IBM

z/OS Version 1 Release 7 Implementation

ServerPac, SMP/E, Installation, BCP, JES2, zFS, z/OS UNIX

IBM Health Checker for z/OS, Security, Language Environment

Communication Server, DFSMS, ISPF

Paul Rogers
Thomas Gabert Diana Nakajima
Amr Khafagy Andre Pradier
Lutz Kuehner Anthony Soares
Luiz Maia Claudia Tomas
Marc Muntane David Welch

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/OS Version 1 Release 7 Implementation

January 2006

SG24-6755-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xv.

First Edition (January 2006)

This edition applies to Version 1 Release 7 of z/OS (5637-A01), and Version 1, Release 7 of z/OS.e (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®
@server®
Redbooks (logo) ™
eServer™
z/Architecture™
z/OS®
z/VM®
z/VSE™
zSeries®
z9™
Advanced Peer-to-Peer Networking®
AIX 5L™
AIX®
APL2®
AS/400®
CICS®
CUA®
DB2 Connect™
DB2®
DFS™
DFSMSdfp™
DFSMSdss™
DFSMShsm™
DFSMSrmm™
DFSORT™
Enterprise Storage Server®
Extended Services®
ESCON®
FICON®
HiperSockets™
Infoprint®
IBM®
IMS™
Language Environment®
Lotus®
Multiprise®
MVS™
NetView®
OS/390®
Parallel Sysplex®
Processor Resource/Systems Manager™
PR/SM™
Redbooks™
RACF®
RMF™
System z9™
Tivoli®
TotalStorage®
Virtualization Engine™
VTAM®
WebSphere®

The following terms are trademarks of other companies:

Java, RSM, Solaris, Sun, Sun Microsystems, VSM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a 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.

Preface

This IBM® Redbook describes the new functions of z/OS® Version 1 Release 7. These functions further strengthen the zSeries® platform with enhancements designed to deliver increased availability of z/OS UNIX® System Services (z/OS UNIX), to support new security standards, and to improve enterprise-wide workload management. Enhancements are made to the following areas:

- Application integration with enhancements to the zSeries File System (zFS)
- Enhanced security with Communications Server support for IP filtering, Internet Key Exchange (IKE), and Virtual Private network (VPN), without requiring the use of the Integrated Security Services Firewall Technologies
- Security Server (RACF®) support for mixed-case passwords
- Improved availability with TCP/IP sysplex autonomics
- Greater scalability with support for sequential data sets larger than 65,535 tracks
- System Logger and XRC+ optimization of duplexing log data
- Greater ease of use with the IBM Health Checker for z/OS
- z/OS UNIX dynamic service activation and a new dynamic service activation function
- JES2 has checkpoint problem recovery and supports TCP/IP for NJE networking
- A new direct access device space management/Common VTOC Access Facility (DADSM/CVAF) device support address space starts during a system IPL
- IDCAMS FROMKEY/TOKEY support in REPRO MERGECAT is planned to help make it easier to move catalog entries from one catalog to another
- VSAM RLS 64-bit virtual can allow continued growth for applications using VSAM RLS.
- DFSMShsm™ extends TTOC records to support more than 330,000 data sets per volume and support DFSMShsm journal data sets larger than 65,535 tracks
- DFSMSrmm™ supports data sets larger than 65,535 tracks for the journal, journal backup, and certain temporary data sets.
- DFSMSdss™ exploits Format-1 Channel Command Words (CCWs) so that it can use storage obtained above the 16-MB line during EXCP processing
- Hardware Configuration Definition (HCD) Input Output Definition File (IODF) improvements

The team that wrote this redbook

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

Paul Rogers is a Consulting IT Specialist at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on various aspects of z/OS JES3, Infoprint® Server, and z/OS UNIX. Before joining the ITSO 18 years ago, Paul worked in the IBM Installation Support Center (ISC) in Greenford, England, providing OS/390® and JES support for IBM EMEA and the Washington Systems Center in Gaithersburg, Maryland.

Thomas Gabert is a z/OS systems programmer working for HVB Info GmbH Munich, Germany. He has 3 years of systems programming experience in mainframe environments on z/OS platforms. His areas of expertise include installation, maintenance, programming, and tools.

Amr Khafagy is a Certified IT Specialist working for IBM Canada. He has 20 years of experience in systems programming. His areas of expertise include MVS™, OS/390, z/OS, WebSphere®, UNIX System Services and Storage. He was an author of four other IBM Redbooks™ and one workshop in the e-business and WebSphere area. He holds a Bachelor's degree in Engineering.

Lutz Kuehner is a z/OS systems programmer working for IBM Germany. He has 19 years of experience in the mainframe operating systems field. His areas of expertise include systems programming, tools, and processes. He particpated in a previous ITSO residency writing about UNIX-related topics.

Luiz Maia is a systems programmer at Banco do Brasil in Brazil. He has 17 years of experience in large systems and holds a degree in Engineering. His areas of expertise include Parallel Sysplex®, Workload Manager, capacity planning, and performance.

Marc Muntane is a telecommunications engineer working for “La Caixa d’Estalvis i Pensions de Barcelona” in Barcelona, Spain. He has six years of experience in mainframe systems. He holds a degree in telecommunications engineering and a master’s in networks and telecommunications services. His area of expertise is communications environments, and he has written extensively on Communications Server.

Diana Nakajima works in IBM Sales and Distribution in Brazil, where she is a technical sales specialist working with large zSeries customers.

Andre Pradier is a z/OS systems programmer working for Banco do Brasil in Brazil. He has 14 years of experience in large systems. He is finishing post graduate study in Information Systems. His areas of expertise include programming, UNIX System Services, and ISPF.

Anthony Soares is a systems programmer at Banco do Brasil in Brazil. He has two years of experience in the z/OS field and holds a degree in Information Systems. His areas of expertise includes z/OS, capacity planning, and programming.

Claudia Tomas is a systems programmer at Banco do Brasil in Brazil, which operates one of the largest sysplexes in South America. She has 11 years of mainframe experience. Her areas of expertise include Parallel Sysplex, Workload Manager, UNIX System Services, and z/OS, as well as various topics including implementation and performance.

David Welch is a z/OS systems programmer at IBM Global Services in New Zealand. He has more than 20 years of experience working in the mainframe area. He holds a Bachelor's degree in Mathematics.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

z/OS Version 1 Release 7

This chapter describes the changes in the Base Control Program (BCP) and other components that provide essential operating system services. The BCP includes the following:

- I/O configuration program (IOCP)
- The workload manager (WLM)
- System management facilities (SMF)
- z/OS UNIX System Services (z/OS UNIX) kernel
- Support for Unicode
- A migration tool, BPXWH2Z, to migrate HFS to zFS

As of z/OS V1R3 and z/OS.e V1R3, the BCP also includes the program management binder, which was formerly in the DFSMSdfp™ base element.

In z/OS V1R7 and z/OS.e V1R7, the IBM Health Checker for z/OS is now also part of the BCP. This tool helps improve availability by reporting on active z/OS and sysplex settings that are different than IBM recommended best practices or customer-defined settings. It was initially available as a prototype via download for z/OS V1R4 and z/OS.e V1R4.

1.1 z/OS Version 1 Release 7 overview

z/OS and z/OS.e consist of base elements and optional features as follows:

- The base elements (or simply *elements*) deliver essential operating system functions. When you order z/OS or z/OS.e, you receive all of the base elements. However, with z/OS.e, some base elements are not functional or not licensed for use, or both.
- The optional features (or simply *features*) are orderable with z/OS or z/OS.e and provide additional operating system functions. Some optional features that are orderable with z/OS are not orderable with z/OS.e. Optional features are unpriced or priced, as follows:
  - Unpriced features

    Unpriced features are shipped to you only if you order them. If you plan to use any unpriced features, IBM recommends that you order them when you order your base elements. You must not wait until the next release becomes available. Once a release's base elements are no longer orderable, neither are its unpriced features. To make ordering easier, the number of unpriced features is reduced from time to time, mainly through consolidation. With the reductions in V1R5 and V1R6, the number of unpriced features is now two:

    - Communications Server Security Level 3
    - z/OS Security Level 3
  - Priced features

    Priced features are always shipped to you. When IBM packages your order, we enable the priced features that you ordered. These features are ready to use after you install z/OS or z/OS.e (and customize them as needed). We disable the priced features that you did not order. Although they are installed on your system, you cannot use them. Later on, if you decide to use them, you notify IBM and you enable them dynamically (which is known as *dynamic enablement*). You dynamically enable by updating parmlib member IFAPRDxx and you notify IBM by contacting your IBM representative.

1.1.1 z/OS V1R7 enhanced availability features

The z/OS operating system, running on System z9™ and zSeries servers, continues to help address requirements for uninterrupted application availability. In z/OS V1R7, support is provided for the following functions:

- Allowing concurrent activation of service for z/OS UNIX System Services
- Dynamic virtual IP address (DVIPA) reclamation
- Improved console message processing
- Improved recovery for JES2, RACF, FICON®, and Unicode

These improvements can help you provide the kind of 24x7 availability needed by today's on demand business applications.

1.1.2 New z9-109 processor support

z/OS continues to support robust vertical and horizontal growth. z/OS V1R7 leverages the new functions of the z9-109 and is designed to provide improved overall performance, including the following:

- Increased I/O device addressability
- Increased I/O bandwidth

- Improved cryptographic performance when the Crypto Express2 feature is configured for Secure Sockets Layer (SSL) in accelerator mode
- A new time synchronization feature, Server Time Protocol (STP)

1.1.3 System limit extensions

z/OS V1R7 also extends system limits in many areas, including support for the following:

- 32-way single-system images, larger sequential data sets
- Support for additional VSAM data set extents
- More Cross System Extended Services® (XES) locks per lock structure connector
- 64-bit VSAM record-level sharing (RLS) support
- An increased number of DASD-only log streams
- Up to 16,384 DASDONLY log streams in System Logger
- Program Management Binder compression for program objects
- **VARY** command processing improvements designed to reduce serialization contention
- Hardware Configuration Definition (HCD) support for larger I/O configurations
- DFSMSdss virtual storage constraint relief

These improvements can help you support larger workloads for today's on demand business requirements.

1.2 z/OS UNIX System Services (USS)

For many workloads, including on demand and existing Enterprise Applications, many customers require 24x7 availability for all system components, including the z/OS UNIX kernel. Installing PTFs often requires a system interruption, even with dynamic LPA and dynamic link list support. Formerly, an IPL was always necessary to refresh the address space when UNIX kernel maintenance was installed, requiring a scheduled outage.

1.2.1 Dynamic service activation

You can take advantage of the UNIX System Services dynamic service activation capability whether you are implementing new enterprise and WebSphere applications or simply have high availability requirements. UNIX System Services dynamic service activation provides a characteristic not typically found on other UNIX platforms: it is designed to provide continuous availability even when certain maintenance is applied. This is an initial step toward addressing planned system outages.

The primary intention of this change is to significantly reduce the number of these planned outages by providing the capability to dynamically activate SMP/E installable service for the UNIX System Services kernel and LFS components without requiring a component shutdown or re-IPL of a given system.

The activation of service is accomplished via a new **F OMVS,ACTIVATE** command to back off service. Additionally, a **D OMVS,ACTIVATE** console command option is provided to display the current set of dynamically activated service.

1.2.2 ISHELL enhancements

New function has been added to the **ISHELL**, **oedit**, and **obrowse** commands, and UNIX System Services REXX functions.

- The ISHELL improvements are the following:
  - A new command retrieval function
  - The ability to create a new file with specified attributes
  - A new display that can be sorted by GID
  - The addition of a reference list
- OEDIT improvements include a higher maximum width for editing (up to 32752) and warning about potential changes to extended file attributes.
- BPXWDYN is enhanced to add the capability to retrieve DD names, data set names, and path names for current allocations.

1.2.3 Mount command support

Starting with z/OS V1R7, the **SET OMVS=xx** command is changed to execute the ROOT and MOUNT statements contained in a specified parmlib member. It is also extended to absorb the functionality of **SETOMVS RESET=(xx)** by also executing the FILESYSTYPE, SUBFILESYSTYPE, and NETWORK statements that are present. This function enhances the capability to perform UNIX System Services systems operations from the console.

This allows an installation to issue **MOUNT** commands contained in a parmlib member specified with **SET OMVS=** from a console. Previously, neither **SET OMVS=xx** nor **SETOMVS RESET=(xx)** supported the **MOUNT** command, so there was no direct way to execute a list of mounts from the console. This makes OMVS different from most of the other MVS components that support **SET** in that z/OS only supports a subset of the commands that one can use during an IPL.

1.2.4 AF_UNIX display support

A new console display, **Display OMVS,Sockets**, is added that displays information about AF_UNIX sockets similar to the information that is displayed by “netstat” commands for AF_INET sockets. This displays who is using AF_UNIX sockets with the following information:

- Job name
- The socket's path name
- The state of the socket
- The socket's ID

AF_UNIX Sockets are local sockets where both ends of a connected socket session are in the local system; there is no network connectivity involved.

1.2.5 Mounts in progress

A new operand is added to the **DISPLAY OMVS** command. **D OMVS,F** shows mounts in progress. Often, these are mounts awaiting a DFSMShsm recall. Also, **DISPLAY OMVS** shows the owning task for the LFS Mount latch along with a few words about why it obtained the latch and what it is doing at the moment. Often, such a task is waiting for some event, such as for the reply to a cross-system message or for another latch.

1.3 zSeries File System (zFS)

In early 2004, IBM announced that Hierarchical File System (HFS) function is stabilized. The zSeries File System (zFS) is the strategic UNIX Systems Services file system for z/OS. IBM has enhanced zFS function in z/OS V1R7 so that you can use zFS file systems at all levels within the file hierarchy.

RMF™ Monitor III uses the zFS monitoring APIs to provide performance information about the zFS environment. You can use this information to tune the zFS environment by monitoring cache sizes, I/O balancing, and the sizes of zFS aggregates. This can help simplify zFS performance management.

The zSeries File System is a UNIX file system that can be used in addition to the HFS. zFS is the strategic file system for z/OS. Therefore, in z/OS V1R7, zFS file system functions are extended beyond those provided by HFS by improving usability and making it easier to migrate your data from HFS file systems to zFS file systems.

zFS file systems contain files and directories that can be accessed with the z/OS and z/OS.e hierarchical file system file APIs. zFS file systems can be mounted into the z/OS UNIX hierarchy along with other local (or remote) file system types (such as HFS, TFS, AUTOMNT, and NFS).

Note: zFS can be used for the root file system. Because zFS has higher performance characteristics than HFS and is the strategic file system, HFS may no longer be supported in future releases and you will have to migrate the remaining HFS file systems to zFS.

1.3.1 Performance monitoring

A new programming interface is provided in pfsctl (BPX1PCT) to provide statistics that were previously available only from the **MODIFY** command. RMF Monitor III uses the zFS monitoring APIs to provide performance information about the zFS environment. You can use this information to tune the zFS environment by monitoring things such as cache sizes, I/O balancing, and the sizes of zFS aggregates. This can help simplify zFS performance management.

1.3.2 Mount processing

Mount processing has been changed to check the file system type. The HFS and zFS file system types in mount statements and command operands are now generic file system types that can mean either HFS or zFS. When mounting file systems, the system will determine which file system type is appropriate to use. Additionally, an ISPF-based tool will help you create new zFS file systems to replace HFS file systems, copy the data from the HFS file systems to the zFS file systems, and mount the new file systems in place of the old ones. Last, pax processing has been improved to copy sparse files as sparse, create mount points at device boundaries, attempt to continue when there is an error processing a source file or directory, and copy File Format and Audit Flags from source files.

1.3.3 Command forwarding support

Command forwarding support is added for zFS. This allows zFS commands to be issued from any system in a sysplex without regard to which system owns the file system. Also, the character set allowed for zFS naming is extended to include all the characters allowed for HFS naming.

1.3.4 Migration from HFS to zFS

z/OS V1R7 provides a migration utility that is invoked via a REXX exec called BPXWH2Z. This utility is a tool that can help you migrate your HFS file systems to zFS file systems. It can do one at a time or build a list and do many. This must run from ISPF to set up the file system migrations, but the actual migration work can be run in UNIX background as well as foreground.

1.3.5 Unquiesce aggregate support

A new **MODIFY** command is provided to unquiesce a specifically named zFS aggregate. This can be useful when DFSMShsm processing has been interrupted after quiescing a data set, leaving it inaccessible. This support is exclusive to zFS.

1.4 ServerPac enhancements

ServerPac (5751-CS9) has been around since March, 1996. It was invented in support of OS/390 release 1. Over time, significant enhancements have been included in the offering and in the dialog, such as:

- Restructuring the install and documentation (Installing Your Order)
- Adding “Software Upgrade” versus “Full System Replace” installation options
- SMS Construct support
- Merge data set
- Recommended System Layout support
- View/Change facility
- HFS and zFS support
- Electronic Delivery

This release introduces the following enhancements:

- Documented and enforced support for merging with saved configurations
- Merge data set support for file systems
- Dialog and install changes in support of Health Checker
- Other dialog enhancements
- Removal of “outdated” PSP buckets
- Support for z/OS R7 Health Checker line item

ServerPac has been enhanced to allow z/OS UNIX file system data sets to be merged. This expands support of file system data sets to include all the facilities that the CustomPac Dialog provides for other data sets.

Also, PSP buckets for products delivered with the ServerPac or CBPDO were formerly provided with the package. This made sense before you had easy access to up-to-date information, but old PSP information is much less useful than current information. Therefore, the PSP buckets have been removed and documentation added to point to the PSP bucket Web site where you can always get the latest available PSP information.

