)' NOT LINE NUMBERED, USING NONUM
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJFV)', FALL BACK JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS1.NEW.SDSNSAMP(DSNTIJUZ)', INSTALL JCL
```

Migrating the first member to compatibility mode

After we reviewed the topics outlined in **Migration Step 1**, we made the following observations:

- Ensured that the IVP jobs and sample database objects for DB2 Version 8 are still available for use. Failure to do so will prevent verifying that a successful migration to DB2 Version 9 compatibility mode has been made.
- Ensured that no utilities are running before migrating to DB2 Version 9. When the migration to Version 9 compatibility mode has been completed, any outstanding utilities that were started under Version 8 cannot be restarted or terminated under Version 9.

Migration Step 2 concerns the optional step of executing DSN1CHKR to verify the integrity of the DB2 directory and catalog table spaces that contain links or hashes. We chose not to run this JOB at this time since we had active applications running on the DB2 V8 members during migration.

Finally, to ensure that there were no STOGROUPs defined with both specific and nonspecific volume ids, we ran the following query:

```
SELECT * FROM SYSIBM.SYSVOLUMES V1
      WHERE VOLID <> '*' AND
      EXISTS (SELECT * FROM SYSIBM.SYSVOLUMES V2
                WHERE V1.SGNAME = V2.SGNAME AND V2.VOLID='*');
```

The query did not return any rows.

Migration Step 3 is an optional step to determine which plans and packages are to be rendered not valid as a result of migrating to DB2 Version 9. To accomplish this, we ran the following queries:

```
SELECT DISTINCT DNAME
                 FROM SYSIBM.SYSPLANDEP
                 WHERE BNAME IN('DSNVVX01','DSNVTH01') AND
                       BCREATOR = 'SYSIBM' AND
                       BTYPE IN ('I','T')
                 ORDER BY DNAME;
               SELECT DISTINCT COLLID, NAME, VERSION
                 FROM SYSIBM.SYSPACKDEP, SYSIBM.SYSPACKAGE
                 WHERE BNAME IN('DSNVVX01','DSNVTH01')
                   AND LOCATION = ' '
                   AND BQUALIFIER = 'SYSIBM'
                   AND BTYPE IN ('I','T')
                   AND COLLID = DCOLLID
                   AND NAME = DNAME
                   AND CONTOKEN = DCONTOKEN
                 ORDER BY COLLID, NAME, VERSION;
```

The first query did not produce any rows, while the second generated the results shown in Figure 74 on page 149.

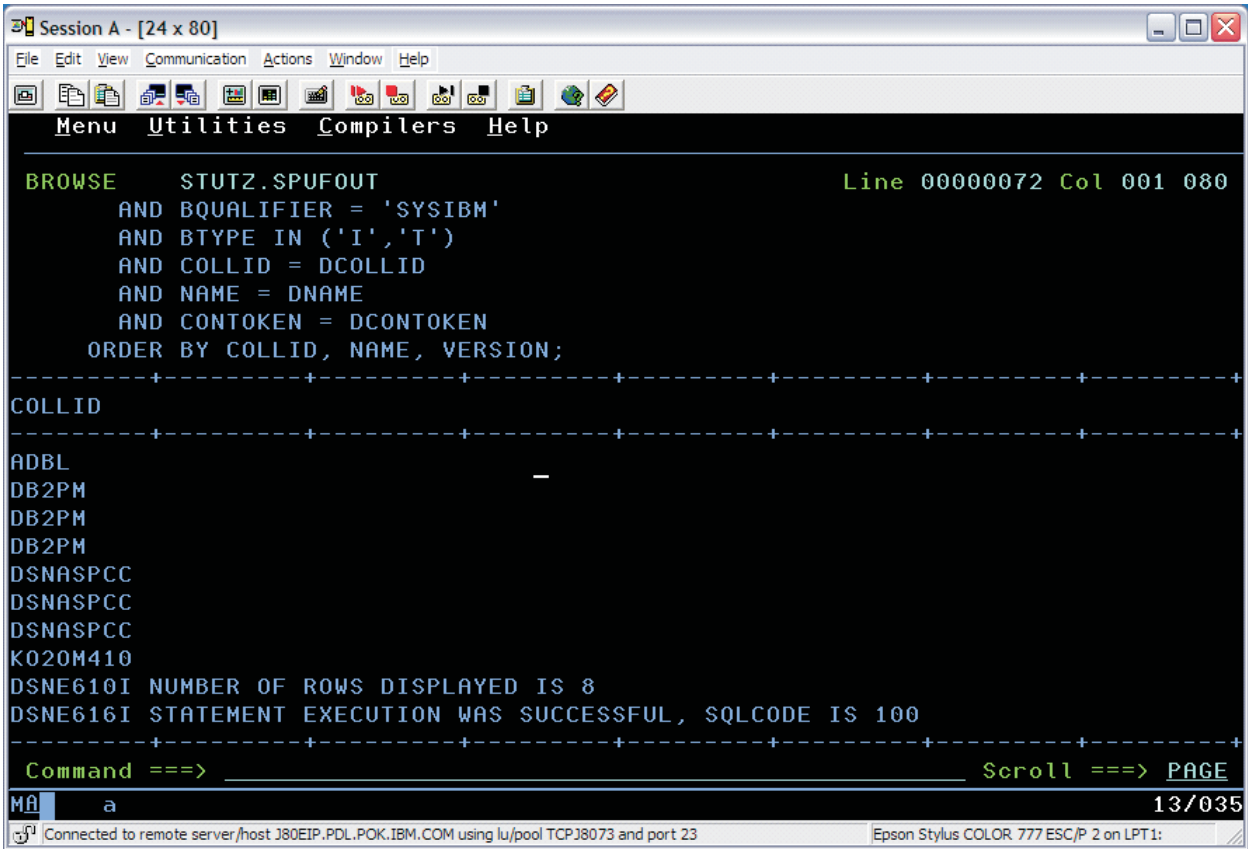


Figure 74. Query output to find packages that will be invalidated when migrating to DB2 Version 9

Migration Step 4 is another optional step to check for consistency between catalog tables through running the queries contained in DB2.DB2910.SDSNSAMP(DSNTESQ). There are a total of 65 queries contained in this data set. We used the data set as input to SPUFI, it ran with no inconsistencies.

Migration Step 5 addresses performing an image copy of the catalog and directory in case of fallback. The *DB2 Installation Guide, (GC18-9846-00)*, recommends using the Version 9 job DSNTIJIC . We followed the recommendation.

Migration Step 6 addresses the following steps necessary to connect DB2 to TSO:

- **Making DB2 load modules available to TSO and batch users -** . Since we run with multiple versions of DB2 V8 and V9 we are using symbolics and extended aliases for TSO and Batch users. SDSNEXIT:

```
DB2.DB2910.DBSG.SDSNEXIT , SDSNLOAD: DB2.DBSG.SDSNLOAD
                  SYMBOLIC:   DB2.&DBSGVER..&DB2PLEX..&DBSGSET..SDSNLOAD
                 EXTENDED ALIAS:
  DEFINE ALIAS (                                          -
  NAME(DB2.DBSG.SDSNLOAD)                                -
  SYMBOLICRELATE(DB2.&DBSGVER..&DB2PLEX..&DBSGSET..SDSNLOAD))
```

- **Making DB2 CLISTs available to TSO and batch users: DSNTIJVC** - Our logon proc DB29PLX1 must again be updated to add DB2.DB2910.NEW.SDSNCLST to the SYSPROC concatenation. We had to do this after we ran the installation job DSNTIJVC (the job that merges tailored CLISTs from prefix.NEW.SDSNTEMP with unchanged CLISTs from prefix.SDSNCLST and places the resulting set of CLISTs in the newly created data set

prefix.NEW.SDSNCLST). Since we currently use fixed-block CLIST libraries (use the SYSPROC concatenation in logon proc DB29PLX1), we had to modify DSNTIJVC as follows:

- Changed the SYSIN DD to DUMMY
- Changed the allocation of prefix.SDSNCLST to match the data control block (DCB) attributes of our other CLIST libraries; this was accomplished by replacing the DCB attributes for DSNTIVB.SYSUT2 with **DCB=*.SYSUT1**.

After DSNTIJVC successfully ran, we updated logon proc DB29PLX1 to add DB2.DB2910.NEW.SDSNCLST to the SYSPROC concatenation.

- **Making panels, messages, and load modules available to ISPF and TSO -** We previously added SDSNSPFP, SDSNSPFM, and SDSNSPFS to the ISPF concatenations. In addition, we updated the logon proc DB29PLX1 to reflect the concatenation of the DB2 English DB2I panels as follows:
  - DB2.DB2910.SDSNPFPE concatenated to ISPPLIB.

Because IMS and CICS connections to DB2 had previously been established, we skipped **Migration Step 7** and **Migration Step 8**.

Migration Step 9 instructs us to stop all DB2 V8 activity or else fallback procedures may fail; prior to stopping data sharing member DBS1, we insured that there were no incomplete utilities (**-DBS1 DISPLAY UTILITY(*)**), and that no databases were in restrict or advisory status (**-DBS1 DISPLAY DATABASE(*) SPACE(*) RESTRICT** and **-DBS1 DISPLAY DATABASE(*) SPACE(*) ADVISORY**, respectively); DBS1 was then brought down.

We skipped optional **Migration Step 10** (**Back Up your DB2 Version 8 volumes**) and performed **Migration Step 11**, which defines DB2 initialization parameters through DSNTIJUZ. After modifying this job by removing the SMP/E step, we submitted it and it ran successfully; expect a return code of 888 if the BSDS has already been converted to the new format.

Special considerations for (Migration Step 11): Step DSBTCNVB converts your BSDS to a New Format. This can be accomplished prior to the migration .We made a decision to convert to the new format prior to the migrations. Following is the DSBTCNVB step:

```
CONVERT THE BSDS TO NEW FORMAT
  NOTE: RC = 888 MEANS BSDS WAS ALREADY CONVERTED
```

As subsystem security had already been established, we skipped **Migration Step 12**.

Migration Step 13 defines DB2 V9 to MVS. We examined job DSNTIJMV to see which modifications to the MVS environment were required; they were implemented accordingly. DSNTIJMV performs the following actions:

- Updates IEFSSNxx, APF, and linklist members, which were deemed not necessary as they had been UPDATED manually RENAME renames the current DB2 procedures in proclib. We skipped this step, however. The DB2 startup procs for DBS1 are renamed manually (see below).
- Step DSNTIPM adds catalogued procedures to proclib; however rather than directing the output of this step to SYS1.PROCLIB, we directed it to a newly created data set, DB2.DB2910.DBSG.PROCLIB.

We renamed the startup procs for DBS1 that reside in PET.PROCLIB (as per the RENAME step of DSNTIJMV). Next, we copied the new V9 startup procs for DBS1 from DB2.DB2910.DBSG.PROCLIB.

For **Migration Step 14**, we successfully ran job DSNTIJIN to define system data sets.

For **Migration Step 15**, we ran the last two steps of job DSNTIJEX to assemble and link edit the access control authorization exit DSNXSXAC and user exit routine DSNACICX (invoked by stored procedure DSNACICS). We skipped the first and second steps that are used to assemble and link edit the signon (DSN3@SGN) and identify (DSN3@ATH) exits because they were not previously implemented.

Because we had previously IPLed the system to pick the V9 early code, we skipped **Migration Step 16**.

Member DBS1 of data sharing group DBSG was then started (**Migration Step 17**) successfully. As the DISPLAY GROUP command shows in the example below, the level of the data sharing group DBSG is now 910 and it is in compatibility mode (MODE(C)); the DB2 level of DBS1 reflects that it is now running DB2 Version 9 code.

```
RESPONSE=J80
 DSN7100I  @DBS1 DSN7GCMD
 *** BEGIN DISPLAY OF GROUP(DSNDBSG ) GROUP LEVEL(910) MODE(C)
                   PROTOCOL LEVEL(2)  GROUP ATTACH NAME(DBSG)
 ------------------------------------------------------------------
 DB2                                    DB2 SYSTEM    IRLM
 MEMBER   ID  SUBSYS CMDPREF   STATUS   LVL NAME      SUBSYS IRLMPROC
 -------- --- ----   --------  -------- --- --------  ----   --------
 DBS1       1 DBS1   @DBS1     ACTIVE   910 J80       IRS1   DBS1IRLM
 DBS2       2 DBS2   @DBS2     ACTIVE   810 JB0       IRS2   DBS2IRLM
 DBS3       4 DBS3   @DBS3     ACTIVE   810 JF0       IRS3   DBS3IRLM
 ------------------------------------------------------------------
```

Migration Step 18. We submitted and ran DSNTIJTC successfully. The job periodically issued message DSNU777I in SYSPRINT to indicate migration progress, as shown in the message DSNU777I which displays CATMAINT progress:

```
DSNU1044I PROCESSING SYSIN AS EBCDIC
DSNU050I            CATMAINT UPDATE
DSNU750I       CATMAINT UPDATE PHASE 1 STARTED
DSNU777I       CATMAINT UPDATE STATUS - VERIFYING CATALOG IS AT CORRECT LEVEL FOR MIGRATION.
DSNU777I       CATMAINT UPDATE STATUS - BEGINNING MIGRATION SQL PROCESSING PHASE.
DSNU777I       CATMAINT UPDATE STATUS - BEGINNING ADDITIONAL CATALOG UPDATES PROCESSING.
DSNU777I       CATMAINT UPDATE STATUS - UPDATING DIRECTORY WITH NEW RELEASE MARKER.
DSNU752I       CATMAINT UPDATE PHASE 1 COMPLETED
DSNU010I       UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

Migration Step 19 is an optional step to ensure that there are no problems with the catalog and directory after running DSNTIJTC. We used the following:

- Ran DSNTIJCX to ensure the integrity of the catalog indexes. The first step produced a return code of 4 as a result of no indexes being found for table space DSNDB06.SYSALTER (these objects will be created during the enabling of New Function Mode). The remaining steps produced a return code of zero.

Indexes can be put into advisory rebuild pending start during migration to DB2 Version 9 when columns are added to the index; DSNTIJRI rebuilds such indexes, and **Migration Step 20** deals with this. DSNTIJRI was executed successfully and we received a return code of 4, the result of several empty indexes.

In **Migration Step 21**, DSNTIJTM was executed to assemble, link-edit, bind, and invoke DSNTIAD. DSNTIJTM ran successfully.

In **Migration Step 22** we ran job DSNTIJSG according to the instructions specified. This step ended as expected.

Special considerations for Migration Step 22:

- In migration mode, job DSNTIJSG does not create any of the objects that are required for XML schema support. You can create these objects only after you have fully migrated to Version 9.
- If you bound special SPUFI packages and plans in Version 8, you need to bind those packages again in Version 9.1. You do not need to bind the plan again. For example, to update special SPUFI packages that were created for use by SPUFI users who require a TSO terminal CCSID of 1047, issue the following commands:

```
BIND PACKAGE(TIAP1047) MEMBER(DSNTIAP) -
     ACTION(REPLACE) ISOLATION(CS) ENCODING(1047) -
     LIBRARY('prefix.SDSNDBRM')
BIND PACKAGE(SPCS1047) MEMBER(DSNESM68) -
     ACTION(REPLACE) ISOLATION(CS) ENCODING(1047) -
     LIBRARY('prefix.SDSNDBRM')
BIND PACKAGE(SPRR1047) MEMBER(DSNESM68) -
     ACTION(REPLACE) ISOLATION(RR) ENCODING(1047) -
     LIBRARY('prefix.SDSNDBRM')
```

- In Version 9.1, SPUFI provides an option to select data with a cursor isolation level of Uncommitted Read. To add a special package and plan with ISO(UR) for SPUFI users who require a TSO terminal of CCSID 1047, issue the following commands:

```
BIND PACKAGE(SPUR1047) MEMBER(DSNESM68) -
     ACTION(REPLACE) ISOLATION(UR) ENCODING(1047) -
     LIBRARY('prefix.SDSNDBRM')
BIND PLAN(SPUR1047) -
     PKLIST(*.SPUR1047.DSNESM68, -
            *.TIAP1047.DSNTIAP) -
     ISOLATION(UR) ENCODING(1047) ACTION(REPLACE)
```

Because some views might have been marked with view regeneration errors during the migration to Version 9 compatibility mode, we performed **Migration Step 23** and identified the views with the following query:

```
SELECT CREATOR,NAME FROM SYSIBM.SYSTABLES
    WHERE TYPE='V' AND STATUS='R' AND TABLESTATUS='V';
```

The query found zero rows. However, if views had been found to have regeneration errors, the following alter command would correct the errors:

```
ALTER VIEW view_name REGENERATE;
```

In **Migration Step 24** we took another image copy of the directory and catalog after they were successfully migrated to V9, and submitted job DSNTIJIC (see **Migration Step 5** on page 149 for details). Execution of DSNTIJIC completed successfully.

The next step verifies the DB2 Version 9 subsystem that is now in Compatibility Mode; only selected Version 8 IVP jobs can be executed as outlined in the DB2 Version 9.1 for z/OS Installation Guide, **Migration Step 25**. After performing the necessary modifications, we ran these IVPs and received the expected results.

Finally, optional **Migration Step 26** deals with enabling WLM stored procedures by either executing the installation CLIST in MIGRATE mode or by editing and executing DSNTIJUZ. Additional information on enabling stored procedures is available in the *DB2 Installation Guide, (GC18-9846-00)*, under Chapter 10 page 361 *"Enabling stored procedures and user defined functions"*. Since we had already enabled WLM stored procedures under DB2 Version 8, this step was skipped.

DB2 V8 and V9 coexistence issues

We allowed the data sharing group to run in coexistence mode for several days while we tested various workloads and products for coexistence issues.

It is recommended that a data sharing group remain in coexistence mode for as brief a time period as necessary.

During this period we did not experience any problems.

Migrating the remaining members to compatibility mode

The next member to migrate in the data sharing group to DB2 Version 9 compatibility mode was DBS2. For us, this was a fairly simple process, which entailed the following steps:

1. Executing the installation CLIST
2. Executing the resultant DSNTIJUZ job
3. Replacing the Version 8 startup procs for the member being upgraded with their Version 9 equivalents. This is performed by executing DSNTIJMV step DSNTIPM
4. Starting the member.

So, beginning with the installation CLIST, we ran DSNTINST from the ISPF Command Shell (ISPF option 6) by entering the following command:

```
ex 'DB2.DB2910.SDSNCLST(DSNTINST)'
```

We filled in the first panel as shown in Figure 75 on page 154.

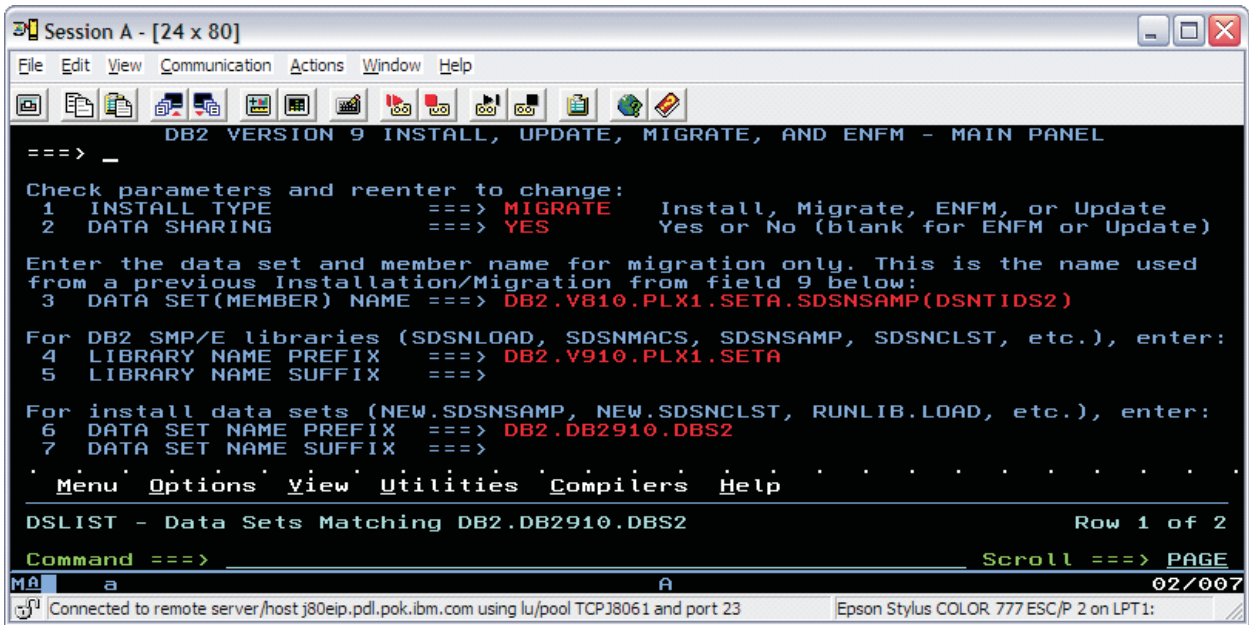


Figure 75. Executing DSNTINST in preparation for migrating the next member of the data sharing group

Pressing enter, we obtained the following pop-up screen as shown in Figure 76.

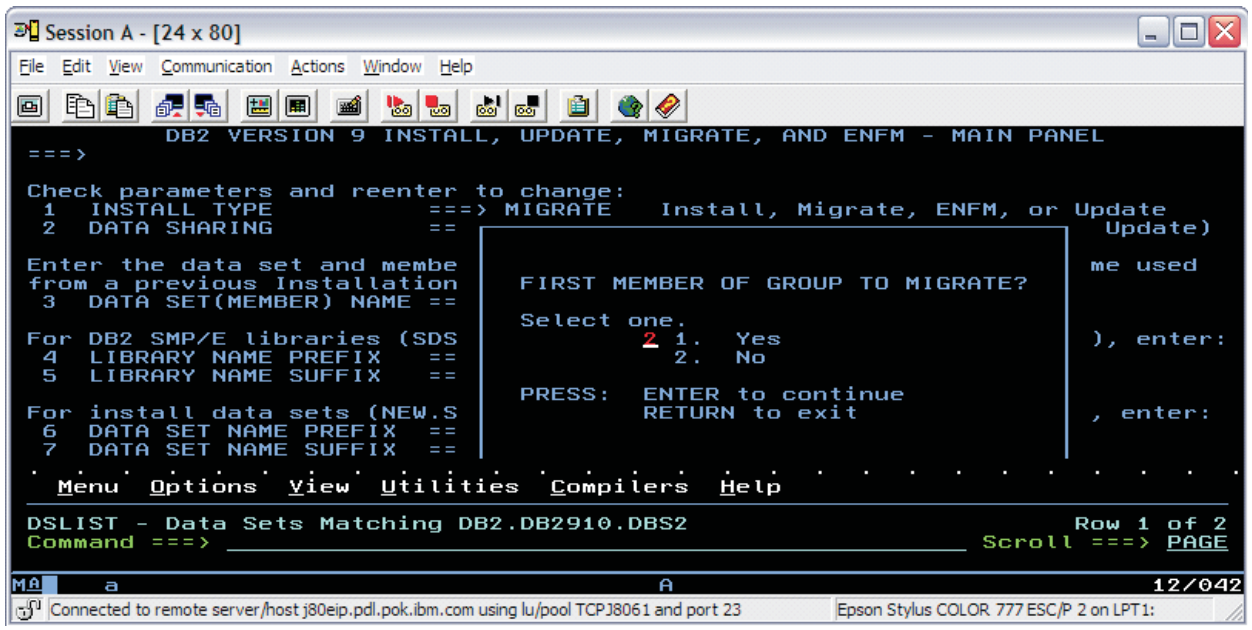


Figure 76. DSNTIPP2 pop-up screen

From this point, we scrolled through the panels and accepted the existing values with the exception of the name of the sample library on panel DSNTIPT. We maintain a separate sample library for each member of the data sharing group, so this field was updated accordingly to reflect DBS2, as shown in Figure 77 on page 155.

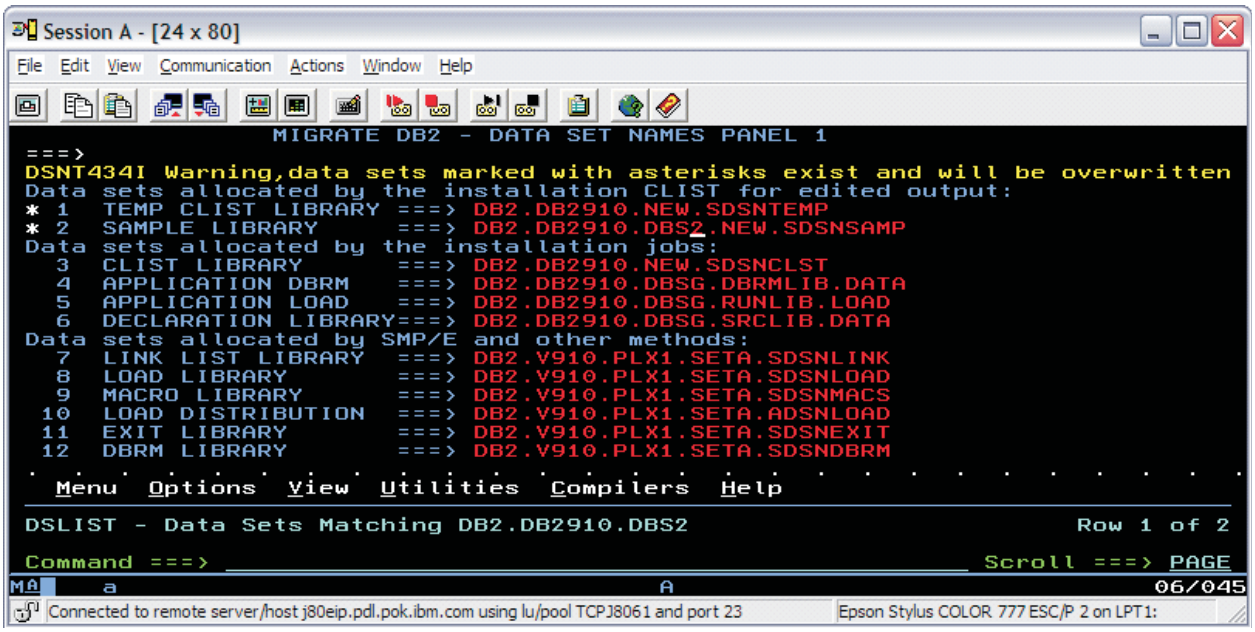


Figure 77. DSNTIPT - Data Set Names Panel 1

We placed the tailored migration JCL in DB2.DB2910.DBS2.NEW.SDSNSAMP as can be seen in the following example:

```
DSNT478I BEGINNING EDITED DATA SET OUTPUT
DSNT489I CLIST EDITING 'DB2.DB2910.DBS2.NEW.SDSNSAMP(DSNTIJMV)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS2.NEW.SDSNSAMP(DSNTIJTM)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS2.NEW.SDSNSAMP(DSNTIJGF)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS2.NEW.SDSNSAMP(DSNTIJFT)', INSTALL JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS2.NEW.SDSNSAMP(DSNTIJFV)', FALL BACK JCL
DSNT489I CLIST EDITING 'DB2.DB2910.DBS2.NEW.SDSNSAMP(DSNTIJUZ)', INSTALL JCL
```

DBS2 was then brought down and DSNTIJUZ was executed after removing the SMP/E step; it ran successfully.

Next, we used steps RENAME and DSNTIPM of job DSNTIJMV to rename the existing Version 8 startup procedures for DBS2 and to add the new Version 8 startup procedures to proclib.

We then started DBS2 successfully in compatibility mode, as can be seen in the following example:

```
RESPONSE=J80
 DSN7100I  @DBS1 DSN7GCMD
 *** BEGIN DISPLAY OF GROUP(DSNDBSG ) GROUP LEVEL(910) MODE(C)
                   PROTOCOL LEVEL(2)  GROUP ATTACH NAME(DBSG)
 ------------------------------------------------------------------------
 DB2                                      DB2 SYSTEM    IRLM
 MEMBER   ID  SUBSYS CMDPREF   STATUS     LVL NAME      SUBSYS IRLMPROC
 -------- --- ---- --------   --------   --- --------  ----   --------
 DBS1       1 DBS1   @DBS1     ACTIVE     910 J80       IRS1   DBS1IRLM
 DBS2       2 DBS2   @DBS2     ACTIVE     910 JB0       IRS2   DBS2IRLM
 DBS3       4 DBS3   @DBS3     ACTIVE     810 JF0       IRS3   DBS3IRLM
 ------------------------------------------------------------------------
```

We followed the same process for the remaining member of the data sharing group, resulting in all members being in compatibility mode as shown below:

```
              RESPONSE=J80
DSN7100I  @DBS1 DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNDBSG ) GROUP LEVEL(910) MODE(C)
                  PROTOCOL LEVEL(2)  GROUP ATTACH NAME(DBSG)
------------------------------------------------------------------------
DB2                                     DB2 SYSTEM    IRLM
MEMBER   ID  SUBSYS CMDPREF   STATUS    LVL NAME      SUBSYS IRLMPROC
-------- --- ----   --------  --------  --- --------  ----   --------
DBS1       1 DBS1   @DBS1     ACTIVE    910 J80       IRS1   DBS1IRLM
DBS2       2 DBS2   @DBS2     ACTIVE    910 JB0       IRS2   DBS2IRLM
DBS3       4 DBS3   @DBS3     ACTIVE    910 JF0       IRS3   DBS3IRLM
------------------------------------------------------------------------
```

Migrating to new function mode

After we migrated all members of the data sharing group to compatibility mode, we had to convert the DB2 catalog to exploit the new functions introduced by DB2 Version 9. The following topics outline the process:

- "Preparing for new function mode"
- "Enabling new function mode" on page 159
- "Running in new function mode" on page 161
- "Verifying the installation using the sample applications" on page 161

Preparing for new function mode

Before enabling-new-function mode, ensure that the following steps are taken:

- **Important:** All members of a data sharing group must have been successfully migrated to Version 9.1 compatibility mode before commencing the enabling-new-function mode process.
- A point of consistency needs to be created for the catalog and directory before enabling new-function mode. The Quiesce Utility should be used to establish a point of consistency for the catalog and directory table spaces; note that DSNDB01.SYSUTILX should be quiesced by itself. Updates to the DB2 catalog and directory should be avoided while in enabling new-function mode. **This is the only time when applications were brought down; planning is therefore essential to reduce the amount of down time.**
- Run the installation CLIST using the ENFM option on panel DSNTIPA1.

After insuring a point of consistency for the catalog and directory , the installation CLIST was executed; panel DSNTIPA1 was completed as shown in Figure 78 on page 157.

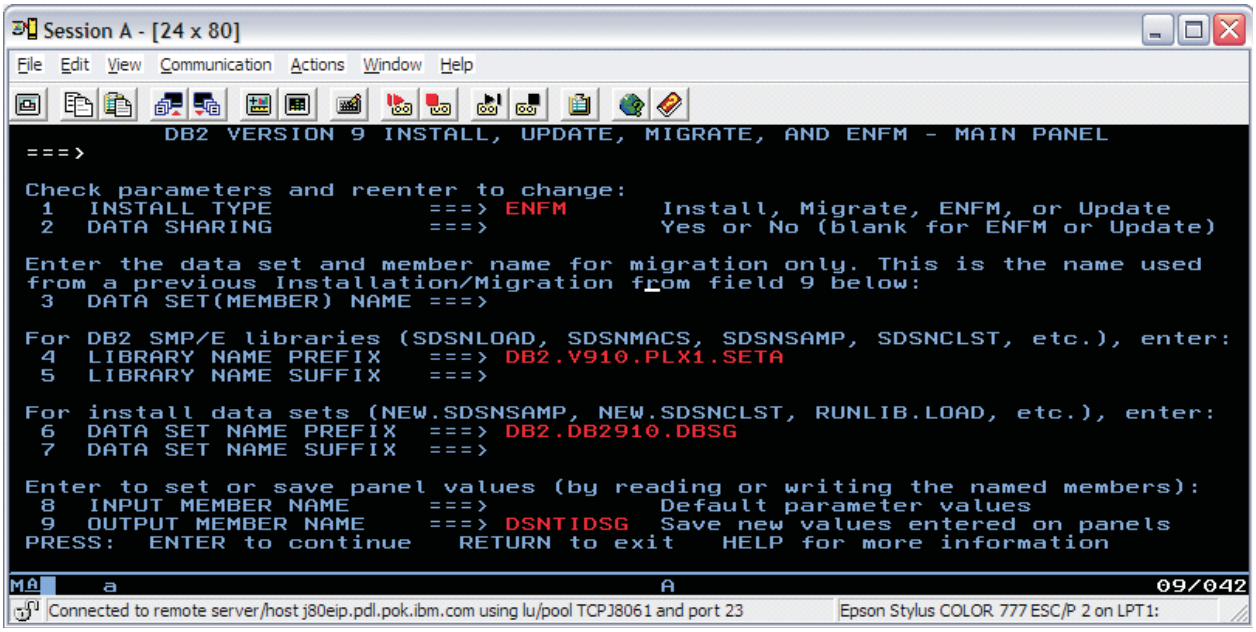


Figure 78. Executing DSNTINST in preparation for enabling-new-function-mode

Press enter twice to display panel DSNTIP00; space was calculated as shown in Figure 79 and Figure 80 on page 158.

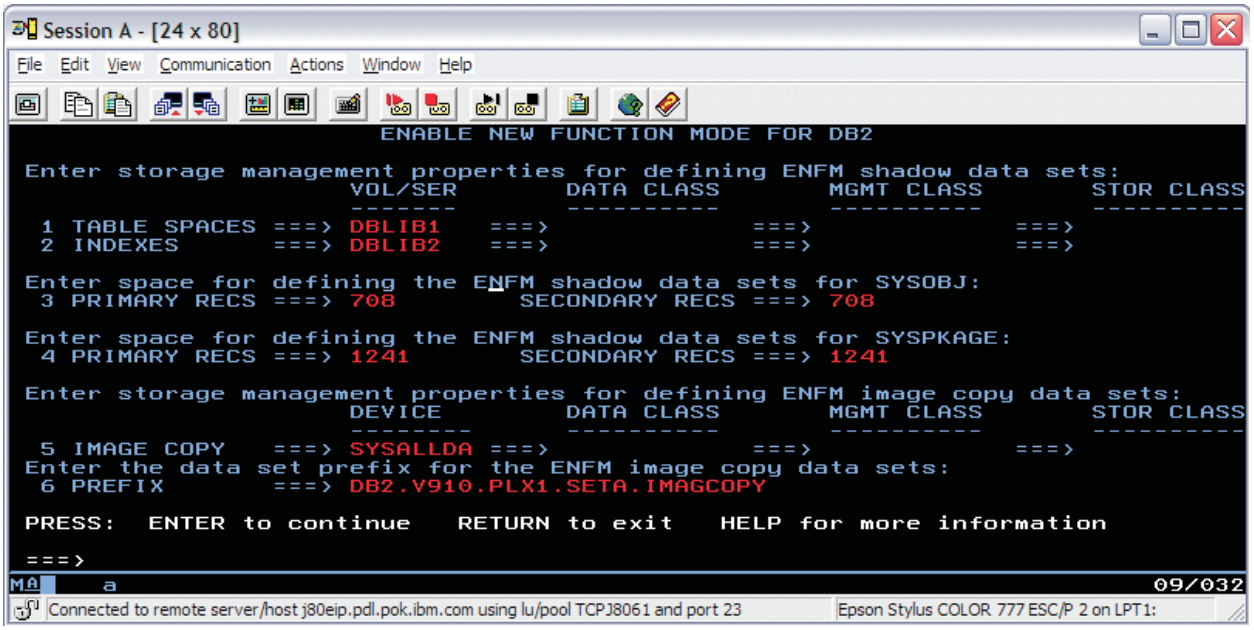


Figure 79. DSNTIP00 first panel

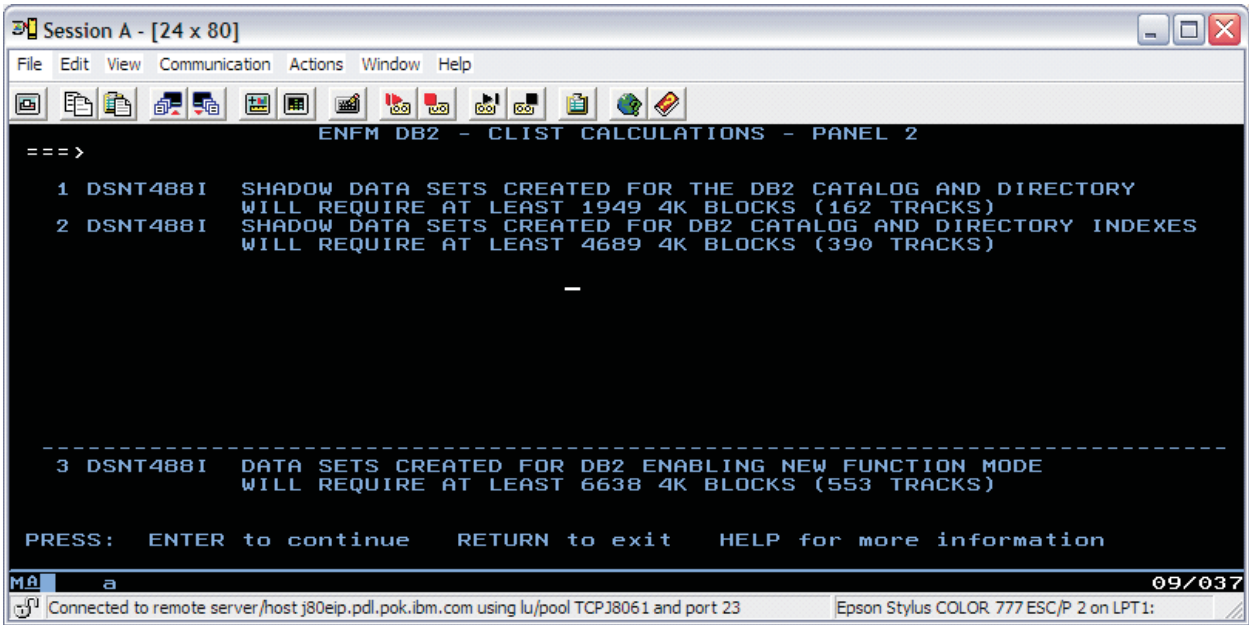


Figure 80. DSNTIP00 second panel

We accepted calculated values and pressed enter to continue.