1.5 Device support in z/OS

IBM intends to continue to support growth by enabling vertical growth within a single image, horizontal growth within a Parallel Sysplex, and direct access device storage growth. IBM continuously analyzes z/OS capability to scale both vertically and horizontally in our efforts to remove constraints to growth. Vertical growth can be accommodated both with increased uniprocessor speed and an increased number of engines per z/OS image. Horizontal growth can be supported by making sure resources shared by systems in a sysplex are not bottlenecked. Storage growth can be supported through the introduction of larger-capacity storage devices and actions intended to reduce the pressure on the 64K device limit.

1.5.1 Storage growth support

In z/OS V1R7, z/OS supports storage growth by:

- Supporting volume sizes of up to 64K cylinders (approximately 54 GB). This allows multiple volumes to be consolidated, reducing the number of device numbers required to support a given amount of storage.
- Supporting sequential (non-extended format) data set sizes of greater than 64K tracks. The new limit on the size of these data sets will be 16,777,215 tracks.
- Both JES2 and JES3 will support spool data sets up to the maximum supported volume size (65,520 cylinders). This allows you to use full-volume spool data sets and exploit the new maximum volume size for spool volumes. A new operand of the DSNTYPE parameter of the JCL DD statement is used to indicate that a data set larger than 64K tracks in size is to be allocated. This will avoid incompatibilities with programs that process Format 1 DSCBs, DEBs, or use previous levels of BSAM NOTE and POINT macros.

1.6 Console restructure

z/OS V1R7 supports console restructuring with the following capabilities:

- Support for deleting unused EMCS consoles
- A new AMRF/ORE service routine (for SDSF and vendors)
- Change in how MONITOR messages are processed so that they are not associated with a console
- Better recovery

1.6.1 1-byte console IDs

z/OS V1R7 is the last release to support 1-byte console IDs. With z/OS V1R7, support for 1-byte console IDs and migration console IDs on external interfaces is removed. The WTO, WTOR and MCSOPER macros now reject attempts to use 1-byte console IDs. Operator commands no longer support the specification of a console ID. A console *name* must be used on those commands. Programs compiled using older versions of the macros will continue to work, but the key interfaces that support 1-byte IDs are being changed now to prepare for future enhancements. In the release following z/OS V1R7, however, all 1-byte console ID support is completely removed.

Note: The console ID tracker introduced in V1R5 will continue in z/OS V1R7 to identify programs that continue to use 1-byte console IDs.

1.7 SMP/E enhancements

SMP/E is a tool for installing and maintaining software, and for managing the inventory of software that has been installed. Prior to z/OS V1R2, SMP/E was an exclusive base element. Beginning with z/OS V1R2, SMP/E is non exclusive because of the introduction of the SMP/E product. The SMP/E product allows customers who are currently licensed for an earlier level of z/OS or z/OS.e to order and install the latest level of SMP/E without having to upgrade their entire operating system. This allows products that run on z/OS or z/OS.e to exploit the packaging and installation enhancements of SMP/E without requiring a later level of the operating system. This also allows customers to exploit new electronic delivery and installation technologies in SMP/E sooner. The SMP/E product is available at no additional charge to customers.

For z/OS V1R7, the new version is SMP/E for z/OS V3R4, 5655-G44.

1.7.1 Automate ordering and delivery of PTFs

With extensions to the **RECEIVE** command, SMP/E can initiate orders for PTFs over the Internet and then download and receive them. They could also be installed in the same job step with the **APPLY** command. This can eliminate manual tasks required for ordering and delivery of IBM PTFs using current methods such as ShopzSeries.

1.7.2 Build processing phase

Improve the load module build processing phase of the **APPLY**, **RESTORE**, and **LINK LMOD** commands by reducing the number of assemblies performed to rebuild a load module or program object. Load module build processing is updated to be more flexible, so distribution libraries are not always required because SMP/E uses modules in the SMPPTS data set if they are available.

1.8 JES2 and SDSF

Prior levels of JES2 can be used with z/OS V1R7 but not with z/OS.e V1R7.

1.8.1 JES2 checkpoint corruption

JES2 is enhanced to detect and correct certain additional kinds of checkpoint control block corruption when JES2 is restarted. This processing occurs with all types of start (including hot start). This new support for detection and recovery for certain kinds of DAS control block corruption adds to prior support for JOE, JIX, and BERT control blocks. This can help prevent cold starts.

1.8.2 SDSF enhancements

SDSF now supports monitoring JES2 system resources, which lets you monitor the same resources described by the $HASP050 JES2 RESOURCE SHORTAGE message. For each resource, SDSF reports the following:

- The total defined number of elements for the resource and the number available for use
- The number of elements currently in use
- The percentage of the total elements currently in use

In addition, SDSF displays the information about JES2 spool volumes that is returned by the **$DSPOOL** command, including total spool utilization and individual spool volume utilization and status, for all members of a Multi-Access Spool cluster (MAS) from any member of the MAS.

These functions help make this information available in one place, can help improve operator and system programmer productivity, and can save JES2 command buffers, command processing CPU time, and SYSLOG space.

1.9 IBM Health Checker for z/OS

Prior to z/OS V1R7 the health checker function was packaged as a prototype (non-SMP/E package) via download. With z/OS V1R7, the Health Checker is a component of the BCP base element, and it has been restructured into a framework with separate checks from the BCP and other elements.

The IBM Health Checker for z/OS and sysplex provides an automated way to check your system's active configuration for conditions that can affect system availability and for values that should be configured for best practices. This tool provides several GRS checks, including a check to see if GRS synchronous reserve processing is enabled.

To help simplify systems management, the z/OS Health Checker is now a base component of z/OS.e, providing an integrated tool for checking on best practices for configuration values. Significantly more checking is provided, and the tool is now easier to use. The dynamic capabilities of z/OS.e are also extended with the new TCP/IP Sysplex Load Balancing Advisor, which provides for better interaction with network-based load balancers and integration between Sysplex Distributor and Workload Manager. In addition, z/OS.e V1R7 simplifies network management with JES2 NJE support for TCP/IP.

1.10 RMF

The following enhancements have been implemented with RMF with z/OS V1R7.

1.10.1 Monitoring zFS UNIX file system activity

RMF Monitor III uses the zFS monitoring APIs to provide performance information about the zFS environment. You can use this information to tune the zFS environment by monitoring things such as cache sizes, I/O balancing, and the sizes of zFS aggregates. This can help simplify zFS performance management. Two new Monitor III reports provide overview and performance information about the zFS activity:

- The zFS Summary Report helps to control the zFS environment and the I/O balancing.
- The zFS Activity Report measures zFS activity on the basis of single file systems, for example, a file system's performance and DASD utilization.

1.10.2 Monitoring storage group and disk space

RMF Monitor III is extended to display the total and free space per DASD and storage group. This information is also exploited by the RMF Common Information Model provider function. These two functions can help you manage your storage resources more easily by providing

this information in additional ways. Two new Monitor III reports support disk space monitoring at the storage group level as well as at the volume level:

- The Storage Space Report displays information about capacity and available space for defined storage groups.
- The Disk Space Report provides capacity and space information for volumes belonging to the defined storage groups.

1.10.3 Support of Crypto Express2 hardware

RMF enhances the Crypto Hardware Activity Report to support the new Crypto Express2 Coprocessor crypto card.

1.10.4 Preferred paths support

For control units supporting path attributes for connected channel paths, RMF enhances the I/O Queuing Activity report, now showing these attributes, if applicable, for CHPIDS connected to these eligible control units.

Note: This functionality is available as SPE since April 2005 and needs to be installed as APAR OA09921 for z/OS releases earlier than V1R7.

1.10.5 Support for IBM System z9 processors

On z900 and z990 processors, special purpose processors IFLs (Integrated Facility for Linux®) and IFAs (Integrated Facility for Applications) are reported as ICFs (Internal Coupling Facility). Starting with IBM System z9 (z9-109) processors, IFLs, IFAs, and ICFs are reported separately in the Postprocessor Partition Data Report and the Monitor III CPC Report.

Also, the reporting about crypto activity is extended. On z9-109 processors, the Integrated Cryptographic Service Facility (ICSF) exploits SHA-256 hashing. RMF enhances the Crypto Hardware Activity Report to provide measurement data for SHA-1 and SHA-256.

1.10.6 RMF eServer zSeries Common Information Model (CIM) monitoring

With z/OS V1.7 base element Common Information Model (CIM), it is possible to use the DMTF CIM open standard for systems management. z/OS CIM implements the CIM server, which is based on the OpenPegasus open source project. A CIM monitoring client invokes the CIM server which, in turn, collects z/OS metrics from the system and returns it to the calling client. To get the z/OS metrics, the CIM server invokes the z/OS RMF monitoring provider, which retrieves the metrics associated with z/OS system resources. The z/OS RMF monitoring provider uses existing and extended RMF Monitor III performance data. The metrics obtained by this new API are common across eServer™ platforms, so you can use it to create end-to-end monitoring applications.

1.10.7 Queue length distribution in the CPU activity report

The presentation of the queue length distribution in the System Address Space Analysis section of the CPU Activity Postprocessor report is changed to reflect the higher number of CPUs available per MVS image.

1.10.8 RMF Monitor III

RMF Monitor III (RMF PM) now allows you to monitor storage and subpool usage for the Master address space. This can help you identify emerging problems more quickly.

In addition, Monitor III will display the amount of unallocated Common Area (CSA and SQA) below the 16MB line. This can help you monitor common storage more easily and spot problems earlier, and satisfies requirement MR031803399 (RMF Monitor III display with the GDACSARE value).

1.11 DFSMS

This section describes the enhancements that are available with DFSMS with z/OS V1R7.

1.11.1 Large format data sets

Before z/OS V1R7, most sequential data sets were limited to 65535 tracks on each volume, although most hardware storage devices supported far more tracks per volume. To support this hardware capability, z/OS V1R7 allows users to create new large format data sets, which are physical sequential data sets with the ability to grow beyond the previous size limit. QSAM, BSAM, and EXCP access methods all support large format data sets, with some limitations for EXCP programs, and for BSAM programs that use the NOTE and POINT macros. Large format data sets reduce the need to use multiple volumes for single data sets, especially very large ones like spool data sets, dumps, logs, and traces. Large format data sets can be either cataloged or uncataloged, SMS-managed or not. Unlike extended-format data sets, which also support greater than 65535 tracks per volume, large format data sets are compatible with EXCP and do not need to be SMS-managed.

1.11.2 Device support address space

z/OS now provides a new device support address space (DEVMAN). It is started during an IPL, and it is cancelable, non-swapable, and restartable with the **START** command. z/OS uses the new address space to:

- Capture CTRACE information for CVAF events for first failure data capture
- Capture CTRACE information for DADSM Create events for first failure data capture

1.11.3 DFSMS subchannel set support

Parallel access volume (PAV) users can define a second set of subchannels to be used with PAV to redefine aliases into the second subchannel set. This frees up device numbers on the first subchannel set for use as additional base devices. PAV users can increase the number of available devices. Device numbers can be duplicated in the same channel subsystem by being in both subchannel sets. PAV alias devices must be defined to an alternative subchannel set using HCD or HCM.

1.11.4 REPRO MERGECAT FromKey/ToKey enhancement

This enhancement enables you to repair damaged catalogs by targeting a range of catalog keys in a REPRO MERGECAT command. This ability is provided through the addition of two new keywords for use with the MERGECAT parameter of the REPRO command. Using the MERGECAT FROMKEY and TOKEY parameters, you can copy unbroken segments of a

damaged catalog to a new catalog, defining both a starting point and an ending point in the range to be copied. This makes it easier to maintain catalogs and recover from problems.

1.11.5 Catalog enhancements

You can specify the amount of space to be used to allocate a VSAM volume data set (VVDS) when the allocation is done implicitly. The default value for an implicit VVDS allocation is TRK(10,10), which may not be adequate for the size of your volumes and number of data sets you are creating. Now, using the implicit VVDS space quantity enhancement, you can specify the amount of space in tracks that you want to use when a VVDS is defined implicitly.

1.11.6 VSAM extent constraint removal

In previous releases, there was a 255-extent limit for VSAM data sets and, for striped VSAM data sets, a limit of 255 extents per stripe. Now, using the VSAM extent constraint removal enhancement, these extent limits are removed for SMS-managed volumes if the extent constraint removal parameter in the data class is set to Y (yes). For non SMS-managed volumes, the previous extent limits still apply. A VSAM data set can be expanded to 123 extents per volume. This is unchanged from previous releases.

1.11.7 VSAM RLS 64-bit data buffers

VSAM record-level sharing (VSAM RLS) is an extension to VSAM that provides direct record-level sharing of VSAM data sets (as opposed to CI-level sharing) from multiple address spaces across multiple systems. VSAM RLS uses the z/OS coupling facility for cross-system locking, local buffer invalidation, and cross-system data caching. Primary users of VSAM RLS include high-volume applications that access VSAM data sets, such as CICS® applications. Now, with z/OS V1R7, you can optionally specify that VSAM RLS uses 64-bit addressable virtual storage for data buffers. Doing so can help you avoid possible buffer space constraints and potentially improve performance for your high-transaction applications.

1.11.8 SMS volume and ACS allocation test enhancements

SMS volume enhancements for this release include volume status change and volume selection messages and traces. There is also the addition of the ACS allocation test environment.

In addition to the summarized analysis messages issued by SMS when volume selection fails for an SMS-managed data set, in z/OS Release V1R7 SMS provides more information to help you analyze the reasons why selection might have failed. The summary analysis messages provide diagnostic information about the number of volumes rejected for each reason during volume selection. However, there are many cases in which you need more detailed failure information to determine why volume selection has failed, or messages about successful allocations to determine why a data set was not allocated. This enhancement provides the following functions:

- Summarized and detailed analysis messages on request
- DADSM failure reasons and diagnostic codes in summarized analysis messages
- Adding volume selection data to SMS trace data
- Adding new trace data for SMS and non-SMS managed VSAM allocations with more-complete information

1.11.9 Object Access Method (OAM) enhancements

A new TAPEDISPATCHERDELAY keyword is added to the SETOAM statement in the CBROAMxx parmlib member to delay processing of certain requests and minimize demounting and remounting.

The MOVEVOL utility is enhanced to accommodate OAM scratch volumes. This allows MOVEVOL to be used with the DELETE option to remove scratch volumes from OAM's inventory.

A new dynamic exit name, CBRUXTVS_EXIT, is invoked to inform the tape management system when a tape volume has been purged from the OAM inventory. This notification exit is patterned after the DFSMSrmm EDGTVEXT exit and the DFSMShsm ARCTVEXT exit.

When an object is moved from an optical or tape volume to DB2® DASD during an OSMC cycle, the volser and sector location, or blockid, is retained in the object directory. This is because the transition to DB2 is usually temporary and the object will later move back to optical or tape. While an optical or tape volser is associated with an object, that volser cannot be expired, even if the object is currently residing in DB2. V1R7 adds a new CLEAROLDLOC keyword to instruct OAM to clear the original volser and sector location or blockid in the object directory for a given object when that object is moved by OSMC to DB2 DASD. This new keyword will be most useful to installations that do not normally transition objects back to tape or optical volumes once they have moved to DB2 DASD.

This release supports the immediate recall of objects, currently residing on removable media, to DB2, for a specified number of days. This allows subsequent requests to read the objects to be satisfied from DB2 DASD rather than another read from the tape or optical volume where the object resides. OSMC will restore the object to its original location after the specified number of days have passed.

1.11.10 DFSMSrmm enterprise enablement

In the past, you could use the high-level language application programming interface from C/C++ and Java™ (using the JNI) code running on the same z/OS system as the DFSMSrmm subsystem. A series of calls to the application programming interface were necessary to run the subcommand and to receive the output. Now, using the DFSMSrmm enterprise enablement enhancement, you can use the high-level language application programming interface as a web service. This enables the high-level language application programming interface to be used from any system or platform that can run Java, C++, or any language that supports the web services standards. Now, it is as if the high-level language application programming interface is available as a locally callable program. A single call to the application programming interface to run a subcommand and receive all the output is all that is needed.

In addition, a plug-in adapter created for the SNIA CIM environment supports removable media. This Java class maps DFSMSrmm resources into those defined in the CIM object model. This plug-in adapter uses the CIM provider interface to provide real-time information about storage resources.

1.11.11 DFSMShsm enhancements

For extended tape table of contents (TTOCs), to make better use of new high capacity tape volumes, DFSMShsm can write more than one million data sets to a migration tape or backup tape. In previous releases, this number was limited to 330000 data sets.

To simplify recycle criteria for connected sets, this release allows connected sets to be recycled sooner, and free up more volumes to scratch or tape pools, DFSMShsm allows you to specify that the entire connected set's average percentage of valid data is to be used to determine whether to recycle a connected set. In previous releases, DFSMShsm required the first volume in the connected set to meet the percent valid criterion before determining the connected set's average percentage of valid data.

The INCLUDE statement is not required for ABARS processing. With this change, ABARS processing no longer requires you to specify an INCLUDE statement in the data set selection list; you need only specify allocation information (an ALLOCATE statement) or tape catalog information (an ACCOMPANY statement).

The LRECL and DS empty indicator can now be queried. For easier analysis of migrated data sets, you can determine the logical record length (LRECL) of a data set that has been migrated, and whether the data set is empty, through DFSMShsm commands (the LRECL and an 'empty data set' flag are now collected in the MCD record for the data set in the MCDS). In previous releases, you needed to recall a migrated data set to learn this information. You can also use the IDCAMS DCOLLECT function to display this information when specifying the MIGRATEDDATA option.

For easier data set migrations, DFSMShsm bypasses already-migrated data sets when you specify the **HMIGRATE** command with a wildcard filter (*). In previous releases, HMIGRATE processing resulted in numerous error messages if your datasetname specification included migrated data sets.

To increase the number of migrated data sets eligible for fast subsequent migration, this function has been changed to use new indicators for reconnection eligibility. This more robust indication of reconnection eligibility can increase your use of fast subsequent migration in two ways. First, the new indication allows you to use fast subsequent migration even if you use a product other than DFSMShsm to back up your data sets. Second, it allows DFSMShsm to reconnect data sets originally migrated to ML2 tape without a valid backup copy (such Tape Mount Management data). In previous releases, these data sets were not eligible for reconnection.

A new command, **V SMS,VOLUME** is provided to alter a volume's status without having to change and reactivate the SMS configuration using ISMF. For example, you can use this command to change the status of a volume from NOTCON (not connected) to another status (like QUIESCE or DISNEW). This function helps improve SMS's ease of use by making it easier to make temporary changes to the status of volumes. In turn, this helps make it easier to manage your storage, which can help keep ownership costs lower.

1.12 Communication Server

There are many major changes in Communication Server in z/OS V1R7; they are described in this section.

1.12.1 Sysplex Distributor

Sysplex Distributor is now designed to distribute incoming traffic to target stacks in a sysplex over the optimal available IP route. This removes the restriction that the Sysplex Distributor must use only dynamic XCF interfaces for packet forwarding, and can allow the use of high-speed interfaces such as OSA Express Gigabit Ethernet.

Sysplex Distributor will also use new metrics to determine how efficiently a target server is processing connections from a TCP/IP perspective. For example, it will now monitor key TCP

server performance statistics, such as size of the TCP connection backlog queue. These metrics will be used to identify servers that may be experiencing performance problems, and will supplement the existing WLM and QoS recommendations to strengthen the overall load balancing decision.

WLM support

Starting with z/OS V1R7, WLM and the Communications Server IP Sysplex Distributor are designed to work together to improve the direction of TCP/IP traffic in a sysplex. WLM now provides a TCP/IP server-specific recommendation to Sysplex Distributor that reflects how well each target server is meeting its WLM service class goals. Server-specific recommendations offer better granularity than previous WLM recommendations based solely upon a system's displaceable capacity. This is intended to improve Sysplex Distributor's workload balancing. Similarly, DB2 can utilize this function if not using DB2 Connect™ sysplex support to route to systems based on how each candidate target server is meeting its goals.

This function provides a new exit for OAM that communicates tape volume status to a tape management program (like RMM). Customers who implement this exit will be able to manage their scratch tape inventory more efficiently and potentially reduce the requirements for scratch tapes.

1.12.2 TCP/IP automatic takeover

Expanding on the TCP/IP Automatic Takeover function introduced in z/OS Release 6, TCP/IP is now designed to be able to automatically rejoin a sysplex when the problems that triggered the takeover have been relieved. When a TCP/IP stack rejoins the sysplex it will automatically restore its original configuration, allowing it to take back ownership of any DVIPAs for which it is designated as the primary owner.

1.12.3 Encryption support

Communications Server will support the Advanced Encryption Standard (AES) using Transport Layer Security (TLS). TLS can be used with TN3270 server and FTP to secure communications. For more information about AES, refer to RFC 3268.

1.12.4 OSA-Express2 support

z/OS Communications Server plans to exploit OSA-Express2 large send (also referred to as TCP segmentation offload). Large send can improve performance by offloading outbound TCP segmentation processing from the host to OSA-Express2 by employing a more efficient memory transfer into OSA-Express2. The z/OS Communications Server support is planned for TCP/IP IPv4 traffic only.

1.12.5 Communications Server FTP enhancements

FTP Client API Support for C/C++ is provided. New C/C++ header files and functions are provided that can allow C/C++ applications to use the FTP Client API.

FTP confidence of success level reporting support is planned for z/OS V1R7. It is intended to help improve the reliability of FTP file transfers performed by the z/OS FTP client and server. For certain types of transfers, the z/OS FTP client and server can be configured to perform additional checks and report a level of confidence that transfers have completed successfully. This is designed to provide an additional safeguard against data loss by including checks not provided for in the FTP protocol.

1.12.6 SNA enhancements

For autologon improvements, a new VTAM® operator command is introduced to initiate autologon sessions for all LUs and their controlling applications. Additionally, enhancements to the **`Display Autologon`** command allow operators to first determine which controlling applications and LUs no longer have sessions. These new operational enhancements provide a VTAM operator with an efficient way to determine autologon LUs that have lost sessions with their controlling application and to initiate logons for the application.

1.12.7 XCF connectivity for TCP/IP communications within a sysplex

Users can now initiate static or dynamic XCF communication between TCP/IP stacks on different pure subarea VTAM nodes within a sysplex. This allows users to utilize the full range of TCP/IP sysplex functions without having to redefine the SNA network to use APPN communications.

Users can also initiate Dynamic XCF communications between TCP/IP stacks on APPN nodes within a Sysplex without also initiating APPN XCF communications automatically.

A new **`MODIFY GR`** operator command provides customers with a mechanism to delete the generic resource representation allowing a USERVAR of the same name to be created.

1.13 System Logger

System Logger is improved by the support described in the next two sections.

1.13.1 XRC+ support

System Logger provides new support for XRC+ by allowing you to choose asynchronous writes to staging data sets for logstreams by using a new logstream attribute and indicating these staging data sets can be used for log data recovery by using a new IPL parameter. Previously, all writes had to be synchronous; this limited the throughput for high-volume logging applications including CICS and IMS™. The ability to do asynchronous writes can allow the use of XRC for some applications for which it was not previously practical.

1.13.2 Deleting logstreams

System Logger now provides support for forcing a logstream connection from logger and forcing deletion of a logstream definition from the LOGR CDS. For example, you can now delete a logstream that has outstanding failed-persistent connections. This can help minimize outages by allowing logger to remove a damaged logstream from its inventory. Also, it offers you a choice between removing resources and remapping to new definitions.

1.14 Resource Recovery Services (RRS)

In z/OS V1R7, Resource Recovery Services (RRS) provides more flexibility for the naming UR selection panel profile data set in the RRS ISPF application. You can now specify your own high-level qualifier in place of the RRS default, your TSO/E prefix (usually, your user ID). For example, instead of the RRS default name of USERID.ATR.PROFILE, you could specify MYRRSHLQ as a high-level qualifier and RRS will allocate MYRRSHLQ.ATR.PROFILE. This satisfies requirement MR0127032216.

1.15 Language Environment

Language Environment® supports specification of run-time options in a file specified via DD statement. This makes it possible to specify Language Environment run-time options while avoiding the 100-character limit for the JCL PARM field. This change also provides a way to specify run-time options for IMS transactions when Library Routine Retention (LRR) is not used.

A new parmlib member, CEEPRMxx, can be used to specify Language Environment run-time options for the system. Operator commands are also provided that allow you to query and update the active run-time options for the system. This can simplify the management of Language Environment options, particularly in multisystem environments, and makes it possible to move Language Environment customization out of assembler language modules maintained using SMP/E usermods.

Several enhancements are being made to the C/C++ compiler, Program Management Binder, Loader, Language Environment Run-Time and class libraries and the dbx debugger. Some changes in this area support enterprise applications, such as the SAP Application Server and Lotus® Domino. Other changes have been made to improve performance and productivity. For example, new architectures are exploited (like the Common Debug Architecture, CDA), which result in a more robust debugging environment and are expected to help improve application developers' productivity.

1.15.1 C compiler and Language Environment run-time

The C compiler and Language Environment run-time are now designed to meet the ISO C99 standard. This provides full ISO/ANSI compliance for application portability. Debugging support in dbx has also been provided. These functions provide IEEE and HFP real and complex math functions, enhancements to the printf and scanf family of functions, enhancements to numeric conversion functions, and xlc utility support.

1.15.2 Preinitialized environments for authorized programs

This is a new facility that supports C/C++ programs executing in supervisor state, PSW keys 0-7, TCB, SRB, or cross-memory mode. The new facility is available as follows:

- It is in the 64-bit version of Language Environment.
- Preinitialized environments for authorized programs can be used by zSeries authorized components or middleware products that need a way to include C/C++ code in their component or product.
- Authorized Language Environment uses the Language Environment "Preinitialized Environments" programming model.
- Preinitialized environments for authorized programs are created, initialized, and terminated, asynchronous to the execution of the Language Environment-conforming C or C++ code.
- It is intended to replace the use of Language Environment's preinitialization facility (CELQPIPI) by APF-authorized components (or products).
- It provides a Language Environment facility that officially supports the execution of Language Environment-enabled C/C++ code in an authorized environment.

1.15.3 Language Environment parmlib member

A new CEEPRMxx parmlib member is added for Language Environment. It can be used to specify Language Environment run-time options for the system. Operator commands are also provided to allow you to query and update the active run-time options for the system. This simplifies the management of Language Environment options, particularly in multisystem environments, and makes it possible to move Language Environment customization out of assembler language modules maintained using SMP/E usermods. However, specifying Language Environment options using CEEDOPT, CEECOPT, and CELQDOPT modules also continue to be supported.

1.16 Enterprise Workload Manager

IBM continues to invest in the self-managing capabilities of z/OS, as well as introducing new support of Virtualization Engine™. In the Release 7 time frame, it is planned that z/OS will provide managed node support for Enterprise Workload Manager, providing part of the infrastructure needed to achieve enterprise-wide performance management.

1.16.1 WLM services

WLM services are added to allow z/OS to contribute to the Developer's Edition and, later, to the Management Edition of the Enterprise Workload Manager. The Developer's Edition of the Enterprise Workload Manager provides services for application instrumentation using the standard group's Application Response Measurement (ARM) standard API. ARM instrumentation in middleware products will become the basis for response time reporting; later, they will also form the basis for resource data collection, address space management, and individual work requests. Furthermore, the ARM services are designed to allow applications that participate in Enterprise Workload Manager to invoke the C-API directly.

At the same time, WLM can invoke the new services on behalf of applications that are already exploiting WLM for transaction management when such applications indicate that they want to participate in Enterprise Workload Manager.

This work represents porting and embedding the ARM reference implementation on z/OS and providing the infrastructure for the Enterprise Workload Manager Agent code on z/OS. New services are provided to inform WLM about individual transaction start and end from an application perspective rather than from the operating system perspective.

1.16.2 Performance measurements

Server-independent, end-to-end performance and workload management are enhanced to provide performance measurements. End-to-end performance measurement information is available in one place so bottlenecks in any tier can be easily identified. Infrastructure for meeting the future objective of self-managing servers is addressed and interoperability among the servers is automated.

1.17 IPCS and system dumps

The interactive problem control system (IPCS) is a tool provided in the MVS system to aid in diagnosing software failures. IPCS provides formatting and analysis support for dumps and traces produced by MVS, other program products, and applications that run on MVS, and it has the following enhancements in z/OS V1R7.

1.17.1 Problem analysis performance

IPCS offers improved performance for contention analysis and WHERE processing. You can now use IPCS options to specify that some parts of the analysis that are not necessary for your immediate needs can be bypassed. This can help speed problem diagnosis.

1.17.2 IPCS select service

The default for the IPCS Select service from is changed from “CURRENT and ERROR” to just “CURRENT.” Having ERROR as a default is more than offset by the additional overhead of processing additional address spaces and the confusion added by presenting address spaces that might be unrelated to the problem at hand. When the system detects errors, there is a very good correlation between the current address space and the error detected. Also, when dumps are initiated via operator action, it is rarely the case that the ERROR criterion will point to the problem.

1.17.3 COPYDUMP command

The **COPYDUMP** command now preserves the prioritized order with which dump data has been captured by SADMP. This can help improve IPCS performance for copied dumps.

1.17.4 IPCS-based ISPF commands

IPCS provides capabilities similar to those in the ISPF editor for browsing reports. This helps to avoid having to route output to a data set to sort and filter it. The new ISPF-based commands are: **EXCLUDE ALL**, **FIND ALL**, **DELETE ALL**, and **EXCLUDE**. Also, a new **REPORT** command allows you to view IPCS reports using ISPF Edit, View, and Browse, in addition to providing other new report processing functions. This can help speed problem diagnosis.

1.17.5 SDUMP enhancements

Zero-suppression records are created for first-referenced pages. When a dump is formatted, storage areas included in the dump are now represented even when they contain only zeroes. This is intended to make it easier to determine which storage areas in a dump are valid.

System trace data is captured earlier when a suspended (but enabled) summary dump request is made while the caller holds locks. This is expected to be particularly helpful for capturing earlier trace data for DB2 SDUMP requests.

2

Migration to z/OS V1R7

This chapter describes what you must do to migrate from any of the three releases that are supported for direct migration to z/OS V1R7:

- z/OS V1R6
- z/OS V1R5
- z/OS V1R4

If you want to migrate to z/OS V1R7 from any other release, contact your IBM representative to find out what alternatives are available.

This chapter includes the following topics:

- Coexistence and migration
- Functions withdrawn in z/OS V1R7
- Functions to be withdrawn in a future release
- Migrations using z9-109 processors

2.1 Coexistence and migration

Coexistence and fallback to previous releases play an important part in planning for migration to the latest release. This section explains what coexistence and fallback are, describes IBM's policy regarding the releases that are supported for coexistence and fallback (as well as migration), and states which specific releases are supported.

While coexistence and fallback might at first seem unrelated, they are very much related in that both deal with an earlier level of a system being able to tolerate changes made by a later level.

2.1.1 Understanding coexistence

Coexistence occurs when two or more systems at different software levels share resources. The resources could be shared at the same time by different systems in a multisystem configuration, or they could be shared over a period of time by the same system in a single-system configuration. Examples of coexistence are as follows:

- Two different JES releases sharing a spool
- Two different service levels of DFSMSdfp sharing catalogs
- Multiple levels of SMP/E processing SYSMODs packaged to exploit the latest enhancements
- An older level of the system using the updated system control files of a newer level (even if new function has been exploited in the newer level)

Releases coexisting with z/OS V1R7

z/OS V1R7 and z/OS.e V1R7 systems can coexist with specific prior releases of z/OS and z/OS.e systems. This is important because it gives you flexibility to migrate systems in a multisystem configuration to z/OS V1R7 or z/OS.e V1R7 using rolling IPLs rather than requiring a systems-wide IPL. The way in which you make it possible for earlier-level systems to coexist with z/OS V1R7 or z/OS.e V1R7 is to install coexistence service (PTFs) on the earlier-level systems. You should complete the migration of all earlier-level coexisting systems to z/OS V1R7 or z/OS.e V1R7 as soon as you can. Keep in mind that the objective of coexistence PTFs is to allow existing functions to continue to be used on the earlier-level systems when run in a mixed environment that contains later-level systems. Coexistence PTFs are not aimed at allowing new functions provided in later releases to work on earlier-level systems. Figure 2-1 shows the coexistence releases with z/OS V1R7.

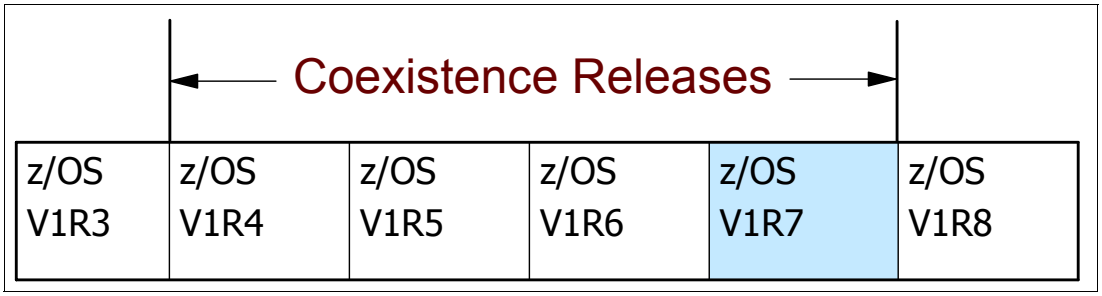


Figure 2-1 Coexistence releases

2.1.2 PTFs for coexistence with z/OS V1R7

PTFs must be applied to the coexistence release shown in Figure 2-1. These PTFs are for the following components:

- BCP – Console, Logger, IODF support
- DFSMS – dfp, dss, hsm, OAM, RLS, VSAM support
- Distributed File Service - zFS and aggregates support
- HCD – IODF support
- JES2 – for MAS support with lower levels
- JES3 – for multisystem complex support with lower levels

Component coexistence PTFs

The following components require coexistence PTFs on a z/OS V1R6 system:

BCP A z/OS V1R7 system cannot join a sysplex of lower-level systems, and a lower-level system cannot join a sysplex of z/OS V1R7 systems, without this PTF. PTF UA16036

BCP z/OS V1R7 introduces the logger log stream DUPLEXMODE(DRXRC) specification. If DUPLEXMODE(DRXRC) is specified in a z/OS V1R7 system in a sysplex, and prior releases are running in the same sysplex, the PTF must be installed. If DUPLEXMODE(DRXRC) is not specified, installing the PTF is not required, but is nevertheless recommended to avoid any complications if the configuration changes. PTF UA17179

BCP Allows a pre-z/OS V1R7 system to use an IODF that was created using z/OS V1R7 HCD. PTF UA17028

BCP Increased CPU utilization in the console address space might be reported on all systems in a sysplex after introducing a z/OS V1R4 system with the z/OS V1R4 z990 Exploitation Support feature, or later-level system, into the sysplex. The coexistence PTF prevents heavy message traffic on the z/OS V1R7 system from impacting the other systems in the sysplex. PTF UA16510

DFSMSdfp An enhancement in the z/OS V1R7 Object Access Method (OAM) component introduces a new valid value (R) for the object location in the object directory to indicate that the object resides on DB2 disk in recalled status. This PTF enables pre-z/OS V1R7 OAMs to coexist in an OAMplex with OAMs at the z/OS V1R7 level. PTF UA16996

DFSMSdfp As part of VSAM extent constraint removal in z/OS V1R7, these PTFs prevent pre-z/OS V1R7 systems from opening data sets that have more than 255 extents. Also, if a data set is open and a z/OS V1R7 job extends it past 255 extents, the close on the pre-z/OS V1R7 system fails. Finally, these PTFs prevent conversion of a data set having more than 255 extents from being converted by DFSMSdss to a non-SMS managed data set on pre-z/OS V1R7 releases. PTFs UA14884, UA16225, and UA16798

DFSMSdfp z/OS V1R7 supports new QNAME and RNAME parameters for VSAM RLS. The coexistence PTFs prevent serialization problems from occurring when the new parameters are used in a GRS complex that includes a pre-z/OS V1R7 system. PTFs UA16635, UA16924, and a PTF for APAR OA09233

DFSMSdfp z/OS V1R7 supports large format sequential data sets. These PTFs cause pre-z/OS V1R7 systems to issue ABEND 213-14 or 213-16 if a program attempts to open data sets that cannot work on those systems. PTFs UA15871 and UA16552

DFSMSdss z/OS V1R7 supports large format sequential data sets. These PTFs cause pre-z/OS V1R7 systems to issue ABEND 213-14 for any large format dump data sets that are provided as input to RESTORE or COPYDUMP, or output for DUMP and COPYDUMP. If DFSMSdss encounters any large format sequential data sets as input to logical data set COPY, logical and physical data set DUMP and RESTORE, data set print, or logical and physical data set RELEASE, the data set will fail with ADR878E rsn 17. PTF UA16798

DFSMShsm z/OS V1R7 supports large format sequential data sets. This PTF causes pre-z/OS V1R7 systems that recall, recover, or ARECOVER a data set to first verify that the data set is not a large format data set. If it is, an error message is issued describing the reason for the failure. PTFs UA16798 and UA16953

DFSMShsm z/OS V1R7 supports large format sequential data sets. This PTF causes a pre-z/OS V1R7 DFSMShsm host that attempts to open a journal in large sequential data set format to fail. The start-up of DFSMShsm continues but journaling is disabled for the failing host, message ARC0025E is issued, and DFSMShsm is placed in emergency mode. If a journal data set is not defined to DFSMShsm at startup via the JOURNAL DD statement, then an attempt to open the journal is not made and ARC0025E is not issued.