This was the last panel displayed. When we pressed enter, the generation of the enabling-new-function mode job along with the DB2 Version 9 sample jobs, occurred as shown in the following three screen images:

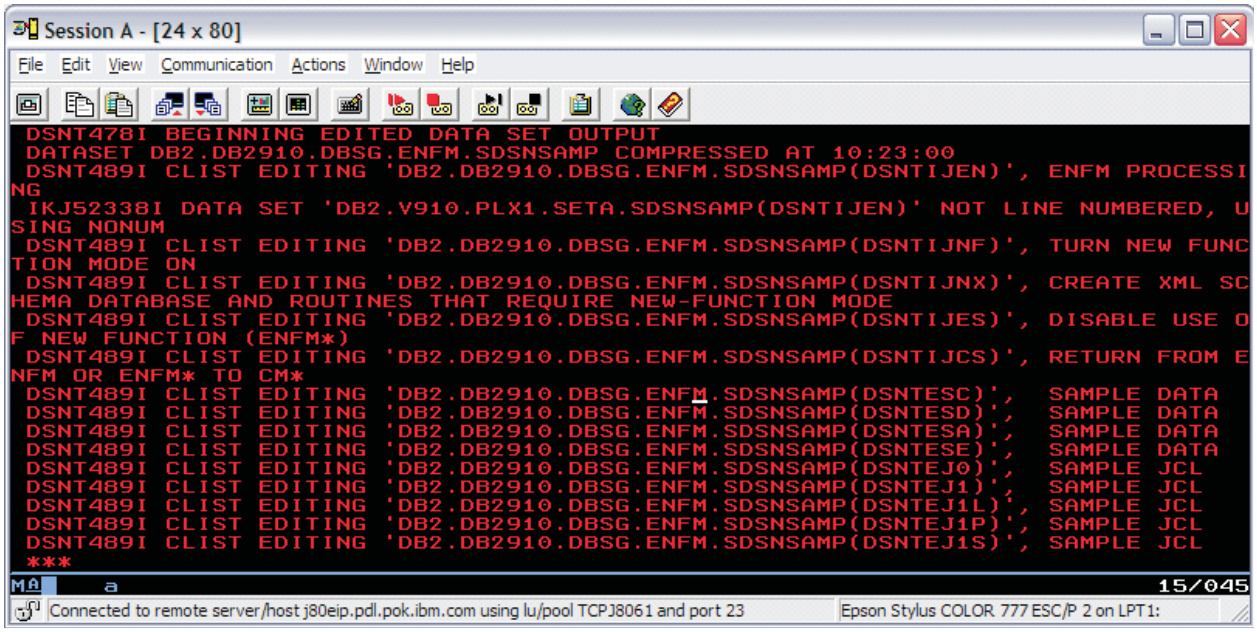


Figure 81. DSNT478I beginning data set output

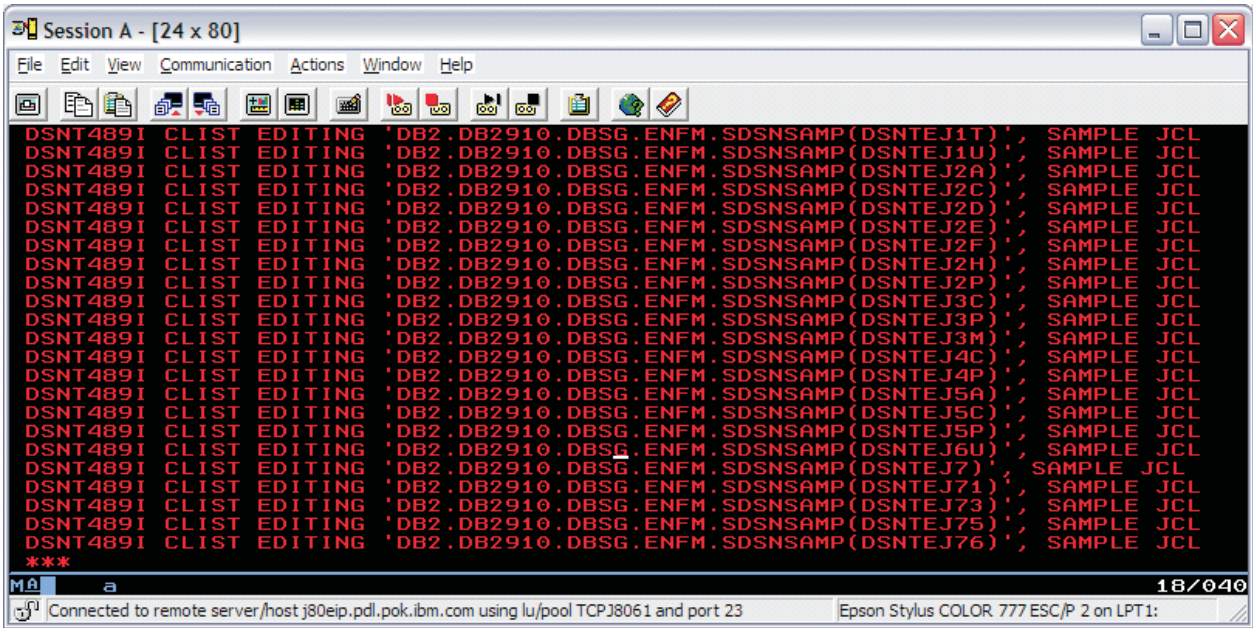


Figure 82. DSNT489I CLIST editing

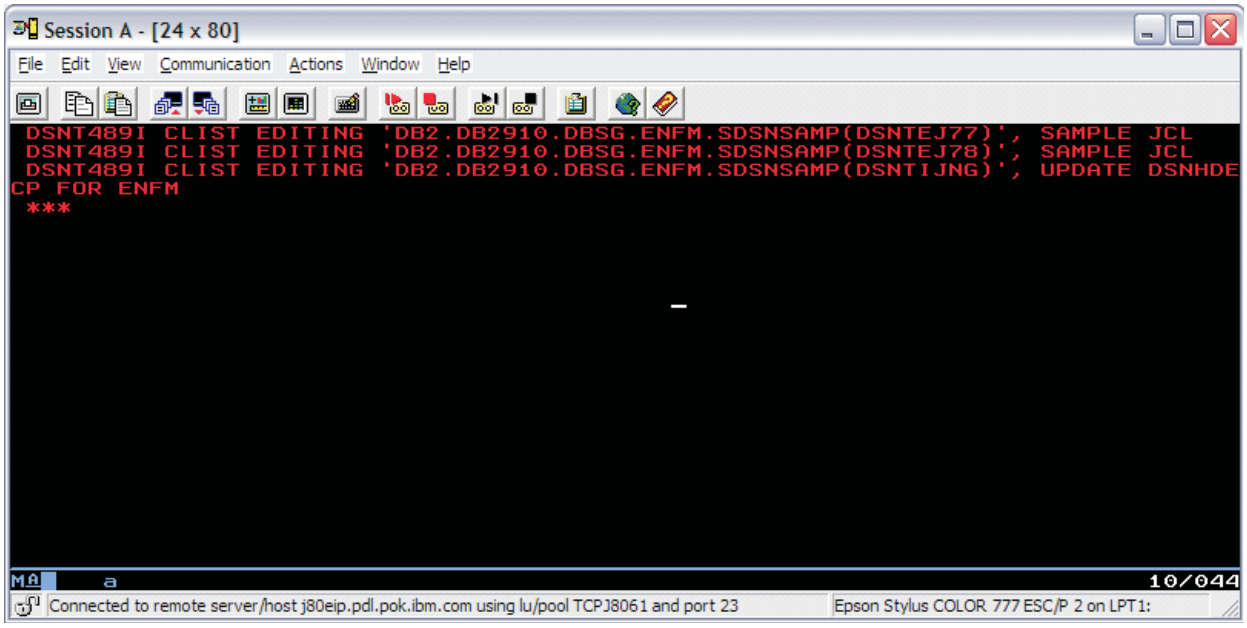


Figure 83. Completion of the preparation before enabling Version 9 new function mode

Enabling new function mode

For **Step 1**, we executed DSNTIJEN and received the messages shown in Figure 27; DSNTIJEN performs the following functions:

- Saves the current RBA or LRSN in the BSDS
- Changes types and lengths of existing catalog columns
- Changes buffer pool for the SYSOBJ table space
- Changes page size of the SYSOBJ table space

- Copies the RTS from the user table spaces to new table spaces in the catalog
- Creates a new index, DSNRTX03, on SYSINDEXSPACESTATS.

The following are the results of the conversion steps:

```
DSNUECM0 - CATENFM START PHASE 1 STARTED
DSNUECM0 - CATENFM START STATUS - VERIFYING CATALOG IS AT CORRECT LEVEL FOR ENFM
DSNUECM0 - CATENFM START PHASE 1 COMPLETED
DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNENFM.ENFM0100
DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNUGUTC -  CATENFM CONVERT INPUT SYSOBJ
DSNUECM0 - CATENFM CONVERT PHASE 1 STARTED
DSNUECM0 - CATENFM CONVERT PHASE 1 COMPLETED
DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0


DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNENFM.ENFM0110
DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNUGUTC -  CATENFM CONVERT INPUT SYSPKAGE
DSNUECM0 - CATENFM CONVERT PHASE 1 STARTED
DSNUECM0 - CATENFM CONVERT PHASE 1 COMPLETED
ALTER COLUMN "SEQNO" SET DATA TYPE
DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0



DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNENFM.ENFM1200
DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNUGUTC -  CATENFM CONVERT INPUT SYSRTSTS
DSNUECM0 - CATENFM CONVERT PHASE 1 STARTED
DSNUECM0 - CATENFM CONVERT PHASE 1 COMPLETED
DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

Step 2 recommends taking an image copy of the catalog and directory at this point.

For **Step 3**, we ran DSNTIJNF which places the DB2 subsystem in new function mode; the job ended with return code zero as shown below:

```
DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNENFM.ENFM9700
DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNUGUTC -  CATENFM COMPLETE
DSNUECM0 - CATENFM COMPLETE PHASE 1 STARTED
DSNUECM0 - CATENFM COMPLETE STATUS - ENTERING NEW FUNCTION MODE (NFM).
DSNUECM0 - CATENFM COMPLETE PHASE 1 COMPLETED
DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

Step 4 is concerned with executing DSNTIJNX, which creates objects for XML Schema Repository (XSR) support. We submitted this job and it ran to completion successfully.

In **Step 5**, DSNTIJNG rebuilds DSNHDECP to specify new function mode as the default by specifying NEWFUN=YES.

Note: If you use more than one DSNHDECP member, modify and update each to use NEWFUN=YES.

To verify the data sharing group was now in new function mode, we issued a DISPLAY GROUP COMMAND; as can be seen in Figure 84 on page 161, MODE(N) has replaced MODE(C):

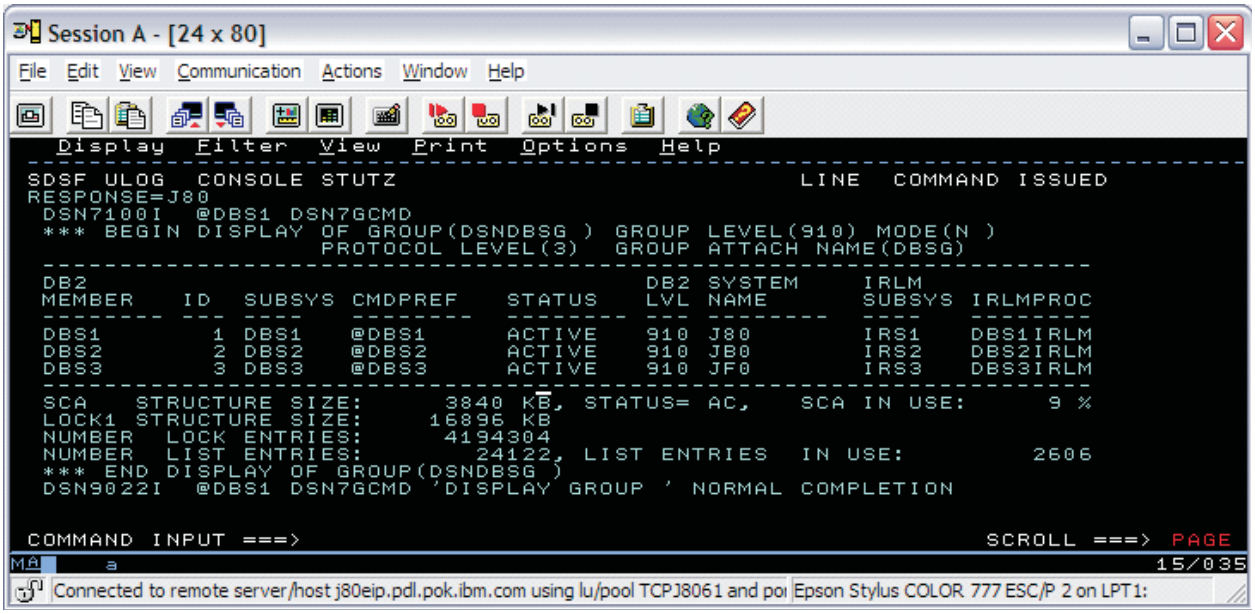


Figure 84. DISPLAY GROUP command showing the data sharing group is now in new function mode

Running in new function mode

Once in new function mode, it is recommended to alter any frequently accessed buffer pools so that their pages are fixed in real storage, thereby avoiding the overhead involved for DB2 to fix and free pages each time an I/O operation is performed. For I/O intensive workloads, this processing time can amount to as much as 10%. To fix pages in storage, the PGFIX parameter of the ALTER BPOOL command is used as shown below:

```
ALTER BPOOL(buffer_pool_name) VPSIZE(virtual_page_size) PGFIX(YES)
```

Note that you should verify that sufficient real storage is available for fixing buffer pool pages before issuing the ALTER BPOOL command.

Verifying the installation using the sample applications

Using the sample applications provided in DB2.DB2910.DBSG.ENFM.SDSNSAMP, we performed verification of DBSG migration to DB2 Version 9 as outlined below. Note that of the seven verification phases available, we ran only those phases and their associated jobs that applied to our specific environment.

Phase 0 is comprised of a single job, DSNTEJ0, that is used to free all objects that were created by running any of the seven verification phases. This permits the verification phases to be executed again in their entirety without the possibility of failure as a result of objects having been previously created.

Phases 1 through **3** are used to test the TSO and batch environments, including user-defined functions.

Phase 4 addresses IMS.

Phase 5 addresses CICS.

Phase 6 initializes sample tables and stored procedures for distributed processing.

Finally, **Phase 7** is used for the testing of DB2's Large Object feature (LOB) using sample tables, data, and programs.

We added the following JCLLIB statement after the JOB statement for all verification jobs that were executed:

```
//         JCLLIB ORDER=DB2.DB2910.DBSG.PROCLIB
```

Recall that in **Migration Step 13** job DSNTIJMV was executed to add catalogued procedures to proclib; however, rather than directing the output of this step to SYS1.PROCLIB, it was directed to the newly created data set DB2.DB2910.DBSG.PROCLIB. This library must be APF authorized (we dynamically added it to the APF authorization list before proceeding).

Planning for verification

Before performing any of the verification phases, you must make certain decisions about your verification strategy. DB2 system administrators and system administrators for ISPF, TSO, batch, IMS, and CICS must be involved in these decisions. Working together, these system administrators do the following:

- Determine the verification phases that you plan to perform. Examine the description of each verification phase in this chapter, and determine which phases apply to your needs.
- Identify any phases that you want to modify before you perform them. Verification is designed to run with little interaction on your part. This chapter does not discuss how to modify any of the phases, but you can adapt any of the seven phases to your needs. If this is your intent, identify and describe any modifications you plan to make.
- Establish additional testing steps to complete the verification. The verification phases and the jobs that you run to perform them are valuable tools for testing DB2. They are not a substitute for a thorough subsystem test. You must plan and perform your own additional testing to complete the verification. To help you assess which additional tests might be necessary, examine the sample applications that are provided with DB2.

We executed the following IVP jobs after every change to the environment (hardware, software).

DSNTEJ1
DSNTEJ1L
DSNTEJ1P
DSNTEJ1S
DSNTEJ2A
DSNTEJ2C
DSNTEJ2D
DSNTEJ2E
DSNTEJ2H
DSNTEJ2P
DSNTEJ3C
DSNTEJ3M
DSNTEJ3P
DSNTEJ6U
DSNTEJ7
DSNTEJ71
DSNTEJ73
DSNTEJ75
DSNTEJ76
DSNTEJ77
DSNTEJ78

You may choose to execute more or less than what we schedule to run. Based on your needs, you may choose to run the IVPs on a different cycle than what we have set up.

Chapter 18. 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 V1R9"
- "z/OS UNIX tools — fsdiruse sample" on page 167
- "Using the _UNIX03 environment variable in the z/OS UNIX shell" on page 168
- "z/OS zFS enhancements in z/OS V1R9" on page 170

z/OS UNIX enhancements in z/OS V1R9

The following topics describe our test experiences with z/OS UNIX enhancements in z/OS V1R9:

- "AUTOMOVE consistency"
- "Unmount of automount file systems" on page 164
- "SMF record type 92 subtype 14 for z/OS file deletion and rename" on page 164
- "z/OS UNIX couple data set BPXOINIT and XCF DISPLAY and message consistency" on page 165

AUTOMOVE consistency

z/OS UNIX improved the way AUTOMOVE settings affect file system movements due to situations such as system failures and shutdowns. Prior to z/OS V1R9, the AUTOMOVE specification was not honored if a file system was mounted in a mode for which the physical file system (PFS) provides *sysplex-aware* capability. The movement of file systems is now more consistent and predictable in these situations.

The levels of sysplex awareness for a PFS are:

sysplex-unaware
Refers to the PFS capability (restriction) that a file system can only be mounted locally on one system in the shared file system configuration. All other systems obtain access to the file system by function shipping operations to the file system server. For example, the HFS PFS is sysplex-unaware for RDWR mounts.

sysplex-aware
Refers to the PFS capability to locally mount a file system on all systems in the shared file system configuration for a particular mount mode, READ or RDWR. For example, the HFS PFS and zFS PFS are sysplex-aware for READ mounts. The zFS PFS can be configured to be sysplex-aware or sysplex-unaware in RDWR mode.

fully sysplex-aware
Refers to the PFS capability to locally mount a file system on all systems in the shared file system configuration for either READ or RDWR mount modes. This is the intended design direction for zFS.

We tested various scenarios that included soft shutdowns, system failures, and PFS termination (zFS). For each test, there were file systems defined to use different AUTOMOVE settings. Our Parallel Sysplex consists of nine active members. The resulting behavior of the settings we tested was exactly as expected. The file

systems that had the AUTOMOVE(YES) attribute had their ownership moved to another system in the sysplex. File systems that had the AUTOMOVE(NO) attribute stayed known to z/OS UNIX; however, they were not accessible since the owning system or PFS was no longer accessible. File systems that were mounted with the UNMOUNT attribute were unmounted, as expected, when the owning system was shutdown or failed.

The final disposition of a file system in the various recovery or shutdown functions is dependent on:

- The AUTOMOVE attribute, as specified at MOUNT time
- The capabilities of the PFS:
  - Whether or not the PFS supports recovering the file system at all on another system. A PFS such as TFS (where the data resides in virtual storage) cannot support moving a file system from one system to another in the various recovery scenarios. This is commonly referred to as a *never move* PFS.
  - Whether or not the PFS provides sysplex-aware capability for the mount mode (RDWR or READ)—thus indicating whether or not the file system is locally mounted.
- Automount managed file systems, which have their own set of behaviors

The above attributes influence the various file systems' recovery and shutdown processing for:

- Member gone recovery (also known as dead system recovery)
- PFS termination
- z/OS UNIX shutdown
- File system shutdown (such as MODIFY BPXOINIT,SHUTDOWN=FILESYS)
- Multiple file system move processing (such as SETOMVS FILESYS,FROMSYS=)

The logic to convert SYSLIST to AUTOMOVE was removed in z/OS V1R9; SYSLIST is now honored for both sysplex-aware and sysplex-unaware file systems. Also, NOAUTOMOVE is now honored for both sysplex-aware and sysplex-unaware file systems.

Unmount of automount file systems

Another change in z/OS UNIX that we tested was the automount facility. Automount was changed to have the UNMOUNT attribute if the parent file system also has the UNMOUNT attribute. Also, automount will not inherit any other AUTOMOVE attributes, such as AUTOMOVE(NO) or SYSLIST, since these are not supported for sysplex-aware file systems. Automount is a sysplex-aware file system. Automount will now do a **getmntent** call for the parent directory to check its AUTOMOVE attributes. If AUTOMOVE(UNMOUNT) is found, it will set that attribute on its **mount** call.

For example, if you set up automount to manage a directory on top of /etc, which has the UNMOUNT attribute, then when the system shuts down or fails for some reason, the automount-managed file system will also be unmounted.

SMF record type 92 subtype 14 for z/OS file deletion and rename

SMF record type 92 reports activity of mounted file systems and files. In z/OS V1R9, z/OS UNIX SMF 92 records were enhanced with a new record subtype 14 to report when a file or directory is deleted or renamed. Information will be collected

in a SMF 92 Subtype 14 record. The installation must set up monitoring for SMF type 92 subtype 14 records in order to collect this information. In a shared file system environment, recording occurs on the user's system where the command was issued.

Some of the operations for which an SMF type 92 subtype 14 record will be written are: **vn_remove** and **v_remove**, **vn_rename** and **v_rename**, and **vn_rmdir** and **v_rmdir**.

For a description of the SMF type 92 subtype 14 record format, see *z/OS MVS System Management Facilities (SMF)*.

z/OS UNIX couple data set BPXOINIT and XCF DISPLAY and message consistency

In z/OS V1R9, the IXC358I message returned by the command DISPLAY XCF,COUPLE,TYPE=BPXMCDS now displays the values for mount entries and automount rules (AMTRULES), as defined in the couple data set (CDS). It also displays the CDS version. The command MODIFY BPXOINIT,FILESYS=DISPLAY will continue to display the values for mounts, automount rules (AMTRULES), and CDS version.

Distributed Byte Range Lock Manager (BRLM) is the only supported byte range locking method since z/OS V1R6. Since the z/OS UNIX couple data set is enabled for distributed BRLM by default, the BRLM references were removed from the F BPXOINIT,FILESYS=DISPLAY display and from some messages. The message BPXF041I was replaced with message BPXF242I, eliminating BRLM information.

The following are display examples. Note the new message ID of BPXF242I for the F BPXOINIT,FILESYS=D display and the elimination of BRLM information. Also, note that message IXC358I, displayed by the D XCF,COUPLE,TYPE=BPXMCDS command, now includes FORMAT DATA for VERSION, MOUNTS, and AMTRULES, under ADDITIONAL INFORMATION.

```
-F BPXOINIT,FILESYS=D
 BPXM027I COMMAND ACCEPTED.
 BPXF242I 2007/06/22 11.46.03 MODIFY BPXOINIT,FILESYS=DISPLAY,GLOBAL
 SYSTEM   LFS VERSION ---STATUS--------------------- RECOMMENDED ACTION
 J80        1.  9.  0 VERIFIED                      NONE
 JF0        1.  8.  0 VERIFIED                      NONE
 JC0        1.  9.  0 VERIFIED                      NONE
 TPN        1.  9.  0 VERIFIED                      NONE
 JB0        1.  9.  0 VERIFIED                      NONE
 J90        1.  8.  0 VERIFIED                      NONE
 JE0        1.  9.  0 VERIFIED                      NONE
 JA0        1.  9.  0 VERIFIED                      NONE
 Z0         1.  9.  0 VERIFIED                      NONE
 CDS VERSION=  2        MIN LFS VERSION=  1.  8.  0
 DEVICE NUMBER OF LAST MOUNT=    8506
 MAXIMUM MOUNT ENTRIES=    1100   MOUNT ENTRIES IN USE=     773
 MAXIMUM AMTRULES=           51   AMTRULES IN USE=            9
MAXSYSTEM=                  16
BPXF040I MODIFY BPXOINIT,FILESYS PROCESSING IS COMPLETE.
```

```
-D XCF,COUPLE,TYPE=BPXMCDS
 IXC358I  11.47.40  DISPLAY XCF 936
 BPXMCDS COUPLE DATA SETS
 PRIMARY    DSN: SYS1.OMVS.CDS10
            VOLSER: CDSOMP     DEVN: 2423
            FORMAT TOD         MAXSYSTEM
```

```
           03/20/2007 13:39:27       16
           ADDITIONAL INFORMATION:
             FORMAT DATA
              VERSION(2)
              MOUNTS(1100)   AMTRULES(51)
 ALTERNATE  DSN: SYS1.OMVS.CDS11
           VOLSER: COUPL4     DEVN: 461A
           FORMAT TOD         MAXSYSTEM
           08/23/2005 10:13:33       16
           ADDITIONAL INFORMATION:
            FORMAT DATA
             VERSION(2)
             MOUNTS(1100)   AMTRULES(51)
```

Message BPXI078I replaces message BPXI050I, eliminating BRLM references (THE VALUE OF DISTBRLM IS 1). This message is issued when the values of the new CDS are greater than the one it replaced. Notice that there are no BRLM references in the following example of message BPXI078I:

```
BPXI078I THE PRIMARY CDS SUPPORTS A LIMIT OF 1200 MOUNTS AND
 A LIMIT OF 52 AUTOMOUNT RULES.  THE CDS VERSION IS 2.
```

If XCF failed the ACOUPLE request, message IXC255I only described the mismatch in terms of internal record names, which did not help the system programmer to figure out how to format the alternate CDS. In z/OS V1R9, additional information for BPXMCDS will be added to message IXC255I to help in determining the error.

For example, if we try to add an alternate CDS (SYS1.OMVS.CDS10) that has values less than the primary CDS for MOUNTS and AMTRULES, we receive the following:

```
-SETXCF COUPLE,ACOUPLE=(SYS1.OMVS.CDS10),TYPE=BPXMCDS

 IXC309I SETXCF COUPLE,ACOUPLE REQUEST FOR BPXMCDS WAS ACCEPTED
 IXC260I ALTERNATE COUPLE DATA SET REQUEST FROM SYSTEM
 J80 FOR BPXMCDS IS NOW BEING PROCESSED.
 DATA SET:  SYS1.OMVS.CDS10
 IXC255I UNABLE TO USE DATA SET
 SYS1.OMVS.CDS10
 AS THE ALTERNATE FOR BPXMCDS:
 ALLOWABLE SIZE OF BPXFSMPT RECORDS IS LESS THAN CURRENT PRIMARY
 RELEVANT BPXMCDS COUPLE DATA SET FORMAT INFORMATION
 PRIMARY
   FORMAT LEVEL:    VERSION(2)
   FORMAT KEYWORDS: MOUNTS(1100)   AMTRULES(51)
 ALTERNATE
  FORMAT LEVEL:    VERSION(2)
  FORMAT KEYWORDS: MOUNTS(1099)   AMTRULES(50)
IXC255I UNABLE TO USE DATA SET
SYS1.OMVS.CDS10
AS THE ALTERNATE FOR BPXMCDS:
ALLOWABLE SIZE OF BPXFSAMT RECORDS IS LESS THAN CURRENT PRIMARY
RELEVANT BPXMCDS COUPLE DATA SET FORMAT INFORMATION
PRIMARY
  FORMAT LEVEL:    VERSION(2)
  FORMAT KEYWORDS: MOUNTS(1100)   AMTRULES(51)
ALTERNATE
  FORMAT LEVEL:    VERSION(2)
  FORMAT KEYWORDS: MOUNTS(1099)   AMTRULES(50)
IXC250I ALTERNATE COUPLE DATA SET REQUEST FAILED FOR DATA SET
SYS1.OMVS.CDS10 FOR BPXMCDS:
CONSISTENCY CHECKING FAILED FOR THE NEW ALTERNATE DATA SET
```

z/OS UNIX tools — fsdiruse sample

It's almost inevitable: File systems fill up. Determining what is using up the most space is often a painful hunt through the file system. When it happens unexpectedly, panic can set in. Complicating the issue, many of our file systems these days have extensive directory structures, have other file systems mounted underneath, or have multiple products, users, or applications using them (such as /tmp and /etc).

As an aid to this perpetual problem, we modified one of the sample programs from the z/OS UNIX tools Web site to help provide us with a means to determine where the usage is within a file system's directories.

Our **fsdiruse** program does the following:

- Displays summary usage of a file system (bytes used, number of files and subdirectories)
- Breakdown of the usage in the first level subdirectories.
- Reports only for the target file system (skips symbolic links and mount points)

Figure 85 shows a sample of the **fsdiruse** output for /Z1/tmp:

```
Bytes used    Dirs      Files       Sub-directory
---------- ---------- ---------- ------------------------------
         0          0          0  /Z1/tmp/IBMRAC
         0          1          0  /Z1/tmp/DB2GWLMJ
         0          1          0  /Z1/tmp/DBXGWLM
     93379          2         36  /Z1/tmp/wwwlogs
         0          0          0  /Z1/tmp/mqsi-CSQ1BRK_servlet_workdir
         0          1          0  /Z1/tmp/mkdir2tmpv1r52
         0          1          0  /Z1/tmp/mkdir2tmpv1r5
    837562          2         36  /Z1/tmp/iwl
         0          2          0  /Z1/tmp/fw
         0          0          1  /Z1/tmp/.cssm
    254325          0          2  /Z1/tmp/logarch
   2689972          6         26  /Z1/tmp/wasusr1
   1563366          0          4  /Z1/tmp/java
         0         14          0  /Z1/tmp/WQ1GWLMJ
         0          0          0  /Z1/tmp/bpxwh2z.WASADM1.temp.zfs
---------- ---------- ---------- ------------------------------
   5438604         45        105   in subdirectories
   3798637                   102   not in subdirectories
   9237241         46        207   total

block spc  char spc   ext links  pipes/fifo  symlinks   sockets
---------- ---------- ---------- ---------- ---------- ----------
         0          1          0         72          0          0
```

Figure 85. Sample output from our fsdiruse tool, run on the /Z1/tmp directory

From the output shown in Figure 85, we can quickly see that there are two subdirectories using most of the space (/Z1/tmp/wasusr1 and /Z1/tmp/java) and are good candidates to warrant a closer look.

Since /Z1/tmp/wasusr1 has six directories, **fsdiruse** can be run again with the /Z1/tmp/wasusr1 directory as its parameter to show the breakdown within those directories.

Downloading, compiling, and running fsdiruse

The source code for **fsdiruse** can be found and downloaded from the Examples section of our Web site at www.ibm.com/servers/eserver/zseries/zos/integtst/.

To compile, ftp this file to a directory of your choice and run **make fsdiruse**. Defaults for **make** should generally be sufficient to compile.

To run, enter:

```
fsdiruse directory
```

where *directory* is the target directory to scan.

The **fsdiruse** program will only report from the requested directory and lower. If you want to see the usage of the entire file system, you will need to point it to the mount point for that file system (for instance, /Z1/tmp versus /Z1/tmp/java).

We placed a copy of the code in the /usrbin directory on both of our sysplexes. Users simply need to add /usrbin to their PATH environment variable to use it (along with other tools) or they can point to it directly.

z/OS UNIX tools

Our **fsdiruse** tool is modified from the **dirsize** tool, available from the z/OS UNIX Tools and toys Web site at www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1toy.html. This site contains many sample programs that you might find helpful, either "as is" or, as we did, modify them to suit our needs. Many can be added to your own toolbox!

Using the _UNIX03 environment variable in the z/OS UNIX shell

The UNIX 03 Product Standard is the mark for systems conforming to Version 3 of the Single UNIX Specification (SUS V3). It is a significantly enhanced version of the UNIX 98 Product Standard. For more information on this standard, go to The Open Group Web site (www.unix.org).

In z/OS V1R8, some z/OS UNIX utilities implemented support for the UNIX 03 specification. _UNIX03 is an environment variable. When _UNIX03 is set to `YES`, the utilities that have implemented support for the UNIX 03 specification will conform to it. Note that this variable is only needed when the syntax or behavior of the new implementation conforming to UNIX 03 conflicts with the existing implementation.

The following are two utilities that support the UNIX 03 specification:

- **cp**
- **mv**

cp utility

In z/OS V1R8, the OMVS shell utility **cp** has three new options (**-H**, **-L**, **-P**) to handle symbolic link processing during a recursive copy (**-R** or **-r** option flags). However, there was already an existing **-P** option for the **cp** utility. It was used for specifying the parameters needed to create a sequential data set. To resolve this conflict, use the _UNIX03 environment variable to specify whether **cp** is to process **-P** for symbolic links handling or **-P** for sequential data set creation. If _UNIX03 is set to `YES`, **cp** will process **-P** for symbolic links handling. If _UNIX03 is set to anything other than `YES`, **cp** will process **-P** for creating a sequential data set.

Another new option for the **cp** utility is **-W**. It works the same way as today's **-P** option. It is provided so that you can create sequential data sets while the _UNIX03 environment variable is set to YES as well.

Here are what the three new **cp** options do:

-H When the **-H** option is specified, **cp** follows symbolic links specified as a source operand on the command line. Following a symbolic link means that an exact copy of the file that is linked will be created rather than a copy of the symbolic link itself.

-L When the **-L** option is specified, **cp** behaves the same way it does when **-H** is specified. However, it also follows the symbolic links that are found during tree traversal.

-P When the **-P** option is specified, **cp** does not follow any symbolic links.

Another new option for the cp utility is:

-W

-W works the same way as today's **-P** option. It is provided so that users can create sequential data sets while the _UNIX03 environment variable is set to YES.

Examples of z/OS UNIX utilities that implement support for the UNIX 03 specification

Set the _UNIX03 environment variable to YES.

```
export _UNIX03=YES
```

Recursively copy directory dir1 to dir2. Use the **-P** option so that no symbolic links are followed.

```
cp –r –P dir1 dir2
```

Set the _UNIX03 environment variable to anything other than YES.

```
export _UNIX03=NO
```

Next, use the **-P** option to specify the parameters needed to create a sequential data set. The command below will copy file1 into a new sequential data set named uss.test0.

```
cp -P "RECFM=U,space=(5,1)" file1 "//'uss.test0'"
```

Leave the _UNIX03 option set to anything other than YES. Use the **-W** option to create a sequential data set called uss.test1.

```
cp -W "seqparms='RECFM=U,space=(5,1)'" file1 "//'uss.test1'"
```

Set the _UNIX03 environment variable to YES. Use the **-W** option to create a sequential data set called uss.test2. The **cp -W** command behaves the same regardless of the value of the _UNIX03 variable.

```
cp -W "seqparms='RECFM=U,space=(5,1)'" file1 "//'uss.test2'"
```

mv utility

In z/OS V1R8, the z/OS UNIX utility **mv** has a new option as well (**-W**). It serves the exact same purpose as the existing **mv** option **-P**. It is implemented purely for consistency between the **cp** and **mv** utilities. Since the **mv** utility does not have any option conflict issues, the _UNIX03 environment variable does not need to be set to YES for **mv** to process the **-W** option.

z/OS zFS enhancements in z/OS V1R9

The following topics describe our experiences with the Distributed File Service zFS enhancements in z/OS V1R9:

- "zFS format authorization"
- "Aggregate full message from zFS"
- "zFS AUDITID" on page 171
- "zFS read-only mount recovery" on page 172

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.PFSCTL 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:

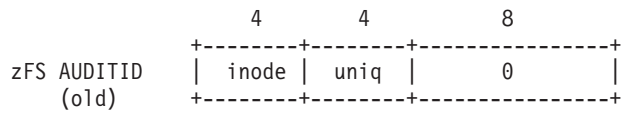
```
IOEZ00551I Aggregate AggrName ran out of space.
```

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.

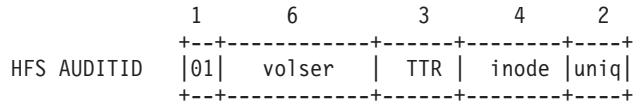
zFS AUDITID

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.

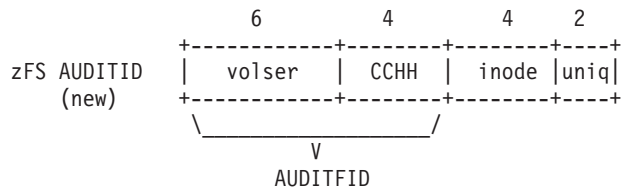
This is how the zFS AUDITFID is determined today:



This is how the HFS AUDITFID is determined today:



Here is how the zFS AUDITFID is determined after the new support:



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:

```
zsadm 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 EF*xx*6271, 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.

```
BPXF002I 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 BPXPRM*xx* 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 BPXF002I 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 19. 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"
- "Running WebSphere MQ implemented shared channels in a distributed-queuing management environment" on page 176
- "Enabling WebSphere MQ Security" on page 179
- "Migrating to WebSphere Message Broker Version 6" on page 181
- "Enabling higher availability for WebSphere MQ" on page 185
- "MQCICS — WebSphere MQ-CICS adapter/bridge workload" on page 185

Using WebSphere MQ shared queues and coupling facility structures

Using Websphere MQ, programs can talk to each other across a network of unlike components, including processors, operating systems, subsystems, and communication protocols, using a simple and consistent application programming interface.

We migrated our WebSphere for z/OS queue managers from V5.3.1 to V6.0. 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/.
- *WebSphere MQ in a z/OS Parallel Sysplex Environment*, SG24-6864, available from IBM Redbooks at www.ibm.com/redbooks/
- *WebSphere MQ Queue Sharing Group in a Parallel Sysplex Environment*, REDP-3636, available from IBM Redbooks at www.ibm.com/redbooks/

Our queue sharing group configuration

We currently have two queue sharing groups: one with three members and another with four 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 four-member production queue sharing group. All of the queue managers in the group run WebSphere MQ for z/OS Version 6.0.

Managing your z/OS queue managers using WebSphere MQ V6 Explorer

Websphere MQ V6 now offers an extensible Eclipse-based graphical configuration tool which replaces the Windows-based MQ Explorer. The Websphere MQ V6 Explorer is supported on both Windows® and Linux operating systems.

This tool, in conjunction with SupportPac™ MO71, has provided us with the ability to monitor as well as perform remote administration and configuration of our entire MQ network. The queue manager being managed does not have to be running WebSphere MQ V6 except when the queue manager is running on z/OS. If you wish to manage your z/OS queue managers using WebSphere MQ V6 Explorer and security is enabled on these queue managers you be will required to install refresh pack 6.0.1.1 or higher. This is because userids on z/OS are validated by RACF security and should be in uppercase. Without refresh pack 6.0.1.1, WebSphere MQ V6 Explorer transmits the userid to the queue manager in lowercase and subsequently the connections are rejected by RACF.

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 273 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 preflist, as expected. Throughout the test, the CICS bridge workload continued to run without any errors.

CF structure rebuild on the alternate coupling facility: With system-managed CF structure duplexing active and a shared queue workload running, we issued the SETXCF STOP,REBUILD command to cause XCF to move the MQ structures to the alternate coupling facility. The queue manager produced no errors and the application continued without any interruption.

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 00C51*nnn*. Many of these are actually coupling facility problems or conditions that result in MQ having a problem and are not necessarily MQ problems in their own right. When such abends occur, we suggest that you analyze the system log for any IXC or IXL messages that might indicate a problem with a coupling facility.

Intra-group queuing: We have all members of the queue sharing group set up for intra-group 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. 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 00F5*nnnn*.

Notes about MQ coupling facility structure sizes:

- All of our MQ coupling facility structures are defined to allow automatic alter (by specifying ALLOWAUTOALT(YES) in the structure definitions in the CFRM policy), whereby XCF can dynamically change the size of a structure, as necessary. This is beneficial because it allows XCF to automatically increase the size of a message structure as needed to hold more messages.
- When we first defined the CSQ_ADMIN structure, we made it 10000K bytes in size. Our original sizing was based on the guidelines in *WebSphere MQ for z/OS Concepts and Planning Guide*, GC34-6051. However, we have since migrated to a higher CFCC level and increased the number of queue managers in the queue sharing group, which increases the size requirement for the CSQ_ADMIN structure. As a result, the queue manager recently failed to start because the CSQ_ADMIN structure was too small and issued the following message:

  ```
  CSQE022E !MQJA0 Structure CSQ_ADMIN unusable, size is too small
  ```

  We used the SETXCF START,ALTER command to increase the size of the structure. The following is an example of the command we issued:

  ```
  SETXCF START,ALTER,STRNAME=MQGPCSQ_ADMIN,SIZE=16000
  ```

  Accordingly, we also increased the value of INITSIZE(24064) and MINSIZE(18668) for CSQ_ADMIN in the CFRM policy to accommodate the increase in usage.