All DFSMShsm hosts within an HSMplex must be at z/OS V1R7 before you can migrate the journal to a large format sequential data set. PTFs UA15871, UA16552, andUA16956

DFSMShsm z/OS V1R7 supports fast subsequent migration (FSM) of data sets. These PTFs cause pre-z/OS V1R7 systems that might potentially open the data sets to check or set the appropriate flags. PTFs UA16851, UA16859, UA16862, and UA16946

DFSMShsm z/OS V1R7 increases the number of data set names supported in a tape table of contents (TTOC). If a system that has the coexistence PTF installed finds that the OCDS RECORDSIZE is 6144 bytes, which indicates that some instances of DFSMShsm in the HSMplex are going to use the extended TTOC, then tape operations on that level of DFSMShsm are inhibited. PTF UA16949

DFS™ Allows pre-z/OS V1R7 systems to tolerate new characters (@#$) allowed in z/OS V1R7 for zFS file systems and aggregates. PTF UA14530

HCD Provides support for HCD users who share IODF data sets between pre-z/OS V1R7 systems and z/OS V1R7 systems. PTF for APAR OA07875

JES2 Allows a back-level JES2 to run with z/OS V1R7 JES2 in a multi-access spool (MAS). Apply the PTFs that are appropriate to the JES2 level that you are using. For z/OS V1R5-V1R6 JES2: PTF UA09501, PTF for APAR OA08145; For z/OS V1R4 JES2: PTFs UA03828 and UA09500

JES3 Allows a back-level JES3 to run with z/OS V1R7 JES3 in a multisystem complex. Apply the PTFs that are appropriate to the JES3 level that you are using. For z/OS V1R4 JES3: PTFs UA04010 and UA07715

Note: For z/OS V1R4 and z/OS V1R5 coexistence PTFs, see *z/OS Migration*, GA22-7499.

2.2 Service policy

The IBM current policy is to provide maintenance (service) for each release of z/OS and z/OS.e for three years following its general availability (GA) date. However, service on the last release of a version might be extended beyond the intended three-year period. Prior to

withdrawing service for any version or release of z/OS or z/OS.e, IBM intends to provide at least 12 months notice. For end-of-service (EOS) dates, see Figure 2-2. Planned EOS dates are based on the three-year service policy.

Release	General Availability	Service Expiration
OS/390 V2R10	29 September 2000	30 September 2004
z/OS V1R1	30 March 2001	31 March 2004
z/OS V1R2	26 October 2001	31 October 2004
z/OS V1R3 and z/OS.e V1R3	29 March 2002	31 March 2005 (announced)
z/OS V1R4 and z/OS.e V1R4	27 September 2002	31 March 2007 (announced) 18 months longer than normal 3-year service period.
z/OS V1R5 and z/OS.e V1R5	26 March 2004	31 March 2007 (planned)
z/OS V1R6 and z/OS.e V1R6	25 September 2004	September 2007 (planned)
z/OS V1R7 and z/OS.e V1R7	30 September 2005	September 2008 (planned)

Figure 2-2 IBM service policy

2.3 DASD storage requirements for z/OS V1R7

If you are migrating to z/OS V1R7 from a very old operating system release, or if you will have a different product set than your previous release, you will see increased need for DASD space. The amount depends on what levels of products you are running. The DASD required for your z/OS system includes *all* elements, *all* features that support dynamic enablement, regardless of your order, and *all* unpriced features that you ordered. This storage is in addition to the storage required by other products you might have installed.

Note: All sizes include 15% free space to accommodate the installation of maintenance.

The total storage required for z/OS data sets is listed in the space table in the z/OS Program Directory and *z/OS and z/OS.e Planning for Installation*, GA22-7504.

2.3.1 z/OS V1R7 DASD space

For z/OS V1R7, the total storage required for all the target data sets is 5225 cylinders on a 3390 device, as shown in Figure 2-3 on page 26. The total storage required for all the distribution data sets listed in the space table is 7286 cylinders on a 3390 device. The total HFS storage is 2,800 cylinders on a 3390 device for the ROOT HFS and 50 cylinders for the /etc HFS. The total storage required for the SMP/E SMPLTS is 0 3390 cylinders (there are no load modules in z/OS V1R7 that are both cross-zone and use CALLLIBs, thus the SMPLTS is not needed for permanent storage).

	z/OSV1R6	z/OS V1R7
Target	5277	5225
DLIB	7338	7286
HFS	2800	2800

** Sizes in 3390 cylinders

Figure 2-3 DASD space requirements for z/OS V1R7

2.4 Supported architecture modes

The only processors that support z/Architecture™ are the zSeries server processors, z800, z890, z900,z990, and the IBM System z9 platform processor z9-109.

Beginning with z/OS V1R6, the G5/G6 and Multiprise® 3000 processors are no longer supported by the z/OS operating system, as shown in Figure 2-4.

zSeries Servers/z9-109

G5/G6

Multiprise 3000

OS/390 V2R10	ESA/390	ESA/390 or z/Architecture
z/OS V1R1	ESA/390	z/Architecture
z/OS V1R2 – V1R4	ESA/390	z/Architecture
z/OS V1R2 – V1R4**	ESA/390	ESA/390 or z/Architecture
z/OS V1R5	ESA/390	z/Architecture
z/OS V1R6, **z/OS V1R7**	not supported	z/Architecture

****Using z/OS Bimodal Migration Accommodation within terms of offering**

Figure 2-4 Processors that support ESA/390 and z/Architecture with operating system releases

Note: z/OS V1R2, V1R3, and V1R4 without the z/OS V1R2/3/4 Bimodal Migration Accommodation installed or after the six-month term has expired.

** z/OS V1R2, V1R3, V1R4 with the z/OS V1R2/3/4 Bimodal Migration Accommodation installed and within the six-month term is supported.

2.5 Ordering Tivoli NetView and System Automation

Parts of two stand-alone products are included in msys for Operations: Tivoli® NetView® for OS/390 V1R4 (5697-B82) and System Automation for z/OS V2R3 (5645-006). If you already have these stand-alone products installed (at the V1R4 and V2R3 levels, respectively), you can install z/OS V1R7 (including msys for Operations) in the same SMP/E zone as the stand-alone products. In this case, it is recommended that you order these stand-alone products in your z/OS ServerPac. They will be installed in the same zones as z/OS, and will not require separate maintenance and duplication of service work (which they would if they were in separate zones).

Older versions of products

However, if you have an earlier level of either stand-alone product installed, you have to put the stand-alone product into a separate zone before installing z/OS V1R7, and maintain its data sets with different names than the z/OS V1R7 msys for Operations data sets. (Use BUILDMCS to move the stand-alone products or else you will have to reinstall them.) Older levels of Tivoli NetView and System Automation than what is included in z/OS V1R7 cannot be ordered with a z/OS ServerPac.

Full-function NetView

If you plan on moving from z/OS V1R7 msys for Operations NetView to a full-function NetView V1R4, there is a sample job to assist you. This sample job will enlarge the msys for Operations data sets to accommodate the extra space needed for a NetView V1R4 installation. For details, see *Tivoli OS/390 Installation: Migration Guide Version 1 Release 4*, SC31-8768.

Newer versions of products

You may order these products to use with z/OS V1R7, as follows:

- Tivoli NetView V5R1 (5697-ENV) or V1R4 (5697-B82) with z/OS V1R7; it is compatible with msys for Operations and will be installed in the z/OS SMP/E zones.
- System Automation V2R3 (5645-006) with z/OS V1R7; it is compatible with msys for Operations and will be installed in the z/OS SMP/E zones.

2.6 Functions withdrawn in z/OS V1R7

The items shown in Figure 2-5 on page 28 are withdrawn in z/OS V1R7. These items were last shipped in z/OS V1R6. You should take this into account as you plan your migration to z/OS V1R7. The removal of these functions may require migration actions that you can perform now, in preparation for z/OS V1R7.

One-byte console IDs on macro interfaces and operator commands and (TRACK, ... commands) (from BCP)	Base element – Use console names instead of 1-byte IDs. Use Console ID Tracking facility to identify one-byte IDs usage, supplied as of z/OS V1R4 Console Enhancement feature.	z/OS V1R7
ECMB=NO circumvention in IEAOPTxx (from BCP)	Base element – ECMB must be used, as the bypass will be removed	z/OS V1R7
JES2 "compatibility" R4 mode (from JES2)	Base element – to avoid a cold start, you must $ACTIVATE to convert the JES2 checkpoint to z2 mode before installing z/OS R7	z/OS V1R7
JOBCAT and STEPCAT facilities (from DFSMSdfp)	Base element – any remaining JCL that use JOBCAT and STEPCAT must change	z/OS V1R7
Support for ISAM data sets (from DFSMS)	Base element – ISAM Compatibility interface will still be provided (which allows you to run an ISAM program against a VSAM KSDS data set)	z/OS V1R7
OS/390 R10 level of the C/C++ compilers (from C/C++)	Priced feature - move to the ISO 1998 Standard level of the compilers (introduced in z/OS R2)	z/OS V1R7
z/OS Optional Source media feature	Not an element, a z/OS orderable feature	z/OS V1R7

Figure 2-5 Functions withdrawn in z/OS V1R7

2.6.1 One-byte console IDs

One-byte console IDs are intended to be removed from macro interfaces and operator commands in z/OS V1R7. You will not be allowed to specify one-byte console IDs on macros (such as WTO/WTOR), or on such operator commands as **`D C,CN=`** or **`D PFK,CN=`** . You should use console names instead. A service called the Console ID Tracking Facility is available to help you identify one-byte ID usage. The use of console names is already a best practice on OS/390 and z/OS. For information about Console ID Tracking facility, see *z/OS MVS Planning: Operations*, SA22-7601. In addition, the following commands are removed from the system:

- **`TRACK`**
- **`STOPTR`**
- **`CONTROL T`**
- **`CONTROL D,U`**
- **`CONTROL D,H,`**
- **`MSGRT TR=A`**

2.6.2 ECMB=NO in IEAOPTxx parmlib member

Previously, the CMB= parameter specified the I/O device classes for which measurement data was to be collected, in addition to the DASD and tape device classes. It also allowed you to specify the number of channel measurement block (CMB) slots to reserve for adding more devices with an ACTIVATE. As of the z990 Exploitation feature on z/OS V1R4, the system ignores the CMB and uses instead the measurement data in the extended channel measurement block (ECMB). As a migration action since then, you must convert user-written programs that make use of the CMB to make use of the ECMB, and contact ISVs to obtain updates to ISV programs that use the CMB. The types of programs especially likely to use the CMB are monitor programs.

Attention: If ISV support is not available, you can specify ECMB=NO in parmlib member IEAOPTxx as a circumvention until the ISV support is available. See APAR OA06164 for additional details. The ECMB=NO circumvention is removed in z/OS V1R7.

2.6.3 JES2 $ACTIVATE for z2 mode to R4 mode

JES2 will no longer support compatibility with pre-z/OS V1R2 systems. The **$ACTIVATE** command, which converts the JES2 checkpoint from z2 mode to R4 mode (compatibility mode), is removed. Before installing z/OS V1R7, you must do one of the following:

1. Use the **$ACTIVATE** command to convert the JES2 checkpoint to z2 mode.
2. Offload the SPOOL, cold start JES2 z/OS V1R7, and reload the jobs on the new SPOOL.

2.6.4 JOBCAT and STEPCAT support

The DFSMSdfp JOBCAT and STEPCAT facilities are removed in z/OS V1R7. The JOBCAT and STEPCAT facilities have been in existence for many years, predating the introduction of ICF (integrated catalog facility) catalogs. JOBCAT and STEPCAT were designed to address some of the functional shortcomings of VSAM catalogs, such as:

- VSAM volume ownership, that is, all data sets on a volume having to be in the same VSAM catalog. Multiple catalogs could not point to data sets on the same volume.
- Performance problems resulting from no multilevel alias support, as well as lack of ability to subset catalog data for recovery purposes.
- Restrictions in the definition of the catalog SVC interface.

ICF catalogs

The introduction of ICF catalogs in the mid-1980s and other catalog enhancements (such as the multilevel alias support) directly addressed those problems. In addition, processes were developed for system build to use system-specific aliases instead of JOBCAT or STEPCAT. CBIPO introduced these processes and they are used today by offerings such as ServerPac to create data set entries in the new master catalog of the system being built. At the time ICF catalogs were introduced, the JOBCAT and STEPCAT facilities were functionally stabilized. Neither SMS-managed data sets nor UCBs above the 16 megabyte line can be used with JOBCAT or STEPCAT. ICF catalogs contain sufficient functional capabilities that all functions that previously could only be performed with JOBCAT or STEPCAT can now be done without them.

Problems using JOBCAT/STEPCAT

The use of JOBCAT and STEPCAT can actually cause significant problems. Data sets are generally not cataloged according to the normal predictable search order when JOBCAT or STEPCAT is used. This impacts the ability to do comprehensive installation storage management and can increase staff requirements. For example, interval migration and recall using DFSMShsm is effectively unusable when the data sets cannot be found using the standard catalog search order. The use of JOBCAT and STEPCAT can also result in noticeable increases in the time required to perform catalog requests.

2.6.5 DFSMS ISAM

Due to ISAM's limited functionality and the capabilities of VSAM, particularly VSAM data sets in extended format, z/OS V1R6 is the last release in which DFSMS ISAM and the utility program, IEABISAM, will be available. IBM has provided the ISAM Compatibility Interface

(ISAM CI) which allows users to run an ISAM program against a VSAM KSDS data set. Details on using this interface and procedures for converting ISAM data sets to VSAM data sets can be found in Appendix E of *z/OS DFSMS: Using Data Sets*, SC26-7410. This compatibility interface program is planned to continue to be provided as part of DFSMS and will not be discontinued when ISAM is removed from DFSMS.

2.6.6 OS/390 R10 C/C++ compiler

From optional feature C/C++ without Debug Tool, the OS/390 R10 level of the C/C++ compilers is removed from z/OS V1R7. The OS/390 R10 C/C++ compilers were shipped as an aid to migration to the C/C++ compilers that were introduced in z/OS V1R2. The z/OS V1R2 level of the C++ compiler supports the ISO 1998 Standard level of C++.

Attention: For information about migrating from the older to the newer level of the compilers, see *z/OS C/C++ Compiler and Run-Time Migration Guide for the Application Programmer*, GC09-4913.

2.6.7 z/OS Optional Source media feature

The z/OS Optional Source media feature is no longer offered in z/OS V1R7. The last release offering these materials will be z/OS V1R6. These features contain macros and source code for some programs in the z/OS BCP, BDT base, BDT SNA NJE, BDT File-to-File, DFSMS, MICR/OCR, BCP JPN, and Security Server RACF elements.