Running WebSphere MQ implemented shared channels in a distributed-queuing management environment

We implemented shared channels within the larger of our two queue sharing groups to bolster our distributed-queuing management (DQM) environment. Previously, we have had a DQM workload that exercised distributed messaging using MQ channels that provided an environment to test channel functionality such as SSL, as well as more general testing such as load stress. We modified the underlying DQM environment to utilize both shared inbound and shared outbound channels without having to change the workload application. We are

now able to handle higher amounts of inbound messages from remote MQ clients and, at the same time. provide transparent failover redundancy for those inbound messages.

Our MQ ″clients″ are in fact full MQ servers on distributed platforms such as Linux and Windows 2000.

Our shared channel configuration

The following sections describe the configuration of our shared inbound and outbound channels. We used information in *WebSphere MQ Intercommunication*, SC34-6059, to plan our configuration.

Shared inbound channels

We decided to implement the shared channel environment on our sysplex using TCP/IP services because our distributed DQM clients are mainly TCP/IP clients. All queue managers in the queue sharing group were configured to start group listeners on the same TCP port (1415), as described in the MQ intercommunication guide.

Example: The following is an example of the command to start group listeners on TCP port 1415:

```
START LISTENER INDISP(GROUP) PORT(1415)
```

The MQ intercommunication guide describes how the group listener port maps to a generic interface that allows the queue sharing group to be seen as a single network entity. For our DQM environment, we configure the Sysplex Distributor service of z/OS Communications Server to serve as the TCP/IP generic interface. This is a slight departure from the intercommunication guide, which utilizes DNS/WLM to provide the TCP/IP generic interface. VTAM® generic resources is another available service that can provide the generic interface for channels defined using LU6.2 connections.

Example: The following is an example of our Sysplex Distributor definition for TCP port 1415:

```
VIPADYNAMIC
VIPADEFINE MOVEABLE IMMED 255.255.255.0  192.168.32.30
VIPADISTRIBUTE DEFINE 192.168.32.30 PORT 1415
DESTIP 192.168.49.30 192.168.49.32 192.168.49.33 192.168.49.38
ENDVIPADYNAMIC
```

We added this definition to the TCP/IP profile of one of our queue sharing groups (in this case 192.168.49.32), but it can be added to any TCP/IP host within the sysplex in which the queue sharing group resides. The IP addresses listed for DESTIP are the XCF addresses of the queue managers in our queue sharing group. The remote client can then specify 192.168.32.30 (or, correspondingly, the host name MQGP.PDL.POK.IBM.COM, which maps to the IP address in our DNS server for our 192.168.xx.xx LAN) on its sender channel, which then causes the receiver channel start to be load-balanced using the WLM mechanisms of Sysplex Distributor.

Example: The following is an example of our definitions for the remote sender channel and the local receiver channel:

```
DEFINE CHANNEL(DQMSSL.CSQ9.TO.MQGP) +
REPLACE +
CHLTYPE(SDR) +
XMITQ(DQMMQGP.QSG.XMITQ) +
TRPTYPE(TCP) +
```

```
DISCINT(10) +
CONNAME('MQGP.PDL.POK.IBM.COM(1415)') +
SSLCIPH(TRIPLE_DES_SHA_US) +
DESCR('DQM SDR CHANNEL TO SHARED RCVR CHANNEL ON MQGP')

DEFINE CHANNEL(DQMSSL.CSQ9.TO.MQGP) +
REPLACE +
CHLTYPE(RCVR) +
QSGDISP(GROUP) +
TRPTYPE(TCP) +
SSLCAUTH(REQUIRED) +
SSLCIPH(TRIPLE_DES_SHA_US) +
DESCR('SHARED RCVR CHANNEL FROM J90 FOR DQM')
```

Note that QSGDISP(GROUP) specifies that a copy of this channel is defined on each queue manager in the queue sharing group. This allows the inbound channel start request to be serviced by any queue manager in the queue sharing group. At this point, messages can be placed on application queues that are either shared or local to the queue manager (as long as they are defined on each queue manager in the queue sharing group, specifying QSGDISP(GROUP) in the definitions).

Shared outbound channels

The MQ intercommunication guide states that an outbound channel is a shared channel if it moves messages from a shared transmission queue. Thus, we defined a shared transmission queue for our outbound channels, along with an outbound sender channel with a QSGDISP of GROUP. This enables the queue managers in the queue sharing group to perform load-balanced start requests for this channel.

Example: The following is our definition for the shared transmission queue:

```
DEFINE QLOCAL(DQMCSQ9.QSG.XMITQ) +
REPLACE +
STGCLASS(DQMSTG) +
DESCR('SHARED XMITQ QUEUE FOR DQM TO J90') +
QSGDISP(SHARED) +
MAXDEPTH(2000) +
TRIGGER +
TRIGDATA(DQMSSL.MQGP.TO.CSQ9) +
INITQ(SYSTEM.CHANNEL.INITQ) +
USAGE(XMITQ) CFSTRUCT(MSGQ1)
```

Example: The following are our definitions for the local sender channel and the remote receiver channel:

```
DEFINE CHANNEL(DQMSSL.MQGP.TO.CSQ9) +
REPLACE +
CHLTYPE(SDR) +
XMITQ(DQMCSQ9.QSG.XMITQ) +
QSGDISP(GROUP) +
TRPTYPE(TCP) +
DISCINT(15) +
CONNAME(J90EIP.PDL.POK.IBM.COM) +
SSLCIPH(TRIPLE_DES_SHA_US) +
DESCR('SHARED SDR CHANNEL TO J90 FOR DQM')

DEFINE CHANNEL(DQMSSL.MQGP.TO.CSQ9) +
REPLACE +
CHLTYPE(RCVR) +
TRPTYPE(TCP) +
SSLCAUTH(REQUIRED) +
SSLCIPH(TRIPLE_DES_SHA_US) +
DESCR('DQM RCVR CHANNEL FROM SHARED SDR CHANNEL ON MQGP')
```

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`
  - **WebSphere:**

    `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 zPET 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
```

Migrating to WebSphere Message Broker Version 6

Before migrating to Websphere Message Broker (WMB) V6 our WBIMB configuration consisted of three brokers at the WebSphere Business Integration Message Broker V5 level on one sysplex and two brokers at the same level on another sysplex. We used the WBIMB V5 toolkit on Windows connecting to a Configuration Manager which was also on Windows. We migrated all of our brokers to WMB V6, created a new broker, and added a z/OS Configuration Manager to one sysplex while the other one is still using the old Configuration Manager in our Windows machine. The option to have a z/OS configuration manager is new with WMB V6.

Changes from WBIMB V5 to WMB V6

The following are some of the changes we had to make from WBIMB V5 to WMB V6.

Directory structure changes

WMB V6 added a new HOME directory separate from the COMP directory used in WBIMB V5. Our new directory structure looks as follows:

```
/wmb60/basecode    contains the product code
/wmb60/COMP        contains a directory for each broker or configmgr
/wmb60/HOME        contains a directory for each broker and config mgr which has files bipprof and ENVFILE
```

Each of the above directories is mounted off of a separate ZFS filesystem.

DB2 DSNAOINI file changes

The WMB V6 started task JCL uses the *dsnaoini* from a dataset instead of using the one in the broker directory as WBIMB V5 did.

/wmb60/HOME/CSQ2BRK/bipprof has a statement:

```
export DSNAOINI=//\'WMB.CSQ2BRK.DSNAOINI\(BIPDSNAO\)\'
```

The broker ENVFILE contains the statement:

```
DSNAOINI=//'WMB.CSQ2BRK.DSNAOINI(BIPDSNAO)'
```

This points to the z/OS dataset member to get the values it needs.

We followed the migration instructions in the WMB V6 Information Center

```
http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp
```

See the section titled ″Migrating from Version 5.0 products″ for the WebSphere Message broker product. We migrated a broker first, then the toolkit, and did the configuration manager last.

XML changes

The new WMB V6 broker requires the XML Toolkit for z/OS, Pgm 5655-J51. This was installed and referenced in the broker bipprof file XMLTOOLKIT=/ixm/ixm/IBM/xml4c-5_5 as well as in the ENVFILE.

Broker migration

Some of the things to watch out for with the Broker migration are:

- Be sure to never edit the files in the broker registry directory. If changes need to be made use **printf "changed value" > filename** . Editing can often add CR or LF characters which the broker does not handle well.
- When migrating, the component directory is the previous version's component directory. The HOME directory is new for WMB V6.

As a pre-migration task we backed up our broker databases and toolkit workspace data. We also backed up the component directories for the brokers. Then we followed the steps outlined in the section "Migrating from WebSphere Business Integration Message Broker Version 5.0 to WebSphere Message Broker Version 6.0" sub-topic "Migrating a Version 5.0 broker to Version 6.0 on z/OS" found at:

`http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp`

Note: The Unix System Services environment variables of the userid running the migration jobs will be copied to the broker ENVFILE in the HOME directory. Be careful and review the ENVFILE to be sure you don't have variables set that you don't want for the broker.

We edited the /wmb60/HOME/CSQABRK/*ENVFILE* to remove all entries it added from */u/lorain0/.profile*. The jobs were run from userid *lorain0*.

All migration jobs ran successfully.

Toolkit migration

On Windows we backed up the WBIMB databases *WBICMDB* and *DWCTRLDB*. Then we used the '**export**' function in the wbimb v5 toolkit to save all projects and create a file structure for them.

We then ran the setup.exe for the new WMB V6 Toolkit. The install was successful and the toolkit was able to connect to the WBIMB V5 Configuration Manager as well as the V5 (not yet migrated) and V6 brokers on z/OS.

Configuration Manager migration on Windows

The following scripts were run as documented in the WMB V6 Information Center:

- *mqsimigratecomponents -c configmgr* pre-check (Note: Don't use the config mgr name.)
- *mqsimigratecomponents configmgr* do the migration
- *mqsimigratecomponents configmgr* do the migration

  These all succeeded so we then started the config mgr
- *mqsistart configmgr*

The conf mgr started successfully but when we started the toolkit it failed to connect to the new configmgr. The event log had:

```
( ConfigMgr ) Unexpected exception in ConfigurationManager class 'initialize' method; exception text:
 ''java.lang.NoSuchFieldError: msgToken'', ''msgToken''.   An exception was caught by the ConfigurationManager class
 'initialize' method while the Configuration Manager was being started or stopped. The exception text is:
 'java.lang.NoSuchFieldError: msgToken', 'msgToken'.
```

We found an IBM technote at:

`http://www.ibm.com/support/docview.wss?rs=849&context=SSKM8N&dc=DB520&uid=swg21229211&loc=en_US&cs=UTF-8&lang=en`

describing this error which says:

Problem

Configuration Manager start up fails with BIP1002E. java.lang.NoSuchFieldError: msgToken exception on Configuration Manager start up.

Cause

This problem occurs if the Config Manager is connected to a WebSphere® MQ V6 queue manager, but does not have the MQ Java™™ Client classes located on the CLASSPATH used by the profile.

Solution

Add the following JARs to the CLASSPATH (all from inside the WMQ installation's lib directory):

```
providerutil.jar
com.ibm.mqjms.jar
ldap.jar
jta.jar
jndi.jar
jms.jar
connector.jar
fscontext.jar
```

We added each of the above jar files to the windows CLASSPATH as shown below:

```
C:\Program Files\IBM\WebSphere MQ\Java\lib\providerutil.jar;C:\Program Files\IBM\WebSphere
MQ\Java\lib\com.ibm.mqjms.jar;C:\Program Files\IBM\WebSphere MQ\Java\lib\ldap.jar;C:\Program Files\IBM\WebSphere
MQ\Java\lib\jta.jar;C:\Program Files\IBM\WebSphere MQ\Java\lib\jndi.jar;C:\Program Files\IBM\
MQ\Java\lib\jms.jar;C:\Program Files\IBM\WebSphere MQ\Java\lib\connector.jar;C:\Program Files\IBM\WebSphere
MQ\Java\lib\fscontext.jar;C:\Program Files\IBM\WebSphere MQ\Java\lib\com.ibm.mq.jar;
```

Then we tried a simple deploy which failed with the message: `the deployment message was addressed to a broker with a UUID 21f01d8e-0a01-0000-0080-ea101ddff920, but this does not match the UUID of the running broker (09fede6a-0a01-0000-0080-d8b172fb79c9).`

This was fixed by altering the universally unique identifier (UUID) using the Configuration Manager Proxy API Exercisor. Start the Configuration Manager Proxy API Exerciser. This is a sample application that demonstrates the capabilities of the Configuration Manager Proxy (a comprehensive Java interface that allows you to control broker domains programmatically). To start this application, we performed the following steps:

- On Windows, click **Start > IBM WebSphere Message Brokers 6.0 > Java Programming APIs > Configuration Manager Proxy API Exerciser**.
- Connect to the configmgr.
- Right click the broker name then select set UUID.
- Enter the new UUID value.

This is a sample application that demonstrates the capabilities of the Configuration Manager Proxy (a comprehensive Java interface that allows you to control broker domains programmatically).

Creating a z/OS configuration manager

We followed the instructions in the WebSphere Message Broker V6 Information Center titled ″Creating a Configuration Manager on z/OS″.

This task went very well with no problems.

Then we switched the WMB V6 Toolkit on Windows to connect to this new z/OS Configmgr. The userid used by the Windows machine was called *'mqtest'* in lowercase. When we deployed to the configmgr it received this RACF error:

```
SYSTEM.BROKER.CONFIG.QUEUE could not be opened (MQ reason code 2035 while trying to open the queue)
ICH408I USER(mqtest  ) GROUP(        ) NAME(???
  LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
```

We had to change the Windows userid to be uppercase as lowercase user IDs will not work on z/OS when connecting to WebSphere MQ resources.

Then we had to authorize the Toolkit user to access the configmgr per the instructions in the WMB Info Center: See the section ″Ensure that your toolkit machine and user ID has the appropriate authorization on the z/OS Configuration Manager.″

In SDSF, grant access to your user ID. For this to work on all machines, enter:

```
'/F <started task name>,CA U=<userID>,A=YES,P=YES,X=F'
```

or to grant access to your user ID for a specific machine, enter:

```
'/F <started task name>,CA U=<userID>,A=YES,M=<machine name>,P=YES,X=F'
```

Verify the above by entering:

```
'/F <configmgrname>,LA
```

We then used the command:

```
'/F MQZ2CMGR,CA U=mqbroker/MQSTEST,A=YES,P=YES,X=F
```

The response message was +BIP8071I MQZ2CMGR 2 Successful command *completion*. We then recycled the configmgr and the toolkit then connected to configmgr successfully.

You cannot switch back and forth between configuration managers when deploying to the same brokers because of the UUID's that are assigned to the brokers. If you do try to deploy using a different configuration manager than was controlling the broker beforehand you will get an error message like:

```
BIP2045E: Broker CSQ2BRK running on WebSphere queue manager CSQ2 did not process
a deployment message, because it was addressed to a broker with a different identifier.
```

This message usually means that an attempt has been made to assign the broker to a second (or a reinitialized) Configuration Manager.

Each broker is identified by a universally unique identifier (UUID) which is allocated when the Message Brokers Toolkit or Configuration Manager Proxy creates a definition for the broker. When deployment occurs, a UUID check is made to help prevent accidental deployment of changes to brokers not under the control of the Configuration Manager. In this case, the deployment message was addressed to a broker with a UUID 7c2e2517-0d01-0000-0080-ae77adc1960f, but this does not match the UUID of the running broker (17794370-0701-0000-0080-c2dfca2e3733).

To switch to the new configmgr we used the Configuration Manager Proxy API Exercisor to change the UUID as described above.

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 DYNAMICVIPA block being processed, message `EZD1214I Initial Dynamic VIPA processing has completed` 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

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 86 demonstrates the cluster environment.

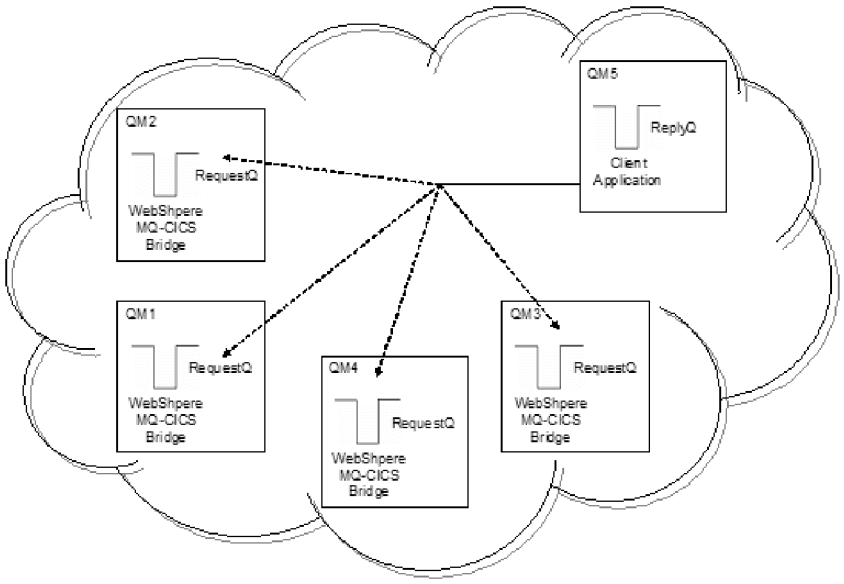


Figure 86. Our MQ cluster configuration for the WebSphere MQ-CICS bridge

WebSphere MQ-CICS adapter using shared queues

For our second workload environment shown in Figure 87 on page 187, 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.

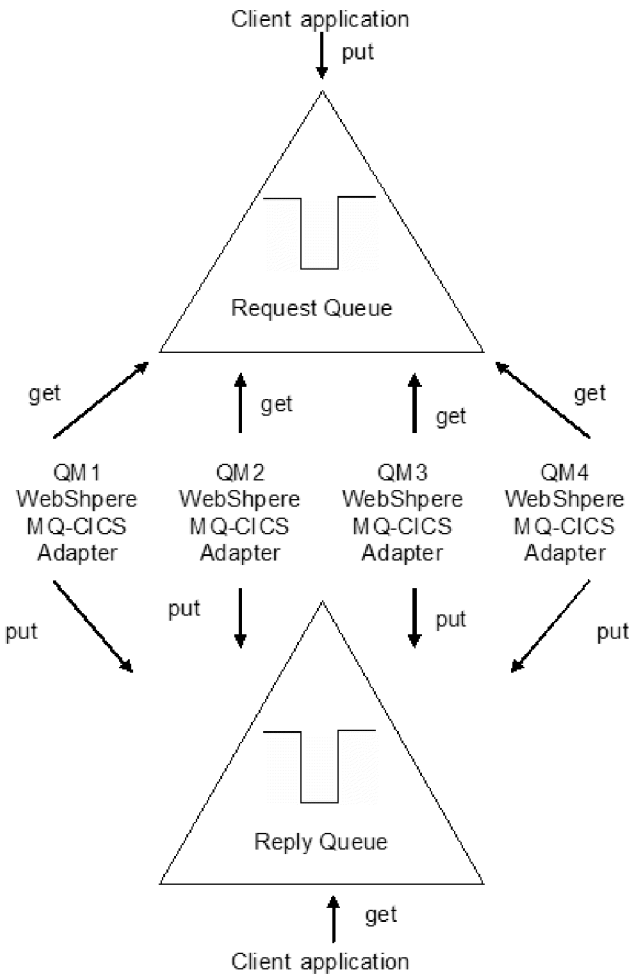


Figure 87. Our queue sharing group configuration for our WebSphere MQ-CICS adapter workload

Chapter 20. 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 have 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 V1R9 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 V1R9 WebSphere test environment

The following topics provide an overview of our z/OS V1R9 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 cf50652.12
- 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
- DB2 V9.1 with JDBC
- CICS TS 3.1
  - CICS Transaction Gateway (CICS TG) V6.1
- IMS V9 with IMS Connector for Java V9
  - IMS Connector for Java V9.1.0.1

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 6.0.1.1
- IBM WebSphere Developer for zSeries Version 6.0.1
- WebSphere Studio Workload Simulator V1.0

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) V6.1
- Added our zBank application (and J2EE server 7 for it)
- Implemented an enhancement of the zBank application, zCredit
- Security enhancements (TAM, TAI++, WebSeal on zLinux)
- Removed Node JH0 from P1 Cell (system removed from our test environment).

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 five cells: T1, T2, and T3 for our test systems, P1 for our WebSphere Application Server for z/OS production systems, and QP for WebSphere Application Server for z/OS applications used by MQ team. All cells are configured as network deployment cells. The QP cell continues 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 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.

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 88 on page 192 shows the server address spaces in our P1 cell.

Note: The wsp1s1 cluster is not shown in the diagram.

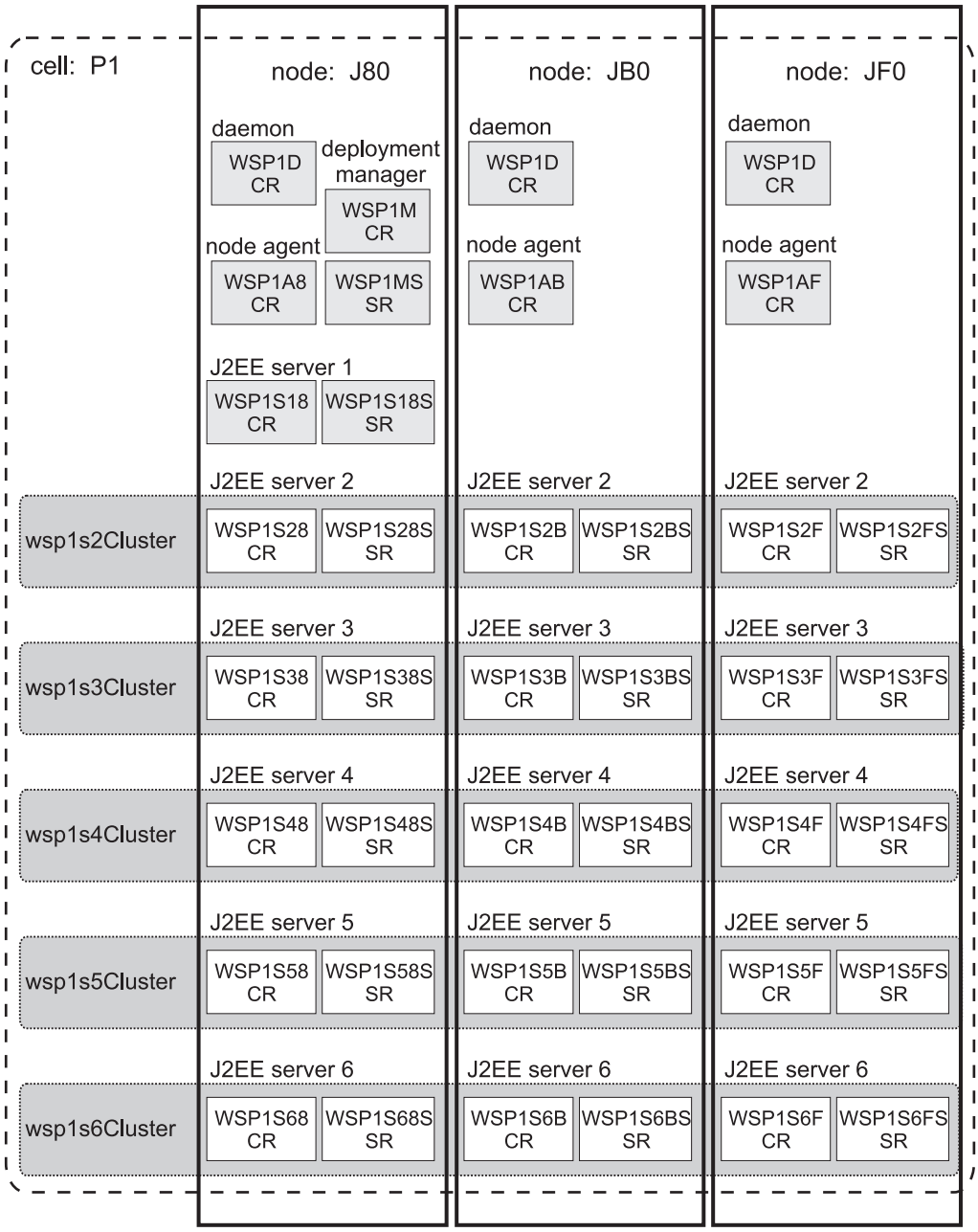


Figure 88. Our WebSphere for z/OS V6 configuration

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:

`WSccs[n]y[S]`

where:

WS The first two characters are always "WS" to identify a WebSphere resource.

cc Cell identifier:
- **T1** Test cell 1
- **T2** Test cell 2
- **P1** Production cell 1
- **QP** MQ Team Production cell

s[*n*] Server type. For J2EE server control regions and server regions, *n* is the instance number of the server within the node:
A Node agent
D Daemon
M Deployment manager
S*n* J2EE server control region, instance *n*

y System identifier:
1 Z1 (test)
2 Z2 (test)
8 J80 (production)
B JB0 (production)
F 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 **W**eb**S**phere **p**roduction cell **1** J2EE server **s**erver region **1** on system J**8**0.

Server short names are specified in upper case. Server long names are the same as the short names, but are specified in lower case.

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

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) 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. 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 InfoCenter at 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
- Also see the WebSphere Integration Test team's current report, which you can find at publibz.boulder.ibm.com/epubs/pdf/e0z1r111.pdf

Migrating to CICS Transaction Gateway V6.1

We made changes to the WebSphere environment when migrating to CICS Transaction Gateway (CICS TG) V6.1. See "Migrating to CICS Transaction Gateway V6.1" on page 139 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. 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 4 on page 195 shows, from the above reference, the methods that you can use to pass additional client information.

Table 4. 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:9.12.20.162
```

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 5.

Table 5. 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
clientAccountingInformation	`BookStoreEJBDB2Entity`	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:9.12.20.162
```

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 196 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:9.12.20.162
```

- 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. 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 client*xxx* 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:9.12.20.162
 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:9.12.20.162
 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 client*xxx* properties:

```
NAME     ST A   REQ ID          AUTHID   PLAN     ASID TOKEN
RRSAF    TD       4 WST2S22S    DB2USR    ?RRSAF   01DC 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 V3.3

We made changes to the WebSphere environment when we installed TPC-R V3.3 in our zPET environment. For specific information about these changes, see Chapter 6, “Using TPC-R V3.3 in our zPET environment,” on page 107.

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, available at `http://www.ibm.com/software/webservers/appserv/zos_os390/library/`
- IBM WebSphere Application Server, Version 6.0 Information Center, available at publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp
- IBM DB2 Information Center, available at publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp
- IBM Techdocs (flashes, white papers, and others), available at www.ibm.com/support/techdocs/
- *Java 2 Platform Enterprise Edition Specification*, available at http://java.sun.com/products/j2ee/
- IBM CICS Transaction Gateway documentation, available at http://www.ibm.com/software/ts/cics/library/
- IBM HTTP Server for OS/390 documentation, available at http://www.ibm.com/software/webservers/httpservers/library/
- IBM WebSphere Studio Workload Simulator documentation, available at www.ibm.com/software/awdtools/studioworkloadsimulator/library/

Part 2. Linux virtual servers

This part describes the experiences of the System z Linux Virtual Servers Platform Evaluation Test (LVS PET) team. the Linux on System z and Linux virtual server aspects of our computing environment.

We discuss the Linux on System z and Linux virtual server aspects of our computing environment, addressing 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 21. About our Linux virtual server environment

In this edition of our test report, we discuss the implementation of a set of reference architectures for systems management. The focus of our integration test lab for the past six months has been almost exclusively on this subject. Systems management encompasses many smaller items including software management, data management, and security. Fortunately, there are many existing tools to ease the burden on systems programmers.

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, as the team managing and exploiting this environment, we are very 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 systems management policies that we follow and the tools for implementing those 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 issue 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 our 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 and z/VM® environments.
- **Preference for packaged (rather than home-grown) solutions:** We strive to use the 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 many of the data management tasks handled by IBM products, while overall systems management is handled by a combination of IBM tools in conjunction with 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 that of 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 disciple 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 topics, so stay tuned.

About our environment

As you might recall from previous reports, we test the material that we present here in a multiple-CPC, multiple-LPAR, highly available environment.

As a result of reader feedback and customer insights, we saw an opportunity to enhance the systems management aspects of our infrastructure and applications. During the construction of our environment, we primarily focused our attention on availability and robustness. Over time, the infrastructure presented in this series of 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.

Our workloads

This topic explains the workloads that we execute here in the 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 EJBs.

- **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, EJBs, and MQSeries, and is populated with an extremely large data set from the Library of Congress.

Overall configuration

Figure 89 shows the logical flow of a transaction. The clouds in the picture depict application clusters.

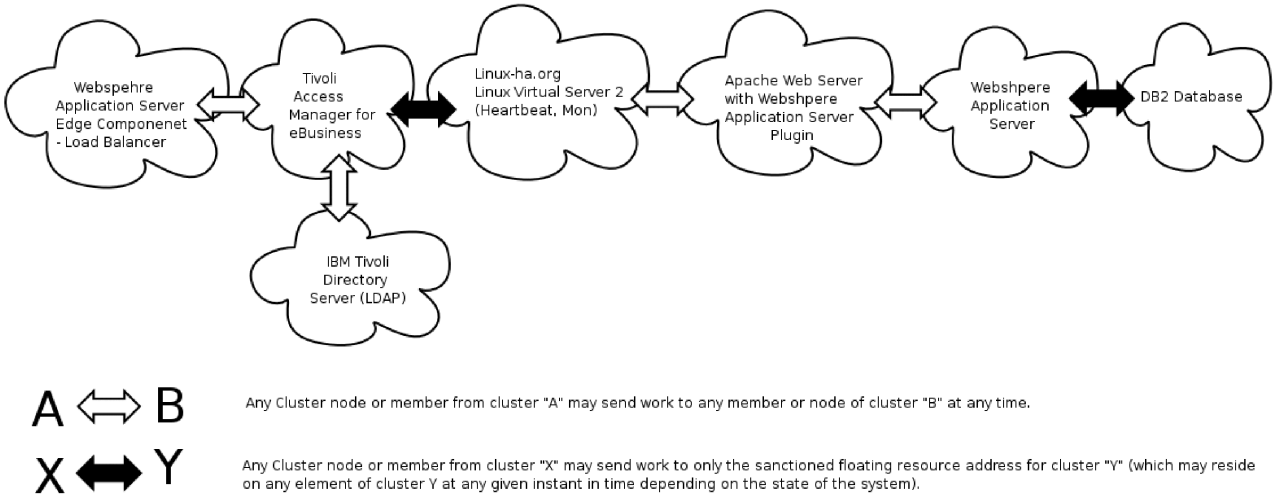


Figure 89. LVS PET application configuration: Logical transaction flow between application clusters

Each cluster has members that are spread across two z/VM LPARs on two different CPCs, as shown in Figure 90 on page 204.

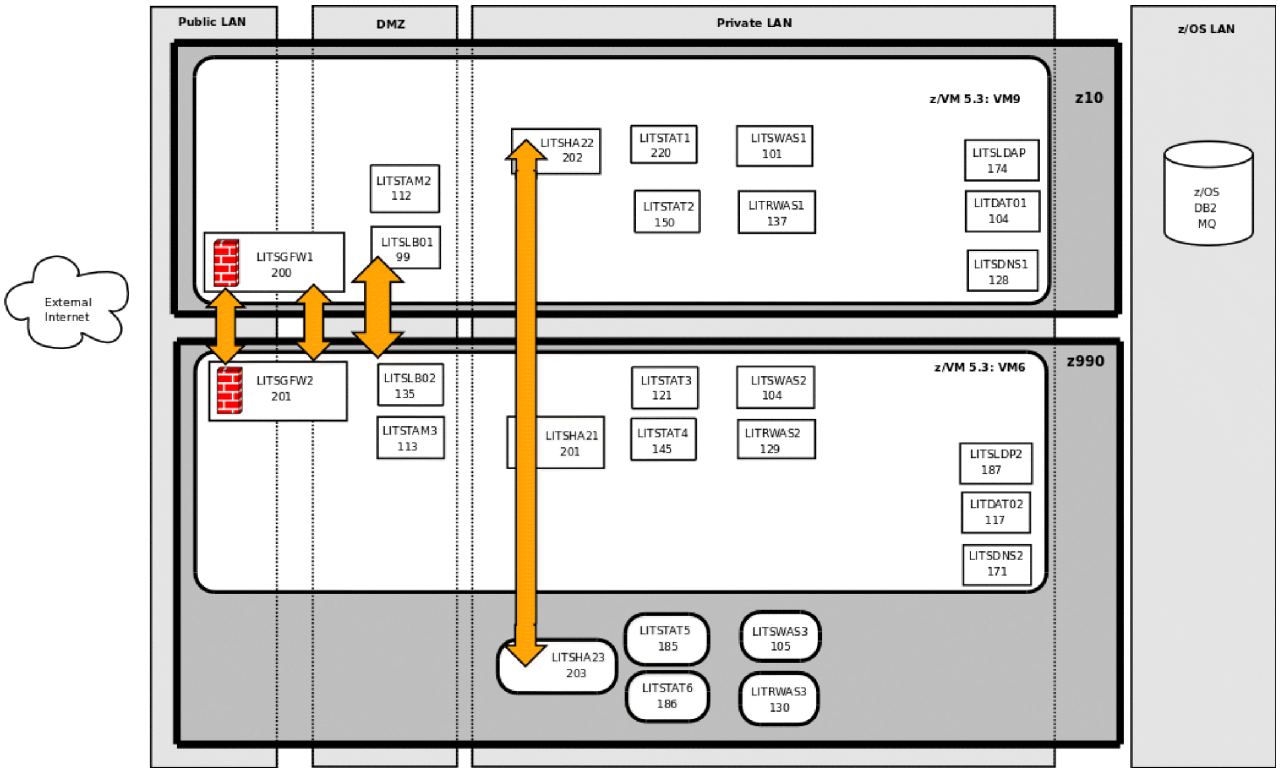


Figure 90. LVS PET system configuration: z/VM LPARs hosting Linux virtual servers on two CPCs

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.

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 TAMe WebSEAL cluster address.
3. 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 inputs this 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 that is installed on the Apache server, transfers the request to an available member in the WebSphere Application Server cluster.
9. WebSphere Application Server fulfills the request. If the request involves a transaction to DB2, then WebSphere Application Server uses the JDBC Type 4 driver to pass the request onto DB2.
   - For Trade, the request goes to the DB2 UDB cluster on Linux.
   - For Bookstore, the request goes to DB2 z/OS data sharing group and shared message queue, depending on the type of transaction.
10. Once the request is fulfilled, the response is bubbled back to the client.

System names and usage

Table 6 lists our systems, their host names, IP addresses, and usage. 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 6. Our Linux system names and usage

Host name	IP address	Usage
litdat01	192.168.71.104	Primary DB2 UDB
litrdat1	192.168.71.148	Backup DB2 - Red Hat
litdat02	192.168.71.117	DB2 UDB - Director
litdcon1	192.168.71.104	Primary DB2 Connect™
litdcon2	192.168.71.136	Secondary DB2 Connect
litdir00	192.168.71.249	IBM Director
lithub	192.168.75.192	Network hub
litrlog1	192.168.71.110	Log server
litrsmb1	192.168.71.108	Samba server
litrwas1	192.168.71.137	WebSphere Application Server - Red Hat
litrwas2	192.168.71.129	WebSphere Application Server - Red Hat
litrwas3	192.168.71.130	WebSphere Application Server - Red Hat
litrwas4	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
litslb01	192.168.74.99	Primary load balancer
litslb02	192.168.74.135	Backup load balancer
litsldap	192.168.71.174	Primary LDAP server
litsldp2	192.168.71.187	Secondary LDAP server

Table 6. Our Linux system names and usage (continued)

Host name	IP address	Usage
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
litstat6	192.168.71.186	Apache Web Server with WebSphere Application Server Plug-in
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
litswas4	192.168.71.106	WebSphere Application Server Network Deployment - SUSE
littam01	192.168.74.120	IBM Tivoli Access Manager for e-business WebSEAL

Chapter 22. Linux 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 presents our experiences with Linux distribution upgrades and the associated caveats that we ran into along the way.

Maintenance strategy and methodology

When applying system maintenance, it is best to perform thorough system backups and ensure that you follow 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 (secondary, failover, standby) systems after removing it from the active configuration. How we perform our backups is detailed in "DFSMS, IBM Tape Manager, and IBM Backup and Restore for z/VM" on page 224. The primary system allows the environment to remain available; however, 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 the regulations and procedures outlined in your data center guidelines and protocols.

Base operating system upgrades

From time to time, mission critical systems will be deemed outdated and require complete replacement. If you have been following along with our reference architecture, you will be well suited to deploy rolling upgrades in many situations. This topic provides an overview of the upgrade processes as tested here in our laboratory. Note that, before you begin, it is imperative that you follow the distributions' supported mechanisms and ensure your backup and restore procedures are in place.

Upgrading Red Hat Enterprise Linux 4.6 to Red Hat Enterprise Linux 5.1

When it came time to migrate our Red Hat Enterprise Linux (RHEL) systems from RHEL 4.6 to RHEL 5.1, we chose to do in-place upgrades. We did run into one exceptionally interesting problem on a few of the upgrades and have chosen to document our workaround here. But before we discuss this particular issue, let us take a moment to examine the statements of support regarding in-place upgrades.

At the time of our migration, the RHEL 5.1 release notes contained the following note indicating that the in-place upgrade procedure was supported:

> In order to upgrade an already-installed Red Hat Enterprise Linux, you must use Red Hat Network to update those packages that have changed. You may use Anaconda to perform a fresh installation of Red Hat Enterprise Linux 5 or to perform an upgrade from the latest updated version of Red Hat Enterprise Linux 4 to Red Hat Enterprise Linux 5.

Comparatively, let us examine the RHEL 5.2 release notes, where we found that this process was now discouraged:

> Red Hat does not support in-place upgrades between major versions of Red Hat Enterprise Linux. Although anaconda's "upgrade" option will allow users to perform an upgrade from earlier major versions of Red Hat Enterprise Linux (such as Enterprise Linux 4 to Enterprise Linux 5), there is no guarantee that the upgrade will result in a working configuration. In-place upgrades across major releases do not preserve all system settings, services, and custom configurations. For this reason, Red Hat strongly recommends that you perform a fresh installation rather than a system upgrade between major versions.

As you can see, this process is no longer recommended or supported. When you plan to migrate your Red Hat systems to the next major version, special consideration will be required on how you want to handle re-installing various vendor-provided middleware products, as well as which application data preservation methods will be used if you chose to perform a fresh installation.

Having chosen to perform in-place upgrades, we discovered that the Anaconda installer does not ask whether you are using FCP (SCSI) devices. This is critical to our deployment as we utilize FCP throughout our environment on many different systems. To get around this issue, we started the installer, started an **ssh** session and manually added the SCSI devices. We did this before responding to any of the upgrade process prompts. Once the SCSI devices were brought online and mounted, it was then possible to resume the installer session and subsequently complete our system upgrade.

Also of critical note is that you must manually run **mkinitrd** to include the scsi modules and then run **zipl** after the upgrade process has finished normally (Complete screen – Congratulations, your Red Hat Enterprise Linux Server installation is complete).

We went though numerous trial and error iterations to get this in-place upgrade to work when SCSI devices were involved. We hope that this advance notice will save some of our readers from the same pitfalls. Once more, it is very important to have good backup and restore procedures when performing upgrades, which we discuss further in "DFSMS, IBM Tape Manager, and IBM Backup and Restore for z/VM" on page 224.

Upgrading SUSE Linux Enterprise Server 9 to SUSE Linux Enterprise Server 10

As with any upgrade, we first took a backup of the system. We then shut down the active system, taking it out of the HA environment. The following are the steps we followed to accomplish the upgrade:

1. Point to the ftp server that contains the initrd, parmfile and vmrdr.ikr files.
2. Start the install and respond to the prompts (enabling our network).
3. Enter the install directory on the ftp server, **ssh** to the target install system and enter **yast** to begin the installation.
4. Configure our DASD disks and select **Update** for installation mode.
5. Once the installation completed, issue the **reboot** command and login to proceed with the installation.
6. Run the command **/usr/lib/YaST2/startup/YaST2.ssh** to complete the installation.

We should point out that upgrades will not preserve all system settings, services, and custom configurations. Depending on what middleware you run, it may be possible that updates made to files during the install or customization of a product could have been overwritten. In our case, that happened on our DB2 system. Modifications that were made to file /etc/services were gone after the upgrade. We opened a bugzilla report and were told that, after any upgrade, it is important to go back and check all files ending in `rpmsave`, as this is how the upgrade process flags differences. The following is an example of how to locate these files:

```
# find / -name *.rpmsave
/etc/X11/xdm/Xservers.rpmsave
/etc/X11/xkb/symbols/cs.rpmsave
/etc/X11/xkb/symbols/gr.rpmsave
/etc/cups/ppds.dat.rpmsave
/etc/snmp/snmpd.conf.rpmsave
/etc/services.rpmsave
/etc/modprobe.conf.rpmsave
/etc/kde3rc.rpmsave
/etc/profile.rpmsave
/opt/kde3/share/config/kdm/kdmrc.rpmsave
```

Operating system updates

Even if you chose not to perform any large scale upgrades of your base Linux distribution, it is wise to apply distribution supported and recommended updates during the lifetime of your virtual server. This is especially critical when security updates are available for your release. This topic briefly describes the process we use to apply routine service. As always, ensure a backup and restore procedure exists before making changes to your systems.

Red Hat routine maintenance

To maintain our Red Hat systems, we use the Red Hat Network (RHN) service. We set up a Squid proxy server to enable access to RHN for our internal network and updated all of our systems to point to it. Originally, we used the **up2date** command to configure and register each system.

Now that we have migrated to RHEL 5.1, we use Yellowdog Updater Modified (YUM). To initiate an update, we issue commands **yum clean all** followed by **yum update** on the Linux system. This can also be accomplished from the RHN Web site.

The System Set Manager allows you work with large numbers of systems to schedule errata updates; upgrade, install, remove, or verify packages; install and remove patches and more. You can also set it up to apply the updates automatically. You can specify which packages you do not want updated, such as kernel updates. Alert notifications are also sent to your registered e-mail address notifying you of newly available maintenance that is applicable to your environment. The Systems Overview displays your systems and any available updates and their severities. The Errata Overview displays any errata that may apply to your environment and the type (security, bug fix, or enhancement). The System Channel Overview displays channels relevant to your organization.

SUSE routine package maintenance

To maintain our SUSE systems, we use YaST online update. Each Linux system is configured to retrieve updates from the ftp server which contains our installation source. This is most easily accomplished by issuing the **yast2 inst_source** command and entering the pertinent information into the on-screen prompts. To

start an update, a system operator only needs to issue the **yast2 online_update** command. You might need to be aware of any firewalls that are in place for your environment before package retrieval can begin.

Chapter 23. Linux data management

The second major component of systems management we dealt with, after software management, was data management. Data management covers the policies, processes, and tools needed to ensure the integrity of data. With properly configured and supported systems, administrators quickly turn their attention to backup and restore procedures to protect the investment. However, there is no simple solution for all backup needs. In this topic, we describe our experiences with a file system level incremental backup solution using Tivoli Storage Manager (TSM), as well as a disaster recovery backup solution for z/VM Linux guests using IBM Backup and Restore Manager for z/VM. Together, these tools can provide a powerful backup solution for Linux virtual servers deployed on z/VM. Note that TSM and Backup Manager for z/VM can also be used to perform file-level backups and disaster recovery backups, respectively, of LPAR-hosted virtual Linux servers, though some additional configuration is necessary.

Our goal is to show you how to back up critical data so that it can be restored quickly, if necessary, as well as how to implement safeguards to checkpoint the file system.

Before we continue with the more sophisticated backup mechanisms, we recommend that anyone using Linux file systems ensure that they are using a journaling file system, such as ext3, for the data. This type of file system is the default with most Linux distributions and there is little reason why one should chose not to use it.

The remainder of this topic describes the tasks within the data management discipline that we implemented and tested within our LVS PET environment.

Tivoli Storage Manager

Tivoli Storage Manager (TSM) is a product designed for file-level backups. These backups can be performed incrementally, on demand, or via regular scheduled intervals depending upon the configuration.

TSM is excellent in overcoming situations where configuration data is corrupted, package installation or software upgrades go awry, or files are accidentally deleted. Restoring from these scenarios is straightforward and easy with TSM, provided the TSM client and configuration file remain operational. It is recommended that you take precautions to ensure that TSM is not easily deleted by using file access permissions or read-only access.

It is important to note that the backups provided by TSM will be of use only if you have a file system to restore the files onto. This is a critical point, as TSM will not be able to directly resolve disaster situations where disks or partition tables are lost. In these cases, some manual intervention is required to restore an underlying disk, partition table, file system, and Linux TSM client before performing a restore operation.

We use an FCP-attached SCSI LTO tape library—specifically, the IBM 3583 with Ultruim2 drives. Our 3583 has six FCP-attached Ultrium2 drives with slots for 72 tape cartridges, which gives it a total uncompressed capacity of 14.4 terabytes. We have partitioned the library into two logical libraries. The smaller logical library

has two drives and 24 of the available slots. The larger library has four drives and 48 of the available slots. We are using the larger logical library for our integration test environment.

We run TSM Server release 5.3.2 on SLES 9 SP3, and TSM client version 5.2.2 on all our managed systems. We are using this old release of SLES due to the availability of drivers for the 3583 library. The drivers include a kernel module, which limits our options of which releases of Linux we can use to run the TSM server. Note that TSM 5.3 is going out of service in the summer of 2008, so we expect to discuss a TSM update in a future release of this test report.

The TSM 5.3 server supports deployment on LPAR or z/VM hosted Linux servers running RHEL 3, SLES 8, and SLES 9. Newer releases of the TSM server support SLES 10 and RHEL 5.

Tivoli Storage Manager Server configuration

The actual installation of the TSM 5.3 server code is a straightforward rpm package install. Therefore, we will forego describing that part and move on to the process of configuring the TSM server once it is up and running.

he first step is to tell the TSM server that there is a SCSI library out there that it needs to know about:

```
TSM:LITSTSM>
define library TICL3583 libtype=scsi

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE LIBRARY TICL3583
libtype=scsi
ANR8400I Library TICL3583 defined.
```

Configure the TSM server to talk to that library. In our case, the 3583 driver created a new device node at /dev/IBMchanger0 for the library manager:

```
TSM:LITSTSM>
define path LITSTSM TICL3583 srct=server autod=yes destt=library
  device=/dev/IBMchanger0 onl=yes

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE PATH LITSTSM
TICL3583 srct=server autod=yes destt=library device=/dev/IBMchanger0 onl=yes

ANR8953I Library TICL3583 with serial number  is updated with the newly
discovered serial number 0000013176581018 .
ANR1720I A path from LITSTSM to TICL3583 has been defined.
```

Tell TSM that there are four drives in the library:

```
TSM:LITSTSM>
define drive ticl3583 drive1 cleanfreq=asneeded

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE DRIVE ticl3583
drive1 cleanfreq=asneeded
ANR8404I Drive DRIVE1 defined in library TICL3583.

TSM:LITSTSM>
define drive ticl3583 drive2 cleanfreq=asneeded

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE DRIVE ticl3583
drive2 cleanfreq=asneeded
ANR8404I Drive DRIVE2 defined in library TICL3583.

TSM:LITSTSM>
```

```
define drive ticl3583 drive3 cleanfreq=asneeded

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE DRIVE ticl3583
drive3 cleanfreq=asneeded
ANR8404I Drive DRIVE3 defined in library TICL3583.

TSM:LITSTSM>
define drive ticl3583 drive4 cleanfreq=asneeded

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE DRIVE ticl3583
drive4 cleanfreq=asneeded
ANR8404I Drive DRIVE4 defined in library TICL3583.
```

Make TSM aware of those four drives:

```
TSM:LITSTSM>
define path LITSTSM DRIVE1 SRCT=server AUTOD=YES destt=drive libr=ticl3583
  device=/dev/IBMtape0n

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE PATH LITSTSM
DRIVE1 SRCT=server AUTOD=YES destt=drive libr=ticl3583 device=/dev/IBMtape0n
ANR8955I Drive DRIVE1 in library TICL3583 with serial number  is updated with
the newly discovered serial number 1110016288 .
ANR1720I A path from LITSTSM to TICL3583 DRIVE1 has been defined.

TSM:LITSTSM>
define path litstsm drive2 SRCT=server AUTOD=YES destt=drive libr=ticl3583
  device=/dev/IBMtape1n

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE PATH litstsm
drive2 SRCT=server AUTOD=YES destt=drive libr=ticl3583 device=/dev/IBMtape1n
ANR8955I Drive DRIVE2 in library TICL3583 with serial number  is updated with
the newly discovered serial number 1110015929 .
ANR1720I A path from LITSTSM to TICL3583 DRIVE2 has been defined.

TSM:LITSTSM>
define path litstsm drive3 SRCT=server AUTOD=YES destt=drive libr=ticl3583
  device=/dev/IBMtape2n

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE PATH litstsm
drive3 SRCT=server AUTOD=YES destt=drive libr=ticl3583 device=/dev/IBMtape2n
ANR8955I Drive DRIVE3 in library TICL3583 with serial number  is updated with
the newly discovered serial number 1110052419 .
ANR1720I A path from LITSTSM to TICL3583 DRIVE3 has been defined.

TSM:LITSTSM>
define path litstsm drive4 SRCT=server AUTOD=YES destt=drive libr=ticl3583
  device=/dev/IBMtape3n

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE PATH litstsm
drive4 SRCT=server AUTOD=YES destt=drive libr=ticl3583 device=/dev/IBMtape3n
ANR8955I Drive DRIVE4 in library TICL3583 with serial number  is updated with
the newly discovered serial number 1110016106 .
ANR1720I A path from LITSTSM to TICL3583 DRIVE4 has been defined.
```

Tell TSM what kind of drives they are:

```
TSM:LITSTSM>
define devcl LTO2 libr=ticl3583 devt=lto format=ultrium2c

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE DEVCLASS LTO2
libr=ticl3583 devt=lto format=ultrium2c
ANR2203I Device class LTO2 defined.
```

Verify the work up to this point with a few query commands:

```
TSM:LITSTSM>
q library

ANR2017I Administrator SERVER_CONSOLE issued command: QUERY LIBRARY

         Library Name: TICL3583
         Library Type: SCSI
               ACS Id:
     Private Category:
     Scratch Category:
WORM Scratch Category:
     External Manager:
               Shared: No
              LanFree:
   ObeyMountRetention:

TSM:LITSTSM>
q drive

ANR2017I Administrator SERVER_CONSOLE issued command: QUERY DRIVE

Library Name     Drive Name       Device Type     On-Line
------------     ------------     -----------     ------------------
TICL3583         DRIVE1           LTO             Yes
TICL3583         DRIVE2           LTO             Yes
TICL3583         DRIVE3           LTO             Yes
TICL3583         DRIVE4           LTO             Yes

TSM:LITSTSM>
q path

ANR2017I Administrator SERVER_CONSOLE issued command: QUERY PATH

Source Name     Source Type     Destination     Destination     On-Line
                                Name            Type
-----------     -----------     -----------     -----------     -------
LITSTSM         SERVER          TICL3583        LIBRARY         Yes
LITSTSM         SERVER          DRIVE1          DRIVE           Yes
LITSTSM         SERVER          DRIVE2          DRIVE           Yes
LITSTSM         SERVER          DRIVE3          DRIVE           Yes
LITSTSM         SERVER          DRIVE4          DRIVE           Yes

TSM:LITSTSM>
q devclass

ANR2017I Administrator SERVER_CONSOLE issued command: QUERY DEVCLASS

Device       Device        Storage    Device       Format     Est/Max     Mount
Class        Access           Pool    Type                   Capacity     Limit
Name         Strategy        Count                              (MB)
---------    ----------    -------    ---------    ------    --------    ------
DISK         Random              3
LTO2         Sequential          0    LTO          ULTRI-                DRIVES
                                                    UM2C
```

Everything looks good, so let's get a tape pool defined and some tapes in the library. We already placed barcodes on the new tapes and entered them in to the front load slot of the library , so we'll just reply to the prompts immediately:

```
TSM:LITSTSM>
define stgpool TAPE_POOL0 LTO2 maxscr=0

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE STGPOOL TAPE_POOL0
LTO2 maxscr=0
```

```
ANR2200I Storage pool TAPE_POOL0 defined (device class LTO2).

TSM:LITSTSM>
LABEL libvolume ticl3583 search=bulk labels=barcode checkin=private overwrite=yes

ANR2017I Administrator SERVER_CONSOLE issued command: LABEL libvolume ticl3583
search=bulk labels=barcode checkin=private overwrite=yes
ANR0984I Process 2 for LABEL LIBVOLUME started in the BACKGROUND at 06:59:01
PM.
ANR8799I LABEL LIBVOLUME: Operation for library TICL3583 started as process 2.
TSM:LITSTSM>
ANR8373I 001: Fill the bulk entry/exit port of library TICL3583 with all LTO
volumes to be processed within 60 minute(s); issue 'REPLY' along with the
request ID when ready.
reply 001

ANR2017I Administrator SERVER_CONSOLE issued command: REPLY 001
ANR8499I Command accepted.
TSM:LITSTSM>
ANR8810I Volume 030004L1 has been labeled in library TICL3583.
ANR8810I Volume 260AEZL2 has been labeled in library TICL3583.
ANR8810I Volume 291AEZL2 has been labeled in library TICL3583.
ANR8810I Volume 030000L1 has been labeled in library TICL3583.
ANR8810I Volume 030002L1 has been labeled in library TICL3583.
ANR8810I Volume 030001L1 has been labeled in library TICL3583.
ANR8801I LABEL LIBVOLUME process 2 for library TICL3583 completed; 6 volume(s)
labelled, 6 volume(s) checked-in.
ANR0985I Process 2 for LABEL LIBVOLUME running in the BACKGROUND completed with
completion state SUCCESS at 07:06:28 PM.
```

The tapes are now in the library and TSM knows about them but we still have to add them to the storage pool that we created:

```
TSM:LITSTSM>
define volume tape_pool0 030000L1

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE VOLUME tape_pool0
030000L1
ANR4502W No files have been defined for automatically storing sequential volume
history information.
ANR2206I Volume 030000L1 defined in storage pool TAPE_POOL0 (device class
LTO2).

TSM:LITSTSM>
define volume tape_pool0 030001L1

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE VOLUME tape_pool0
030001L1
ANR4502W No files have been defined for automatically storing sequential volume
history information.
ANR2206I Volume 030001L1 defined in storage pool TAPE_POOL0 (device class
LTO2).

TSM:LITSTSM>
define volume tape_pool0 030002L1

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE VOLUME tape_pool0
030002L1
ANR4502W No files have been defined for automatically storing sequential volume
history information.
ANR2206I Volume 030002L1 defined in storage pool TAPE_POOL0 (device class
LTO2).

TSM:LITSTSM>
```

```
define volume tape_pool0 030004L1

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE VOLUME tape_pool0
030004L1
ANR4502W No files have been defined for automatically storing sequential volume
history information.
ANR2206I Volume 030004L1 defined in storage pool TAPE_POOL0 (device class LTO2).
```

TSM has tapes in a tape pool now. Let's see if it can use one of the other tapes not in the pool to back up its database. This will tell us if TSM can talk to the tape drives in the library. Note that we will tell TSM to use one of the tapes that was not added to the tape pool. There are two tapes not in the pool and we will alternate between these two tapes for TSM database backups until the database gets larger than a single tape. If this occurs, a choice will have to be made: For example, to either add more tapes or start doing incremental dumps every night and full volumes once a month.

```
TSM:LITSTSM>
q vol

ANR2017I Administrator SERVER_CONSOLE issued command: QUERY VOLUME

Volume Name              Storage      Device      Estimated    Pct   Volume
                         Pool Name    Class Name   Capacity   Util   Status
-----------------------  -----------  ----------  ---------  -----  --------
/opt/tivoli/tsm/server/-  ARCHIVEPOOL  DISK              5.0    0.1  On-Line
 bin/archive.dsm
/opt/tivoli/tsm/server/-  BACKUPPOOL   DISK             10.0    0.0  On-Line
 bin/backup.dsm
030000L1                 TAPE_POOL0   LTO2              0.0    0.0   Empty
030001L1                 TAPE_POOL0   LTO2              0.0    0.0   Empty
030002L1                 TAPE_POOL0   LTO2              0.0    0.0   Empty
030004L1                 TAPE_POOL0   LTO2              0.0    0.0   Empty

TSM:LITSTSM>
ba db t=f dev=lto2 vol=260AEZL2

ANR2017I Administrator SERVER_CONSOLE issued command: BACKUP DB t=f dev=lto2
vol=260AEZL2
ANR0984I Process 3 for DATABASE BACKUP started in the BACKGROUND at 09:19:25
PM.
ANR2280I Full database backup started as process 3.
TSM:LITSTSM>
ANR8337I LTO volume 260AEZL2 mounted in drive DRIVE3 (/dev/IBMtape2n).
ANR0513I Process 3 opened output volume 260AEZL2.
ANR1360I Output volume 260AEZL2 opened (sequence number 1).
ANR4554I Backed up 640 of 1350 database pages.
ANR4554I Backed up 1280 of 1350 database pages.
ANR1361I Output volume 260AEZL2 closed.
ANR0515I Process 3 closed volume 260AEZL2.
ANR4502W No files have been defined for automatically storing sequential volume
history information.
ANR4550I Full database backup (process 3) complete, 1350 pages copied.
ANR0985I Process 3 for DATABASE BACKUP running in the BACKGROUND completed with
completion state SUCCESS at 09:20:05 PM.
ANR8336I Verifying label of LTO volume 260AEZL2 in drive DRIVE3 (/dev/IBMtape2-
n).
ANR8468I LTO volume 260AEZL2 dismounted from drive DRIVE3 (/dev/IBMtape2n) in
library TICL3583.

ba volh F=/opt/tivoli/tsm/server/bin/volhistory.save

ANR2017I Administrator SERVER_CONSOLE issued command: BACKUP VOLHISTORY
```

```
F=/opt/tivoli/tsm/server/bin/volhistory.save
ANR2462I BACKUP VOLHISTORY: Server sequential volume history information was
written to /opt/tivoli/tsm/server/bin/volhistory.save.
```

Now we have to tell TSM that tape_pool0 is the pool to overflow to from the default backuppool. We will also tell TSM to go ahead and migrate all the content of the backuppool at once, but to cache the files on the backuppool in case they are needed again soon.

```
TSM:LITSTSM>
upd stg backuppool next=tape_pool0 lowmig=0 cache=yes

ANR2017I Administrator SERVER_CONSOLE issued command: UPDATE STGPOOL backuppool
next=tape_pool0 lowmig=0 cache=yes
ANR2202I Storage pool BACKUPPOOL updated.
```

At this point, we've got the library defined, tapes in the library and defined to TSM, and the hierarchy of disk pool to tape pool for backups defined. We only have the default disk pool with 10M bytes of space, so we will fill that very quickly before overflowing to the tape pool. The next step is to add more disk space to the TSM server for use as disk pools and increasing the size of the TSM database. A large disk pool will allow multiple TSM clients to run backups simultaneously with good performance.

Before adding a disk device to the backuppool, make sure that the disk is formatted and mounted on the Linux system. Then, add the space to the TSM backuppool. The **format** operand will allocate the pool.002 file on the disk device mounted at /opt/tivoli/tsm/diskpool/pool2.

```
def vol backuppool /opt/tivoli/tsm/diskpool/pool2/pool.002 format=17000

ANR0984I Process 24 for DEFINE VOLUME started in the BACKGROUND at 01:48:00 PM.
ANR2491I Volume Creation Process starting for /opt/tivoli/tsm/diskpool/pool2/p-
ool.002, Process Id 24.
TSM:LITSTSM>
ANR2206I Volume /opt/tivoli/tsm/diskpool/pool2/pool.002 defined in storage pool
BACKUPPOOL (device class DISK).
ANR0986I Process 24 for DEFINE VOLUME running in the BACKGROUND processed 1
items for a total of 17,825,792,000 bytes with a completion state of SUCCESS at
01:55:22 PM.
ANR1305I Disk volume /opt/tivoli/tsm/diskpool/pool2/pool.002 varied online.
```

Before adding a disk device to be used for the TSM database, make sure that the disk is formatted and mounted on the Linux system. The **define dbvolume** command works similarly to the **define volume** command. After defining the volume to TSM, use the **extend db** command to make the space available to TSM.

```
def dbvol /opt/tivoli/tsm/diskpool/database3/db3.dsm format=2000

ANR2017I Administrator SERVER_CONSOLE issued command: DEFINE DBVOLUME
/opt/tivoli/tsm/diskpool/database3/db3.dsm format=2000
ANR0984I Process 27 for DEFINE DBVOLUME started in the BACKGROUND at 05:05:35
PM.
ANR2491I Volume Creation Process starting for /opt/tivoli/tsm/diskpool/databas-
e3/db3.dsm, Process Id 27.
TSM:LITSTSM>
ANR2240I Database volume /opt/tivoli/tsm/diskpool/database3/db3.dsm defined.
ANR0986I Process 27 for DEFINE DBVOLUME running in the BACKGROUND processed 1
items for a total of 2,098,200,576 bytes with a completion state of SUCCESS at
05:05:55 PM.

extend db 2000
```

```
ANR2017I Administrator SERVER_CONSOLE issued command: EXTEND DB 2000
ANR2248I Database assigned capacity has been extended.
```

Adding clients to TSM

We are running our TSM 5.2 clients on SLES 9, SLES 10, RHEL 4, and RHEL 5. Fortunately we did not run into any problems even though SLES 10 and RHEL 5 are not yet on the official supported environment list for the TSM 5.2 client.

On the TSM server, we added the node by connecting to the console and issuing:

```
reg node nodename password passexp=0 comp=no
```

We set client side compression of the backup objects to OFF because compression is an expensive process CPU-wise, and our Ultrium2 drives support hardware-based compression .

Tivoli Storage Manager client configuration

Remember that before you can perform a TSM client backup, you must define the client to the TSM server.

The installation of the TSM client is a straightforward rpm install:

```
rpm -ivh /root/tsm/tsmcli/linux390/*.rpm
```

We created a file in /etc/profile.d to ensure that the environment variables that the TSM client needs are defined automatically at system startup. We created a file named /etc/profile.d/tsmclient.sh which contains the following two lines:

```
export DSM_CONFIG=/opt/tivoli/tsm/client/ba/bin/dsm.opt
export DSM_DIR=/opt/tivoli/tsm/client/ba/bin
```

We made sure that the new file has execute permission by running the following command:

```
chmod +x /etc/profile.d/tsmclient.sh
```

Now we have to create the new configuration files for the TSM client. By default, the configuration files reside in /opt/tivoli/tsm/client/ba/bin. The directory contains skeleton examples, named dsm.opt.smp and dsm,sys.smp, which can be copied and altered. Keeping the configuration files in the same place as the binary files is a violation of the practice of separating of code and configuration, though. The configuration files can be relocated if you alter the DSM_CONFIG environment variable in /etc/profile.d/tsmclient.sh.

The dsm.sys file defines the options used by the TSM client, such as the IP address of the TSM server, the port name where the server is running, the areas of the file system that are excluded from being backed up, and the name of this TSM client node, as in the following example:

```
*********************************************************************
* IBM Tivoli Storage Manager
*
* Sample Client System Options file for UNIX (dsm.sys.smp)            *
*********************************************************************
*  This file contains the minimum options required to get started
*  using ITSM.  Copy dsm.sys.smp to dsm.sys.  In the dsm.sys file,
*  enter the appropriate values for each option listed below and
*  remove the leading asterisk (*) for each one.
*  If your client node communicates with multiple ITSM servers, be
```

```
*  sure to add a stanza, beginning with the SERVERNAME option, for
*  each additional server.
************************************************************************

SErvername  LITSTSM
   COMMmethod         TCPip
   TCPPort            1500
   TCPServeraddress   192.168.71.177
   exclude.dir /sys
   exclude.dir /proc
   exclude.dir /dev

NODENAME lithub

PASSWORDACCESS GENERATE
```

Note that the NODENAME does not need to be a fully qualified domain name—just the short host name will do.

The PASSWORDACCESS GENERATE statement causes the TSM client to hash the client password the first time it is used so that it does not need to be entered for every backup. This will allow automatic backups to happen later when we define a backup schedule on the TSM server.

The dsm.opt file defines which server definition will be used out of dsm.sys, as in this example:

```
************************************************************************
* IBM Tivoli Storage Manager
*
* Sample Client User Options file for UNIX (dsm.opt.smp)               *
************************************************************************
*  This file contains an option you can use to specify the ITSM
*  server to contact if more than one is defined in your client
*  system options file (dsm.sys).  Copy dsm.opt.smp to dsm.opt.
*  If you enter a server name for the option below, remove the
*  leading asterisk (*).
************************************************************************

SErvername       LITSTSM
```

After the configuration files are in place, we can perform the initial client backup, which will archive files from the whole machine, by issuing the following command:

```
/opt/tivoli/tsm/client/ba/bin/dsmc ba
```

The command prompts the user for the node name (the name in your configuration file with corresponding server side definition) and the associated password the first time. Subsequent backups will not require password authentication since we set PASSWORDACCESS GENERATE in dsm.sys.

A successful client backup will end with a status statement similar to the following example:

```
--------------------------------------------------------------------
Total number of objects inspected:   64,862
Total number of objects backed up:        6
Total number of objects updated:          0
Total number of objects rebound:          0
Total number of objects deleted:          0
```

```
Total number of objects expired:          0
Total number of objects failed:           0
Total number of bytes transferred:    63.41 KB
Data transfer time:                    0.00 sec
Network data transfer rate:        47,327.57 KB/sec
Aggregate data transfer rate:          2.43 KB/sec
Objects compressed by:                    0%
Elapsed processing time:           00:00:26
------------------------------------------------------------------------
```

We are now free to configure a **cron** job or some other means for automating backups, if we like. Note that server side configuration can also prompt clients for backups at scheduled intervals depending on the server configuration.

Configuring DB2 to back up via TSM

After the TSM client is installed and the system has had a file level backup run, copy the dsm.opt and dsm.sys configuration files to /opt/tivoli/tsm/client/api/bin64 so that the TSM client API can figure out where to point itself. If you are running on a 31-bit Linux version then copy them to /opt/tivoli/tsm/client/api/bin.

Then, set the following environment variables, which allow DB2 to use the TSM client API to dump itself to the TSM server:

```
export DSMI_CONFIG=/opt/tivoli/tsm/client/api/bin64/dsm.opt
export DSMI_DIR=/opt/tivoli/tsm/client/api/bin64
export DSMI_LOG=/opt/tivoli/tsm/client/api/bin64
```

Since our DB2 is running on SLES 10 SP1, we create a file named /etc/profile.d/db2tsmcli.sh and put these three statements in it. That way those variables are set every time the Linux system gets started.

Now we need to set up DB2 to actually dump itself to TSM.

From the DB2 instance, we set the log archive method to allow DB2 to ship its transaction logs off to TSM:

```
db2inst1@litdat02:~> db2 update db cfg for tsmtest using logarchmeth1 tsm
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
```

The following command will place the database in a backup pending state. All applications must disconnect from the database and an offline backup must be run to get a baseline backup. Once the offline backup has been run, the applications can reconnect and all future backups can be run while the database is online.

```
db2inst1@litdat02:~> db2 force applications all
DB20000I  The FORCE APPLICATION command completed successfully.
DB21024I  This command is asynchronous and may not be effective immediately.

db2inst1@litdat02:~> db2 backup db tsmtest use tsm
Backup successful. The timestamp for this backup image is : 20080522154302
```

Once the offline backup is complete, we can then allow connections to the database and run online backups using this version of the command:

```
db2inst1@litdat02:~> db2 connect to tsmtest

   Database Connection Information

 Database server        = DB2/LINUXZ64 9.1.0
 SQL authorization ID   = DB2INST1
 Local database alias   = TSMTEST
```

```
db2inst1@litdat02:~> db2 "select * from tsm_test"

CHAR50                                             DATE
-------------------------------------------------- ----------
testchars                                          -

  1 record(s) selected.

db2inst1@litdat02:~> db2 backup db tsmtest online use tsm

Backup successful. The timestamp for this backup image is : 20080522154339
```

Restoring files from TSM

Restoring files from TSM is relatively painless. Use the **dsmc** command to specify the files to restore and where to put them. It's also possible to see a list of how many versions of a file the TSM server contains using the **-pick** operand (as shown in a later example).

The following is an example of restoring a single file to its original location:

```
db2inst1@litdat02:~> dsmc rest /home/db2inst1/dsmerror.log
IBM Tivoli Storage Manager
Command Line Backup/Archive Client Interface - Version 5, Release 2, Level 2.0
(c) Copyright by IBM Corporation and other(s) 1990, 2003. All Rights Reserved.

Restore function invoked.

Node Name: LITDAT02
Session established with server LITSTSM: Linux/s390x
  Server Version 5, Release 3, Level 2.0
  Data compression forced off by the server
  Server date/time: 05/22/2008 16:13:19  Last access: 05/22/2008 16:12:46

Restoring           6,569 /home/db2inst1/dsmerror.log [Done]

Restore processing finished.

Total number of objects restored:         1
Total number of objects failed:           0
Total number of bytes transferred:     6.43 KB
Data transfer time:                       0.00 sec
Network data transfer rate:        83,616.57 KB/sec
Aggregate data transfer rate:          2.15 KB/sec
Elapsed processing time:           00:00:02
```

The following is an example of restoring a directory to a new location:

```
db2inst1@litdat02:~> dsmc rest "/home/db2inst1/sqllib/*" /home/db2inst1/sqllib_new/ -subdir=yes
```

Note: The asterisk must be enclosed within double quotation marks; otherwise, the bash shell will expand it before passing it to the **dsmc** command. Also, the trailing slash on the destination directory is required.

```
IBM Tivoli Storage Manager
Command Line Backup/Archive Client Interface - Version 5, Release 2, Level 2.0
(c) Copyright by IBM Corporation and other(s) 1990, 2003. All Rights Reserved.

Restore function invoked.

Node Name: LITDAT02
Session established with server LITSTSM: Linux/s390x
  Server Version 5, Release 3, Level 2.0
  Data compression forced off by the server
  Server date/time: 05/22/2008 16:15:57  Last access: 05/22/2008 16:15:35
```

```
ANS1247I Waiting for files from the server...
Restoring           4,096 /home/db2inst1/sqllib --> /home/db2inst1/sqllib_new/sq
llib [Done]
Restoring           4,096 /home/db2inst1/sqllib/.netls --> /home/db2inst1/sqllib
_new/sqllib/.netls [Done]
Restoring           4,096 /home/db2inst1/sqllib/adm --> /home/db2inst1/sqllib_ne
w/sqllib/adm [Done]
Restoring           4,096 /home/db2inst1/sqllib/backup --> /home/db2inst1/sqllib
_new/sqllib/backup [Done]
Restoring           4,096 /home/db2inst1/sqllib/cfg --> /home/db2inst1/sqllib_ne
w/sqllib/cfg [Done]

... output trimmed ...

Restoring             212 /home/db2inst1/sqllib/sqldbdir/sqldbbak --> /home/db2i
nst1/sqllib_new/sqllib/sqldbdir/sqldbbak [Done]
Restoring             212 /home/db2inst1/sqllib/sqldbdir/sqldbdir --> /home/db2i
nst1/sqllib_new/sqllib/sqldbdir/sqldbdir [Done]
Restoring             540 /home/db2inst1/sqllib/sqldbdir/sqldbins --> /home/db2i
nst1/sqllib_new/sqllib/sqldbdir/sqldbins [Done]
Restoring             288 /home/db2inst1/sqllib/uif/lic_comp_NODE0000.dt --> /ho
me/db2inst1/sqllib_new/sqllib/uif/lic_comp_NODE0000.dt [Done]

Restore processing finished.

Total number of objects restored:        83
Total number of objects failed:           0
Total number of bytes transferred:     7.87 MB
Data transfer time:                    0.07 sec
Network data transfer rate:        110,314.17 KB/sec
Aggregate data transfer rate:      1,612.82 KB/sec
Elapsed processing time:           00:00:05
db2inst1@litdat02:~>
```

The following is an example of restoring a file that has changed over time by picking from the available versions of the file to restore:

```
db2inst1@litdat02:~> dsmc rest "/home/db2inst1/*" -pick -inactive


TSM Scrollable PICK Window - Restore

     #    Backup Date/Time        File Size A/I  File
        ---------------------------------------------------------------------
     1. | 05/22/2008 16:19:13          6  B  A   /home/db2inst1/a_file
     2. | 05/22/2008 16:19:02          4  B  I   /home/db2inst1/a_file
     3. | 05/22/2008 16:18:49          4  B  I   /home/db2inst1/a_file
     4. | 05/22/2008 01:52:19       1.31 KB  A   /home/db2inst1/.bash_history
     5. | 05/22/2008 16:12:46       4.00 KB  A   /home/db2inst1/db2inst1
     6. | 05/22/2008 16:12:46       6.41 KB  A   /home/db2inst1/dsmerror.log
     7. | 05/22/2008 01:52:19        763  B  I   /home/db2inst1/dsmerror.log
     8. | 05/22/2008 01:52:19         35  B  A   /home/db2inst1/.lesshst
     9. | 05/22/2008 01:52:19       1.04 KB  A   /home/db2inst1/.profile
    10. | 05/22/2008 01:52:19       4.00 KB  A   /home/db2inst1/sqllib
    11. | 05/22/2008 01:52:19        693  B  A   /home/db2inst1/.viminfo
        |
        |
        |
        0---------10--------20--------30--------40--------50--------60--------7
<U>=Up  <D>=Down  <T>=Top  <B>=Bottom  <R#>=Right  <L#>=Left
<G#>=Goto Line #  <#>=Toggle Entry  <+>=Select All  <->=Deselect All
<#:#+>=Select A Range <#:#->=Deselect A Range  <O>=Ok  <C>=Cancel
pick>
```

At this point, you can select the version of the file you want to restore. If you want to pick the first revision of /home/db2inst1/a_file then enter a 3 and press Enter. Enter 0 and press Enter to process your selection. If the file currently exists on the file system, the following prompt will appear before TSM overwrites the current file:

```
--- User Action is Required ---
File '/home/db2inst1/a_file' exists

Select an appropriate action
  1. Replace this object
  2. Replace all objects that already exist
  3. Skip this object
  4. Skip all objects that already exist
  A. Abort this operation
Action [1,2,3,4,A] : 1
Restoring               4 /home/db2inst1/a_file [Done]

Restore processing finished.

Total number of objects restored:         1
Total number of objects failed:           0
Total number of bytes transferred:       28  B
Data transfer time:                     0.00 sec
Network data transfer rate:         2,485.79 KB/sec
Aggregate data transfer rate:           0.00 KB/sec
Elapsed processing time:            00:00:50
```

Restoring DB2 from TSM

First, restore the database to a temporary format from TSM:

```
db2inst1@litdat02:~> db2 list database directory
SQL1057W  The system database directory is empty.  SQLSTATE=01606
db2inst1@litdat02:~> db2 restore database tsmtest use tsm
DB20000I  The RESTORE DATABASE command completed successfully.
```

The database is now on the system but it is in a roll-forward pending state. If you are restoring the database into the same DB2 server, the same DB2 instance, with the same DB2 database name, then you can immediately roll forward the database to make it available. DB2 will be able to get all its transaction logs off of TSM if all of the above parameters match between what was backed up and the target to which you are restoring.