2.7 Functions to be withdrawn in a future release

Figure 2-6 lists items that IBM has announced it intends to remove in a future z/OS release. You are encouraged to consider these removals when making your plans for system upgrades. These statements represent IBM's current intentions. IBM development plans are subject to change or withdrawal without further notice.

zFS multi-file system aggregates shared across a sysplex (from Distributed File Service)	**Base element - zFS compatibility mode aggregates (which have a single file system per data set) will continue to be supported in all environments.**	**Planned for a release after R7**
Firewall Technologies (from Integrated Security Services)	**Base element - Many Firewall Technologies functions have been been stabilized and can be replaced w/ Communications Server functions. Some functions won't have replacements.**	**Planned for a release after R7**
Some Communications Server Functions	**Base element - TCP/IP Configuration profile block definitions, PAGTSNMP subagent, EE TGs definition by specifying multiple SAP addrs, and AnyNet**	**Planned for a release after R7**
Any remaining one-byte console ID support (from BCP)	**Base element - Use console names instead of one byte console IDs**	**Planned for a release after R7**
Bind DNS 4.9.3 (from Communications Server)	**Base element – implement BIND 9.2.0 as a replacement (available since z/OS R4)**	**Future release**
OROUTED (from Communications Server)	**Base element – use OMPROUTE as the dynamic routing daemon**	**Future release**
English and Japanese panels from DFSORT	**Priced feature – no replacement offered**	**Future release**
zFS multi-file system aggregates (from Distributed File Service)	**Base element - zFS compatibility mode aggregates will still be supported**	**Future release**
Support for VSAM data sets with IMBED, REPLICATE, or KEYRANGE attributes (from DFSMS)	**Base element – plan to redefine any affected VSAM data sets. Use tool to assist in identifying affected VSAM data sets**	**Future release**

Figure 2-6 Functions to be withdrawn in future releases

2.7.1 Multi-file mode zFS aggregates

z/OS V1R7 is planned to be the last release to allow mounting zFS file systems contained in multi-file system aggregates that are to be shared across systems in a sysplex. IBM has previously recommended that these multi-file system aggregates not be shared in a sysplex environment. Once this support has been removed, attempts to mount zFS file systems contained in multi-file system aggregates will fail in a z/OS UNIX shared file system environment. Mounting zFS compatibility mode aggregates, which have a single file system per data set, will continue to be supported in all environments.

2.7.2 Firewall technologies

z/OS V1R7 is planned to be the last release to include the Firewall Technologies component of the Integrated Security Services element. Many Firewall Technologies functions have been stabilized for some time and can be replaced using comparable or better functions provided by or planned for Communications Server, notably, IPSecurity. In addition, a functionally rich downloadable tool is planned to replace the IPSecurity and IP Filtering configuration GUI support. The following functions will be removed without replacement:

- FTP Proxy services
- Socks V4 services
- Network Address Translation (NAT)
- RealAudio (TM) support

2.7.3 Communication Server

z/OS V1R7 is planned to be the last release in which z/OS Communications Server will support the following functions, after which they will be removed from the product:

- TCP/IP configuration profile block definition statements
- ASSORTEDPARMS
- ENDASSORTEDPARMS
- KEEPALIVEOPTIONS,
- ENDKEEPALIVEOPTIONS

2.7.4 One-byte console IDs

In the release following z/OS V1R7, the remaining support for one-byte console IDs will be removed from control blocks CIB, CSCB, ORE, WQE, XSA. This completes the removal of one-byte console IDs in z/OS. Instead of using one-byte console IDs, console names should be used.

2.7.5 BIND DNS 4.9.3

In a future release the support for BIND DNS 4.9.3 will be removed from Communications Server. Customers should implement BIND DNS 9.2.0 as a replacement. BIND DNS 9.2.0 is included in the product beginning with z/OS V1R4. Customers exploiting the Connection Optimization (DNS/WLM) feature of BIND 4.9.3 should investigate alternative solutions, such as the Sysplex Distributor function.

2.7.6 OROUTED daemon

In a future release the support for OROUTED will be removed from Communications Server. Customers should use OMPROUTE as their dynamic routing daemon.

2.7.7 DFSORT panels

The English and Japanese ISPF panels will be removed from DFSORT™ in a future release. This limited function interactive facility will no longer be provided, and there will be no replacement.

2.7.8 zFS multi-file system aggregates

In a future release, IBM plans to withdraw support for zFS multi-file system aggregates. When this support is withdrawn, only zFS compatibility mode aggregates will be supported. A zFS compatibility mode aggregate has a single file system per data set.

2.7.9 VSAM data sets

From DFSMS, support for the VSAM IMBED, REPLICATE, and KEYRANGE attributes will be withdrawn in a future release. No supported release of z/OS or OS/390 allows you to define new VSAM data sets with these attributes. Using them for existing data sets can waste DASD space and can often degrade performance. When this support is withdrawn, you will not be able to process data sets with these attributes. It is best to plan for this removal now, with the aid of a tool that will help you identify affected data sets.

2.8 z9-109 processors

The z/OS capabilities that you have on the z9-109 server depend on the level of z/OS that you execute on the z9-109. More z/OS capabilities exist on the higher z/OS releases than on the lower z/OS releases. The lowest supported z/OS release for the z9-109 is z/OS V1R4 with the z990 Compatibility feature.

The z990 Compatibility feature is no longer orderable, and has been replaced with the z990 Exploitation feature. The z990 Exploitation feature remains orderable until December 2006.

Note: All z/OS V1R4 orders placed after 24 February 2004 automatically included the z990 Exploitation feature.

Software requirements differ depending on z/OS release and functions exploited. Support is provided via a combination of features, Web deliverables, and PTFs. The required PTFs are documented in hardware and software PSP buckets depending on what server and z/OS release you are coming from.

The following list identifies the supported z/OS levels that can run on a 2094 (z9-109) processor shown in Figure 2-7 on page 34. This does not imply all 2094 functions are available in all z/OS supported releases. The following releases are supported:

- z/OS V1R4 with z990 Exploitation Support feature + PTFs
- z/OS V1R4 with z990 Compatibility Support feature + PTFs
- z/OS V1R5 plus PTFs
- z/OS V1R6 plus PTFs
- z/OS V1R7 plus PTFs

Note: The bimodal migration accommodation is not available for z/OS V1R5 and all later releases; therefore, z/OS V1R5 can only run in 64-bit mode on a z9-109, z990, and z890 processor. The z/OS V1R4 z990 Compatibility Support feature is no longer available and this function is now included in the z/OS V1R4 z990 Exploitation Support feature.

2.8.1 z990 functions available with 2094

The functions available on the z990 when using the z/OS V1R4 z990 Exploitation Support feature installed are carried forward to the 2094, as follows:

- IPL with any LCSS
- IPL with any LPAR ID
- ECMB (Extended Channel Measurement Block)
- External spanned channels
- 60 logical partitions

60 logical partitions

IBM is once again doubling the number to 60 logical partitions (LPARs). This allows the ability to define up to 60 LPARs with 15 LPARs per Logical Channel Subsystem. This may provide more flexibility in allocating hardware resources. With Processor Resource/Systems Manager™ (PR/SM™) and Multiple Image Facility (MIF) you can share ESCON® and FICON channels, ISC-3s, and OSA ports across LPARs. Support for up to 30 LPARs became available in October 2003.

Note: Support of up to 60 LPARs is exclusive to the z9-109 and is supported by z/OS, z/VM®, z/VSE™, TPF, z/TPF, and Linux on System z9. Introduction of 60 LPARs satisfies the Statement of General Direction in Hardware Announcement 103-142, (RFA38035) dated May 13, 2003.

2.8.2 z/OS V1R7 function support for z9-109

Many of the new features with the z9-109 processor are only available when z/OS V1R7 is the operating system. These features are as follows:

- 63.75K subchannels is exclusive to the z9-109
  - zSeries addresses a maximum of 64K subchannels in subchannel set 0 (zero) with 1024 (1K) of these previously reserved for system use. IBM is making available 768 of these 1K reserved subchannels for customer use. The increased addressable storage this represents may be significant as follows:

    For example, if you are using 3390 volume sizes and have 768 volumes of 54 GB per volume, this is equal to 41 terabytes of increased storage addressability.

    ```
    54GB/volume * 768 volumes = 41 TB
    ```

  - The IBM TotalStorage® DS8000 series can be defined as follows:

    To attach 63.75K unit addresses and with 63.75K now supported in the host, there is symmetry between the server and storage subsystems.

Note: 63.75K subchannels is supported by all channel types, and by z/OS V1R7 and z/VM.

- Request node identification data (RNID)

  RNID is designed to assist with the isolation of ESCON and FICON cabling-detected errors. In a fiber optic environment when using extended distances, z/OS V1R7 can request the RNID data for a specified device and control unit attached to ESCON or native FICON channels and display the RNID data using an operator command.

  Note: RNID is exclusive to z9-109, and is supported by ESCON, FICON Express2, and FICON Express features when configured as CHPID type FC, and by z/OS V1R7.

- Separate PU management for PUs

  A new flexibility for managing processor units (PUs) allows the following PUs to now be managed separately:

  – Internal Coupling Facility (ICF) processors
  – Integrated Facility for Linux on System 9 (IFL) processors
  – zSeries Application Assist Processors (zAAPs)

  Note: In the past, ICF processors, IFL processors, and zAAPs were grouped together for allocation within and across the LPARs. The separate management of PU types enhances and simplifies capacity planning and management of the configured LPARs and their associated processor resources.

Figure 2-7 z9-109 processor

2.8.3 Migration support for z9-109 processor

Migrating to a z9-109 processor from a z990 or z890 requires fewer migration actions than are necessary for users coming from servers older than z990 or z890 who have not yet performed the migration actions associated with z990 or z890.

It is important to note that you can migrate directly to a z9-109 without going through to intermediate servers, but you still do need to ensure that any migration considerations are satisfied for those servers that you “skipped.”

Cryptographic support

The support (excluding the cryptographic support) for the z9-109 is delivered entirely via service, unlike the support that was required for the z890 and z990. The z890 and z990 support was delivered with service and FMIDs (Web deliverables and features). The

cryptographic support for the z9-109, as well as for the z890 and z990, continues to be FMIDs, many of which are available in Web deliverables.

Migration considerations

Migration actions are documented in *z/OS and z/OS.e Planning for Installation*, GA22-7504. For migrations using the z9-109 processors, consider the following changes:

- Update CFRM policies.

  If a coupling facility image resides on a z990, then the LPAR ID (from the HMC Image Profile) is used in the CFRM policy. In order to use an LPAR ID > 15 (X'F'), the z990 compatibility (or exploitation) code is required on all systems in the sysplex.

  Use the **`SETXCF START,POLICY,TYPE=CFRM`** command to make the updated policy active.

- Update automation for new and changed messages.

  Several messages and command output are updated for 2 digit LPAR IDs. Notify those affected by changed command output.

- Update PARMLIB members.
  - IPCS support, there is a new ICSF member, CSFIPCSP.
  - With exploitation, with the IEASYSxx parmlib member, the CMB= parameter is now ignored.
  - IEAOPTxx ECMB=NO is *not* supported in z/OS V1R7 and higher. Previously, this was merely a circumvention.

    Note: With the z990 Exploitation feature on z/OS V1R4, the CMB was ignored and uses instead the measurement data in the extended channel measurement block (ECMB). As a migration action since then, you must convert user-written programs that make use of the CMB to make use of the ECMB, and contact ISVs to obtain updates to ISV programs that use the CMB. Now, ECMBs only are supported.

  - With exploitation, the IEASYSxx parmlib member may require an increase (by 1) for the value of MAXCAD.
  - With exploitation, the SMFPRMxx parmlib member specifying the description of the serial number in the SID parameter is changed when running on a z990.
- Rebuild standalone dump.
- OSA/SF has a new GUI which requires Java 1.1.2 help files and the Java 1.4 run-time library loaded on the workstation.

Migration APARs

The APARs shown in Figure 2-8 on page 36 provide migration support for the z9-109 processor. The z9-109 compatibility support is for all the z/OS supported environments.

BCP Support for IODF Size Reduction	OA08197(*)
SMF recognizes new processor SU values, based on LSPR information for Measured Usage Reporting Program - IFAURP/IFAUMCCT	OA11730
EREP Support for new z9-109	IO00735 IR53369
HCD Processor Support Module (PIT)	OA07875(*)
IOCP	OA11665
RMF for Enhanced PR/SM Diagnose 204 Output	OA10346
HLASM support for new hardware instructions	PK02660
ICSF (Crypto toleration)	OA09157(*) OA11946
OSA/SF - OSA-Express2 CHPID type OSN	OA11007

(*) Integrated into z/OS V1R7 FMIDs

Figure 2-8 APARs for migration support

Note: Compatibility support PTFs must be installed in all z/OS environments.

2.8.4 New HCD V5 level for z/OS V1R7

HCD APAR OA07875 is required to define a 2094 for the following z/OS systems:

- z/OS V1R4 with the z990 Exploitation Support feature (HCD FMID HCS7708)
- z/OS V1R5 (HCD FMID HCS7708)
- z/OS V1R6 (HCD FMID HCS7708)

APAR OA07875 provides support for HCD users who share IODF data sets between pre-z/OS V1R7 systems and z/OS V1R7 systems.

IOS APAR OA08197 is required for IPL when sharing an IODF built with the z/OS V1R7 HCD. IPL will fail without this APAR installed.

Migration to z/OS V1R7

When planning a migration to z/OS V1R7, the IODF has important coexistence and fallback requirements, as follows:

- If you are running z/OS V1R4 without the z990 Compatibility feature, then you have an additional coexistence and fallback requirement that must be followed.

  It is not possible to read from or dynamically activate a V5 IODF from that z/OS V1R4 system. If you wish to read from or dynamically activate a V5 IODF from that z/OS V1R4 system, then you may install the z/OS V1R4 z990 Compatibility feature (which has been replaced with the z/OS V1R4 z990 Exploitation feature), and install the PTFs for APARs OA07875 and OA08197.

- To read from, IPL with, and dynamically activate an IODF at the V5 level, the PTFs for coexistence APARs OA07875 and OA08197 are required on back-level systems. If you attempt to IPL with a V5 IODF from a lower level, the z/OS system that does not have the PTF for APAR OA08197 installed goes into a wait state.

The coexistence PTF for APAR OA07875 does not allow you to update the V5 IODF from back-level systems. Once the IODF has been upgraded to V5, the z/OS V1R7 HCD libraries must be used to process updates to it. A STEPLIB or JOBLIB from a back-level system is acceptable.

Important: APAR OA07875 is available for z/OS V1R4 systems that have the z/OS V1R4 z990 Compatibility feature installed (that is, HCD FMID HCS7708). z/OS V1R4 HCD without the z990 Compatibility feature (that is, HCD FMID HCS6091) does not have an applicable coexistence PTF. Therefore, HCD FMID HCS7708 shows up as “z/OS V1.4 HCD” on its primary panel, and is described as “z/OS V1.4 HCD” in the documentation. HCD FMID HCS6091 shows up as “OS/390 Release 9 HCD” on its primary panel.

IODF to V5

An upgrade of the IODF to V5 is only in z/OS V1R7 HCD FMID HCD7720. It is not in HCD APAR OA07875 for z/OS releases V1R4 to V1R6. The upgrade option is shown in Figure 2-9.

IOS APAR OA08197 is required for IPL when sharing an IODF built with the z/OS 1R7 HCD. IPLs will fail without this APAR installed.

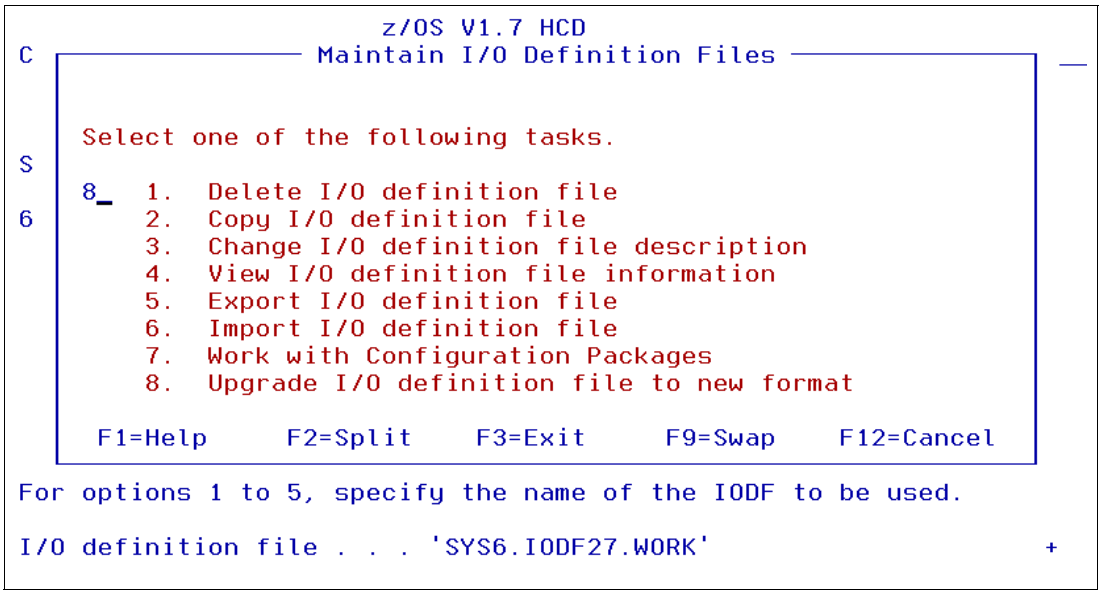


Figure 2-9 HCD panel to upgrade an existing IODF to V5 level

2.8.5 Defining devices in subchannel set 1 (SS1)

The problem that is solved with z/OS V1R7 is that currently installations have to give up a lot of device numbers for 2105 alias addresses, and have run out of room to define other devices. Up to this point, (z/OS R1R6 and earlier and z990 and earlier) there was only a single subchannel set (now known as SS0) in each channel subsystem (CSS). With the 2094 and z/OS V1R7, we have now introduced the capability of defining a second subchannel set (SS1) in one or more CSSs. The new HCD from z/OS V1R7 is required in order to be able to define SS1 and its devices. Also, a POR is required in order to define SS1 for the first time in

any given CSS. The only devices that can be defined in SS1 are 3390A (2105 alias) devices. By moving alias devices out of SS0 and into SS1, there is now more room for additional devices in SS0. Again, each CSS may be defined to have only SS0, or both SS0 and SS1.