```
db2inst1@litdat02:~> db2 rollforward database tsmtest to end of logs

                                  Rollforward Status

 Input database alias                   = tsmtest
 Number of nodes have returned status   = 1

 Node number                            = 0
 Rollforward status                     = DB  working
 Next log file to be read               = S0000002.LOG
 Log files processed                    = S0000001.LOG - S0000001.LOG
 Last committed transaction             = 2008-05-22-19.43.40.000000 UTC

DB20000I  The ROLLFORWARD command completed successfully.

db2inst1@litdat02:~> db2 rollforward database tsmtest complete

                                  Rollforward Status

 Input database alias                   = tsmtest
 Number of nodes have returned status   = 1
```

```
 Node number                            = 0
 Rollforward status                     = not pending
 Next log file to be read               =
 Log files processed                    = S0000001.LOG - S0000001.LOG
 Last committed transaction             = 2008-05-22-19.43.40.000000 UTC

DB20000I  The ROLLFORWARD command completed successfully.
db2inst1@litdat02:~> db2 connect to tsmtest

   Database Connection Information

 Database server        = DB2/LINUXZ64 9.1.0
 SQL authorization ID   = DB2INST1
 Local database alias   = TSMTEST

db2inst1@litdat02:~> db2 "select * from tsm_test"

CHAR50                                             DATE
-------------------------------------------------- ----------
testchars                                          -

  1 record(s) selected.
```

If you are restoring the database onto a different node or a different database instance, or you are using a different database name, then you will have to work around some limitations in the DB2 and TSM integration. DB2 will not be able to get the transaction logs out of TSM, so you will have to retrieve them manually. We were not able to use the TSM command line interface (CLI) utility to do this since it appears that there is no way to point the CLI at the DB2 file space in TSM. The TSM GUI apparently does not have this limitation, so we were able to start up **dsmcad** on our DB2 system, point a Java enabled Web browser at it, and retrieve the DB2 transaction logs that way. Once we had all the DB2 transaction logs somewhere that the DB2 instance could get to them, we had to manually add read permissions to all the files by using the **chmod** command:

```
db2inst1@litdat02:~> chmod +r ./logs/*
```

Then, we were able to roll forward by pointing DB2 at the logs:

```
db2 "rollforward database newname to end of logs overflow log path (/home/db2inst1/logs/)"
db2 "rollforward database newname complete"
```

DFSMS, IBM Tape Manager, and IBM Backup and Restore for z/VM

As part of our Systems Management / Data integrity phase we installed IBM Backup and Restore Manager for z/VM (BKRM). This is part of the same suite of products in which Operations Manager resides. As you may recall, we documented our install and configuration of Operations Manager in the 2007 December Report.

BKRM performs full system backups, incremental backups and user initiated backups. This includes any CKD device defined as a MDISK in the VM directory or files stored in a SFS filepool. BKRM can dump CMS formatted minidisks at the file level and non-cms formatted disks such as a Linux system as an image.

BKRM works by processing templates you design providing the flexibility to perform system image backups, incremental file backups and specialized jobs for critical data backup using these templates. In addition, the tool provides a number of options and views when it comes to restoring data. These include, restoring by

backup ID, z/VM Guest, Volume, Volume Extent and Minidisk List. These options allow the user or system programmer to quickly locate the file or image they need to restore.

Our environment has access to an Automatic Tape Library (ATL) and so we will configure BKRM to utilize it. However, BKRM can not communicate directly with the ATL. To do so BKRM uses IBM Tape Manager and DFSMS™ removable media services. In this chapter we will discuss how we installed, configured and tested each of these products and how we set them up to work together.

These products are layered on top of each other with DFSMS at the base, IBM Tape Manager in the middle, and BKRM on top. DFSMS communicates directly with the ATL and is configured with the ATL name, real device addresses of the Tape Drivers, list of authorized users, and which tape volumes are accessible via the ATL. The DFSMS service machine, RMSMASTR, is the removable media services virtual machine that handles all ATL requests.

IBM Tape Manager is used to define and manager the tape pools, track tape expiration and request tape mounts request from DFSMS on behalf of the user. The IBM Tape Manager is made up of three main components. The tape management service machine (TMTMM), the device manager (TMDMM) and the library manager (TMLM1).

BKRM performs the actual backup and restore operations. It is made up of four components, the administrative user (BKRADMIN), the backup catalog (BKRCATLG), the controller (BKRBKUP) and the worker machines, (BKRWRK01-BKRWRK04).

In hind sight, we feel the best approach to setting up this backup solution would have been to start with DFSMS, completely configuring and testing it before moving to the next layer. Sadly, we started with BKRM and then discovered the need for IBM Tape Managaer and DFSMS since we are using an ATL.

To save you the headaches that we encountered, our test report is written as if we installed, configured and tested the components in the correct order. DFSMS, then IBM Tape Manager followed by BKRM. If you are not using a ATL you can start with the section on installing BKRM.

DFSMS

This topic discusses our experiences with the DFSMS feature of z/VM.

Configuring and testing DFSMS

DFSMS is an orderable feature of z/VM, which available at no charge to customers of z/VM 5.3 and 5.4. When we initially built our z/VM system, DFSMS was installed using a shared file system (SFS) for the configuration files. Durning our customization of DFSMS, we needed to update files BRUNOMNT CTLDATA, DGTVCNTL DATA, FSMRMSHR CTLDATA and RMCONFIG DATA, all of which reside in the shared file system VMSYS:DFSMS.CONTROL.

Configuring the BRUNOMNT CTLDATA file: The BRUNOMNT CTLDATA file controls which z/VM users are permitted to communicate with DFSMS. Later, when configuring IBM Tape Manager, it also will need to be added to the list of authorized users.

Configuring the DGTVCNTL DATA file: The DGTVCNTL DATA file is where the ATL is defined to DFSMS. In the following example, lines 152 and 153 show the definition for our ATL, SUBVTS1 to DFSMS:

```
136 *====================================================================
137 *
138 *  PART 5 - PARAMETERS PERTAINING TO THE RMS MASTER VIRTUAL MACHINE
139 *
140 *  To perform Removable Media Services, the parameters specified
141 *  in section 1 must be specified.  In addition to that, the
142 *  following parameters are required:
143 *
144 *           DFSMSRM_MASTER_VM
145 *           RM_AUTO_LIBRARY
146 *====================================================================
147
148  DFSMSRM_MASTER_VM        RMSMASTR       * Userid of RMS master
149
150  RM_ACCOUNTING            Y              * Accounting function on
151
152 *RM_AUTO_LIBRARY  710_1 10002 OPERATOR   * Automated
153  RM_AUTO_LIBRARY  SUBVTS1  13038 OPERATOR   * SUBVTS1 native library
```

Configuring the FSMRMSHR CTLDATA file: The FSMRMSHR CTLDATA file is where we specify which tape volumes can be accessed by DFSMS. Our system NODE ID is LTICVM9 and the tape volumes we will be using are LX00111–LX0150, as in the following example:

```
* This file specifies which tape volumes and SCRATCH categories
* can be accessed by each system.  The format is:
*   node   <volid<-volid>> <SCRATCHx<-SCRATCHy>> ...
* where:
*   node      is the node of the processor, or ALL for all processors
*   volid     is a tape volume label
*   SCRATCHx  is a SCRATCH category (n = 0 to F)
*
* If a range of volumes is specified, the second volume must be higher
*   (standard sort sequence) than the first volume.
* The file can be common for multiple processors.
 LTICVM9 LX0111-LX0150
```

If you attempt to mount a tape volume not defined in FSMRMSHR CTLDATA, the following error will occur when you request the mount:

```
FSMBEC2103E Request identifier = FSMRMSHR:
 Installation-wide exit 8 indicates that processing should not be allowed to continue
```

Configuring the RMCONFIG DATA file: The RMCONFIG DATA file defines the real device addresses of the tape drives in our ATL, as in the following example:

```
*      If a range, the beginning and ending addresses of the range
*      must be separated by a dash (-), and optionally by one or
*      more spaces.  Ranges can not span lines.
*
*      The ending address of the range must be greater than the
*      beginning address.
* ====================================================================
 B5A0-B5A3                  * SUBVTS1 native library 3590 drives
 B5F4-B5FF                  * SUBVTS1 native library 3590 drives
```

We use RACF in our environment so we need to permit users who will be allowed to issue tape mounts, as in the following example. Later, when setting up BKRM, we will need to permit TMTMM.

```
RAC PERMIT STGADMIN.RM.* CLASS(FACILITY) ID(BEYER) ACCESS(ALTER)
```

Verifying the setup of DFSMS removable media services

The best way to verify the setup is to mount a tape in the ATL. Before we can do this, we have a couple of things we need to do:

1. If not already done, physically load the tapes into the ATL.
2. Autolog the DFSMS removable media services service machine, RMSMASTR:

   ```
   XAUTOLOG RMSMASTR
   ```

This is also a good time to add RMSMASTR to your system autolog procedure. In our case, we use Operations Manager to autolog our z/VM guests after system IPL. This is done via a MONITOR statement in Operations Manager that will bring up DFSMS after the system is IPLed. For example:

```
DEFMMON NAME(DFSMS),USER(RMSMASTR),ACTION(XAUTOLOG),PARM(RMSMASTR),+
DELAY(6)
```

Testing tape mounts

From an account authorized to issues tape mounts, we used the mount command, DFSMSRM. In this example, we have requested tape volume LX0111 to be mounted read/write at virtual device 181:

```
dfsmsrm mount vol lx0011 ( attach * READWRITE vdev 181
Ready; T=0.01/0.01 16:59:10
TAPE B5F4 ATTACHED TO BEYER 0181
 16:59:18  * MSG FROM RMSMASTR:    FSMBCR2120I Request 21 complete; volume LX0011,
 category 000F, mounted on device B5F4 in library SUBVTS1
 16:59:18  * MSG FROM RMSMASTR:    FSMBEC2125I Request 21: device B5F4 attached
 to BEYER as 0181; access mode = READWRITE
```

A simple tape query will verify that the tape drive has been attached to the user as vdev 181:

```
q ta
TAPE B5F4 ATTACHED TO BEYER    0181 R/W
```

For more information about configuring DFSMS removable media services, see IBM Redbook, *Lights out! Advanced Tape Automation using VM/ESA®*, GC24-4347, available from the IBM Redbooks Web site at www.ibm.com/redbooks/.

IBM Tape Manager

This topic discusses our experiences with IBM Tape Manager.

Installing and configuring IBM Tape Manager

Tape Manager for z/VM Program Directory is clear and easy to follow, allowing for an error free installation. After the installation was complete, we followed the instructions in *Tape Manager for z/VM Installation and Administration Guide* to customize the environment. The Tape Manager documentation is available at www.ibm.com/software/stormgmt/zvm/tape/library.html.

In this topic, we will describe the configuration files that we needed to modify to allow Tape Manager to communicate with DFSMS. However, before we do that, we need to understand what each of the IBM Tape Manager Service machines do.

- TMTMM is the service machine that manages the tape pool database. It also responds to users requests.

- TMDMM is the device management machine and is responsible for the control of the physical tape. When a tape mount is received, the tape drive is attached to TMDMM and the tape volume is mounted. At that point, TMDMM is responsible for verifying that the mounted volume is correct. Upon verification, it uses the CP GIVE command to transfer the drive and tape to the requestor, thus guaranteeing that the correct volume was mounted.
- TMLM1 is the library management service machine. It is the interface from Tape Manager to DFSMS and communicates directly with RMSMASTR. Therefore, TMLM1 must be authorized to DFSMS for tape operations. In our environment, we accomplished this with the following RACF command:

  ```
  RAC PERMIT STGADMIN.RM.* CLASS(FACILITY) ID(TMLM1) ACCESS(ALTER)
  ```

Configuring the SYS CONFIG file: The SYS CONFIG file resides on TMTMM's 191 minidisk. Most of the statements in the SYS CONFIG are intuitive; however, we did run into a problem with the DEVPOOL statement. Since we attempted to customize IBM Tape Manger before the completion of DFSMS, we did not realize that the name used for the ATL on the DEVPOOL statement for Tape Manager must match that used on the RM_AUTO_LIBRARY statement in the DFSMS file DGTVCNTL DATA. You can see from our DEVPOOL statement below that we elected to name our device pool the same name as our ATL to keep it simple. The tape drives available to Tape Manager are also defined in this statement of the SYS CONFIG file.

```
DEVPOOL    SUBVTS1    ATL SUBVTS1 B5F0-B5FF
```

Configuring the SYS MEDIA file: The SYS MEDIA file on TMTMM's 191 minidisk defines the media type that IBM Tape Manager will use along with the device pool. The example below defines a media type of 3590K in DEVPOOL SUBVTS1:

```
MEDIA=3590K DEVPOOL=SUBVTS1  MODE=RW
```

Starting Tape Manager's service machines

From an authorized user ID, we issued the XAUTOLOG commands:

```
XAUTOLOG TMTMM
XAUTOLOG TMDMM
XAUTOLOG TMLM1
```

In our environment, we used Operations Manger to autolog IBM Tape Manager with the monitor statements shown below. However you could use AUTOLOG2 or another automation tool to do this.

```
DEFMMON NAME(TMTMM),USER(TMTMM),ACTION(XAUTOLOG),PARM(TMTMM),+
DELAY(7)
DEFMMON NAME(DMTMM),USER(DMTMM),ACTION(XAUTOLOG),PARM(DMTMM),+
DELAY(7)
DEFMMON NAME(TMLM1),USER(TMLM1),ACTION(XAUTOLOG),PARM(TMLM1),+
DELAY(7)
```

Sending commands to Tape Manager

Commands and queries are sent to IBM Tape Manager from an authorized user ID via the TAPCMD EXEC. For example, to add tapes to the tape pool with the TAPEADD command, we would prefix the TAPEADD command with the TAPCMD EXEC, like so:

```
TAPCMD TAPEADD parameters
```

Defining a tape pool

Before a tape mount can take place, the tape pool must be defined and tape volumes added to it. The tape pool is defined by specifying the owner, media type, ATL name, and retention period.

We defined our tape pool as TICLPOOL, owner BEYER, containing 3590K tape media, using ATL SUBVTS1, and a maximum retention of 63 days. Why 63 days? We plan on performing full system backups every month and incremental backups daily. We also want to insure that we have two months of backup data available at any time. Twice a year, between December/January and July/August, we have 31-day months back-to-back. So, we figured, 31 days + 31 days = 62, plus one day to be safe. This way, we ensure that we have least two complete months' worth of full system backups available.

The tape pool definition is performed with the POOLDEF command:

```
TAPCMD POOLDEF BEYER TICLPOOL MEDIA 3590K RETNMAX 63
Ready; T=0.01/0.01 13:48:59
 EUMTAP0010I POOLDEF message 000126 received from BEYER.
 EUMTAP0083I POOLDEF request 000126 complete - RC 0.

TAPCMD POOLMOD BEYER TICLPOOL DEVPOOL SUBVTS1
Ready; T=0.01/0.01 14:00:22
 EUMTAP0010I POOLMOD message 000133 received from BEYER.
 EUMTAP0402I Tape pool BEYER TICLPOOL has no tapes and no free pool.
 EUMTAP0083I POOLMOD request 000133 complete - RC 0.
```

The above procedure could have been done with a single command, however we wanted to demonstrate the use of the pool modifier command POOLMOD.

Adding tapes to the tape pool

With the pool definition done, we are ready to add tapes to the pool. When doing so, not only do we specify the tape volumes, but also the media type and which device they use. The TAPEADD command below adds volumes LX0111-LX0118 with a media type of 3390K using device ALT SUBVTS1.

```
TAPCMD TAPEADD VOL LX0111-LX0118 POOL BEYER TICLPOOL MEDIA 3590K FREE NOHOLD ATL SUBVTS1
Ready; T=0.01/0.01 14:10:02
 EUMTAP0010I TAPEADD message 000142 received from BEYER.
 EUMTAP0083I TAPEADD request 000142 complete - RC 0.
```

Note: The tape volumes added to the pool must reside within the tape range that was defined to DFSMS in FSMRMSHR CTLDATA. Additional tapes can be added to the file at any time.

We can now verify that the tapes were correctly added with the TAPEQRY command:

```
TAPCMD TAPEQRY POOL BEYER TICLPOOL SHORT
Ready; T=0.01/0.01 14:11:03
 EUMTAP0010I TAPEQRY message 000144 received from BEYER.
 Volume Owner    Name     Flags  MEDIA  Exp Date R/W ID   R/W Date Dev  DevT
 LX0111 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0112 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0113 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0114 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0115 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0116 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0117 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0118 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 EUMTAP0083I TAPEQRY request 000144 complete - RC 0.
```

Verifying the setup of IBM Tape Manager

At this point, we are ready to test mounting a tape. In the following example, a request is made for a tape from pool TICLPOOL to be mounted and the drive attached to our test user, BEYER. This is done with the TAPEMNT command:

```
TAPCMD TAPEMNT SCR BEYER TICLPOOL
Ready; T=0.01/0.01 14:17:11
 EUMTAP0072I Mount request 000146 received.
 EUMTAP0079I Mount request 000146 pending.
Ready; T=0.01/0.01 14:17:53
 EUMTAP0113I Volume LX0118 is READY on device B5FA.
Tape 0181 attached
 EUMTAP0109I Device B5FA attached to BEYER as 0181 with volume LX0118.
```

Once the drive is attached to the user, we can issue tape commands to verify that we can read from and write to the tape. When done testing, we need only to detach tape 181 from the user and Tape Manager will update the tape status in the database and return the tape to the ATL for safe keeping.

After the tape has been detached, we check the pool with the TAPEQRY command. Note that tape LX0118 is no longer FREE and is set to expire on June 13, 2008. Also listed is the user who last wrote to the tape and the drive that was used.

```
TAPCMD TAPEQRY POOL BEYER TICLPOOL SHORT
Ready; T=0.01/0.01 14:19:57
 EUMTAP0010I TAPEQRY message 000148 received from BEYER.
 Volume Owner    Name     Flags  MEDIA  Exp Date R/W ID   R/W Date Dev  DevT
 LX0111 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0112 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0113 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0114 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0115 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0116 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0117 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0118 BEYER    TICLPOOL AUNNIN 3590K  20080613 BEYER    20080411 B5FA 3590
 EUMTAP0083I TAPEQRY request 000148 complete - RC 0.
```

Since this was only a test, we want to free the volume in the pool with the TAPEMOD command:

```
TAPCMD TAPEMOD VOL LX0118 STATUS FREE NODSE
Ready; T=0.01/0.01 14:22:38
 EUMTAP0010I TAPEMOD message 000150 received from BEYER.
 EUMTAP0083I TAPEMOD request 000150 complete - RC 0.
```

Using the TAPEQRY command, we can verify that tape is now free with an expire date of 00000000:

```
TAPCMD TAPEQRY POOL BEYER TICLPOOL SHORT
Ready; T=0.01/0.01 14:22:42
 EUMTAP0010I TAPEQRY message 000151 received from BEYER.
 Volume Owner    Name     Flags  MEDIA  Exp Date R/W ID   R/W Date Dev  DevT
 LX0111 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0112 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0113 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0114 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0115 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0116 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0117 BEYER    TICLPOOL AFNNIN 3590K  00000000          00000000
 LX0118 BEYER    TICLPOOL AFNNIN 3590K  00000000  20080411 B5FA 3590
 EUMTAP0083I TAPEQRY request 000151 complete - RC 0.
```

For more information about IBM Tape Manager commands, see *Tape Manager for z/VM User's Guide and Reference* at www.ibm.com/software/stormgmt/zvm/tape/library.html.

IBM Backup and Restore Manager for z/VM

With DSFMS and Tape Manager configured, installing and configuring IBM Backup and Restore Manager for z/VM (BKRM) with an ATL will be straightforward. Before we start with the install, we will describe the five types of service machines that make up BKRM.

- BKRSVSFS, although not actually part of BKRM, is the Shared File System (SFS) server that will be used to contain the catalog files used by Backup Manager.
- BKRADMIN is the administrative user ID used to control and configure BKRM.
- BKRCATLG manages the catalog.
- BKRBKUP manages backup and restore tasks and responds to user requests.
- BKRWRK01–BKRWRK04 perform the actual backup and restore operations. When a backup or restore is initiated, BKRBKUP assigns the work to one of these BKRWRK*xx* service machines.

Installing IBM Backup and Restore Manager

We suggest that you review *Backup and Restore Manager for z/VM Program Directory* and *Backup and Restore Manager for z/VM Administration Guide*. However, for the actual installation, we recommend the procedure documented in the presentation, *Getting Started with IBM Backup and Restore Manager for z/VM*. This presentation, as well as the Program Directory and Administration Guide, are available at www.ibm.com/software/stormgmt/zvm/backup/library.html.

The Getting Started presentation is a well written, step-by-step process detailing the installation and configuration of BKRM. It also contains detailed instructions on creating the SFS server that will contain the catalog.

Rather the repeat what has been done, we will assume that have followed the instructions in the presentation "Getting Started with IBM Backup and Restore Manager for z/VM." If you are not using an ATL in your environment, the presentation will have all the steps you need to get BKRM up and running.

If you are planning on using an ATL in your environment, follow the steps in the document, "Getting Started with Backup and Restore Manager for z/VM" to slide 41, "Configuration is Complete." Then, follow the documentation below to configure BKRM to work with IBM Tape Manger and DFSMS.

Tailoring IBM Backup and Restore Manager to work with IBM Tape Manager

In this topic, we discuss how to configure BKRM to work with Tape Manager. Chapter 2, "Configuring Backup and Restore Manager to work with IBM Tape Manager," of *Backup and Restore Manager for z/VM Administration Guide* details the process. The following is a summary of those steps:

1. Authorize BKRM's service machines to use IBM Tape Manager.
2. Modify BKRSYSTM CONFIG enabling the use of IBM Tape Manager.
3. In order for BKRM to submit tape requests to IBM Tape Manger it needs the authority to access the tape pool. This is done with the IBM Tape Manager command POOLACC (pool access). The following POOLACC command authorizes all of the BKRM service machines for access to the Tape Manager pool:

```
TAPCMD POOLACC BEYER TICLPOOL USER BKRADMIN BKRBKUP BKRCATLG BKRWRK01 BKRWRK02
  BKRWRK03 BKRWRK04 TAPE
Ready; T=0.01/0.01 14:29:41
 EUMTAP0010I POOLACC message 000153 received from BEYER.
 EUMTAP0083I POOLACC request 000153 complete - RC 0
```

Now that BKRM has access to the Tape Manager tape pool, we need to tell it to use Tape Manager for all tape requests and not to look for standalone tape drives. This is done with the BKRSYSTM CONFIG file on BKRADMIN's 192 minidisk in the Tape Handling section. The following sample section of BKRSYSTM CONFIG shows the changes we made to the Tape Handling section:

```
*   COMMENT OUT THIS NEXT STATEMENT
*Tape_Handled_Via_EUM = 0
*
* Handshaking with Tape Manager; Tape Manager in RMM mode:
*  ADD THESE STATEMENTS
Tape_Handled_Via_EUM = 1
EUM_Pool_Owner = BEYER
EUM_Pool_Name  = TICLPOOL
*
```

For details about these steps, see the section "Configuring Backup and Restore Manager to work with Tape Manager" in chapter 2 of *Backup and Restore Manager for z/VM Administration Guide*.

Starting Backup and Restore Manager

Now that we have completed the configuration of BKRM, we are ready to autolog the service machines and run a test.

```
XAUTOLOG BKRSVSFS
```

Once the SFS server has completed initialization, we can autolog the BKRCATLG and BKRBKUP service machines.

Just like with DFSMS and IBM Tape Manager, we are using Operations Manager to autolog the service machines. The following are the monitor statements that we added to Operations Manager to perform this task:

```
DEFMMON NAME(BKSFS),USER(BKRSVSFS),ACTION(XAUTOLOG),PARM(BKRSVSFS),+
DELAY(8)
DEFMMON NAME(CATLG),USER(BKRCATLG),ACTION(XAUTOLOG),PARM(BKRCATLG),+
DELAY(10)
DEFMMON NAME(BKUP),USER(BKRBKUP),ACTION(XAUTOLOG),PARM(BKRBKUP),+
DELAY(10)
```

BKRBKUP will autolog the BKRWRK*xx* service machines as tasks are assigned to them.

Checking the status of the servers

From the BKRADMIN user ID, we can check the status of the catalog and BKRBKUP server by sending the status request via a CP SMSG. The following examples show how to query the status of the BKRBKUP and BKRCATLG servers:

```
cp smsg bkrbkup status
Ready; T=0.01/0.01 13:42:09
 SVM Name   : BAKSRVR - 5697-J06 IBM Backup and Restore Manager for z/VM - Master
   Backup SVM - Version 1.2.0
 Compiled on: 20 Sep 2006 - 12:10:15
 SVM Owner  : System Administrator - dbeyer@us.ibm.com
 SVM Started: Friday, 16 May 2008 09:26:32
 Catalog SVM: BKRCATLG
```

```
cp smsg bkrcatlg status
Ready; T=0.01/0.01 13:42:45
 SVM Name    : CATSRVR - 5697-J06 IBM Backup and Restore Manager for z/VM - Backup
   Catalog SVM - Version 1.2.0
 Compiled on : 18 Jan 2007 - 17:04:50
 SVM Owner   : System Administrator - dbeyer@us.ibm.com
 SVM Started : Friday, 16 May 2008 09:25:32
 Backup SVM  : BKRBKUP
 Catalog base: BKRSFS:BKRCATLG.
 Userid    Storage Group  4K Block Limit  4K Blocks Committed  Threshold
```

Creating a backup template

Backup and Restore Manager uses template files to define the backup jobs. You can create customized templates for each type of backup that you want to run.

Our IT environment consists of two z/VM systems on separate CPCs for HA. Linux guests are strategically placed so that if one z/VM system is down, the workload can continue to run on Linux guests on the other z/VM system. We created a template for each z/VM system containing just the Linux guests running on that particular system. This allows us to shutdown all the Linux guests on one system and get a clean image backup and continue to process work on the other system. When the backup has completed on system A, we then IPL the Linux guests and start the process on system B. With most of our guests being Linux systems, we need to shutdown the guests to get a clean backup.

In addition, we only run BKRM on system A. To allow the backup of systems on system B from system A, all z/VM users are defined in a common, shared z/VM directory, allowing us to run all backups from a single z/VM system. In a future test report, we will discuss how we setup z/VM with a shared directory and shared RACF using CSE.

The template files reside on the BKRADMIN 199 minidisk (which we have accessed as the E disk). Complete details on how to set up a template is described in Appendix B of *Backup and Restore Manager for z/VM Administration Guide*. In this test report, we will focus on the include/exclude section of the template.

The following example shows a segment of the include/exclude section for our backup template for system A:

```
  FUNCTION   MEDIATYPE  OWNER   VDEV VOLUME DEVTYPE START END SIZE RESERVED
 |----------|----------|--------|--|----|------|--------|--|----------|--|

  INCLUDE    MINIDISK   LITSGFW2 =     * *      *        =  *          =
  INCLUDE    MINIDISK   USER     =     * *      *        =  *          =
  EXCLUDE    MINIDISK   USER     =  5*   *      *        =  *          =
```

We have defined an entry for each of the Linux systems that run on system A. The first statement specifies that all minidisks associated with the Linux guest, regardless of the virtual address, are to be backed up. The last two entries define the backup criteria for the z/VM user, USER. The second to last line specifies that all minidisks for USER should be backed up. The last statement excludes any minidisk that has a virtual address starting with a 5. The include/exclude section is processed from top to bottom allowing the exclude statement to override the previous include statement. We have setup a separate template that deals with the 5*xxx* devices for this user. These volumes are our master Linux images that we use for cloning.

Reviewing templates

BKRM provides a review facility that will process the template file and output the actual minidisks that will be backed up. It will also verify the syntax of the template allowing you to make corrections before submitting the backup job.

Submitting a template for review simply involves sending an SMSG to BKRBKUP with the template name you want processed. Any user with Backup and Restore Manager system administrator authority can process a template for review using the command SMSG BKRBKUP REVIEW *templatename*. The following example processes the template name LIT_VM9 for review:

```
SM BKRBKUP REVIEW LIT_VM9
Ready; T=0.01/0.01 11:44:17
 BKRBAK8529I Processing REVIEW LIT_VM9 command for BKRADMIN.
RDR FILE 3643 SENT FROM BKRBKUP  PUN WAS 0115 RECS 0070 CPY  001 A NOHOLD NOKEEP
 BKRMAK8559I INCLUDE / EXCLUDE processing for job LIT_VM9 selected 70 objects
 BKRMAK8559I for backup processing.
 BKRMAK8563I Worker count for job LIT_VM9 has been set to 1.
 BKRMAK8568I CMS files will be filterd against file mask "* * *".
 BKRMAK8566I SFS filespaces will be filtered with path mask "*".
 BKRMAK8583I Sending results to BKRADMIN for review.
 File LIT_VM9 JOB D1 sent to BKRADMIN at LTICVM9 on 05/22/08 11:44:18
 Return code "0" from command REVIEW LIT_VM9 at 05/22/08 11:44:18.
```

Output from the review job is returned to the reader of the user who requested the template review. In the LIT_VM9 template, we specified that we wanted all minidisks for LITSGFW2 to be backed up:

```
INCLUDE    MINIDISK   LITSGFW1 =     * *      *        =  *          =
```

We can see in the review job output below how that single line in the template has expanded into 3 separate backup tasks. The fist line, starting with `DUMPEDF` (Enhanced Disk Format), specifies that a file-level backup will take place for the 191 disk. The second and third lines, starting with `DUMPCKD`, specify that a CKD image dump will take place for the 201 and 202 minidisks:

```
DUMPEDF LITSGFW2 0191 $$FMASK$$ $$DRIVER$$
DUMPCKD LITSGFW2 0201 $$DRIVER$$
DUMPCKD LITSGFW2 0202 $$DRIVER$$
```

Using template files allows you to quickly define which users and minidisks you want to back up without knowing the minidisk details for each user. Using templates also allows your system administrators the freedom to add and delete minidisks from the directory without having to update the backup templates each time a change is made. Only when new users are added to the directory would you need to update a template. However, you could specify a template that simply dumps all minidisk for a certain type of user, such as `LIT*`.

Submitting a backup: After viewing the review job output and verifying no errors occurred, you are ready to submit the backup. This is done in the same manner as reviewing the template except that you change the word *review* to *submit*.

We have created a sample template that backs up two users, BEYER and USER. Both of these users contain both CMS minidisks and Linux images. We have previously tested the template with a review and now we are ready to submit the backup. Any user with system administrator authority can submit the backup. In the following example, the backup was submitted from the BKRADMIN user ID and both USER and BEYER were logged off at the time.

```
sm bkrbkup submit system
Ready; T=0.01/0.01 14:04:59
 BKRBAK8532I Processing SUBMIT SYSTEM command for BKRADMIN at 05/22/08 14:04:59

 BKRMAK8559I INCLUDE / EXCLUDE processing for job SYSTEM selected 23 objects
 BKRMAK8559I for backup processing.
 BKRMAK8563I Worker count for job SYSTEM has been set to 1.
 BKRMAK8571I Instance tracking started for new job SYSTEM; the initial instance
 BKRMAK8571I number is "00000001".
 BKRMAK8568I CMS files will be filterd against file mask "* * *".
 BKRMAK8566I SFS filespaces will be filtered with path mask "*".
 BKRMAK8584I Sending SYSTEM JOB D to worker task BKRWRK04.
 File SYSTEM JOB D1 sent to BKRWRK04 at LTICVM9 on 05/22/08 14:04:59
 Return code "0" from command SUBMIT SYSTEM at 05/22/08 14:04:59.
 14:05:00  * MSG FROM BKRWRK04: BKRWRK9080I Worker Task: Entering processing
 loop at 14:05:00. Idle timeout is set to +00:02:00.
```

How backup jobs are processed: The following example outlines the general sequence of events to process a backup job:

1. When the job is submitted, BKRBKUP assigns the backup task to one of the worker machines, BKRWRK04 in this example.
2. The worker machine contacts IBM Tape Manager requesting a free tape from the TICLPPOL pool to be mounted and attached to the worker.
3. IBM Tape Manager service machine TM1LM forwards the tape mount request to DFSMS, specifying which volume to mount.
4. IBM Tape Manager requests DFSMS to attach the tape drive with the requested volume to Tape Manager's TMDMM service machine.
5. DFSMS Removable Media Services machine, RMSMASTR, attaches a free tape drive to itself and requests that the specified tape volume be mounted by the ATL. This is why, if you query your tape drives immediately after submitting the backup, you see the tape drive attached to RMSMASTR, for instance:

   ```
   q ta
   TAPE B5F6 ATTACHED TO RMSMASTR B5F6 R/W
   Ready; T=0.01/0.01 14:05:41
   ```

6. Once the tape volume has been mounted, RMSMASTR transfers the tape drive with the volume mounted to TMDMM, IBM Tape Manager's device manager. At this point, IBM Tape Manager verifies that the correct volume has been mounted.
7. TMDMM uses CP GIVE to give the tape and mounted volume to the BKRM worker machine that initiated the request (BKRWRK04, in this example).

   If we query that tape drive at this point we can see that the tape drive has been attached to BKRWRK04 from TMTMM.

   ```
   q ta
   TAPE B5F6 ATTACHED TO BKRWRK04 0181 R/W GIVEN BY TMDMM
   Ready; T=0.01/0.01 14:06:39
   ```

Checking backup status: At any point after the backup has been submitted, you can query the status of the backup by sending a status query to BKRBKUP. The results in the example query below indicate that worker BKRWRK04 is processing step 2 of 23, a CMS file-level back of BEYER's 0291 disk:

```
sm bkrbkup status
Ready; T=0.01/0.01 14:06:18
 SVM Name   : BAKSRVR - 5697-J06 IBM Backup and Restore Manager for z/VM - Master
  Backup SVM - Version 1.2.0
 Compiled on: 20 Sep 2006 - 12:10:15
 SVM Owner  : System Administrator - dbeyer@us.ibm.com
 SVM Started: Friday, 16 May 2008 09:26:32
```

```
 Catalog SVM: BKRCATLG
   BKRWRK04 - Processing job SYSTEM/00000001, task 2 of 23; EDF file-level back
  up of MDISK BEYER 0291 since 14:06:14 on Thursday, 22 May 2008.
```

In the following status query, we see that BKRWRK04 is performing a CKD image backup of BEYER's 0201 disk, which is step 4 of 23:

```
sm bkrbkup status
Ready; T=0.01/0.01 14:07:02
 SVM Name   : BAKSRVR - 5697-J06 IBM Backup and Restore Manager for z/VM - Master
  Backup SVM - Version 1.2.0
 Compiled on: 20 Sep 2006 - 12:10:15
 SVM Owner  : System Administrator - dbeyer@us.ibm.com
 SVM Started: Friday, 16 May 2008 09:26:32
 Catalog SVM: BKRCATLG
 Worker info as of 14:07:02:
   BKRWRK03 - Logged out at 12:35:15 on Friday, 16 May 2008.
   BKRWRK04 - Processing job SYSTEM/00000001, task 4 of 23; CKD image backup of
  MDISK BEYER 0201 since 14:06:25 on Thursday, 22 May 2008.
```

Reviewing the backup: When the backup has completed, two files will be returned to the submitter's reader: A summary of activity and a console log from the worker machine performing the backup.The following example shows these files in the reader after completion of backups:

```
BKRADMIN RDRLIST  A0  V 164  Trunc=164 Size=2 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Class User  at Node     Hold  Records  Date
      SYSTEM   20080522 CON T BKRWRK04 LTICVM9  NONE      142  5/22
      WORKER   OUTPUT   CON T BKRWRK04 LTICVM9  NONE       22  5/22
```

The first file in the reader, SYSTEM 20080522, is the console log from the worker machine. It contains details of each step performed during the backup. The following sample segment of output shows that each step of the backup completed with a return code of 0. Also noted is the type of backup performed.

```
BEYER    0191  RR EDF 4096 BEY191 00010800 00004659 00000060 00000060 00000126 1
4:05:00 05/22/08 14:06:13 05/22/08 Dump OK.
BEYER    0291  RR EDF 4096 BEY191 00018000 00000182 00000100 00000100 00000097 1
4:06:17 05/22/08 14:06:18 05/22/08 Dump OK.
BEYER    0391  RR EDF 4096 USR191 00009000 00003211 00000050 00000050 00000132 1
4:06:21 05/22/08 14:06:23 05/22/08 Dump OK.
BEYER    0201  RR CKD 3390 3990   00003339     0015 00058786 00050084 00050085 1
4:06:27 05/22/08 14:10:32 05/22/08 Dump OK.
```

For BEYER 291 and 391, an EDF (enhanced disk format) backup, or file-level backup, was performed. For BEYER's 201, a Linux image, a CKD image dump was performed.

The file name assigned to the console log is set within the backup template on the following CP_QUIET line:

```
CP_QUIET SPOOL CONSOLE CLOSE NAME SYSTEM   $$SDATE$$
```

The second file in the reader, WORKER OUTPUT, is the summary report. It summarizes the number of tasks, backup type, and highest return code for each type of backup. The following sample shows that a total of 23 tasks were performed, 4 of them were of type CKD, 19 were of type EDF, and the highest return code for EDF backups was 4:

```
*** End-of-Job Summary:
***
*** Start time: 05/22/08 14:05:00
*** Ended time: 05/22/08 14:19:33
***
```

```
*** DUMPCKD tasks, Max RC: 4, 0
*** DUMPEDF tasks, Max RC: 19, 4
*** DUMPSFS tasks, Max RC: 0, 0
*** RESTORE tasks, Max RC: 0, 0
***
```

Restoring data

BKRM provides a number of avenues for restoring data. The easiest is to use one of the catalog browser interface routines: BKRJOB, BKRLIST, BKRUSER or BKRVOL. Each panel-driven browser gives a different view of the backup catalog. Details for each can be found in Appendix A of *Backup and Restore Manager for z/VM User's Guide*.

Restoring CMS files to the reader: In this example, we are using BKRLIST to restore the PROFILE EXEC on LITDAT02's 191 disk. We begin by launching the catalog browser view BKRLIST from administrative user BKRADMIN, which displays a panel that lists every backed up file in the catalog, as shown in Figure 91. In this example, there are 99962 files in the catalog (top, right corner of the panel). These are only EDF, or CMS files, stored in the backup catalog. CKD image backups are not displayed in this view. We cover that in another example.

```
                            Files for owner(s): *
 Selection: Name: * Type: * Mode: *                99962 of 99962 shown
 Current filters: Name: *        Type: *        Mode: * Owner: *_

 Owner   Filename Filetype Fm  Date     Time   Device or Path
MAINT    ATS010I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS011I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS013I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS017I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS018I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS019I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS021I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS022I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS023I  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS024W  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS025W  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS075E  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS076E  HELPMSG  1 04/12/13 13:25:22 019D
MAINT    ATS077E  HELPMSG  1 04/12/13 13:25:23 019D
MAINT    ATS078E  HELPMSG  1 04/12/13 13:25:23 019D
MAINT    ATS079I  HELPMSG  1 04/12/13 13:25:23 019D
MAINT    ATS080I  HELPMSG  1 04/12/13 13:25:23 019D
 1= Help                      3= Quit        4= Return      5= Sort Up
 6= Sort down   7= Backward   8= Forward    10= Restore    11= Details
```

Figure 91. The BKRLIST panel

Looking for LITDAT02's PROFILE EXEC in the list of 99962 files might seem a bit daunting. To remedy this, enter the filter string `LIT*` next to the **Owner** tag in the top, right corner of the panel. The new list of 44 files, as shown in Figure 92 on page 238, is much easier to work with. The PROFILE EXEC we are looking for is at the top of the list.

```
                            Files for owner(s): *
 Selection: Name: * Type: * Mode: *                    44 of 99962 shown
 Current filters: Name: *        Type: *        Mode: * Owner: LIT*_

 Owner   Filename Filetype Fm  Date     Time   Device or Path


LITDAT02 PROFILE  EXEC     1 06/09/22 09:40:59 0191
LITDCON2 PROFILE  EXEC     1 06/09/22 09:40:59 0191
LITRDAT1 BACKUP   PARM     1 07/06/01 17:19:02 0191
LITRDAT1 RESCUE   EXEC     1 07/06/01 17:06:10 0191
LITRDAT1 RHEL4U5  IMAGE    1 07/06/01 16:46:27 0191
LITRDAT1 RHEL4U5  INITRD   1 07/06/01 16:46:23 0191
LITRDAT1 RHEL4U5  PARM     1 07/06/01 17:31:59 0191
LITRDAT1 RHEL5GA  IMAGE    1 07/06/19 20:24:17 0191
LITRDAT1 RHEL5GA  INITRD   1 07/06/19 20:24:32 0191
LITRDAT1 RHEL5GA  PARM     1 07/06/19 20:24:42 0191
LITRLOG1 PROFILE  EXEC     1 06/09/22 09:40:59 0191
LITRSMB1 PROFILE  EXEC     1 06/09/22 09:40:59 0191
LITRWAS1 PROFILE  EXEC     1 06/09/22 09:40:59 0191
LITRWAS2 PROFILE  EXEC     1 06/09/22 09:40:59 0191
LITRWAS4 PROFILE  EXEC     1 06/09/22 09:40:59 0191
 1= Help                      3= Quit        4= Return      5= Sort Up
 6= Sort down   7= Backward   8= Forward    10= Restore    11= Details
```

Figure 92. The BKRLIST panel, filtered by owner

We placed the curser on the line for LITDAT02 PROFILE EXEC and pressed F10 to display the EDF Minidisk Restore Specifications panel, as shown in Figure 93.

```
                  CMS EDF Minidisk Restore Specifications

 From LITDAT02 0191 date 06/09/22 time 09:40:59 (job TICLFULL 00000001 ).