The HCD screen showing the definitions in place for a 2094 appears in Figure 2-10. Note that SS1 is defined in CSS2 only and that the settings for the number of devices in SS0 and SS1 (in CSS2) are at their maximum values. Changing the maximum value for any SSx in any CSS requires a POR, so you can define values at the allowed maximum right from the start.

```
                          Channel Subsystem List                 Row 1 of 3
Command ===> ____________________________________________ Scroll ===> CSR

Select one or more channel subsystems, then press Enter.  To add, use F11.

Processor ID . . . : SCZP101

  CSS Devices in SS0    Devices in SS1
/ ID  Maximum + Actual  Maximum + Actual  Description
/ 0   65280     7364    0         0       ________________________________
_ 1   65280     7108    0         0       ________________________________
_ 2   65280     8371    65535     1       ________________________________
****************************** Bottom of data ******************************
```

Figure 2-10 Channel Subsystem List panel in HCD

Only z/OS V1R7 systems can use CSS2 and take advantage of the alias devices defined in SS1. Additionally, because device numbers for operating systems in CSS0 and CSS1 may be different from device numbers in CSS2, we will have a different operating system definition (defined in the IODF) in place for the systems running in CSS2.

Adding alias devices

To add alias devices in SS1, use the Add Device dialog specifying a device type of 3390A, as shown in Figure 2-11.

```
  Goto  Filter  Backup  Query  Help
 ------------------------------ Add Device ------------------------------


 Specify or revise the following values.

 Device number  . . . . . d340  + (0000 - FFFF)
 Number of devices  . . . . . 192_
 Device type  . . . . . 3390a_____  +

 Serial number  . . . . . __________
 Description  . . . . . ________________________________________

 Volume serial number . . . . . _______  (for DASD)

 Connected to CUs . . D300 ___  ____  ____  ____  ____  ____  ____  +
```

Figure 2-11 Adding alias devices in SS1

Add devices to an operating system

Alias devices are added to each selected operating system configuration, as shown in Figure 2-12.

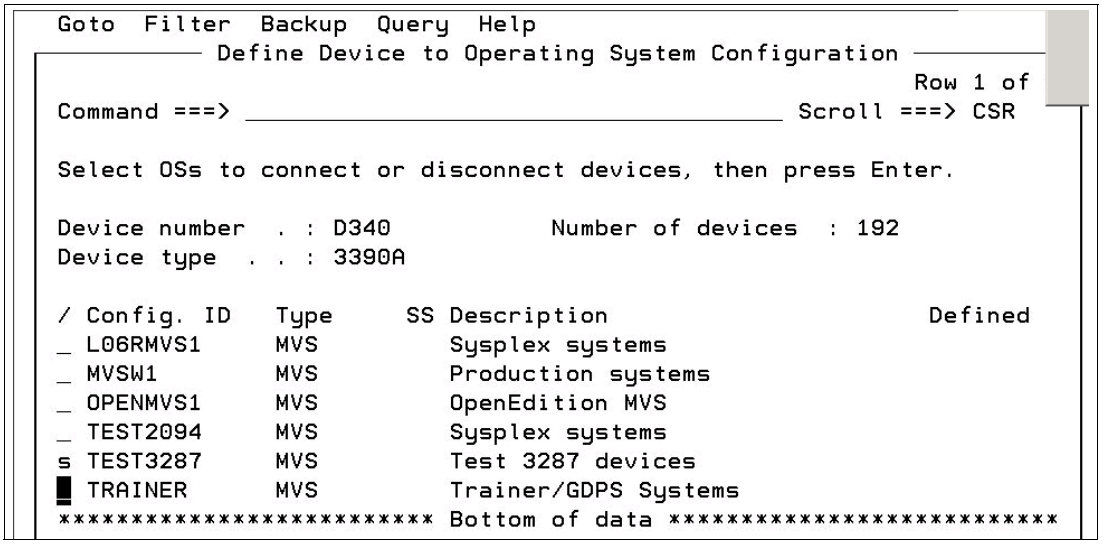


Figure 2-12 Add devices to an operating system

Specify Subchannel Set ID panel

The Specify Subchannel Set ID panel is used to set the Subchannel Set ID of the alias devices, as shown in Figure 2-13.

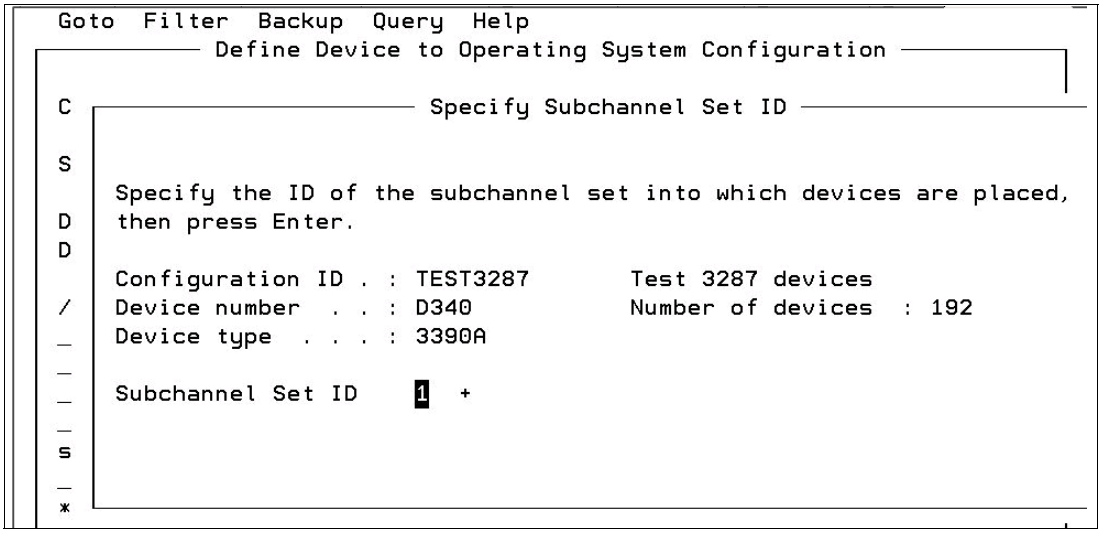


Figure 2-13 Specify a Subchannel Set ID

3

z/OS V1R7 ServerPac enhancements

A ServerPac order includes an ISPF application called CustomPac that is used to customize and install z/OS.

This chapter discusses the enhancements and changes to the ServerPac installation process including:

- Merge data set support for filesystems
- Other dialog changes for merge
- Support for zFS root
- Dialog changes for secondary space
- CONSOL00 parmlib member support
- z/OS Health Checker support
- Removal of PSP bucket
- Migration considerations

3.1 Improvements to the installation dialog

Over the last 10 plus years, there has been a steady increase in the amount of code being installed into the z/OS UNIX file systems on z/OS. Over time, significant enhancements have been included in the offering and in the dialog, such as:

- Restructuring the install and documentation (Installing Your Order)
- Adding “Software Upgrade” versus “Full System Replace” installation options
- SMS construct support
- Merge data set
- Recommended system layout support
- View / Change facility
- HFS and zFS support
- Electronic delivery

Previously the CustomPac dialog had limited support for handling filesystems and their mountpoints, and there was no way for the dialog user to manage and configure the physical filesystem structure. The ServerPac for z/OS V1R7 introduces:

- Merge data set support for filesystems
- Other dialog changes for merge
- Support for zFS root
- Dialog changes for secondary space
- CONSOL00 parmlib member support
- z/OS Health Checker support
- Removal of PSP bucket
- Migration considerations

3.2 Merge data set support for filesystems

Filesystem merge is a new capability in the CustomPac dialog. The CustomPac dialog is updated to provide functionality similar to *data set merge* for management of UNIX filesystems. With z/OS V1R7 ServerPac, the dialog allows you to merge some of the eligible HFS and zFS data sets in the ServerPac order. The benefit to users of the dialog is that they are able to merge filesystems, if desired, when availability of DASD space is not an issue.

Using the new capabilities of filesystem merge, you can:

- Merge filesystems to minimize the number of them and maximize DASD utilization
- Be shielded from creating the BPXPRMFS parmlib member
- Be protected from creating an invalid software configuration

The RESTORE job is enhanced to create the BPXPRMFS member.

3.2.1 Data set merges

From the dialog Installation Menu, enter `M` (Modify) to begin the next dialog function. The Modify System Layout Options panel is displayed, as shown in Figure 3-1 on page 43.

If filesystem merge is to be used, IBM recommends that you modify your configuration in the following order:

1. Data set merges, if any
2. Data set space changes, if any
3. Specifying Reserved Space, if it will be used
4. Specifying which volumes are not to be initialized
5. Everything else

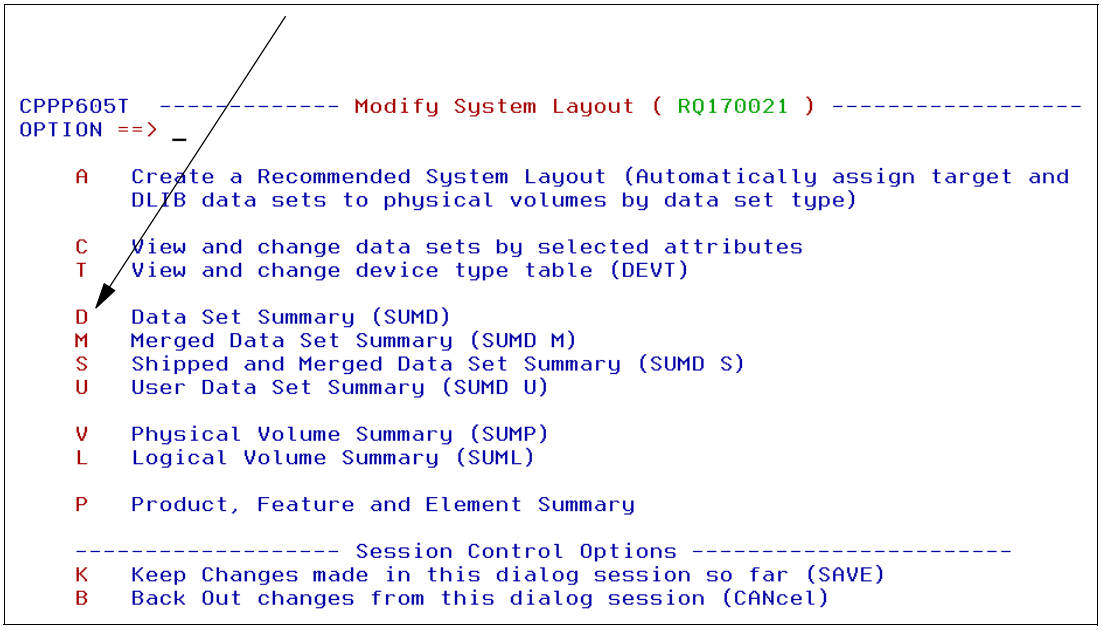


Figure 3-1 Modify System Layout panel - Select D (SUMD)

Displaying filesystem merge candidates

The merge data set function, which now includes filesystems, can be driven from any data set list display. For illustration purposes, the SUMD display shown in Figure 3-2 was accessed by entering a **D** on the command line in the panel shown in the previous screen.

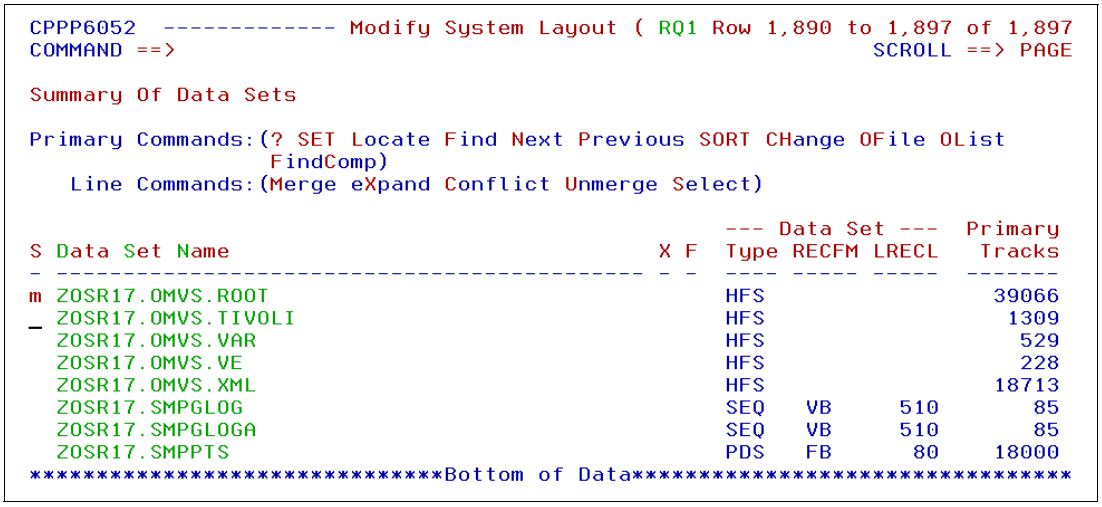


Figure 3-2 Summary of Data Sets panel with new option Merge

Eligible merge filesystems

Figure 3-3 shows the new panel CPPP605V that is displayed when the filesystem ZOSR17.OMVS.ROOT is selected from the Summary of Data Sets panel shown in the previous figure. The resulting display shows all filesystems that are eligible to be merged.

Note: When installing and with large DASD devices defined, you can choose to use this function to minimize the post-installation filesystem management.

Merge filesystems into target

Figure 3-3 shows which of the data sets from the display list are eligible for merging with the selected data set (the target data set). Enter **S** to the left of any data sets that you want to merge with the target data set.

```
CPPP605V  ------------ Modify System Layout ( RQ170021 ) ---- Row 1 to 2 of 2
COMMAND ==> _                                                 SCROLL ==> PAGE

Data Set Merge Candidates for ZOSR17.OMVS.ROOT

Primary Commands:(? SET Locate Find Next Previous Merge)
   Line Commands:(Information Select)

                                                 DS
S Data Set Name                                  Type     Mount Point
- ----------------------------------------------- ---- --------------------------
  ZOSR17.OMVS.XML                                HFS  /usr/lpp/ixm
  ZOSR17.OMVS.JV390                              HFS  /usr/lpp/java/J1.4
******************************** Bottom of data ********************************
```

Figure 3-3 Data set merge candidates for filesystems

Merging filesystems

The merge filesystems capability is structured (ruled by) their mountpoints, as follows:

- A filesystem can only be merged into a parent mountpointed filesystem.
- Intermediate mountpointed filesystems may be included in a single merge operation.
- Candidates must be eligible for merging.

A new panel CPPP605V is displayed when the target of the merge function is a filesystem.

3.2.2 Other dialog changes for merge

When selecting a data set such as ISP.SISPLOAD for its merge candidates, Figure 3-4 on page 45 is displayed. z/OS V1R7 includes the following modifications to this panel:

- The **I** (Information line command) has been added and when used displays the CPPP605D panel.
- The RECFM column has been removed from the panel.
- The APF flag column has been added. The APF column displays a Y if the data set requires APF authorization.

```
CPPP605M  ------------- Modify System Layout ( RQ170021 ) -- Row 1 to 21 of 51
COMMAND ==> _                                                 SCROLL ==> PAGE

Data Set Merge Candidates for ISF.SISFLOAD

Primary Commands:(? SET Locate Find Next Previous SORT Merge)
   Line Commands:(Information Select Conflict)

                                                  Element    DS    MCAT  IPL
S Data Set Name                                   Type       Type  Reqd  Vol APF
- ----------------------------------------------  ---------- ----  ----  --- ---
  APK.ACIF.SAPKMOD1                               LMOD       PDS     Y    N   N
  ASM.SASMMOD1                                    LMOD       PDS     Y    N   Y
  CBC.SCLBDLL                                     LMOD       PDS     Y    N   Y
  CSF.SCSFMOD0                                    LMOD       PDS     Y    N   N
  DIT.V1R3M0.SDITMOD1                             LMOD       PDS     Y    N   Y
  DVG.NFTP230.SDVGLMD0                            LMOD       PDS     Y    N   Y
  DVG.NFTP230.SDVGLMD2                            LMOD       PDS     Y    N   N
  EOX.SEPHLOD1                                    LMOD       PDS     Y    N   N
```

Figure 3-4 Data set merge modifications on panel CPPP605M

3.3 Support for zFS root

ServerPac has removed the restriction on using a zFS root for your root filesystem. The shipped HFS data set for the root can be “switched” to a zFS root on the data set modification attributes panel shown in Figure 3-5, by doing the following:

- Change the Data Set Type field from HFS to ZFS.
- Change the Data set Name if required.