 To EDF minidisk, userid:          and virtual address:
  FORMAT: OK if needed? NO   FORMAT regardless? NO

 Or to RDR of userid: LITDAT01 node: _        (defaults to this node).

 Or to SFS filepool:          and filespace:
 and path:

 File filters:  Filename: PROFILE  Filetype: EXEC     mode number: 1


 Master backup userid: BKRBKUP




                               3= Quit        4= Return      10= Restore
```

Figure 93. CMS EDF Minidisk Restore Specifications panel

The restore panel has a number of options for restoring the file. We can write it directly to the user's minidisk or send it to the user's reader. In this example, we are sending the file to the user's reader. We do this by placing the name of the user, LITDAT02, next to the **Or to RDR of userid** field, as shown in Figure 93.

With all of the required information for a restore entered on the panel, we pressed F10 to begin the restore job. When you exit the panel, information about the restore job will be displayed on the console, as in the following example:

```
Your command "RESTORE TICLFULL 00000001 LITDAT02 EDF $DEV0191 TO RDR LITDAT02
 - PROFILE EXEC 1" is being processed at 05/23/08 11:27:38.
 BKREST9029I Sending RESTORE request 00000017 to worker task BKRWRK03...
 File RESTORE JOB D1 sent to BKRWRK03 at LTICVM9 on 05/23/08 11:27:38
 *** Restore: Request 00000017 submitted to worker BKRWRK03 for processing.
 Return code "0" from command RESTORE TICLFULL 00000001 LITDAT02 EDF $DEV0191
 TO RDR LITDAT02 - PROFILE EXEC 1 at 05/23/08 11:27:39.
```

You can request status of the restore job at any time by querying BKRM with the command SMSG BKRBKUP STATUS.

Restoring a LINUX image: In this example, we are using the BKRUSER view to restore a Linux image. In this case we will be restoring LITSDNS1's 201 disk. We launched the catalog browser BKRUSER from the administrative user BKRADMIN, which displays a panel that lists all backed up users in the catalog, as shown in Figure 94.

```
                     Catalog:BKRSFS:BKRCATLG.USERCAT.
                                              50 of 50 ownerids displayed
Ownerid filter: *

                                 Ownerids


BEYER    BKRSVSFS LAC0001  LAC0002  LAC0003  LAC0004  LAC0005
LAC0006  LAC0007  LAC0008  LAC0014  LEFEVRE  LITDAT01 LITDAT02
LITDCON1 LITDCON2 LITDIR00 LITHUB   LITRDAT1 LITRDAT2 LITRLOG1
LITRSMB1 LITRWAS1 LITRWAS2 LITRWAS4 LITSDNS1 LITSDNS2 LITSGFW1
LITSGFW2 LITSHA21 LITSHA22 LITSLB01 LITSLB02 LITSLDAP LITSLDP2
LITSPRXY LITSTAM2 LITSTAM3 LITSTAT1 LITSTAT2 LITSTAT3 LITSTAT4
LITSTSM  LITSWAS1 LITSWAS2 LITSWAS4 LITTAM01 LITWORK  MAINT
USER




1= Help        2=              3= Quit        4= Return      5= Sort Up
6= Sort Down   7= Backward     8= Forward    10= Restore    11= Details
```

Figure 94. The BKRUSER panel: Selecting an owner ID

We placed the cursor over the user (ownerid) LITSDNS1 and pressed F11 to get a detailed view of the minidisk backups that are available for LITSDNS1, as shown in Figure 95 on page 240.

```
                      Catalog:BKRSFS:BKRCATLG.USERCAT.
 Devices for ownerid LITSDNS1                    3 of 3 devices displayed
 Ownerid filter: *       Type filter: *

 Device  Type   Instance in catalog


$DEV0191 EDF  1 instances
$DEV0201 CKD  1 instances
$DEV0202 CKD  1 instances









 1= Help        2=              3= Quit        4= Return      5= Sort Up
 6= Sort Down   7= Backward     8= Forward    10= Restore    11= Details
```

Figure 95. The BKRUSER panel: Backed up devices for the selected owner ID

We can see that LITSDNS1 has one backup instance of LITSDNS1's 191 disk in EDF format, or CMS file-level backup. We also see one CKD image instance for the 201 and 202 disks. We want to restore the Linux image on the 201 disk, so we moved the cursor over $DEV0201 CKD and pressed F11. This will display details about the backup instance for that disk, as shown in Figure 96.

```
                      Catalog:BKRSFS:BKRCATLG.USERCAT.
 For LITSDNS1 $DEV0201 CKD                       1 of 1 instances displayed
 Jobname filter: *

 Jobname Instance Date/time completed


LIT_VM9 00000003 2008/04/24 18:06:58









 1= Help        2=              3= Quit        4= Return      5= Sort Up
 6= Sort Down   7= Backward     8= Forward    10= Restore    11= Details
```

Figure 96. The BKRUSER panel: Backups for the selected owner ID and device

In Figure 96, we see that a single backup for the 201 disk is available for restore and the backup was performed on 2008/04/24 as part of backup job LIT_VM9. At this point, we placed the cursor over the backup job name LIT_VM9 and pressed F10 to display the restore options. The restore options are shown in Figure 97 on page 241.

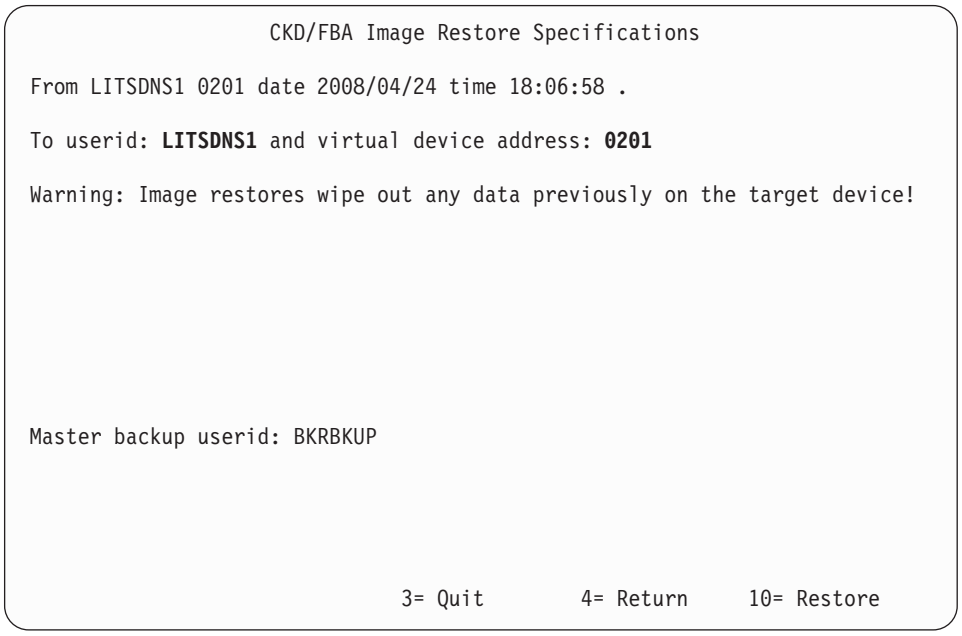


Figure 97. BKRUSER: CKD/FBA Image Restore Specifications

We are now at the CKD/FBA Image Restore Specifications panel. At the top of the panel is user and minidisk information, as well as the date and time the backup was performed. You have the option of restoring the image in place, on the original minidisk, or to a different location. We are going to restore the image back to the original minidisk. To do this, we entered LITSDNS1 in the **To userid** field followed by the virtual address as 0201. We are now ready to start the restore by pressing F10.

Just like in the previous restore, status of the restore progress can be obtained with the command SMSG BKRBKUP STATUS.

When the restore job completes, a summary report and console log from the worker virtual machine will be sent to BKRADMIN's reader.

For more information about restoring data, see Appendix A of *Backup and Restore Manager for z/VM User's Guide*.

Disk space utilization warnings

A remaining major component of a data management strategy is to ensure that your Linux file systems do not fill up. This is easily accomplished by the use of space management tools to periodically archive data. For those would like a refresher, see our previous test report edition that covers the deployment of the logrotate command. Other strategies, such as SNMP monitoring of disk utilization, are also strongly recommended.

Chapter 24. Installing IBM Director 5.20 with z/VM Center

IBM Director is a systems management console that provides an integrated, easy-to-use suite of tools for monitoring and managing a large number of systems. IBM Director has been supported on System z since version 5.10. The basic functionality allows monitoring and connectivity to managed nodes through a single unified console.

Monitored systems are defined by one of three *agent levels*, or *tiers*:

- **Tier 0 systems**

  Tier 0 systems can be connected to remotely via secure shell connectivity (SSH). Through this interface, commands can be executed on the remote host. Additionally, presence checking and some machine statics can be retrieved and displayed. There is no system overhead or additional software installation needed for tier 0 systems.

- **Tier 1 systems**

  Tier 1 systems are running a CIM server which provides enhanced management capabilities on System x™ and System p™ platforms. The tier 1 agent provides hardware level monitoring and management for the systems that support it. This is the first of the clients requiring package installation on the managed nodes. On the Linux for System z platform, the tier 1 agent itself does not provide any extra functionality over what you can do with a tier 0 system, since System z does not expose hardware monitor events to Linux in a way that the tier 1 agent can handle. It does, however, provide an open, standard environment for other systems management tools to build upon. For example, the z/VM Management Access Point is installed on top of a tier 1 agent to provide the ability to monitor and manage a z/VM system.

- **Tier 2 systems**

  The highest level of management and integration of IBM Director comes with the installation of the tier 2 agent. This agent is installed via simple **rpm** commands and enables the managed nodes to have all the functionality of tier 1 and tier 2 agents as well as the ability to monitor processes running within the operating system, OS-level performance metrics, and advanced software distribution.

Extensions to IBM Director

You can purchase extensions to IBM Director to provide additional functionality. The extensions we used are z/VM Center and Software Distribution Premium Edition.

z/VM Center

Additional System z specific functionality can be added via the purchase of z/VM Center. The z/VM Center extension provides a unified way to deploy virtual Linux servers running on z/VM. The major benefit to many is the GUI interface which allows the provisioning without requisite z/VM knowledge. The extension uses the Systems Management API provided by z/VM. The underlying technology that unifies the information is referred to as the Common Information Model (CIM) which provides access to z/VM system management functions.

While we will not deal with provisioning directly in this test report, we acknowledge that maintaining the systems list and various sources of resource allocation for virtual servers can provide a burden to even the most seasoned system administrator. IBM Director is a key component to the management of server resource provisioning. We will be using it extensively in the lab over the next few months as we transition our integration test infrastructure to use the z/VM Center concept of virtual server templates and operating system templates for all of our provisioning needs. As usual, we will document the process for the next edition of this test report. Director also employs the concept of disk pools, which can alleviate the need for manually maintaining DASD allocation by creating appropriately sized minidisks on the fly from a pre-allocated pool. We have found this to be very useful functionality in our early evaluations.

Software Distribution Premium Edition

The second purchased extension to IBM Director is the Software Distribution Premium Edition component. This software provides a mechanism for the creation and distribution of software packages to the managed systems. Note that the standard software distribution included with IBM Director enables you to distribute IBM-provided software packages while the Premium Edition allows system administrators to build and distribute custom packages for Windows and Linux environments.

Installing IBM Director

The installation of IBM Director itself is fairly straightforward—it's just a bash script that installs several rpms for you. The tough part is configuring z/VM Center, such as getting DIRMAINT to talk to the Systems Management API, then configuring the z/VM MAP server to talk to the Systems Management API. In other words, the bulk of the effort is in making z/VM Center actually do something.

The multifaceted installation is best covered at length in an IBM Redbook, *Implementing IBM Director 5.20*, available at www.redbooks.ibm.com/redbooks/pdfs/sg246188.pdf. In fact, some of the members of our integration test team co-authored this authoritative and highly recommended Redbook. Rather than attempt to repeat all of that content here, we'll simply urge you to read the Redbook for yourself.

Configuring Linux virtual servers for SNMP monitoring by IBM Director

Configuring our Linux virtual servers to report SNMP information to a Director server was a fairly straightforward operation. We simply began by ensuring that we had an snmpd distribution package installed, such as net-snmp as obtained from **yast** on the SLES 10 distribution.

Next, we needed a valid SNMP configuration, such as the sample snmpd.conf file that we provide below. On SLES 10, this file can be found at /etc/snmp/snmpd.conf. On RHEL 4 distributions, the file can be found at /etc/snmpd.conf.

Here is our sample snmpd.conf file:

```
##############################################################################
#
# snmpd.conf
#
#   - created by the snmpconf configuration program
#
##############################################################################
```

```
#doDebugging 1
# SECTION: Access Control Setup
#
#   This section defines who is allowed to talk to your running
#   snmp agent.
# rwuser: a SNMPv3 read-write user
#   arguments:  user [noauth|auth|priv] [restriction_oid]
rwuser ticlsnmp_rw
# rouser: a SNMPv3 read-only user
#   arguments:  user [noauth|auth|priv] [restriction_oid]
rouser ticlsnmp_ro
# rocommunity: a SNMPv1/SNMPv2c read-only access community name
#   arguments:  community [default|hostname|network/bits] [oid]
rocommunity  ticlsnmp_ro_comm
# rwcommunity: a SNMPv1/SNMPv2c read-write access community name
#   arguments:  community [default|hostname|network/bits] [oid]
rwcommunity ticlsnmp_rw_comm
###########################################################################
# SECTION: Trap Destinations
#
#   Here we define who the agent will send traps to.
# trapsink: A SNMPv1 trap receiver
#   arguments: host [community] [portnum]
trapsink  192.168.71.249 ticlsnmp_rw_comm
# trap2sink: A SNMPv2c trap receiver
#   arguments: host [community] [portnum]
#trap2sink  192.168.71.249 ticlsnmp_rw_comm
# informsink: A SNMPv2c inform (acknowledged trap) receiver
#   arguments: host [community] [portnum]
#informsink  192.168.71.249 ticlsnmp_rw_comm
# trapcommunity: Default trap sink community to use
#   arguments: community-string
trapcommunity  ticlsnmp_ro
# authtrapenable: Should we send traps when authentication failures occur
#   arguments: 1 | 2   (1 = yes, 2 = no)
authtrapenable  1
###########################################################################
# SECTION: Monitor Various Aspects of the Running Host
#
#   The following check up on various aspects of a host.
# proc: Check for processes that should be running.
#     proc NAME [MAX=0] [MIN=0]
#
#     NAME:  the name of the process to check for.  It must match
#            exactly (ie, http will not find httpd processes).
#     MAX:   the maximum number allowed to be running.  Defaults to 0.
#     MIN:   the minimum number to be running.  Defaults to 0.
#
#   The results are reported in the prTable section of the UCD-SNMP-MIB tree
#   Special Case:  When the min and max numbers are both 0, it assumes
#   you want a max of infinity and a min of 1.
# This particular example configuration monitors the httpd service on one of our Apache servers
proc  /usr/sbin/httpd2-prefork 100 1
# disk: Check for disk space usage of a partition.
#   The agent can check the amount of available disk space, and make
#   sure it is above a set limit.
#
#    disk PATH [MIN=100000]
#
#    PATH:  mount path to the disk in question.
#    MIN:   Disks with space below this value will have the Mib's errorFlag set.
#           Can be a raw byte value or a percentage followed by the %
#           symbol.  Default value = 100000.
#
#   The results are reported in the dskTable section of the UCD-SNMP-MIB tree
#
#   This particular example configuration monitors the "/" and "/opt" partitions
#   for 5 and 10% minimum remaining capacity
disk  / 5%
disk  /opt 10%
# load: Check for unreasonable load average values.
#   Watch the load average levels on the machine.
#
#    load [1MAX=12.0] [5MAX=12.0] [15MAX=12.0]
#
#    1MAX:   If the 1 minute load average is above this limit at query
```

```
#           time, the errorFlag will be set.
#    5MAX:   Similar, but for 5 min average.
#    15MAX:  Similar, but for 15 min average.
#
#   The results are reported in the laTable section of the UCD-SNMP-MIB tree
load  10 7 5
# file: Check on the size of a file.
#   Display a files size statistics.
#   If it grows to be too large, report an error about it.
#
#    file /path/to/file [maxsize_in_bytes]
#
#      if maxsize is not specified, assume only size reporting is needed.
#
#   The results are reported in the fileTable section of the UCD-SNMP-MIB tree
file  /var/log/messages 50000
###########################################################################
# SECTION: Agent Operating Mode
#
#   This section defines how the agent will operate when it
#   is running.
# agentaddress: The IP address and port number that the agent will listen on.
#   By default the agent listens to any and all traffic from any
#   interface on the default SNMP port (161).  This allows you to
#   specify which address, interface, transport type and port(s) that you
#   want the agent to listen on.  Multiple definitions of this token
#   are concatenated together (using ':'s).
#   arguments: [transport:]port[@interface/address],...
#agentaddress udp:161,tcp:161
###########################################################################
# SECTION: System Information Setup
#
#   This section defines some of the information reported in
#   the "system" mib group in the mibII tree.
# syslocation: The [typically physical] location of the system.
#   Note that setting this value here means that when trying to
#   perform an snmp SET operation to the sysLocation.0 variable will make
#   the agent return the "notWritable" error code.  IE, including
#   this token in the snmpd.conf file will disable write access to
#   the variable.
#   arguments:  location_string
#syslocation Server Room
syslocation  710RaisedFloor
# syscontact: The contact information for the administrator
#   Note that setting this value here means that when trying to
#   perform an snmp SET operation to the sysContact.0 variable will make
#   the agent return the "notWritable" error code.  IE, including
#   this token in the snmpd.conf file will disable write access to
#   the variable.
#   arguments:  contact_string
#syscontact Sysadmin (root@localhost)
syscontact  systemadmins@ltic.pok.ibm.com
```

The following command enables snmpd to start automatically at boot:

```
chkconfig snmpd on
```

The following command queries the recent load average for a host (here, we specified host litstat7) from some other snmpd configured machine:

```
snmpget -v1 -Cf -c "ticlsnmp_ro_comm" litstat7  .1.3.6.1.4.1.2021.10.1.3.2
```

Similarly, the following command queries the system contact string:

```
snmpget -v1 -Cf -c "ticlsnmp_ro_comm" litstat7  system.sysContact.0
```

Chapter 25. Energy management with Active Energy Manager

This topic discusses our test experiences with the Active Energy Manager (AEM) product. We used AEM on zLinux over a six-month period, managing the power usage of a wide variety of hardware, including IBM System z10 and other IBM platforms.

AEM is a major part of IBM's Big Green initiative. It provides users with a single view of the actual power usage across multiple platforms. It also provides the capability, on some hardware platforms, to control energy consumption and improve energy efficiency, which can result in substantial savings and reduction in operating costs.

AEM is an IBM Director extension, a member of the IBM System Director Family. It runs on various hardware platforms and operating systems, including Linux on System z (RHEL 4 and RHEL 5, and SLES 9 and SLES 10).

AEM supports the power management of a number of different types of managed objects, including the System z10 platform. A few items to note here:

- System z10 is the only System z platform that AEM directly supports. Older System z platforms can be monitored by AEM by attaching them to metering devices such as an IBM PDU+.
- System z10 supports power and thermal monitoring. Power capping and power saver features are not supported.
- It is not necessary to run AEM on the same system that you want to monitor. For example, you can run it on a System z9 and use it to monitor a System z10 and other non-System z hardware on your network. See the reference material for the complete list of supported hardware and software.

There are no agents that need to be installed on the managed objects. AEM measures, monitors, and manages power consumption using the hardware and software built into the managed objects that it supports.

Reference material for Active Energy Manager

We found the following references to be very useful for working with Active Energy Manager:

- IBM Redpaper, *Going Green with IBM Active Energy Manager*, available at www.redbooks.ibm.com/redpieces/abstracts/redp4361.html?Open
- *IBM Systems Director Active Energy Manager™ Installation and User's Guide* and the IBM Director Information Center, available on the IBM Director Documentation and resources Web page at www.ibm.com/systems/management/director/resources/
- Project Big Green Web page at www.ibm.com/systems/optimizeit/cost_efficiency/energy_efficiency/

Experiences with Active Energy Manager

Active Energy Manager (AEM) is a useful tool for System z that monitors both power and thermal data. Figure 98 shows an example of trend data from one of our z10 EC CPCs.

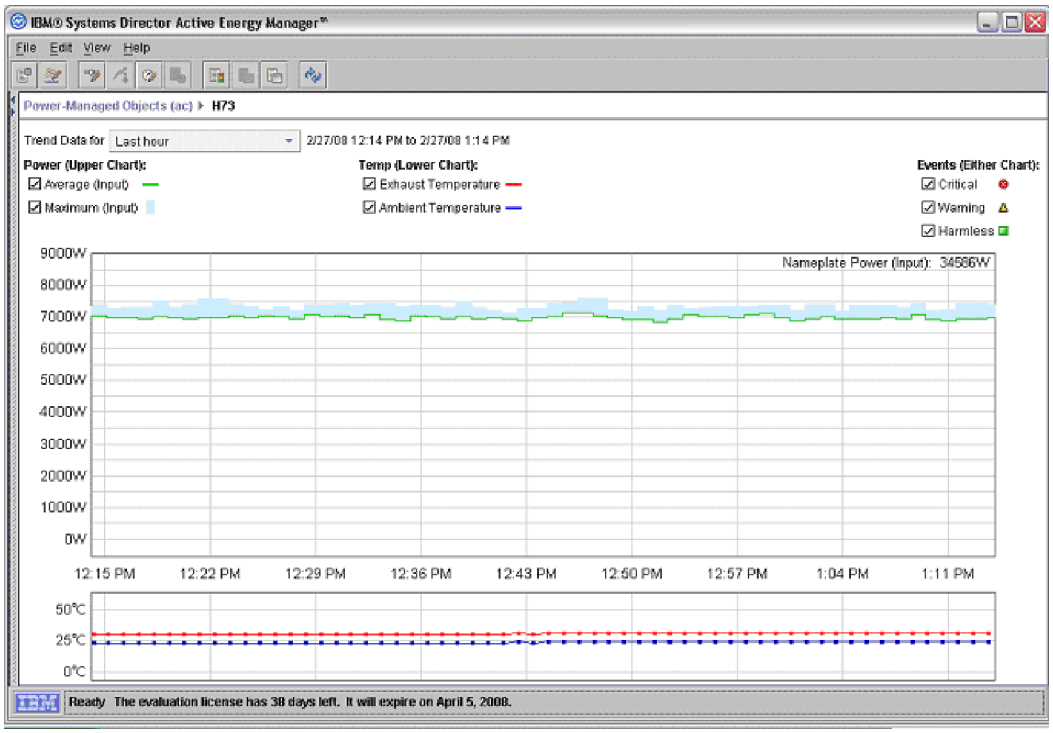


Figure 98. Example of Active Energy Manager power and thermal trend data

Figure 99 on page 249 shows another example of power trend data for a z10 EC CPC. Note the event icons on the screen and that event details can be seen by placing the cursor over an icon.

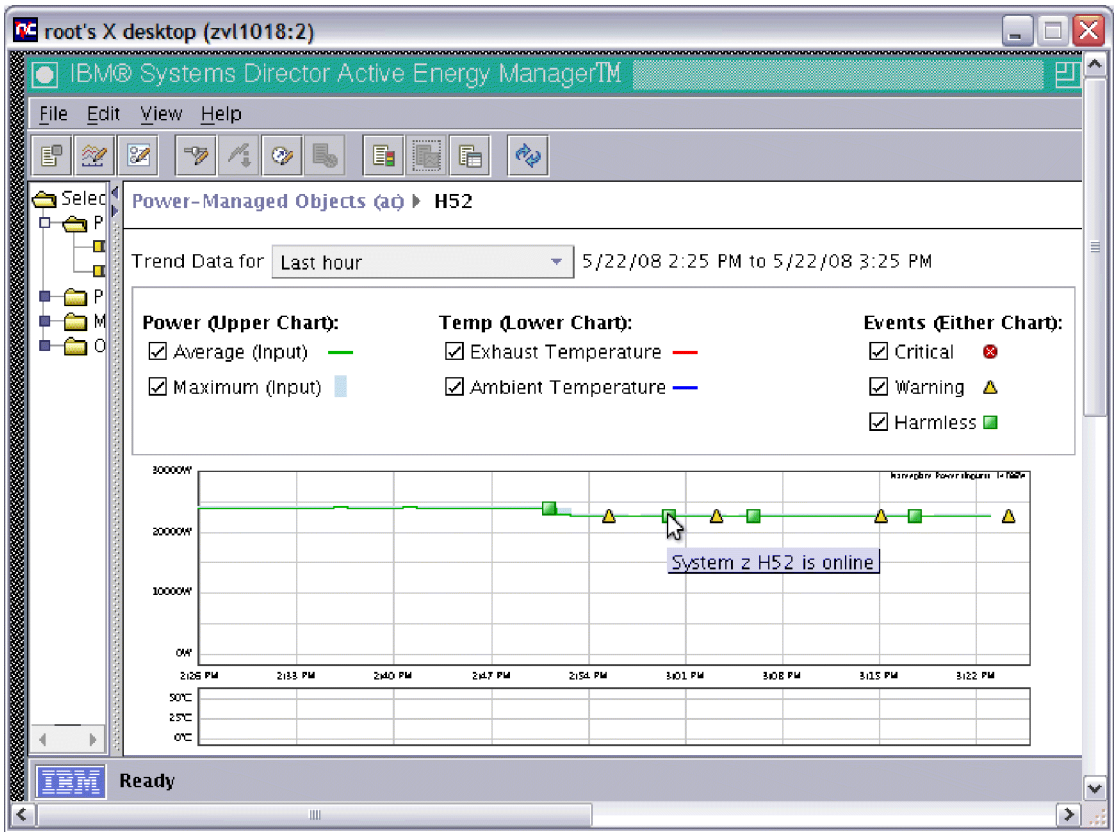


Figure 99. Example of Active Energy Manager power and thermal trend data with event icons and descriptions

System z users can take advantage of the AEM watt-hour meter function for comparing actual power used to the specified nameplate power. Nameplate power is the power specification that the manufacturer is required to put on the back of a server. It represents the absolute worst case power that the system could ever draw and is based on the capacity of the power supplies. Figure 100 on page 250 shows an example of a cost comparison of powering a z10 EC server over a 12-hour time period. Note that both the price per kilowatt-hour and the cooling rate factors are user-defined variables and can vary significantly from site to site.

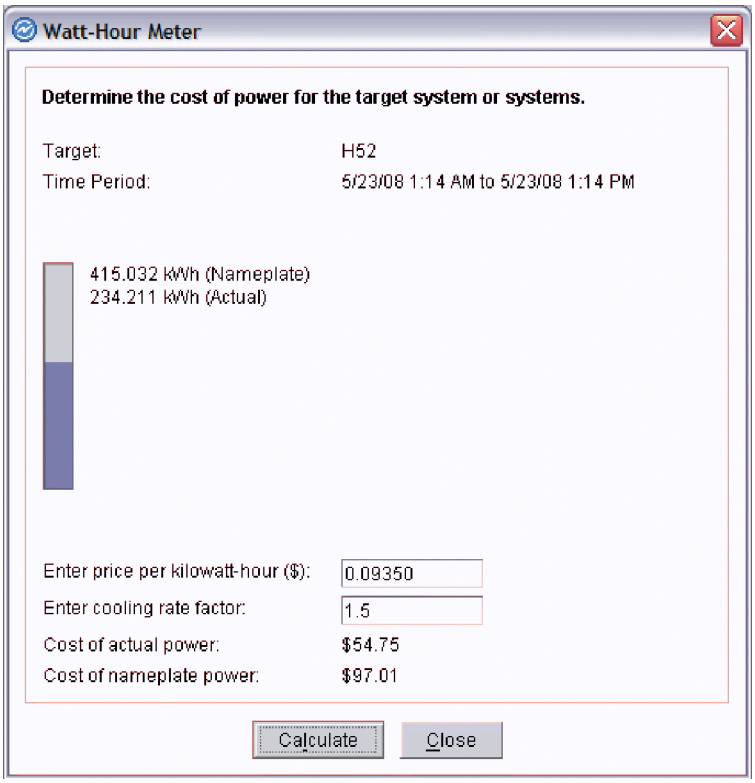


Figure 100. Example of Active Energy Manager Watt-Hour Meter

Installing the AEM 3.1.1 server and console on Linux is a very simple matter of using **rpm** with the **-vih** options. We ran into a few problems with the installation of the prerequisites, which include:

- IBM Director 5.20.2
- BladeCenter® Management extension (required only if you are going to manage BladeCenter servers)
- Hardware Management Console extension for IBM Director

Not all operating systems support both the Director server and console. Keep this in mind when planning for installation. (See the section on "Specified operating environments" in the Information Center, then follow the "Supported operating systems" link, then choose the version of Director you are installing.)

Also, make sure that the requirements that are applicable to zLinux have been met. (See the section on "Preparing to install IBM Director Server on Linux for System z" in the Information Center.)

We ran into a few problems installing IBM Director on RHEL 5. The Director console could not be opened because it required the 32-bit versions of the following X libraries:

- libXmu-1.0.2-5.s390.rpm
- libXtst-1.0.1s390.rpm

Our RHEL 5 installation only included the 64-bit versions of these libraries.

We encountered another problem due to the fact that we are managing a BladeCenter server while installing the Director console on zLinux. In the console

response file, we did not choose the optional component, DirSeriesLib, to be installed. This is a co-requisite for the BladeCenter management extension and must be marked for installation (set to 1) with BladeCenter, for the Director console installation to be successful.

In our test environment, we ran a number of different configurations which included running AEM and the Director Server on both RHEL and SLES. We ran the Director/AEM console both locally, on the same system as the server, and remotely on zLinux and Windows 2003 systems running on System x.

Regarding the performance of the server, the critical factor has to do with the number of managed objects that you plan to monitor and/or manage. In our test environment, we are continually monitoring close to 1000 objects on Director/AEM server zLinux running on a z900 with 1G bytes of memory and 2.5G bytes of disk space. These end points include a variety of systems ranging from System x BladeCenter servers to intelligent PDUs. They reside both locally within our site and remotely, half way across the country. Our experience has been that screen refreshes, especially when we're viewing the aggregate power data for the entire complex, are noticeable, in the range of a few seconds, but are acceptable.

Chapter 3, "Planning for Active Energy Manager" in the installation and user's guide is worth reading. It gives detailed recommendations regarding memory and storage requirements which are based upon the number of objects to be managed. Two items to note are the AEM polling refresh and console refresh rates. The default for both is one minute. Changing them to five minutes reduces the frequency of the delays and the disk space required to save the data.

One of the features of AEM that makes it relatively easy to install and configure is that there are no remote agents; nothing needs to be installed on any of the supported objects. This feature does present the challenge of keeping track of which objects are supported and to what extent. The information center has a section that lists supported hardware, including managed systems and metering objects. In some cases, it includes the levels of firmware that need to be installed. We experienced this when trying to manage JS20 and LS20 blades in an 8677 BladeCenter server. The power capping feature of AEM would not work until we installed the proper level of firmware on each of the blades.

One minnor issue to note is that, in Director 5.20.2, when adding a System z10 CPC, you need to add it as an SNMP device type. It will appear on the Director console as an SNMP printer, as shown in Figure 101 on page 252. It is *not* a printer; it is a z10 CPC! This should be fixed in future Director releases.

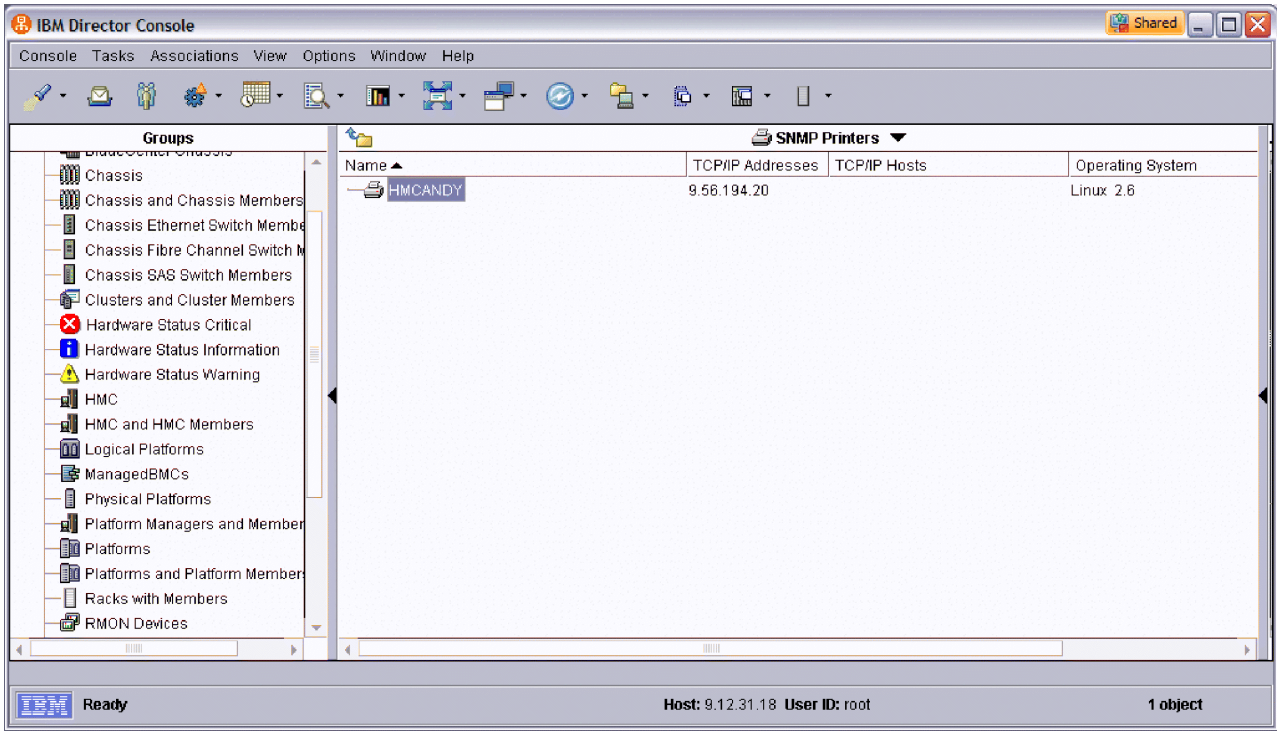


Figure 101. Example of a System z HMC icon (erroneously shown as a printer) in IBM Director 5.20.2

We found problems when attempting to start an AEM task on a managed object. The task would fail to start with an error message stating that the object is not supported. In these cases you need to make sure that:

- The object type is included on the list of supported hardware.
- If it is, as is the case of an HMC managed IBM Power 550 (System p 8204-E8A), make sure you are *not* selecting the HMC icon. In the case of an HMC managed System p server, you need to start AEM from the server icon. Note that this not the case with System z. You can start an AEM task on a System z CPC directly from the HMC icon. (See the figures of sample AEM screen Shots in this topic for examples.)
- Check the firmware levels on managed objects. As stated earlier, we had this problem with servers in a BladeCenter server.
- In Director, make sure you have gained access to the managed object or that you added the object using an ID with the correct level of authority. For example, with a PDU+ you need to add it using a community name that has write access to the device.

Figure 102 on page 253 shows an example of adding a PDU+ into Director. The powerwrite community name must have write access to the device.

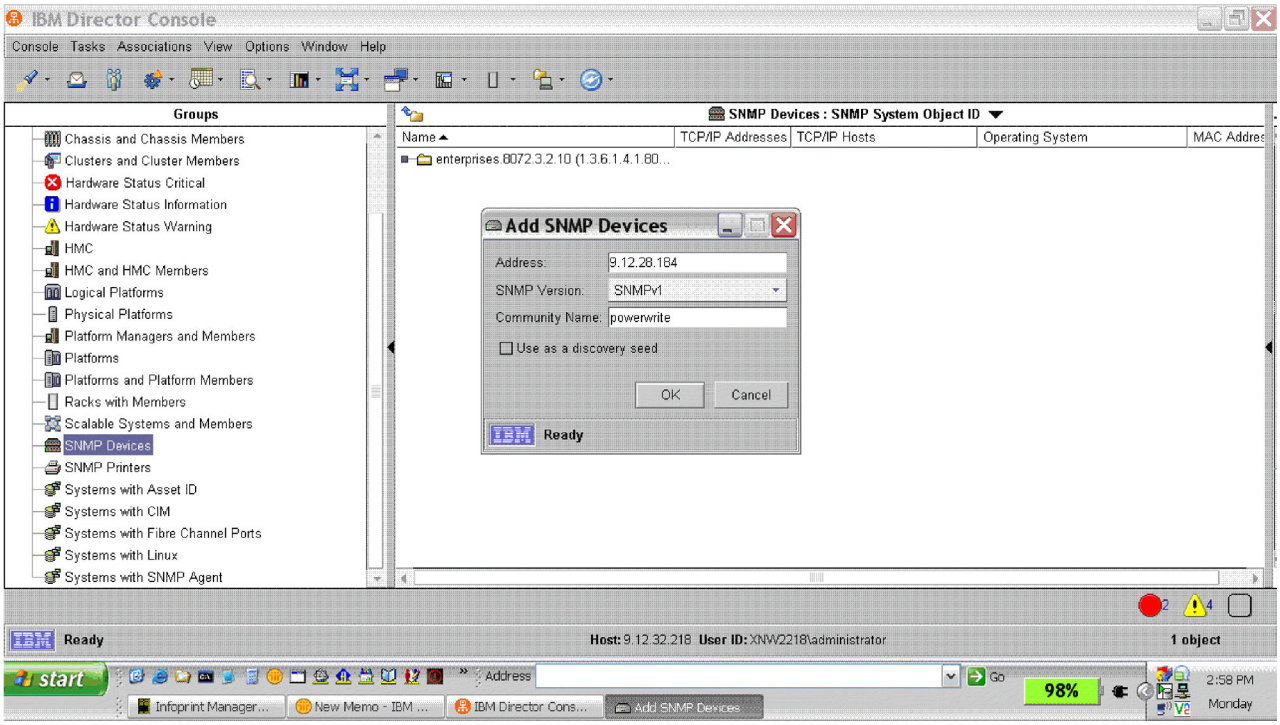


Figure 102. Example of adding a PDU+ into Director

As the cost of energy increases and becomes a more significant part of the IT budget, understanding its use throughout the data center becomes more important. With the combination of supported hardware and externally connected metering devices, AEM can provide a single view of the power and thermal trend data for the entire data center. Also, for additional savings, it allows you to take advantage of the power saving and power capping functions provided by some of the newer hardware.

Chapter 26. Linux security

Security is another critical aspect of systems management. Though we have discussed some items previously regarding things like fire walls and construction of our demilitarized zone (DMZ), there are some fairly easy security steps that should be deployed for systems management and systems operations purposes.

3270 encryption

Before we could successfully enable 3270 encryption, we gathered all of the necessary documentation from www.vm.ibm.com/related/tcpip/vmsslinf.html. There you can find a set of links to the Linux requirements for the various z/VM releases. Locate the link for your VM release, read it, and keep it handy. Most importantly, make sure you choose a supported Linux version for the z/VM release you are running. This will most likely be an older release, since the SSL enabler contains kernel modules that will have to be recompiled to support a current release. For example, on z/VM 5.3 running SLES 9 SP3 with the GA SP3 kernel level, 2.6.5-7.244, applying any updates at all from the YaST online updates tool will change the kernel level and prevent the ssl enabler from working correctly.

Also see chapter 22 about configuring the SSL server in *z/VM TCP/IP Planning and Customization*.

Experiences setting up the SSL server

The following steps describe our experience with setting up the SSL server:

1. Set up the directory entry for the SSL server guest user ID. The following is an example of the directory entry for our SSL server, SSLSERV:

```
USER SSLSERV password 128M 512M BG
   INCLUDE TCPCMSU
   IUCV ALLOW
   OPTION ACCT MAXCONN 1024 QUICKDSP SVMSTAT
   LINK TCPMAINT 0591 0591 RR
   LINK TCPMAINT 0592 0592 RR
   LINK TCPMAINT 0198 0198 RR
   LINK 5VMTCP30 0491 0491 RR
   LINK 5VMTCP30 0492 0492 RR
*  LINK 5VMTCP30 0493 0493 RR
   MDISK 0191 3390 8027 1 +VMRES MR RSSLSERV WSSLSERV MSSLSERV
*  MDISK 0201 3390 8028 1 +VMRES MR RSSLSERV WSSLSERV MSSLSERV
   MDISK 0201 3390 0001 3338 VM6SSL MR
   MDISK 0203 3390 8029 1 +VMRES MR RSSLSERV WSSLSERV MSSLSERV
```

   Note that the commented out 201 minidisk was in the original directory entry as it was shipped by IBM. We commented it out and added our own 201 disk. We will IPL Linux from the new 201 disk that was added. In addition, 5VMTCP30's 493 disk is commented out because it is a service disk and the service disks are all living in the shared file system (SFS) on this z/VM system. This is important to remember later.

2. Define the RACF segment for SSLSERV, if it does not exist, and log on.
3. Install Linux onto the 201 disk. You can do this by cloning from another Linux system using DDR or FlashCopy, or you can do a new install directly on SSLSERV.

   We chose to install the 31-bit version of SLES 9 SP3 directly on SSLSERV.

a. SSLSERV has a PROFILE EXEC on its 191 disk that does a lot of TCPIP setup. Somewhere in there, a VDISK at 152 gets allocated. It is intended to be used for Linux swap space but we formatted it for CMS to use as a scratch disk for the Linux install.

```
FORMAT 152 X
1
SWP152
```

b. Defined a NIC and coupled it to a vswitch for the installer to use. SSLSERV will not need a network connection once it is running; this is only used to get Linux installed. We did not define the NIC in the VM directory, so it is all temporary. The vswitch has access to a LAN that can get to the SLES 9 SP3 install server.

c. FTPed the SLES 9 SP3 32-bit installer kernel, parmfile, and initrd to SSLSERV's 152 vdisk at file mode X. Punched to SSLSERV's reader and started the install process normally.

d. Configured the installer with an IP address—again, just for the Linux install process.

e. Selected 152 and 201 in the DASD panel and activated them, then formatted the 201 disk in **yast**.

Do *not* touch the 203 disk, as it contains an ext2 file system with data on it already. If you format it, you will have to figure out how to recover it from the install tapes. We will hook it up in Linux later.

f. Selected **New Install**.

g. Clicked on the partitioner section and set up 152 as a swap device and allocated a new ext3 partition on 201.

h. Changed the software selection to **Minimal Install**, went into the Special Configuration panel and added the compat rpms:

```
compat-libstdc++-lsb-4.0.2_20050901-0.4
compat-2004.7.1-1.2
```

i. Clicked the button to start the install process.

4. After the Linux installation completed successfully, we customized the Linux system to serve as a VM SSL server.

On 5VMTCP30 493 there are binary files containing VMSSL rpms and the linux customization docs. On our system, the service disks are in SFS, so this content was actually at VMSYS:5VMTCP30.TCPIP.BINARY.

We are running z/VM 5.3, so we used the table at www.vm.ibm.com/related/tcpip/vmsslinf.html to identify the files that are needed on our Linux release and z/VM release. There are also README and INSTALL files containing the Linux customization instructions.

a. Linked the disk containing the files (we used MAINT) and copied the appropriate two packages to the Linux system which will become the SSLSERV server. We chose VMSS9 RPMBIN and IBMGSK RPMBIN as they match the SLES 9 SP3 s390 kernel level 2.6.5-7.244.

The VMSS9 INSTALL file contains the instructions that we are following at this point.

You can use FTP or your 3270 emulator to move those files around.

b. Altered the zipl.conf file to include the 203 disk. We added the **dasd** parameter to the **parameters** line in /etc/zipl.conf:

```
parameters = "dasd=152,201,203 root=/dev/dasdb1 selinux=0 TERM=dumb elevator=cfq
```

c. Ran **zipl** to write the new IPL record out to the boot disk. This allows the 203 disk to be visible at IPL time. The 203 disk contains an ext2 file system

already, so it is important *not* to format it during the install process. That's why we're adding it now, instead of during the Linux install.

d. Re-IPLed and made sure that **lsdasd** reports that the 203 disk is available.

e. Renamed both of the files that we copied to Linux, as:

```
VMSS9.RPMBIN -> vmssld-2.6.5-2.s390.rpm
IBMGSK.RPMBIN -> gsk7bas-7.0-3.13.s390.rpm
```

f. Installed both of the rpms:

```
rpm -ivh vmssld-2.6.5-2.s390.rpm gsk7bas-7.0-3.13.s390.rpm
```

g. Added a line to /etc/fstab so that 203 gets mounted automatically at IPL time:

```
/dev/dasdc1      /opt/vmssl/parms     ext2       defaults      1 1
```

We ended up with 203 being /dev/dasdc since we used **dasd=152,201,203** on the kernel parameter line. If your parameter line is different, use **lsdasd** to determine the proper /dev/dasd*x* designation for your 203 disk.

h. Ran **mount -a** to mount everything defined in **fstab**. Verified that 203 is mounted at /opt/vmssl/parms now.

i. Ran the **modsymlinks** shell script provided by vmssld:

```
cd /opt/vmssl/bin
./modsymlinks -m
```

That script renames everything in /etc/init.d so that when Linux starts up it only runs the vmssl daemon. No network stack, no cron jobs, no logrotate, nothing. There is a getty running on the 3270 console, and that's it. Do not IPL Linux or you will lose your SSH session and have to do the rest via 3270.

j. Make sure that /etc/inittab has a line like this:

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -h now
```

This will make sure SSLSERV can catch shutdown signals from VM and come down cleanly.

k. Re-IPL Linux. When it comes up, there won't be any services running but you can still log on as *root* from the console.

l. From an authorized user such as TCPMAINT, ran the following command:

```
SSLADMIN QUERY CERT *
```

If the output displays a whole list of CA certificates, then the VM-SSLSERV communication link is working. For instance:

```
ssladmin query cert *
Labels are:
  1  - Thawte Personal Premium CA
  2  - Thawte Personal Freemail CA
  3  - Thawte Personal Basic CA
  4  - Thawte Premium Server CA
  5  - Thawte Server CA
  6  - RSA Secure Server Certification Authority
  7  - VeriSign International Server CA - Class 3
  8  - VeriSign Class 4 Public Primary Certification Authority - G3
  9  - VeriSign Class 3 Public Primary Certification Authority - G3
 10  - VeriSign Class 2 Public Primary Certification Authority - G3
 11  - VeriSign Class 1 Public Primary Certification Authority - G3
 12  - VeriSign Class 4 Public Primary Certification Authority - G2
 13  - VeriSign Class 3 Public Primary Certification Authority - G2
 14  - VeriSign Class 2 Public Primary Certification Authority - G2
 15  - VeriSign Class 1 Public Primary Certification Authority - G2
 16  - VeriSign Class 3 Public Primary Certification Authority
 17  - VeriSign Class 2 Public Primary Certification Authority
 18  - VeriSign Class 1 Public Primary Certification Authority
```

```
19  - Entrust.net Global Secure Server Certification Authority
20  - Entrust.net Global Client Certification Authority
21  - Entrust.net Client Certification Authority
22  - Entrust.net Certification Authority (2048)
23  - Entrust.net Secure Server Certification Authority

Ready; T=0.03/0.03 17:11:38
```

At this point, we're following along in chapter 22 of *z/VM TCP/IP Planning and Customization*, picking up with step 2:

Step 2: Update PROFILE TCPIP

Get write access to the PROFILE TCPIP file and add or edit the following changes:

```
AUTOLOG
    SSLSERV  PASSWORD   ; VM SSL Server ENDAUTOLOG

SSLSERVERID SSLSERV TIMEOUT 60

INTERNALCLIENTPARMS
  PORT 23
  PORT 923
  SECURECONNECTION PREFERRED ENDINTERNALCLIENTPARMS

PORT
    23 TCP INTCLIEN            ; TELNET Server
    923 TCP INTCLIEN SECURE LTICVM6    ; SSL protected TELNET Server
    9999 TCP SSLSERV             ; SSL Server - Administration
```

Make sure to autolog SSLSERV and keep it running. SSLSERV is the SSLSERVER ID to be used for TLS processing. Setup the internal Telnet server to listen on both ports 23 and 923. Port 23 is the normal unencrypted Telnet; port 923 is secure Telnet and is secured by the key labeled LTICVM6. (Note that we have not yet created the LTICVM6 key.)

Step 3: Update the DTCPARMS file

We did not need to change anything since the default DTCPARMS will work for us. We did not change the IPL address for Linux or anything else.

Step 4: Update ETC SERVICES

We did not need to change anything in ETC SERVICES since they only ask you to add port 9999 for the SSL Server administration and it was already there on our system.

Step 5: Set up the certificate database

Now we can generate a self-signed key for testing.

1. Created a file called LTICVM6 X509INFO on TCPMAINT and added the following:

   ```
   COMMON LTICVM6.PDL.POK.IBM.COM
   ORGANIZATION LTIC
   COUNTRY US
   ```

2. Issued the following command to have SSLSERV generate a self-signed key:

   ```
   SSLADMIN SELF LTICVM6 A 1024 LTICVM6
   ```

3. Created a self signed certificate using LTICVM6 X509INFO on A, 1024 bits strong, labeled it LTICVM6 in the database. This is the LTICVM6 key referenced in the PORT section of PROFILE TCPIP.

We are basically done here. A restart of TCPIP will pick up all the changes. We recommend reviewing the rest of chapter 22 of *z/VM TCP/IP Planning and Customization* and see if anything else looks like it may pertain to your environment.

Client-side configuration notes

We have the following notes to offer about configuring the 3270 client-side SSL exploiters, specifically x3270 and IBM Personal Communications (PCOMM):

- **x3270**

  x3270 will actually work as is, as long as it was built with SSL support. We did not have to import certificates or anything. We were able to connect to the system using the following magic address format:

  ```
  L:9.12.20.103:923
  ```

  The `L:` means SSL encrypted and the `:923` is the port number of the secure Telnet server.

- **IBM Personal Communications (PCOMM)**

  We had to get the LTICVM6 certificate from TCPMAINT on LTICVM6 and get it onto our workstation. Actually, SENDFILE works fine for this.

  1. We used SENDFILE to send the LTICVM6 X509CERT file from TCPMAINT to a Lotus Notes ID (for instance, *notes_id* at IBMUS).
  2. Detached the certificate to a plain text file and followed the instructions under "Receiving Certificates Into the PCOM Certificate Database" at www.vm.ibm.com/related/tcpip/tcsslcfx.html#crtrcv to import the certificate into PCOMM.
  3. Turned on security in PCOMM—either SSL or TLS will work.

Assigning a cryptographic domain to an LPAR

Cryptographic domains are assigned to an LPAR in the zSeries Activation Profile. The profiles are edited from the HMC or SE from the Customize/Delete Activation Profile Panel. You can assign many domains to a single LPAR. However, a domain can only be assigned to one LPAR.

Figure 103 on page 260 shows how we have assigned Control Domain 0 and Usage Domain 0 to LPAR LP01 on X04.

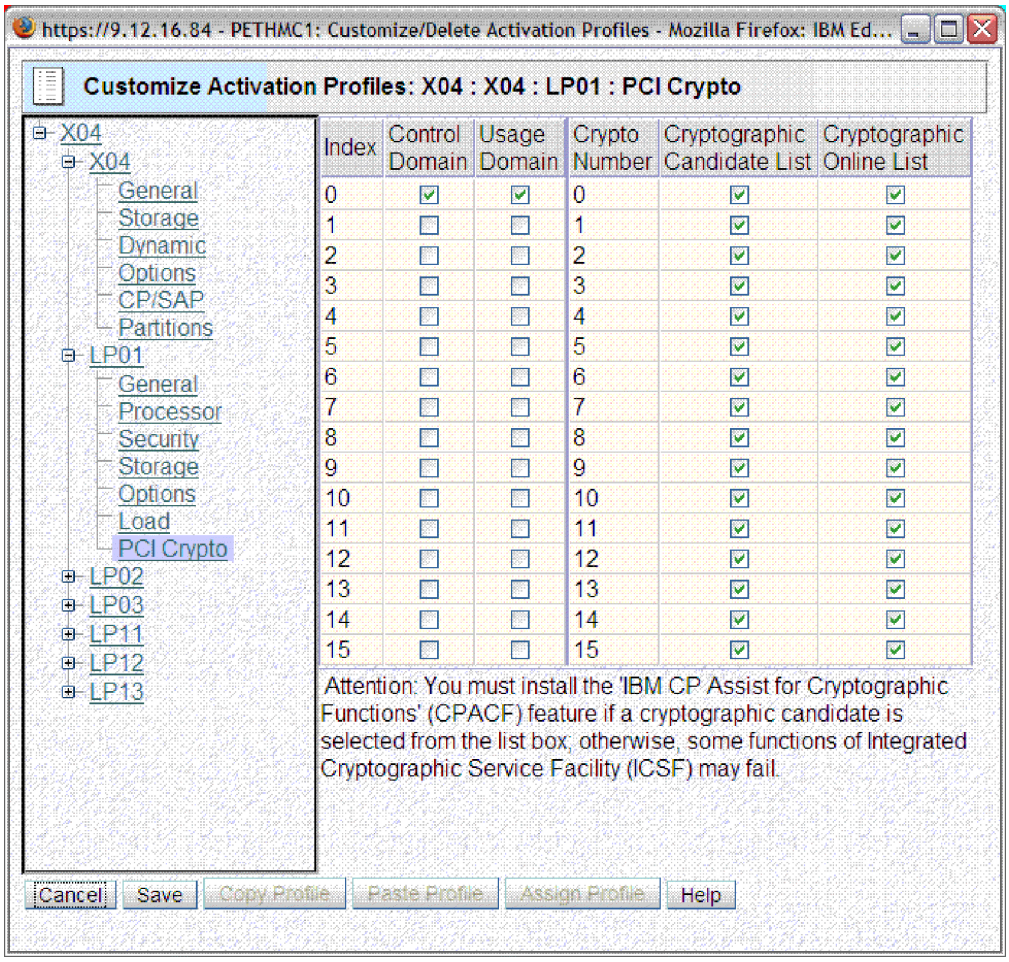


Figure 103. Customize Activation Profiles: Assigning cryptographic domains to an LPAR

LP01 on X04 is our integration test VM6 LPAR. From z/VM, we can verify that the Crypto domain 00 is available by using the QUERY CRYPTO command, as in the following example:

```
q crypto ap

AP 00 CEX2C Queue 00 is installed
AP 01 CEX2C Queue 00 is installed
AP 02 CEX2C Queue 00 is installed
AP 03 CEX2C Queue 00 is installed
Ready; T=0.01/0.01 16:56:37
```

This particular CPC has four Crypto cards installed and available for use. Since we have assigned Crypto Domain 00 to LPAR LP01, we see in the output from the QUERY CRYPTO command that domain 00 from all four cards has been assigned to LP01, as expected.

Since we have only assigned a single domain to our LPAR, we are going to share the domain with all of our z/VM guests. We do this by placing the CRYPTO APVIRT statement in the z/VM directory for each guest that we want to have shared access to the card.

To simplify the directory statements, we elected to place the CRYPTO APVIRT statement in a directory profile and then include that profile in each guest that needs access to the Crypto domain. The following is an example of our directory profile with the APVIRT statement:

```
PROFILE LNXPOOL
  CPU 0 BASE
  IPL CMS
  IUCV ALLOW
  MACHINE ESA 64
  CRYPTO APVIRT
```

You can also elect to dedicate a Crypto domain to a z/VM guest. However, you can only dedicate the domain to a single guest. If you want to have a shared Crypto domain and dedicated Crypto domain on the same LPAR, you would need to assign at least two domains to the LPAR. The dedicated domain would be reserved for the guest when the z/VM system is IPLed. The remaining domains would then be used for the shared CRYPTO APVIRT statement.

For more information about assigning Crypto devices to a z/VM LPAR, see *z/VM CP Planning and Administration*.

Apache SSL configurations to exploit IBM cryptographic hardware acceleration

If your virtual servers have been granted access to cryptographic hardware acceleration hardware, this topic demonstrates how to configure Apache to use it.

To verify that our Linux virtual server is already configured with cryptographic acceleration, we issued the following command:

```
litstat1: # rcz90crypt status
 Checking for module z90crypt:
 running
```

If you see `running` in the response, you most likely have the cryptographic module running and hardware available for use with Apache.

Also, to see if the acceleration configuration is currently being used, the following command shows the number of open connections to the cryptographic hardware:

```
litstat1: # cat /proc/driver/z90crypt

zcrypt version: 2.1.0
Cryptographic domain: 7
Total device count: 1
PCICA count: 0
PCICC count: 0
PCIXCC MCL2 count: 0
PCIXCC MCL3 count: 0
CEX2C count: 0
CEX2A count: 1
requestq count: 0
pendingq count: 0
Total open handles: 1


Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
           0000000000000000 0000000000000000 0600000000000000 0000000000000000


Waiting work element counts
```

```
                0000000000000000 0000000000000000 0000000000000000 0000000000000000


Per-device successfully completed request counts
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 0000034E 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

If the **total open handles** value reports a value greater than 0, some existing process, such as an active Web server, is already making use of the cryptographic device.

If you have no /proc/driver/z90crypt device, you can try to manually insert the module with:

```
litstat1: # modprobe z90crypt
```

If the module loads successfully, you should be set to continue configuring your Apache2 installation. If you receive errors about not being able to create a device or other negative indications in the output, it more than likely indicates that your virtual machine does not have access to cryptographic hardware. This could be a result of a missing entry in your guest's directory entry or improperly configured hardware.

You can begin the process of adding SSL support to a stock Apache installation on SLES 10 by edting the Apache2 configuration file, /etc/sysconfig/apache2. Ensure that the APACHE_MODULES line contains `ssl` in the list. For instance:

```
APACHE_MODULES="actions alias auth_basic authn_file authz_host authz_groupfile
 authz_default authz_user authn_dbm autoindex cgi dir env expires include
 log_config mime negotiation setenvif ssl suexec userdir php5"
```

Next, you should ensure you have the IBM Open SSL certificate authority package installed. This can be accomplished easily from **yast** by selecting the openssl-ibmca package for normal installation. Then, if you do not have an existing SSL configuration file, simply copy the sample SSL-specifc Apache configuration file, as shown here:

```
litstat1: # cp /etc/apache2/vhosts.d/vhost-ssl.template  /etc/apache2/vhosts.d/main-ssl.conf
```

Additionally, configure the ssl-global configuration for Apache, specifying to use the ibmca device for SSL cryptography. On a default configuration, the change needs to be done right before the closing **<IfModule></IfModule>** tag (in a default configuration, this is around line 70):

```
litstat1: # vi /etc/apache2/ssl-global.conf +70

--------------------------------------------------------------------------------
        #Add the following
 line to enable ibmca exploitation
        SSLCryptoDevice ibmca<IfModule></IfModule>
--------------------------------------------------------------------------------
```

As indicated in the documentation contained in the ssl-global.conf file, we must also generate a new certificate using the provided Apache scripts. To generate a certificate, simply use the script provided with the Apache installation:

```
litstat1:/usr/share/doc/packages/apache2 # ./certificate.sh

SSL Certificate Generation Utility (mkcert.sh)
Copyright (c) 1998 Ralf S. Engelschall, All Rights Reserved.
Generating test certificate signed by Snake Oil CA [TEST]
WARNING: Do not use this for real-life/production systems

STEP 0: Decide the signature algorithm used for certificate

The generated X.509 CA certificate can contain either
RSA or DSA based ingredients. Select the one you want to use.

Signature Algorithm ((R)SA or (D)SA) [R]:


STEP 1: Generating RSA private key (1024 bit) [server.key]

36364 semi-random bytes loaded

Generating RSA private key, 1024 bit long modulus

..................++++++

.....++++++

e is 65537 (0x10001)


STEP 2: Generating X.509 certificate signing request [server.csr]

You are about to be asked to enter information that will be incorporated
into your certificate request. What you are about to enter is what is called a
Distinguished Name or a DN. There are quite a few fields but you can leave some blank.
For some fields there will be a default value.
If you enter '.', the field will be left blank.

-----

1. Country Name             (2 letter code) [XY]:

2. State or Province Name   (full name)     [Snake Desert]:

3. Locality Name            (eg, city)      [Snake Town]:

4. Organization Name        (eg, company)   [Snake Oil, Ltd]:

5. Organizational Unit Name (eg, section)   [Webserver Team]:

6. Common Name              (eg, FQDN)      [www.snakeoil.dom]:litstat1

7. Email Address            (eg, name@FQDN) [www@snakeoil.dom]:


STEP 3: Generating X.509 certificate signed by Snake Oil CA [server.crt]

Certificate Version (1 or 3) [3]:

Signature ok

subject=/C=XY/ST=Snake Desert/L=Snake Town/O=Snake Oil, Ltd/OU=Webserver
```

```
Team/CN=litstat1/emailAddress=www@snakeoil.dom

Getting CA Private Key

Verify: matching certificate & key modulus

Verify: matching certificate signature

/etc/apache2/ssl.crt/server.crt: /C=XY/ST=Snake Desert/L=Snake Town/O=Snake Oil,
Ltd/OU=Certificate Authority/CN=Snake Oil CA/emailAddress=ca@snakeoil.dom

error 10 at 1 depth lookup:certificate has expired

OK

STEP 4: Enrypting RSA private key with a pass phrase for security [server.key]

The contents of the server.key file (the generated private key) has to be kept secret.
So we strongly recommend you to encrypt the server.key file with a Triple-DES cipher
and a Pass Phrase.

Encrypt the private key now? [Y/n]: n

Warning, you're using an unencrypted RSA private key.

Please notice this fact and do this on your own risk.

RESULT: Server Certification Files


o  conf/ssl.key/server.key

   The PEM-encoded RSA private key file which you configure
   with the 'SSLCertificateKeyFile' directive (automatically done
   when you install via APACI). KEEP THIS FILE PRIVATE!



o  conf/ssl.crt/server.crt

   The PEM-encoded X.509 certificate file which you configure
   with the 'SSLCertificateFile' directive (automatically done
   when you install via APACI).


o  conf/ssl.csr/server.csr

   The PEM-encoded X.509 certificate signing request file which
   you can send to an official Certificate Authority (CA) in order
   to request a real server certificate (signed by this CA instead
   of our demonstration-only Snake Oil CA) which later can replace
   the conf/ssl.crt/server.crt file.

WARNING: Do not use this for real-life/production systems
```

With the certificate generated, to verify that the Apache configuration file is correctly enabled for SSL, you can use the following command:

```
httpd2 -S -DSSL
 VirtualHost configuration:
 Syntax OK
```

To start the Apache2 process with SSL for test purposes, simply stop any currently executing Apache2 processes and issue:

```
litstat1: # /etc/init.d/apache2 startssl
```

Now we can ensure that the Linux kernel driver which provides the underlying support the modssl configuration via the /proc Linux kernel interface:

```
litstat1:~ # cat /proc/driver/z90crypt

zcrypt version: 2.1.0
Cryptographic domain: 13
Total device count: 1
PCICA count: 1
PCICC count: 0
PCIXCC MCL2 count: 0
PCIXCC MCL3 count: 0
CEX2C count: 0
CEX2A count: 0
requestq count: 0
pendingq count: 0
Total open handles: 1

Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
           0000000000000000 0000000000000000 0000000000000000 0000000000100000

Waiting work element counts
           0000000000000000 0000000000000000 0000000000000000 0000000000000000

Per-device successfully completed request counts

    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

If no errors on Apache startup and total open handles is 1 or more, you should be able to proceed with a Web browser-based test or a command line test (such as the Linux **wget** command) to confirm that SSL is operational. Performing repeated SSL accesses while monitoring this proc interface will show changes in the per-device count section.

To ensure the z90crypt module is automatically loaded at boot, you must issue the following command:

```
chkconfig z90crypt on
```

You must also use **chckconfig** to make Apache start automatically at boot time.

Chapter 27. Future Linux on System z projects

For the next phase of our testing, we plan to continue implementing systems management architectures throughout our environment based on the core infrastructure demonstrated in this report. In addition to the areas we have already touched upon, such as availability management, data management, security management, and related systems programmer tasks, we plan to explore the critical aspects of performance management and capacity management, which we have not yet covered. In order to continue our emphasis on software management, we will continue our established process of documenting necessary upgrades to our middleware and operating systems to maintain current, supported levels. To expand in the area of security management, we plan to integrate several Tivoli products that we are currently investigating.

In addition, we have a unique opportunity scheduled for our laboratory in the fall. We have planned 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 reports. It is our hope that, with time permitting, we will be able to document some of our lessons learned and best practices for that migration, including any data management tasks related to unforeseen problems caused by the migration.

It is our hope that this report has provided you with some valuable insight into the tools and strategies we have discussed. As always, we encourage you to send us 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:

- Four *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)
  - One IBM System z9 Business Class (z9 BC)
  - One IBM System z9 Enterprise Class (z9 EC)
  - One IBM System z10 Enterprise Class (z10 EC)

  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.)
- Six *coupling facilities* (CFs):
  - One 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 104 on page 270 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.

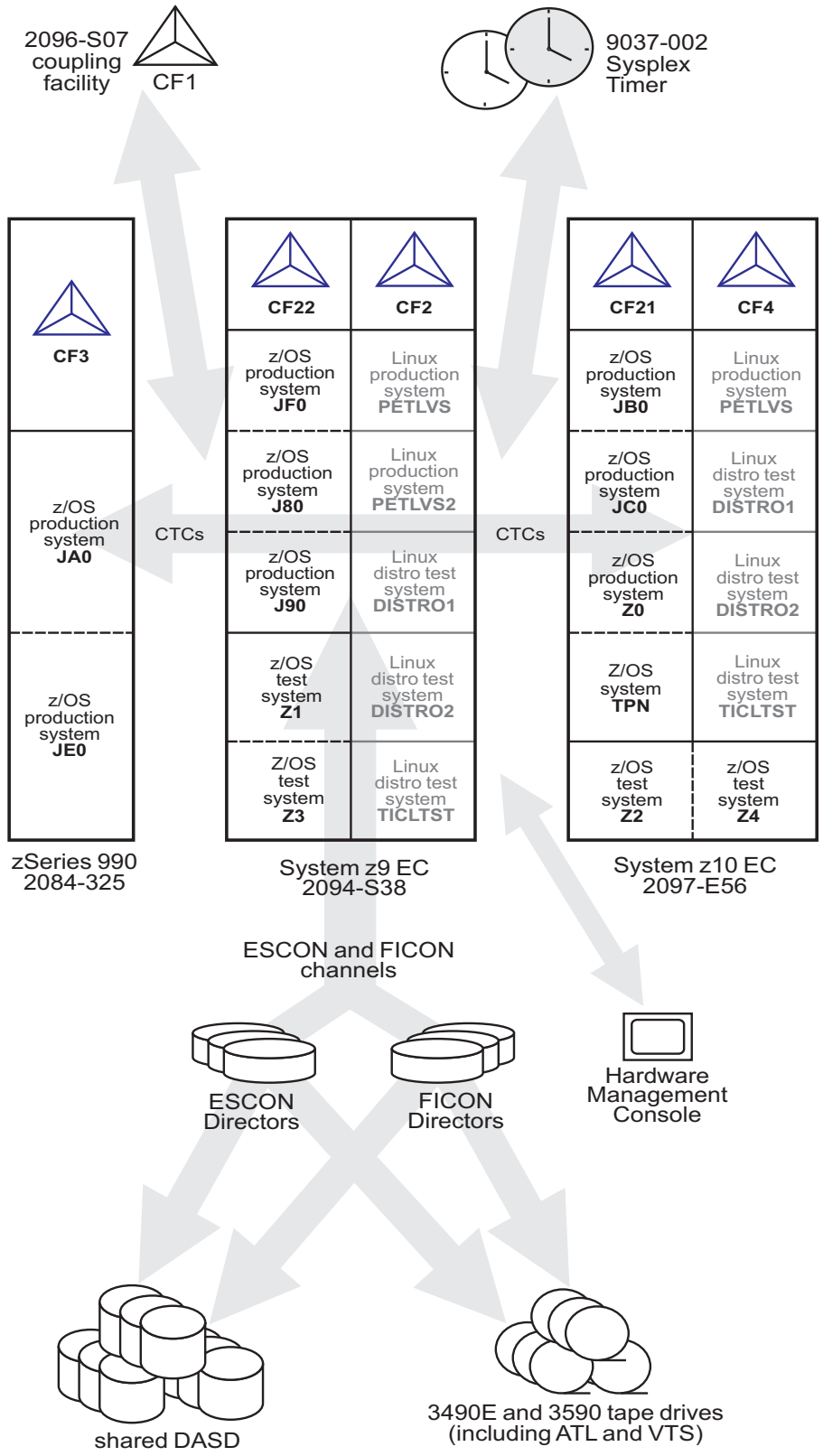


Figure 104. Our sysplex hardware configuration

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 104 on page 270.

Mainframe server details

Table 7 provides information about the mainframe servers in our sysplex environment.

Table 7. 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	**JAO**, z/OS production system Plex 1 20 shared CPs 2 shared zAAPs
				22528M	2	**JE0**, 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	**J80**, z/OS production system Plex 1 42 shared CPs 2 shared zIIPs 2 shared zAAPs
				112640M	0	**J90**, z/OS production system Plex 1 41 shared CPs 2 shared zIIPs 2 shared zAAPs
				15360M	0	**JF0**, z/OS production system Plex 1 16 shared CPs 2 shared zIIPs 2 shared zAAPs
				30720M	0	**Z1**, z/OS test system Plex 2 8 shared CPs 2 shared zIIPs 2 shared zAAPs
				30720M	0	**Z3**, z/OS test system Plex 2 8 shared CPs 2 shared zIIPs 2 shared zAAPs
				256M	1	**PETLVS**, Linux production system 1 shared CP weight of 10
				4096M	1	**PETLVS2**, Linux production system 4 shared CPs weight of 10
				8192M	1	**DISTR01**, Linux distribution test system 2 shared IFLs weight of 10
				2048M	1	**DISTR02**, Linux distribution test system 2 shared IFLs weight of 10
				1024M	1	**TICLTST**, Linux distribution test 1 shared IFL weight of 10

Table 7. 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 EC Model E56 (2097-E56)	1 CPC 45 CPs 5 ICFs	LPAR mode 4 LPARs (1 LP is a coupling facility)	256M	9216M		**Z0**, z/OS production system Plex 1 8 shared CPs
				30720M	0	**JB0**, z/OS production system Plex 1 16 shared CPs 2 shared zAAPs
				22528M	0	**JC0**, z/OS production system Plex 1 16 shared CPs 2 shared zAAPs
				6144M	0	**TPN**, z/OS system for TPNS Plex 1 12 shared CPs
				10752M	0	**Z2**, z/OS test system Plex 2 8 shared CPs
				30720M	0	**Z4**, z/OS test system Plex 2 8 shared CPs
				4096M	1	**PETLVS**, Linux production system 4 shared CPs weight of 10
				3072M	1	**DISTRO1**, Linux distribution test system 2 shared IFLs weight of 10
				2048M	1	**DISTRO2**, Linux distribution test system 2 shared IFLs weight of 10
				1024M	1	**TICLTST**, Linux distribution test 1 shared IFL weight of 10

Coupling facility details

Table 8 provides information about the coupling facilities in our sysplex. Figure 104 on page 270 further illustrates the coupling facility channel distribution as described in Table 8.

Table 8. Our coupling facilities

Coupling facility name	Model description CPCs and CPs CFLEVEL (CFCC level) Controlled by	Storage
CF1 (Plex 1)	IBM System z9 BC Model 2096-S07 standalone coupling facility 1 CPC with 4 CPs CFLEVEL=15 (CFCC Release 15.00, Service Level 00.29) Controlled by the HMC	14G
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 00.29) Controlled by the HMC	14G
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	14G
CF4 (Plex 1)	Coupling facility LPAR on a System z10 EC Model E56 (2097-E56) 3 dedicated ICF CPs CFLEVEL=15 (CFCC Release 15.00, Service Level 04.01) Controlled by the HMC	14G
CF21 (Plex 2)	Coupling facility LPAR on a System z10 EC Model E56 (2097-E56) 1 dedicated ICF CP CFLEVEL=15 (CFCC Release 14.00, Service Level 00.17) 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 00.29) Controlled by the HMC	6G

Table 9 illustrates our coupling facility channel configuration on Plex 1.

Table 9. Coupling facility channel configuration on Plex 1

	Coupling facility (CF) images			
Machine type **z/OS images** **CF images**	**2096-S07** **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 1 ICB	4 CFP 1 CBP	4 CFP	4 ICP
2094-S54 **J80, J90, JF0** **CF2**	3 CFP 1 ICB	8 ICP	1 CBP 3 CFP	5 ICP 1 ICB
* = Same links				

Table 10 on page 274 illustrates our coupling facility channel configuration on Plex 2.

Table 10. Coupling facility channel configuration on Plex 2

	Coupling facility (CF) Images	
Machine type **z/OS images** **CF images**	**2094-S54** **CF22**	**2097-E56** **CF21**
2097-E56 **Z2, Z4CF21** **CF21**	2 CFP * 1 CBP *	2 ICP
2094-S54 **Z1, Z3** **CF22**	2 ICP	2 CFP * 1 CBP *
* = Same links		

In addition to our coupling facility channel configuration listed in Table 9 on page 273 and Table 10, we configured 1 ISC-3 link and 1 ICB-3 link between our 2084-325 CPC and our 2096-S07 CPC which will be used as Server Time Protocol (STP) timing-only links in our new STP environment. See Chapter 4, "Migrating to a Server Time Protocol Coordinated Timing Network," on page 33 for more information about STP timing-only links.

Other sysplex hardware details

Table 11 highlights information about the other hardware components in our sysplex.

Table 11. Other sysplex hardware configuration details

Hardware element	Model or type	Additional information
External Time Reference (ETR)	Sysplex Timer (9037-002 with feature code 4048)	We use the Sysplex Timer with the Expanded Availability feature, which provides two 9037 control units connected with fiber optic links. We don't have any Sysplex Timer logical offsets defined for any of the LPs in our sysplex.
Channel subsystem	CTC communications connections	We have CTC connections from each system to every other system. We now use both FICON® and ESCON® CTC channels on all of our CPCs. **Note:** All of our z/OS images use both CTCs and coupling facility structures to communicate. This is strictly optional. You might choose to run with structures only, for ease of systems management. We use both structures and CTCs because it allows us to test more code paths. Under some circumstances, XCF signalling using CTCs is faster than using structures. See *S/390 Parallel Sysplex Performance* for a comparison.
	Coupling facility channels	We use a combination of ISC, ICB, and IC coupling facility channels in peer mode. We use MIF to logically share coupling facility channels among the logical partitions on a CPC. We define at least two paths from every system image to each coupling facility, and from every coupling facility to each of the other coupling facilities.
	ESCON channels	We use ESCON channels and ESCON Directors for our I/O connectivity. Our connections are "any-to-any", which means every system can get to every device, including tape. (We do not use any parallel channels.)
	FICON channels	We have FICON native (FC) mode channels from all of our CPCs to our Enterprise Storage Servers and our 3590 tape drives through native FICON switches. (See *FICON Native Implementation and Reference Guide*, SG24-6266, for information about how to set up this and other native FICON configurations.) We maintain both ESCON and FICON paths to the Enterprise Storage Servers and 3590 tape drives for testing flexibility and backup. Note that FICON channels do not currently support dynamic channel path management. We have also implemented FICON CTCs, as described in the IBM Redpaper *FICON CTC Implementation* available on the IBM Redbooks Web site.

Table 11. Other sysplex hardware configuration details (continued)

Hardware element	Model or type	Additional information
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. **Note:** 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) V3R1
- 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 for z/OS V6.0.2
- WebSphere MQ for z/OS V6
- Websphere Message Broker V6

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 105 on page 276 shows a high-level view of our sysplex software configuration.

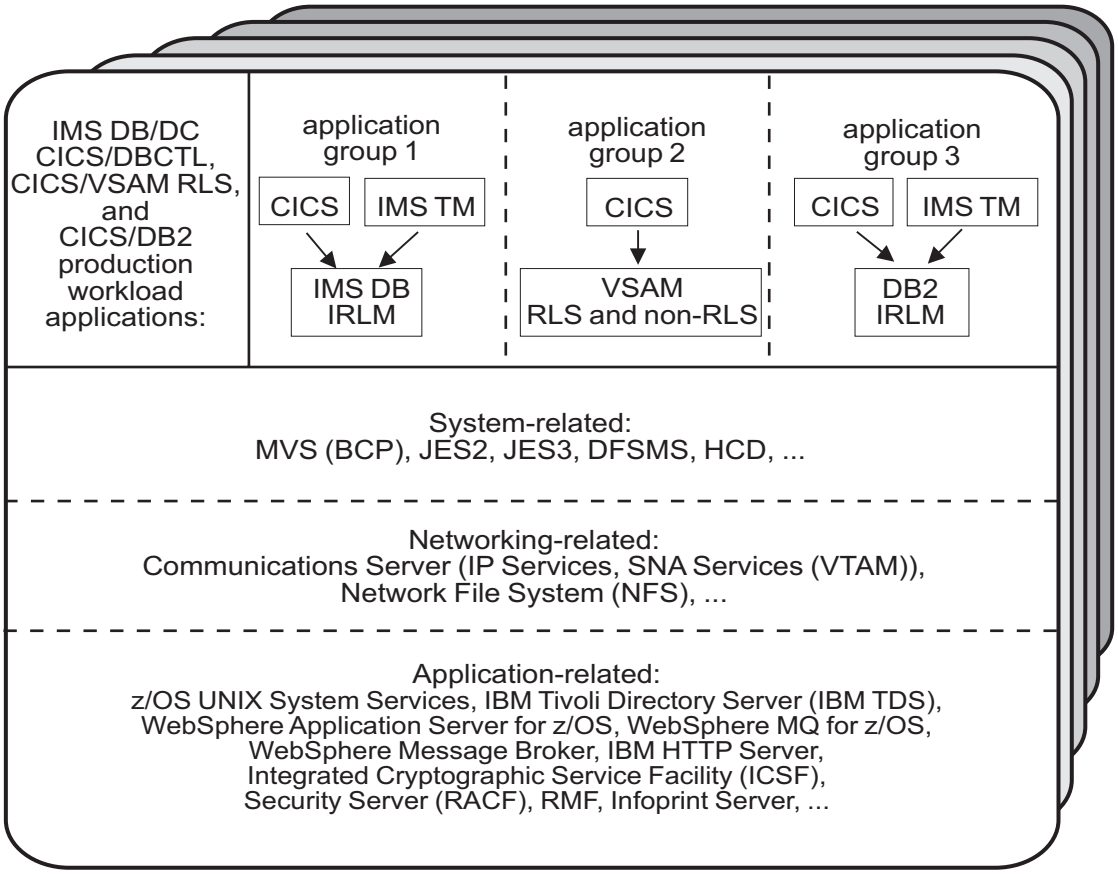


Figure 105. Our sysplex software configuration

We run three separate application groups in one sysplex and each application group spans multiple systems in the sysplex. Table 12 provides an overview of the types of transaction management, data management, and serialization management that each application group uses.