ServerPac will then create a zFS aggregate and format it, and move the root into the zFS aggregate.

```
CPPP605D  ------------ Modify System Layout ( RQ170021  ) ------------------
COMMAND ==> _

Data Set Modification - Attributes


       Data set Name ==> ZOSR17.OMVS.ROOT
        Shipped      :  OMVS.ETC

       Placement     :  C       (DLIB, Target, Catalog, or User-Defined)

       Data Set Type ==> ZFS     (HFS, PDS, PDSE, SEQ, VSAM, or ZFS)
        Shipped      :  HFS

       SMS-Managed   ==> NO      (Yes or No)
       SMS-Eligible  :  YES
       SMS-Required  :  NO

     Logical Volume ==> HLB002  Shipped   :  CAT001
    Physical Volume  :  T6Z5H1
      Storage Class  :
```

Figure 3-5 Panel to change the root filesystem from HFS to zFS

3.4 Dialog changes for secondary space

The CustomPac dialog and ServerPac are modified in support of z/OS Health Checker. The dialog is changed to support LNKLSTed data sets to be allocated with no secondaries. The following changes are made:

- LNKLSTed data sets are going to be shipped with no secondary space allowed. The default *free space* for data sets with no secondaries is 20% since the default for data sets with secondaries is normally 10%. Other data sets may also be shipped with no secondaries.
- Support is added to the Select Data Set View panel, shown in Figure 3-6 on page 46, which has a new selection to allow a *find* for data sets shipped with no secondaries.
- The **CHange SECOND** command is enhanced to change data sets with no secondaries to allow secondaries. These changes support the installation for using SMP/E maintenance.

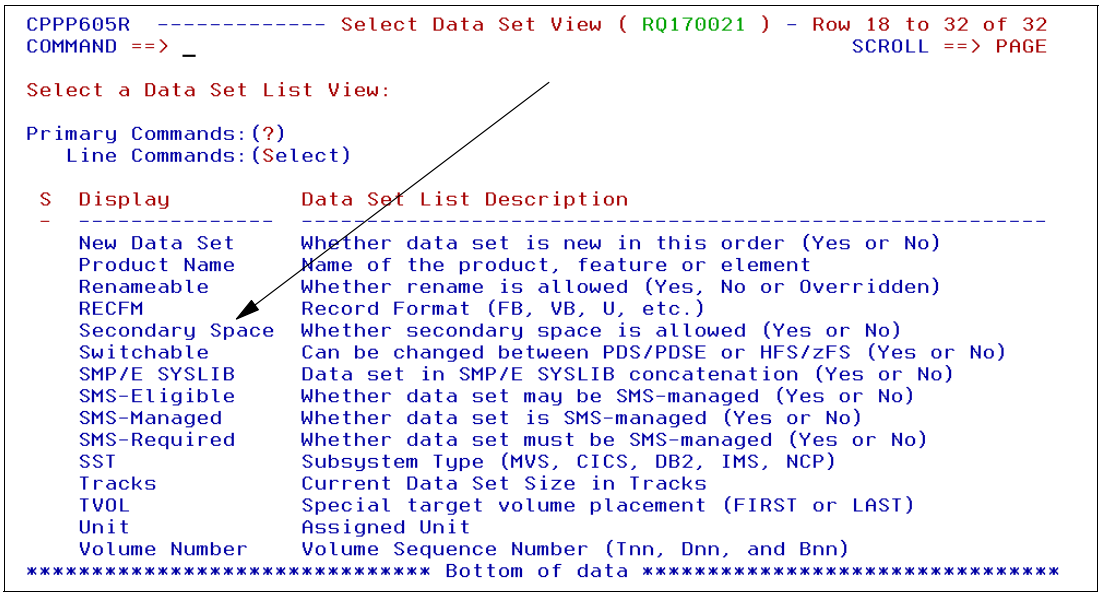


Figure 3-6 Select Data Set View panel for secondary space selection

3.4.1 Data sets with secondary space

To find data sets with or without secondary space, enter an **S** on the panel shown in Figure 3-6, on the Display for Secondary Space line. The panel shown in Figure 3-7 is displayed.

Command changes for secondary space

ServerPac ships the link list data sets with no secondary space allocations. The **CHange SECOND** command allows you to override this and allocate secondary space for these data sets. Only the data sets in the list that were originally shipped with no secondary are eligible to have secondary space allocation changed. This command has no effect on data sets with shipped secondaries.

If you use the **CHange SECOND Y** command to allocate secondary spaces for data sets that were shipped with no secondary space, you can subsequently use the **CHange SECOND N** command to restore the no secondary attribute. You cannot, however, use this **CHange SECOND N** command to remove the secondary attribute for data sets that were shipped with secondary space allocated.

Display data sets with no secondary space

To see a list of data sets that do not have secondary space allocated, select the No option; to see those that do have secondary space allocated, select Yes.

```
CPPP605S  ----------- Select Values to Display ( RQ170021 ) -- Row 1 to 2 of 2
COMMAND ==> _                                                 SCROLL ==> PAGE

Select list for: Secondary Space

Select values for which data sets are to be listed and press Enter

Primary Commands:(?)
   Line Commands:(Select SS)

   S   Values
   --  ----------------------------------------
       No
       Yes
******************************Bottom of Data********************************
```

Figure 3-7 Select Values to Display panel for secondary space

For example, entering an **S** on the No line in the panel in Figure 3-7 returns the panel shown in Figure 3-8.

```
CPPP605U  --------------- Data Set List ( RQ170021 ) ------ Row 53 to 72 of 81
COMMAND ==> _                                                 SCROLL ==> PAGE

Data Set List for: Secondary Space

Primary Commands:(? SET Locate Find Next Previous SORT CHange OFile OList
                  FindComp)
   Line Commands:(Merge eXpand Conflict Unmerge Select)

                                                                      Physical
S Data Set Name                                 Selected Value        Volume
- ---------------------------------------------- --------------------- --------
  REXX.SFANLMD                                  No                    Z17RS1
  SCRIPT.R40.DCFLOAD                            No                    Z17RS1
  SYS1.CMDLIB                                   No                    Z17RS1
  SYS1.CSSLIB                                   No                    Z17RS1
  SYS1.DFQLLIB                                  No                    Z17RS1
  SYS1.DGTLLIB                                  No                    Z17RS1
  SYS1.LINKLIB                                  No                    Z17RS1
  SYS1.MIGLIB                                   No                    Z17RS1
  SYS1.NFSLIBE                                  No                    Z17RS1
  SYS1.NUCLEUS                                  No                    Z17RS1
```

Figure 3-8 Data sets with no secondary space

Changes to format and flow of Attribute panel

The data set attributes are displayed on two logically connected panels whose content is built dynamically and depends on the data set type. Use the **S** (Select) line command on the panel in Figure 3-8 to specify the data set about which you want more information. The resulting panels are shown in Figure 3-9 and Figure 3-10 on page 48.

The two logically connected panels (“1 of 2” and “2 of 2”) replace the constructs used in panel CPPP605D in previous releases. The two panels are:

- Data Set Modification - Attributes - (CPPP605D)
- Data Set Modification - Space - (CPPP605E)

If a filesystem is selected, panel CPPP605F is displayed.

```
CPPP605D  --------- Data Set Attributes 1 of 2 ( RQ170021 ) ---------------
COMMAND ==> _

  Data set Name  :  SYS1.NUCLEUS
   Shipped Name  :  SYS1.NUCLEUS
  Data Set Type  :  PDS                          Shipped   : PDS

Data Set Element Type  :  LMOD
             Category  :  Target

       Logical Volume  :  IPLVOL                 Shipped  :  IPLVOL
          SMS Managed  :  No

      Physical Volume  :  Z17RS1

       Primary Tracks ==> 924                    Shipped  :  924
     Secondary Tracks  :  0                      Shipped  :  0
     Directory Blocks ==> 123                    Shipped  :  123

   Product, Element, or Feature :  BCP
```

Figure 3-9 Data Set Modification - Attributes (1 of 2)

```
CPPP605E  --------- Data Set Attributes 2 of 2 ( RQ170021 ) ---------
COMMAND ==> _

   Data set Name  :  SYS1.NUCLEUS


                          RECFM  :  U
                          LRECL  :
                APF Authorized   :  No
                  LPA Eligible   :  No
                  LPA Required   :  No
             Link List Eligible  :  No
          SMP/E SYSLIB Data Set  :  Yes
         Required on IPL Volume  :  Yes
                   SMS-Eligible  :  No
                   SMS-Required  :  No
                     Renameable  :  No
     Required in Master Catalog  :  Yes
```

Figure 3-10 Data Set Modification - Space (2 of 2)

3.5 CONSOL00 parmlib member

The CONSOL00 parmlib member that is provided by ServerPac now has an option, AMRF(N), that is recommended and checked for by the console check in the z/OS Health Checker.

3.6 z/OS Health Checker support

Support has been added so that the new z/OS Health Checker can have the following functions available:

- Updates are included to the installed system to allow Health Checker to be active on the first IPL.
- Required “operational” data sets in ALLOCDS job are allocated.

- The **START** command for Health Checker in COMMND00 has been added.

3.7 Removal of PSP bucket

PSP buckets have been removed from the ServerPac and CBPDO documentation to resolve the problems of redundant, conflicting, and obsolete information. To ensure that the most current PSP bucket is used, installers are directed to access the PSP buckets from the internet.

3.8 Migration considerations

There are no new migration considerations with the z/OS V1R7 dialog. A change was introduced in the z/OS V1R6 dialog, so if you skipped that release, an upgrade of your dialog will be required to receive the order for z/OS V1R7.

Merging with saved configurations is now being limited to three releases prior to the level of the executing dialog. Attempting to merge with a saved configuration that is older than this will result in the error message:

```
CPPP0601009E – Configuration selected for merge is not supported
```

Apart from the enhancements discussed in this chapter the overall installation process is unchanged in this release compared with z/OS V1R6. The order's dialog is used to install a ServerPac z/OS V1R7 order and the "master dialog" will be automatically updated if it is lower than the order's level.

4

SAPI and extended status call enhancements

The SYSOUT Application Program Interface (SSI function code 79) allows JES to function as a server for applications needing to process SYSOUT data sets residing on JES spool. Use of the SAPI SSI call allows a user-supplied program to access JES SYSOUT data sets independently from the normal JES-provided functions (such as print or network). Users of this function are application programs operating in address spaces external to JES. SAPI supports multiple, concurrent requests from the applications' address spaces. Each issuer of the IEFSSREQ macro is referred to as an *application thread*.

This chapter describes the changes made to the Subsystem Interface in the following areas:

- SYSOUT Application Programming Interface (SAPI), SSI function code 79.
- Extended status function call, SSI function code 80.

4.1 SAPI enhancements

The following changes to SAPI processing have been made in z/OS V1R7:

- SSI Function Code 79 is enhanced to allow an application to make a read-only (non-update) call by setting a new flag SSS2SRON in IAZSSS2. A new error return code SSS2RRON is given back to the application if it tries to update data obtained with the read-only flag set.
- A new flag and new field allow the user to update the sysout priority via SSSI79 (SAPI).

This section covers the following topics:

- Overview of SAPI processing
- Support read-only access for spool files
- Return job-level ABEND code and highest condition code
- Support to modify SYSOUT priority
- Support to set forms code to the installation default
- Example of using the new functions

4.1.1 Overview of SAPI processing

The SYSOUT Application Program Interface (SSI function code 79) allows JES to function as a server for applications needing to process SYSOUT data sets residing on the JES spool. Use of the SAPI SSI call allows a user-supplied program to access JES SYSOUT data sets independently from the normal JES-provided functions (such as print or network). Users of this function are application programs operating in address spaces external to JES. SAPI supports multiple, concurrent requests from the applications' address spaces.

4.1.2 Using the SAPI interface

An application thread can make three types of requests with SAPI. Each is independent of, and mutually exclusive with the others. Field SSS2TYPE indicates which of these three possible types of requests the application thread is issuing:

- SSS2PUGE - indicates a SAPI PUT/GET request
- SSS2COUN - indicates a SAPI COUNT request
- SSS2BULK - indicates a SAPI BULK MODIFY request

PUT/GET request

This request, shown in Figure 4-1 on page 53, initiates data set selection, and optionally can provide disposition processing for the data set returned in the previous SAPI PUT/GET call.

PUT/GET request processing occurs when an application thread issues the IEFSSREQ macro to initiate data set selection. The input SSOB and SSS2 control blocks, provided by the application thread, specify the selection criteria used to select a data set. The application thread can use a wide variety of selection criteria to select a SYSOUT data set to be processed.

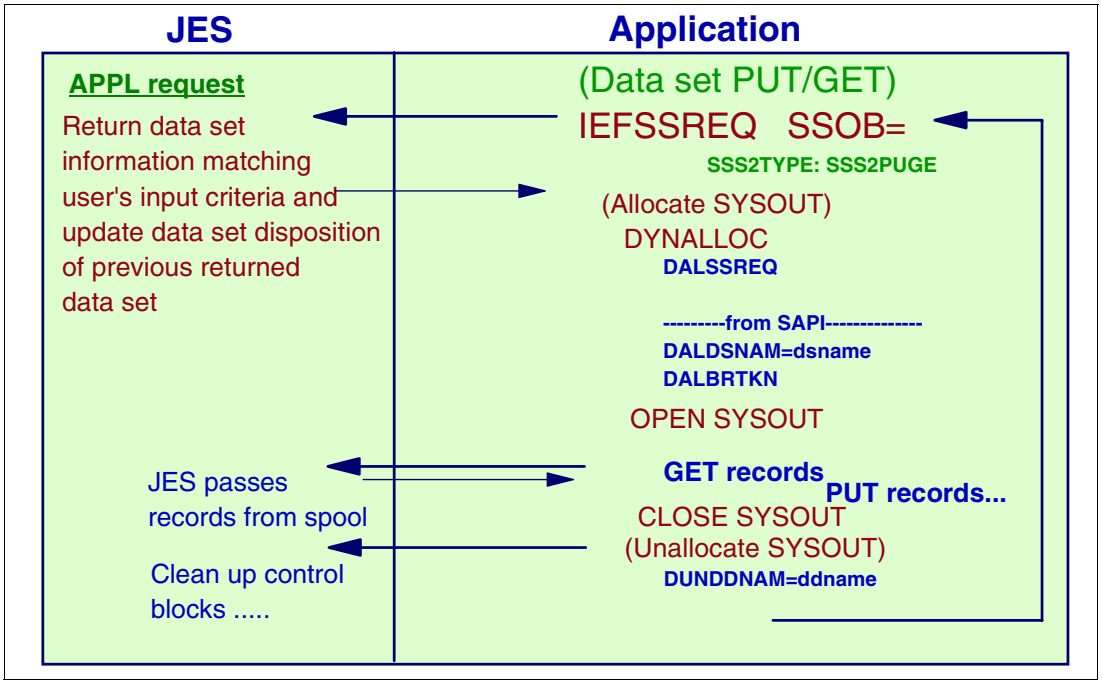


Figure 4-1 Application requesting a SYSOUT data set from JES

For more information about this interface see *z/OS MVS Using the Subsystem Interface*, SA22-7642.

4.1.3 Support read-only access for spool files

Prior to z/OS V1R7, applications must provide at least SAF UPDATE authority for the JESSPOOL resource class to the application thread to issue the SAPI PUT/GET call, using the IEFSSREQ macro correctly. In this release the interface is updated to allow the caller to specify that it will only read the spool files and never change or delete them. So the applications do not need to have high authority to read spool files.

This new function is invoked by setting the new flag SSS2SRON in field SSS2SEL5, one of five data set selection flags, in macro IAZSSS2. These flags, when set by the application, are used by JES to determine which data set matches the application's request. If this bit is on, JES performs read access requests for the data sets selected and gives an error return code if an attempt is made to update the status of a data set that was obtained with read access. The error causes the current thread to be terminated. The application must have SAF READ authority.

IAZSSS2 macro

The IAZSSS2 (SSS2) mapping macro, available in SYS1.MACLIB, is used as input to the IEFSSREQ request for SAPI processing. Fields in the SSS2 macro are differentiated into input, output, and disposition fields, as follows:

- An issuer's application thread sets input fields upon each IEFSSREQ invocation.
- JES manages output fields.
- JES updates the output-defined fields in response to each IEFSSREQ invocation.
- An issuer's application thread sets the disposition fields on an obtain data set request to inform JES of the disposition processing to occur for the data set returned on the prior obtain data set request.

4.1.4 Return job-level ABEND and highest condition codes

As of z/OS V1R7, additional job-related data is returned. There are new fields in macro IAZSSS2 that are passed back to the application in the SSOB extension; they are as follows:

- SSS2MXRC contains the highest return code.
- SSS2LSAB contains the last ABEND code.

4.1.5 Support to modify SYSOUT priority

In this release the caller can modify the SYSOUT priority on PUT requests by setting the new flag SSS2RPRI in field SSS2DSP2 and by specifying the new priority in a new field SSS2DPRI in macro IAZSSS2. A value of 0 through 255 is accepted. This field is meaningful only if SSS2RPRI is on in SSS2DSP2.

Attention: This specification is supported by JES2 only.

The new field SSS2DPRI is just one of the fields that can be used on the return call to JES to change output data set characteristics. The following fields (SSS2DCLS, SSS2DFOR, SSS2DPGM, SSS2DDES, SSS2CLFT, and SSS2DPRI) are used to change a subset of the data set characteristics.

Note: These fields only have meaning if the data set is kept (by setting SSS2DKPE on in SSS2DSP1).

4.1.6 Support to set forms code to the installation default

The caller can now set the forms code to the installation default by setting the new flag SSS2DNFO in a new field SSS2DSP2 in macro IAZSSS2.

4.1.7 Example use of this function

Appendix A.1, “Change the SYSOUT priority with a SAPI call” on page 476 provides an example of how SAPI can be called.

4.2 Extended status function call enhancement

The extended status function call (SSI function code 80) allows a user-supplied program to obtain detailed status information about jobs and SYSOUT in the JES queue. Both JES2 and JES3 subsystems support job status information.

Extended status is more than just an extension to the original STATUS SSI. It is intended as a programming interface into both JES2 and JES3 to obtain not only status information, but general information about jobs and SYSOUT. It can be used to obtain information to present on a front end such as SDSF, or to programmatically check the status of jobs. It can be also used as a screener for a SAPI print application to select work to process next.