Table 12. 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 299.

About our naming conventions

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

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

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

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

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

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

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 106 provides a logical view of our networking configuration.

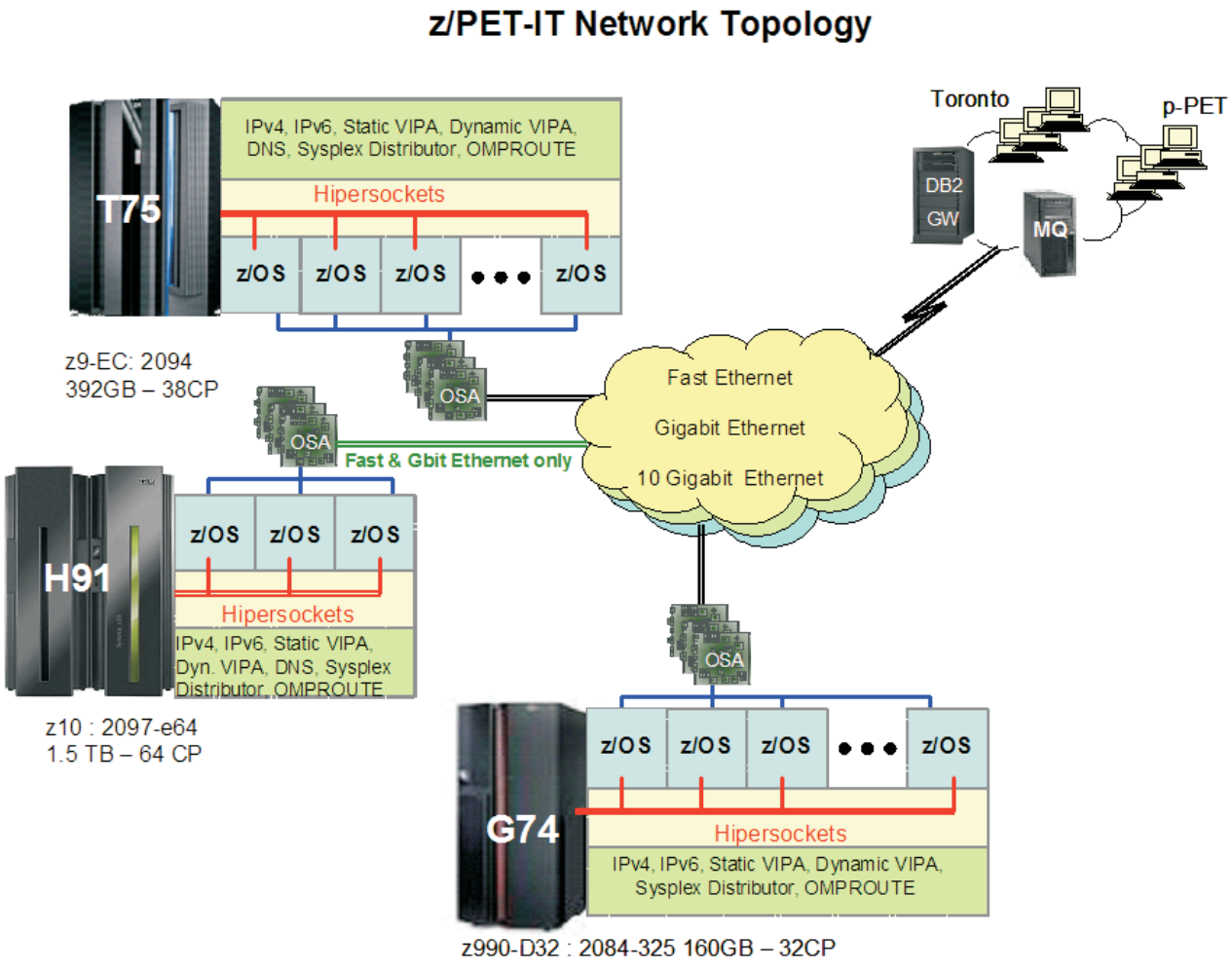


Figure 106. Our networking topology

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

TCPIP Profile changes

We made the following additions to our IPv6 INTERFACE statements:

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

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

Dynamic XCF addition

We made the following addition for our Dynamic XCF:

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

Dynamic VIPA additions

The following statement was added to our VIAPDYNAMIC section:

```
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 107 illustrates our VTAM configuration.

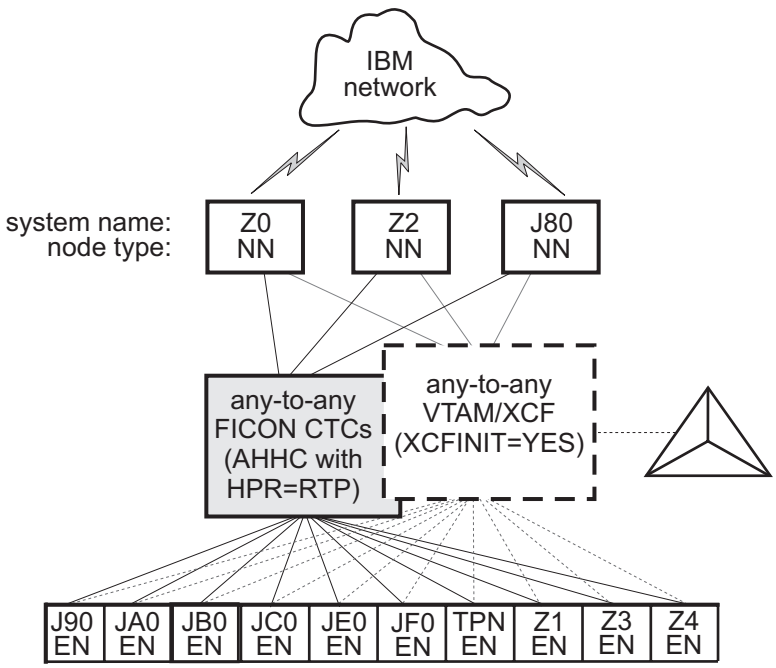


Figure 107. Our VTAM configuration

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

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

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 293.

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 108 on page 284: 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 108 on page 284 : 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.

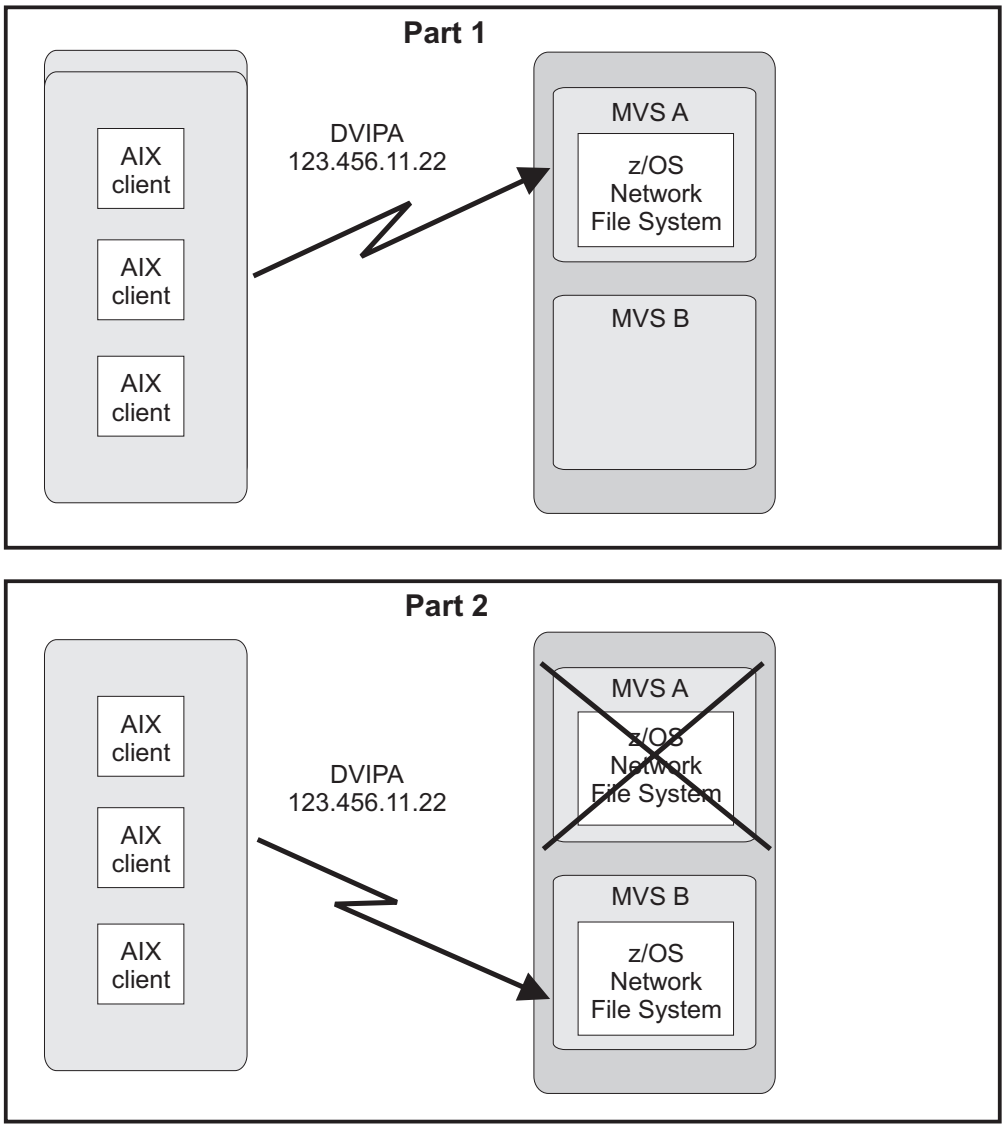


Figure 108. NFS configuration

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

Step for setting up our NFS environment

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

1. Acquiring dynamic VIPA:

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

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

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

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

   ---

2. Defining the NFS element:

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

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

---

3. Loading the ARM policy:

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

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

---

4. Registering NFS using an ARM policy:

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

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

---

5. Terminating the address space:

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

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

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 288
- "Our LDAP configuration" on page 289

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 7, "Migrating to and using ICSF HCR7750," on page 109.

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 our December 2006 test report.

Network Authentication Service configuration

Figure 109 shows an overview of our Network Authentication Service (NAS) configuration.

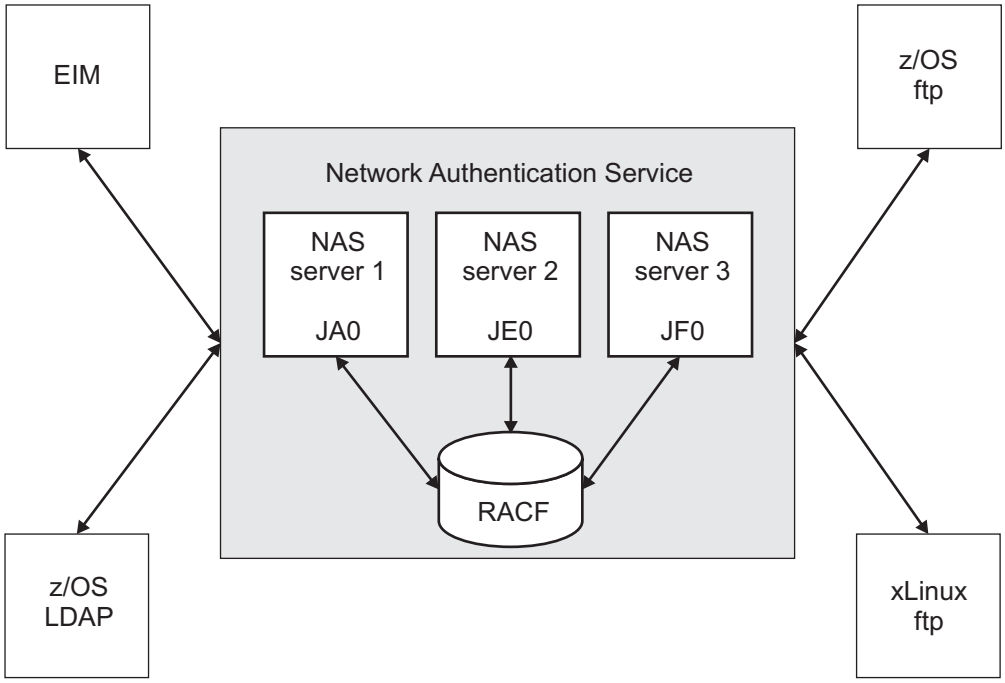


Figure 109. Overview of our Network Authentication Service configuration

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.

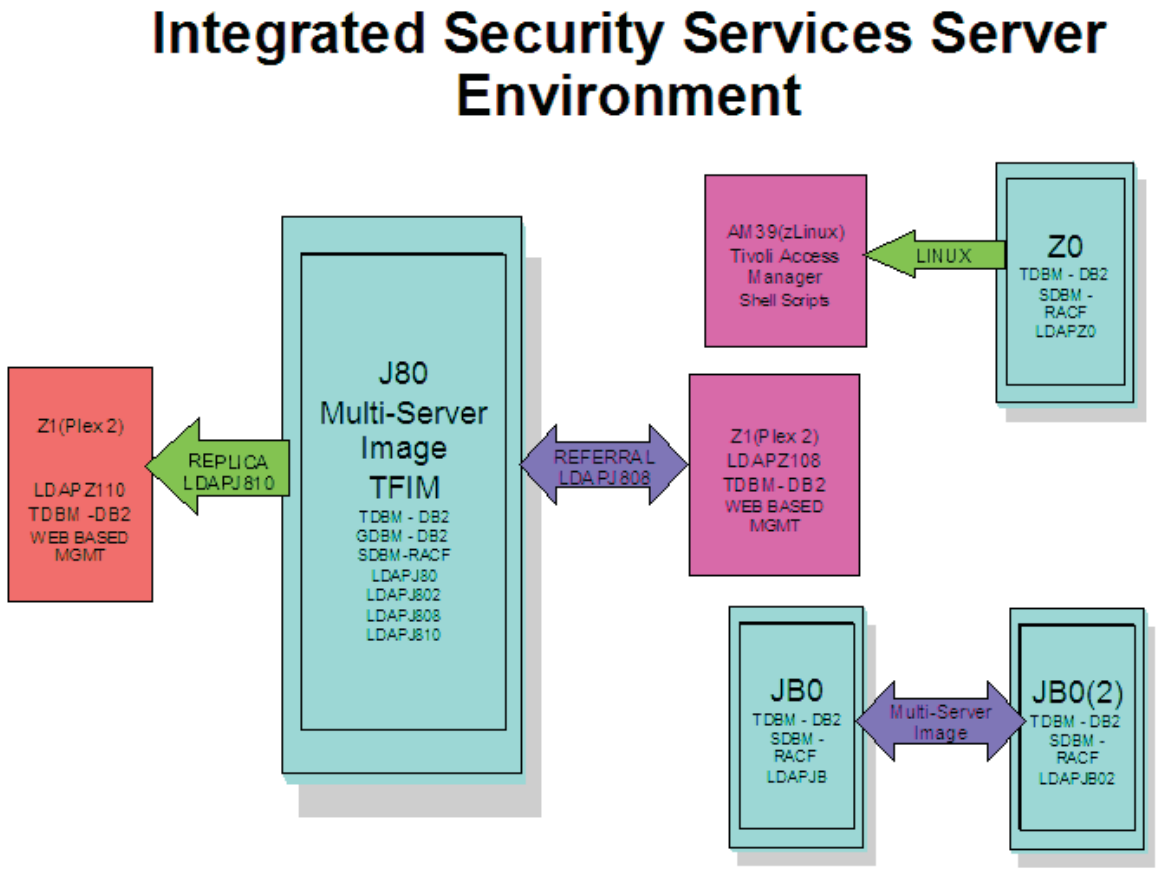


Figure 110. Integrated Security Services (ISS) LDAP environment

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.

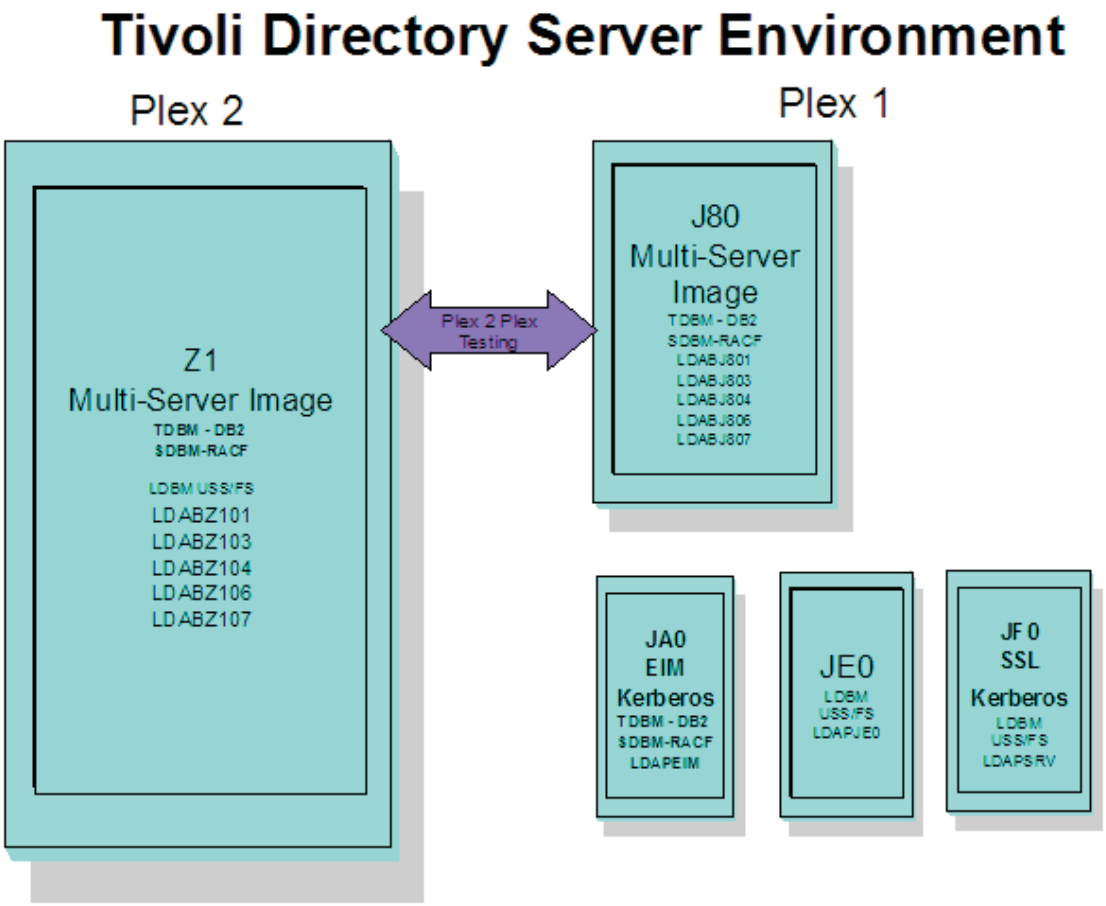


Figure 111. IBM Tivoli Directory Server (IBM TDS) environment

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 one of our z/OS images and Apache running on an xLinux box. Both of these http servers access the LDAPJE0 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 13 summarizes the workloads we run during our prime shift and off shift. We describe each workload in more detail below.

Table 13. Summary of our workloads

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

Base system workloads

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

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

Automatic tape switching: We run 2 batch workloads to exploit automatic tape switching and the ATS STAR tape sharing function. These workloads use the Virtual Tape Server and DFSMSrmm™, as described in our December 1998 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.

EQQXIT51 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.

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 queueing 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.

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

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

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

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

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

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

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

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

Database product workloads

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 record level sharing (RLS) and non-RLS
- Application group 3 — DB2 data sharing (four different OLTP workloads, as well as several batch workloads).

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

IMS data sharing workloads: In application group one, we run 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 use VSAM files.

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

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

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

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

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

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

Database product batch workloads

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

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

We run our batch workloads under 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" on page 304
- "RMF Workload Activity in WLM goal mode" on page 306

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.

```
                                R M F   S U M M A R Y   R E P O R T
                                                                                PAGE  001
           z/OS  V1R9              SYSTEM ID J80             START 07/17/2007-10.45.00  INTERVAL 00.14.59
                                   RPT VERSION V1R9 RMF      END   07/17/2007-11.00.00  CYCLE 0.100 SECONDS

NUMBER OF INTERVALS 1
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
07/17 10.45.00 14.59  89.9   2.6 18474   0.0   176   175    42    41   493   485     2     0    49    38 0.00   0.02


                                R M F   S U M M A R Y   R E P O R T
                                                                                PAGE  001
           z/OS  V1R9              SYSTEM ID J90             START 07/17/2007-10.45.00  INTERVAL 00.15.00
                                   RPT VERSION V1R9 RMF      END   07/17/2007-11.00.00  CYCLE 0.100 SECONDS

NUMBER OF INTERVALS 1
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
07/17 10.45.00 15.00  72.1   1.8 12678   0.0   177   174    14    14   386   381     1     0    23    18 0.00   0.00


                                R M F   S U M M A R Y   R E P O R T
                                                                                PAGE  001
           z/OS  V1R9              SYSTEM ID JB0             START 07/17/2007-10.45.00  INTERVAL 00.15.00
                                   RPT VERSION V1R9 RMF      END   07/17/2007-11.00.00  CYCLE 0.100 SECONDS

NUMBER OF INTERVALS 1
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
07/17 10.45.00 15.00  76.5   1.9 10608  1213   179   178     7     7   690   685     0     0    29    22 0.00   0.00


                                R M F   S U M M A R Y   R E P O R T
                                                                                PAGE  001
           z/OS  V1R9              SYSTEM ID JC0             START 07/17/2007-10.45.00  INTERVAL 00.15.00
                                   RPT VERSION V1R9 RMF      END   07/17/2007-11.00.00  CYCLE 0.100 SECONDS

NUMBER OF INTERVALS 1
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
07/17 10.45.00 15.00  65.5   6.4 994.1   0.0   175   174     0     0   416   413     1     0    24    19 0.00   0.02


                                R M F   S U M M A R Y   R E P O R T
                                                                                PAGE  001
           z/OS  V1R9              SYSTEM ID JA0             START 07/17/2007-10.45.00  INTERVAL 00.14.59
                                   RPT VERSION V1R9 RMF      END   07/17/2007-11.00.00  CYCLE 0.100 SECONDS

NUMBER OF INTERVALS 1
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
07/17 10.45.00 14.59  81.3   3.0 12716   0.0    33    31    32    32   420   407     2     0    85    70 0.00   0.00
```

```
                                    R M F   S U M M A R Y   R E P O R T
                                                                                                  PAGE  001
            z/OS  V1R9              SYSTEM ID JE0             START 07/17/2007-10.45.00  INTERVAL 00.15.00
                                    RPT VERSION V1R9 RMF      END   07/17/2007-11.00.00  CYCLE 0.100 SECONDS

 NUMBER OF INTERVALS 1
 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
 07/17 10.45.00 15.00  26.1   1.0  4660   0.0     3     1     1     1   371   366     1     0    39    28 0.00   0.00


                                    R M F   S U M M A R Y   R E P O R T
                                                                                                  PAGE  001
            z/OS  V1R9              SYSTEM ID JF0             START 07/17/2007-10.45.00  INTERVAL 00.15.00
                                    RPT VERSION V1R9 RMF      END   07/17/2007-11.00.00  CYCLE 0.100 SECONDS

 NUMBER OF INTERVALS 1
 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
 07/17 10.45.00 15.00   8.3   4.4 293.5   0.0     3     1     1     1   389   387     1     0    36    28 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 1972 CICS transactions/second.

```
           HARDCOPY  RMF V1R9   Sysplex Summary - UTCPLXJ8       Line 1 of 84
Command ===>                                                   Scroll ===> CSR
WLM Samples: 239     Systems: 7  Date: 07/17/07 Time: 10.45.00 Range: 60    Sec

                     >>>>>>>>XXXXXXXXXXXXXXXXXX<<<<<<<<

Service Definition: WLMDEF02              Installed at: 07/12/07, 20.00.32
     Active Policy: WLMPOL01              Activated at: 07/12/07, 20.00.40
               ------- 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           30                              0.117 1.662  2.12M  2.13M
BATI1V90 S  1    90  69                         1.31  0.000 0.000  0.000  0.000
BATI2V50 S  2    50  66                         0.76  0.000
DISCR    S  D      9.6                               0.117
CICS     W           71                               1972 0.000  0.843  0.982
CI2V60   S  2    60  71                         0.85  0.033
CI280%P6 S  2       N/A  0.600 80%          95%  0.50   1471 0.000  0.201  0.426
CI350%10 S  3       N/A  10.00 50%          95%  0.70  2.150 0.000  6.081  6.081
CI390%01 S  3       N/A  1.000 90%          85%  **** 498.6 0.000  2.282  1.137
ICSS     W           42                              51.25 0.001  0.086  0.086
ICI2V50  S  2    50  70                         0.71  41.02 0.001  0.092  0.093
IC14V50  S  4    50  13                         3.75  10.23 0.001  0.060  0.060
IMS      W          N/A                              98.10 0.000  0.349  0.606
II290%P5 S  2       N/A  0.500 90%          94%  0.70  31.38 0.000  0.158  0.177
II390%P7 S  3       N/A  0.700 90%          69%  2.00  65.28 0.000  0.444  0.824
II490%01 S  4       N/A  1.000 90%         100%  0.50  1.433 0.000  0.108  0.114
STC      W           34                              27.70 0.001  3.997  3.998
STCI1V40 S  1    40  90                         0.44  0.400 0.002  0.256  0.258
STCI1V90 S  1    90  60                         1.50  0.000 0.000  0.000  0.000
STCI2V30 S  2    30  10                         2.90  0.000
STCI2V40 S  2    40  68                         0.59  0.233 0.002  0.509  0.511
STCI2V50 S  2    50  74                         0.67  0.000
```

```
STCI2V60 S  2    60  71                          0.85  0.000
STCI2V70 S  2    70  85                          0.82  0.000 0.000  0.000  0.000
STCI5V05 S  5     5  61                          0.08  0.750 0.000  0.019  0.019
STCI5V10 S           16                                0.000 0.000  0.000  0.000
         3  5    10  16                          0.63  0.000 0.000  0.000  0.000
STCOMVS  S           17                                26.32 0.001  4.198  4.199
         1  2    30  44                          0.68  4.900 0.000  2.468  2.468
         2  3    20  58                          0.34  7.600 0.000  3.730  3.730
         3  5    10  16                          0.64  13.82 0.002  5.070  5.072
SYSTEM   W           78                                0.017 0.432  0.663  1.095
SYSSTC   S      N/A  83    N/A                         0.017
SYSTEM   S      N/A  68    N/A                         0.000 0.000  0.000  0.000
TSO      W           51                                8.283 0.000  6.840  6.840
TSO      S  2        51  2.000 AVG  6.840  AVG  3.42  8.283 0.000  6.840  6.840
WAS      W           11                                278.0 0.181  0.478  0.659
WI1VEL30 S  1    30  81                          0.37  0.000 0.000  0.000  0.000
WI180%01 S  1       9.5  1.000 80%          81%  1.00  278.0 0.181  0.478  0.659
WI2VEL50 S  2    50  22                          2.33  0.000
CICSCONV R          N/A                               2.150 0.000  6.081  6.081
CICSCPSM R          N/A                               1.200 0.000  0.002  0.002
CICSMISC R          N/A                               213.0 0.000  0.036  0.036
CICSWEB  R           46                               0.000 0.000  0.000  0.000
Name     T  I  Goal Act  ---Goal--- --Actual--  Indx  Rate   Time   Time   Time
CQSREP   R           71                                0.000 0.000  0.000  0.000
CSQCCHIN R           76                                0.000 0.000  0.000  0.000
CSQCMSTR R           71                                0.000 0.000  0.000  0.000
DB2IRLM  R           90                                0.000 0.000  0.000  0.000
DB2REP   R           80                                0.000 0.000  0.000  0.000
DB2WLM   R           50                                0.000 0.000  0.000  0.000
DDF      R           61                                0.750 0.000  0.019  0.019
EWLMWORK R           48                                49.72 0.002  0.156  0.158
FDBRREP  R           40                                0.000 0.000  0.000  0.000
GRS      R           76                                0.000 0.000  0.000  0.000
IMSREG   R           60                                0.000 0.000  0.000  0.000
IMSREP   R           73                                0.000 0.000  0.000  0.000
MQ       R           43                                0.000 0.000  0.000  0.000
OMVS     R           77                                0.000 0.000  0.000  0.000
OMVSINIT R           90                                0.400 0.002  0.256  0.258
OMVSTPNS R          0.0                                0.000 0.000  0.000  0.000
         3       10 0.0                          N/A   0.000 0.000  0.000  0.000
RMF      R           97                                0.000 0.000  0.000  0.000
RMFGAT   R           83                                0.000 0.000  0.000  0.000
RWASCR   R           22                                0.000 0.000  0.000  0.000
RWASDFLT R          0.0                                82.78 0.024  0.145  0.170
RWASMED  R          0.0                                145.5 0.331  0.777  1.108
RWASWEB  R           81                                0.000 0.000  0.000  0.000
SLOWICSS R          0.0                                10.23 0.001  0.060  0.060
SYSSTC   R           49                                0.083 0.447  2.44H  2.44H
         1           49                                0.083 0.447  2.44H  2.44H
         3       10  16                          0.63  0.000 0.000  0.000  0.000
TCPIP    R           62                                0.000 0.000  0.000  0.000
TPNSREP  R           85                                0.000 0.000  0.000  0.000
TSOREP   R           51                                8.283 0.000  6.840  6.840
WCSQBRK  R          0.0                                0.000 0.000  0.000  0.000
WCSQMSTR R           56                                0.000 0.000  0.000  0.000
WLM      R           97                                0.000 0.000  0.000  0.000
WPDB2    R           17                                0.000 0.000  0.000  0.000
WPOMV    R          0.0                                0.233 0.002  0.509  0.511
WT9STDM  R          0.0                                0.000 0.000  0.000  0.000
WT9STDMS R          0.0                                0.000 0.000  0.000  0.000
```

```
WT9ST00  R          0.0                              0.000 0.000  0.000  0.000
WT9ST00S R          0.0                              0.000 0.000  0.000  0.000
XCFAS    R           92                              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 94.3% of our CICS transactions are completing in 0.5 seconds, and our CICS workload is processing 1941.35 transactions per second.

```
1                                    W O R K L O A D   A C T I V I T Y
                                                                                                                PAGE   6
        z/OS V1R8              SYSPLEX UTCPLXJ8          START 01/26/2007-10.30.00 INTERVAL 000.14.59   MODE = GOAL
                             CONVERTED TO z/OS V1R8 RMF      END   01/26/2007-10.44.59

                                        POLICY ACTIVATION DATE/TIME 10/25/2006 13.29.47

   REPORT BY: POLICY=WLMPOL01   WORKLOAD=CICS       SERVICE CLASS=CICS       RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=2
                                                    CRITICAL     =NONE

   TRANSACTIONS   TRANS-TIME HHH.MM.SS.TTT
   AVG      0.00  ACTUAL                22
   MPL      0.00  EXECUTION             18
   ENDED  346638  QUEUED                 0
   END/S  385.16  R/S AFFIN              0
   #SWAPS      0  INELIGIBLE             0
   EXCTD  473510  CONVERSION             0
   AVG ENC  0.00  STD DEV              112
   REM ENC  0.00
   MS ENC   0.00

              RESP  ------------------------------- STATE SAMPLES BREAKDOWN (%) -------------------------------  ------STATE------
   SUB    P   TIME  --ACTIVE-- READY IDLE  ----------------------------WAITING FOR-----------------------------  SWITCHED SAMPL(%)
   TYPE        (%)   SUB  APPL             CONV  I/O LOCK MISC PROD LTCH                                        LOCAL SYSPL REMOT
   CICS  BTE  77.6   3.3   0.0   0.3  0.1  95.8  0.4  0.0  0.0  0.1  0.0                                         88.2   8.0   0.0
   CICS  EXE  28.8  60.6   0.0   6.9  0.0   0.0 29.8  0.0  0.2  2.5  0.0                                          7.2  42.5   0.0
   DB2   BTE   0.0   0.0   0.0   0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0                                          0.0   0.0   0.0
   DB2   EXE  19.3  86.2   0.0   0.0  0.0   0.0  0.4  0.6 12.6  0.0  0.2                                          0.0   0.0   0.0
   IMS   BTE   0.9  98.9   0.0   0.0  0.0   0.0  0.0  1.1  0.0  0.0  0.0                                          0.0   0.0   0.0
   IMS   EXE   5.4  90.9   0.0   0.0  0.0   0.0  0.0  9.1  0.0  0.0  0.0                                          0.0   0.0   0.0
   SMS   BTE   0.0   0.0   0.0   0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0                                          0.0   0.0   0.0
   SMS   EXE  31.6  12.1   0.0   0.0  0.0   0.0 55.4 32.5  0.0  0.0  0.0                                          0.0   0.0   0.0

   GOAL: RESPONSE TIME 000.00.00.600 FOR  80%

            RESPONSE TIME EX   PERF
   SYSTEM      ACTUAL%    VEL% INDX

   *ALL           100      N/A  0.5
   JE0            100      N/A  0.5
   J80            100      N/A  0.5
   J90            100      N/A  0.5
   Z0             100      N/A  0.5

1                                    W O R K L O A D   A C T I V I T Y
                                                                                                                PAGE   6
        z/OS V1R8              SYSPLEX UTCPLXJ8          START 01/26/2007-10.30.00 INTERVAL 000.14.59   MODE = GOAL
                             CONVERTED TO z/OS V1R8 RMF      END   01/26/2007-10.44.59

                                        POLICY ACTIVATION DATE/TIME 10/25/2006 13.29.47

                                     ----------RESPONSE TIME DISTRIBUTION----------
      ----TIME----    --NUMBER OF TRANSACTIONS--    -------PERCENT-------  0    10   20   30   40   50   60   70   80   90   100
      HH.MM.SS.TTT    CUM TOTAL       IN BUCKET    CUM TOTAL   IN BUCKET  |....|....|....|....|....|....|....|....|....|....|
   <  00.00.00.300          344K            344K        99.3        99.3  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
   <= 00.00.00.360          345K            741         99.5         0.2  >
   <= 00.00.00.420          345K            461          100         0.1  >
   <= 00.00.00.480          346K            324          100         0.1  >
   <= 00.00.00.540          346K            229          100         0.1  >
   <= 00.00.00.600          346K            151          100         0.0  >
   <= 00.00.00.660          346K            128          100         0.0  >
   <= 00.00.00.720          346K             86          100         0.0  >
   <= 00.00.00.780          346K             52          100         0.0  >
   <= 00.00.00.840          346K             36          100         0.0  >
   <= 00.00.00.900          346K             39          100         0.0  >
   <= 00.00.01.200          346K            102          100         0.0  >
```

```
<= 00.00.02.400          347K              95          100         0.0  >
>  00.00.02.400          347K             134          100         0.0  >
================================================================================================== WORKLOAD

REPORT BY: POLICY=WLMPOL01   WORKLOAD=CICS
                                    cics workload

TRANSACTIONS   TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE----  SERVICE TIMES  ---APPL %---  PAGE-IN RATES   ---STORAGE---
AVG     41.00  ACTUAL               143  SSCHRT  2536  IOC     23626K  CPU   2079.9   CP    297.69  SINGLE     0.0  AVG  17179.02
MPL     41.00  EXECUTION             62  RESP     1.7  CPU    341640K  SRB    607.7   AAPCP   3.00  BLOCK      0.0  TOT  704281.3
ENDED  496115  QUEUED                 0  CONN     0.7  MSO      1444M  RCT      0.0   IIPCP   0.28  SHARED     0.0  CEN  704281.3
END/S  551.25  R/S AFFIN              0  DISC     0.8  SRB     97488K  IIT     13.6                 HSP        0.0  EXP      0.00
#SWAPS      0  INELIGIBLE             0  Q+PEND   0.3  TOT      1907M  HST      0.0   AAP     0.97  HSP MISS   0.0
EXCTD  524444  CONVERSION             0  IOSQ     0.0  /SEC     2119K  AAP      8.7   IIP     1.47  EXP SNGL   0.0  SHR    227.98
AVG ENC  0.00  STD DEV            1.130                                 IIP     13.3                 EXP BLK    0.0
REM ENC  0.00                                           ABSRPTN    52K                               EXP SHR    0.0
MS ENC   0.00                                           TRX SERV   52K
```

Appendix F. Availability of our test reports

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 z/OS 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.

You can access our test reports on the Internet or on IBM Softcopy collections.

Availability on the Internet: You can view, download, and print the most current edition of our test report from our System z Platform Test Web site at www.ibm.com/servers/eserver/zseries/zos/integtst/.

You can also find our test reports on the z/OS Internet Library at www.ibm.com/servers/eserver/zseries/zos/bkserv/.

Each edition is available in the following formats:

- IBM BookManager® BOOK format

  On the Web, BookManager documents are served as HTML via IBM LibraryServer. You can use your Web browser (no plug-in or other applications are needed) to view, search, and print selected topics. You can also download individual BOOK files and access them locally using the IBM Softcopy Reader or IBM Library Reader™. You can get the Softcopy Reader or Library reader from the IBM Softcopy Web site at www.ibm.com/servers/eserver/zseries/softcopy/.
- Adobe® Portable Document Format (PDF)

  PDF documents require the Adobe Reader to view and print. Your Web browser can invoke the Adobe Reader to work with PDF files online. You can also download PDF files and access them locally using the Adobe Reader. You can get the Adobe Reader from www.adobe.com/products/acrobat/readstep.html.

Softcopy availability: BookManager BOOK and Adobe PDF versions of our test reports are included in the z/OS softcopy collections on CD-ROM and DVD. For more information about softcopy deliverables and tools, visit the IBM Softcopy Web site.

A note about the currency of our softcopy editions

Because we produce our test reports twice a year, June and December, we cannot meet the production deadline for the softcopy collections that coincide with the product's GA release and the softcopy collection refresh date six months later. Therefore, there is normally a one-edition lag between the release of our latest test report edition and the softcopy collection in which it is included. That is, the test report that appears in any given softcopy collection is normally one edition behind the most current edition available on the Web.

Other related publications: From our Web site, you can also access other related publications, including our companion publication, *z/OS V1R8.0 System z Parallel Sysplex Recovery*, GA22-7286.

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/servers/eserver/zseries/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 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 (® or ™), 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.

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

P

R

S

T

U

V

W

Z

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 9

Publication No. SA22-7997-07

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.

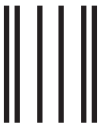
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
SA22-7997-07

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

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

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in USA

SA22-7997-07