The mapping macro used for SSI 80 is IAZSSST. z/OS V1R7 introduces a new version level of the IAZSSST (version 4).

Note: The new information presented here is only available if a version 4 IAZSSST is passed on the interface.

4.2.1 Extended status request types

The extended status interface is designed to be a general purpose interface to obtain information from JES. Callers use the STATTYPE field to indicate the type of data they require. This SSI call returns job information and SYSOUT status for information that is stored on the JES spool. This enhancement was requested by the Infoprint Server development. In z/OS V1R7, callers can now request either terse or verbose information, as follows:

Terse requests This request type returns less data but has lower overhead because no I/O is required.

Verbose requests This request type returns more detailed data, but involves multiple I/O requests. For this reason, verbose requests are limited in how much data can be obtained in a single SSI invocation. A verbose job request returns information related to the entire job, for example, the programmer name, or the message class.

To return additional job-related information to the caller, two new types of status calls were created, as follows:

- New status call: Job verbose
- New status call: Output verbose

In addition, the following changes have been implemented:

- A new external output structure
- 3 new status filters

For more detailed information about this function and these two new status calls see *z/OS MVS Using the Subsystem Interface*, SA22-7642.

4.2.2 New status extended call types

This new type of request is invoked by setting the STATVRBO flag in field STATTYPE in the IAZSSST macro. Selection filters must be limited to a single job.

Using job verbose calls

When you make the request, specify one the following:

- Must select by a range of one job
- A client token
- Request to expand terse element (JES2 only)

This can be done by selecting one of the following options:

- Set STATTRSA to zeros and ensure that the job ID filters specified by STATSJBI refer to the same job ID in STATJBIL and STATJBIH (or that STATJBIH is set to zero). Both terse and verbose job data is returned.
- Set STATTRSA to zeros and STATSCTK has STATCTKN set to the SYSOUT token you want verbose data for. Both terse and verbose data are returned.
- Set STATTRSA to a STATJQ or STATSE (obtained previously with no intervening memory management call). The related STATJQ will chain to a verbose element (STATVE), as shown in Figure 4-2 on page 59.

Note: The IAZSSST macro defines the SSOB extension used to request status information for jobs in the JES queue.

Fields returned to the caller

Table 4-4 shows all the returned fields. There is a new section STATJQVB in mapping macro IAZSSST.

Table 4-1 Fields returned in new DSECT STATJQVB

Field name	Meaning
STVBJCPY	Job Copy count
STVBXTSD	Execution start date
STVBLNCT	Job Line count
STVBXTE	Execution end Time/Date
STVBIDEV	Input device name
STVBXTET	Execution end time
STVBISID	Input system/member
STVBXTED	Execution end date
STVBJCIN	Job input count
STVBJUSR	JMRUSEID field
STVBJLIN	Job line count
STVBMCLS	Message class (Job card)
STVBJPAG	Job page count
STVBNOTN	Notify Node
STVBJPUN	Job card (output) count
STVBNOTU	Notify Userid
STVBRTST	Input start time
STVBPNAM	Programmer's name (Job card)
STVBRTSD	Input start date
STVBACCT	Account number (Job card)
STVBRTET	Input end time
STVBDEPT	NJE department
STVBRTED	Input end date
STVBBLDG	NJE building
STVBSYS	Execution MVS system name
STVBROOM	Job card room number
STVBMBR	Execution JES2 member name
STVBJDVT	JDVT name for job
STVBXTST	Execution start time
STVBSUBU	Submitting userid

Using output verbose status calls

A verbose SYSOUT request returns information relevant to particular SYSOUT data sets in a job. Some examples of these information fields are:

- Lines
- Pages
- Procedure name
- Step name
- DD name
- ...

The new request is invoked by setting the STATOUTV flag in field STATTYPE of macro IAZSSST and also setting one of the following input fields:

- Set STATTRSA to zero and ensure that the job ID filters specified by STATSJBI refer to the same job ID in STATJBIL and STATJBIH (or that STATJBIH is set to zero). Both terse and verbose data are returned. Verbose data is also returned for all valid SYSOUT data sets (chained into the STATJQ). If the job is still executing, STATVOs for data sets that are still open may also be returned. Lastly, terse SYSOUT data is returned. The STATVOs are chained into the STATSEs with which they are associated, as shown in Figure 4-2 on page 59.
- Set STATTRSA to zero and ensure that STATSCTK has STATCTKN set to the SYSOUT token of the SYSOUT group for which you want verbose data. Both terse and verbose job and SYSOUT data are returned (only for the data sets represented by the token passed).
- Set STATTRSA to a STATJQ (obtained previously with no intervening memory management call). Similar to the case in which STATSJBI is set, verbose job data will be chained into the STATJQ, STATVOs are obtained for all valid SYSOUT data sets, and STATSEs are obtained for all SYSOUT groups for the job.
- Set STATTRSA to a STATSE (obtained previously with no intervening memory management call). Similar to the case in which STATSCTK is set, verbose job data is obtained for the job, and all the STATVOs related to the STATSE.

Selection filters

Selection filters must be limited to a single job. This can be accomplished by doing one of the following:

- Selecting STATSJBI and using a jobid range of one job (STATJBIL = STATJBIH)
- Selecting STATJBIH = binary 0
- Supplying a client token in STATCTKN and selecting the filter STATSCTK

Note: In JES2, the application can also point to a terse element and request verbose data for that element.

Fields returned to the caller

Table 4-2 on page 58 shows all returned fields. There is a new section STATSEVB in mapping macro IAZSSST.

Table 4-2 Fields returned in new DSECT STATSEVB

Field name	Meaning
STVSPRCD	Procname for the step
STVSSTPD	Stepname for the step
STVSDDND	DDNAME for the data set
STVSTJN	APPC Transaction jobname
STVSTJID	APPC Transaction jobid
STVSTOD	Date/time data set available
STVSSEGM	Segment ID(JES2 only)
STVSDSKY	Data set number (key)
STVSMLRL	Maximum LRECL
STVSLNCT	Line count
STVSPGCT	Page count
STVSBYCT	Byte count after truncation
STVSRCCT	Record count (JES3 only)
STVSDSN	SYSOUT data set name
STVSCTKN	SYSOUT data set token

4.2.3 New external output structure

The external output structure has been modified. There were two new sections added, which are mapped by macro IAZSSST. The new structure is shown in Figure 4-2 on page 59.

The results of these verbose requests are returned in new segments:

- A job verbose request returns a job verbose section (SJVE) that is chained to the job queue element (SJQE)
- A SYSOUT verbose request returns a SYSOUT Verbose Section (SSVE) that is chained to the SYSOUT data section (SOUT) and to the job queue element (SJQE), and a job verbose section (SJVE) that is chained to the job queue element (SJQE). Several SSVEs can be returned for one SOUT.

Both terse and verbose segments are returned.

Figure 4-2 on page 59 shows the output structure when an application has previously done a terse output request and wants to request information for all the data sets for the job. The application puts the address of the job-level terse into STATTRSA. Upon return from the expansion request, all SOUTs are expanded and their associated SSVEs are chained in. Since a request type of STATOUTV returns both job and SYSOUT verbose information, the SJQE is also expanded and an associated SJVE is chained in.

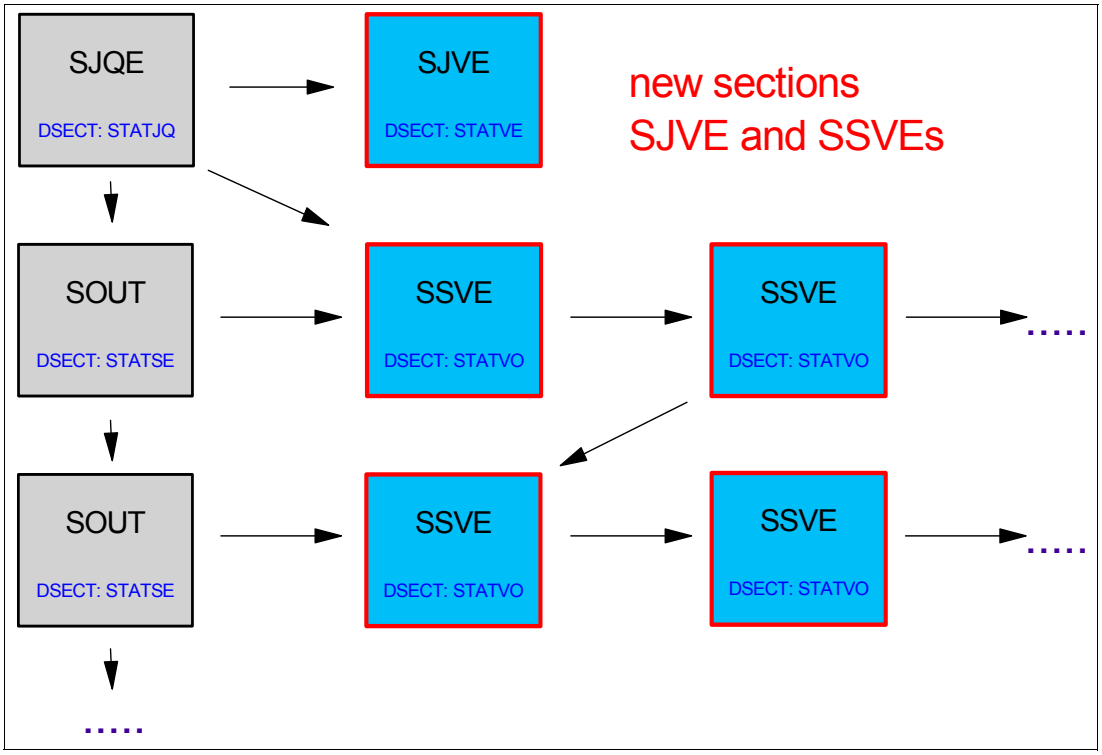


Figure 4-2 New output structure

4.2.4 New status filters

In addition to providing the verbose output sections, z/OS V1R7 provides three new status filters. Filters can be made as follows:

- Forms
- Process mode
- Submitter ID

Note: The submitter ID filter is available only in JES3.

Table 4-3 New status filters

Filter	How it is invoked
SYSOUT FORMS as a filter	Setting STATSSFR flag in field STATSSL2 and specifying SYSOUT FORMS in field STATSFOR.
Submitter ID as a filter	Setting STATSSUB flag in field STATSEL1 and specifying a submitter ID in field STATSUBR. Only in JES3.
Process Mode as a filter	Setting STATSSPR flag in field STATSSL2 and specifying Process Mode in field STATSPRM.

4.2.5 Differences between JES2 and JES3

For the most part, JES2 and JES3 return the same information where it makes sense. Although JES2 and JES3 both group verbose output elements under a terse element, the grouping works differently. Certain fields make sense in JES2 but not in JES3.

JES3 differences

JES3 differences are as follows:

- JES3 grouping is by like output characteristics and residence within the same spool buffer.
- JES3 fills in zero or a reasonable default for the copy count.
- Complex member names are the same as the system name in JES3.
- JES3 provides SYSOUT information for active jobs only when the output in question is spun off and closed. (This is also true of terse output, but is more obvious when verbose output is requested.)
- JES3 is also expanding the JES3-dependent terse job extension which APAR OA06535 started by filling in the spool token (JCT spool address).
- JES3 will now fill in a list of system names and corresponding reason codes indicating why a job is not running on the given system.

JES2 differences

JES2 differences are as follows:

- JES2 grouping is done under an externally named group.
- JES2 copy counts is always filled in as 1.
- JES2 provides SYSOUT information for active jobs.

Functional differences between JES2 and JES3

Table 4-4 shows the functional differences between JES2 and JES3.

Table 4-4 Function differences between JES2 and JES3

Function	How it is invoked	JES2	JES3
Expand terse elements to verbose elements	Setting STATTRSA to STATJQ or to STATSE	Yes	No, but planned in a future release
Submitter ID as a filter	Setting STATSSUB flag in field STATSEL1 and specifying a submitter ID in field STATSUBR	No	Yes

4.2.6 Job verbose call example

Appendix A.2, “Issue a Job Verbose call” on page 479 is an example of how the job verbose call can be issued.

5

JES2 V1R7 enhancements

This chapter describes the changes to JES2 V1R7 with regard to scalability and availability. The following topics are discussed:

- >64K track for JES2 spool
- Internal reader enhancements
- Checkpoint recovery
- JES2 resource display for the health monitor
- Migration from older levels of JES2

5.1 >64K tracks for JES2 spool

Currently, the size of a JES2 spool data set is limited to 64K tracks due to the limitations on the size of a sequential data set. This release of JES2 supports larger volumes and the 64K track limitation is now removed. This solves the problem of new volumes with more capacity that cannot be dedicated entirely to spool. This eliminates the possibility of either wasting space, defining smaller volumes, or sharing the spool volume with other data sets.

DFSMS has been enhanced to support large sequential data sets on a volume. JES2 spool support has been updated in z/OS V1R7 to allow spool data sets to have up to 1M tracks. Parallel Access Volume (PAV) ensures that large spools perform well.

In order to support more track addresses, some changes have been made in the format of spool addressing. Pre-z/OS V1R7 JES2 uses MTTR to represent spool addresses; z/OS V1R7 JES2 uses the new MTTtr addressing format.

5.1.1 New JES2 spool addresses

Spool addressing in previous releases uses the 4-byte MTTR as follows:

M This is the spool extent number.

TT This is the track address. With 16 bits, JES2 has up to 64K for track addresses.

R This is the record number in 8 bits.

In z/OS V1R7, 4 bits from R was given to TT, meaning that now JES2 has 20 bits for track address, or greater than 1 million tracks. The new format MTTtr is described as follows:

M This is the spool extent number, same as in older JES2 releases.

TTt This is the track address. With 20 bits, JES2 has up to 1 million plus track addresses.

r This is the record number in 4 bits.

So an MTTtr of 01234567 is extent 01, track 23456, and record 7. This new addressing scheme will support up to x'FFFFF' or 1,048,575 tracks.

Attention: The MTTr format (as well as the support for large spool data sets) will *not* be supported on volumes that have more than 15 records per track (buffer sizes less than 2943 on a 3390).

Other control block changes

Additional changes have been made in the data areas that track spool usage. The TGR is an area in the checkpoint that tracks what entries are in each systems BLOB. Entries in the TGR were only 3 bytes long (MTT) in pre z/OS V1R7 releases. In addition, allocation IOTs have 3 byte MTTs in them to track how much space a job is using. The 3-byte MTT fields in the IOT and TGR have been increased to a 5 byte MTTTT, also called an MQT.

Note: Back-level members do not support these new MQTs.

5.1.2 Initialization statements

To activate this new function, a new parameter called LARGEDS has been added to the SPOOLDEF initialization statement. LARGEDS specifies whether large spool data sets (greater than 65,535 tracks) can be used or not. It also indicates when to use the new format

spool record addresses (MTTtr). It is MAS scope and the statement is honored on cold starts. A warm start uses the previous values.

LARGEDS specifications

The SPOOLDEF initialization statement defines the JES2 spooling environment and in this release a new keyword is added, as follows:

```
LARGEDS=FAIL | ALLOWED | ALWAYS
```

The LARGEDS options and their meanings are as follows:

FAIL Any attempt to start a spool data set with more than 65,535 tracks is to be failed. The new format spool record address is never used. FAIL is the default and it permits compatibility with older releases of JES2.

ALLOWED Specifies that a spool data set with up to the current size limit (1,048,575 tracks) is to be allowed. New format spool addresses are used only for spool data sets with more than 64 K tracks. The setting of LARGEDS=ALLOWED has the following effects:

- The TGR data area on the checkpoint will be expanded to support 5 bytes MTTTTs (from 3 byte MTTs). The total size goes from 3*255*32=24,480 bytes to 5*255*32=40800 bytes. This change will be reversed if LARGEDS=FAIL is set.
- All new allocation IOTs will be built with 5 byte MTTTTs instead of 3 byte MTTs. A new version field in allocation IOT will indicate it is using 5 byte MTTTTs (version 2), 3 byte MTTs (version 0) or a combination of both (version 1). Setting LARGEDS=FAIL will stop creating version 1 or 2 IOTs, but existing IOTs will still exist on SPOOL.
- Pre-z/OS V1R7 JES2 members can never be warm started again.
- When a new volume is started, it uses the new MTTtr format of SPOOL addresses if it is larger than 64K tracks, and the old MTTR format if not.

ALWAYS Processing is the same as ALLOWED, except the new format spool addresses (MTTtr) are used when using any volume regardless of data set size. This is intended for testing applications that may use spool addresses to access records on spool when an actual large data set is not available.

See *z/OS JES2 Initialization and Tuning Reference,* SA22-7533, for more details about the LARGEDS parameter.

Note: The RELADDR=NEVER|ASNEEDED|ALWAYS parameter is deleted on the SPOOLDEF initialization statement as a spool volume definition.

LARGEDS considerations

Once LARGEDS is set to ALLOWED or ALWAYS, pre-z/OS V1R7 JES2 members cannot join the MAS until a cold start is performed. IBM recommends not setting LARGEDS to ALLOWED or ALWAYS until all MAS members have stabilized on a z/OS V1R7 level of JES2 or later.

Even if LARGEDS is set to ALLOWED or ALWAYS, spool volumes with more than 15 records per track (for example, when using buffer sizes less than 2943 on a 3390) cannot use the new format for spool record addresses, thus they do not support data sets larger than 65535 tracks.

5.1.3 Large spool data set commands

The **$TSPOOLDEF** command can be used to dynamically activate support for large spool data sets and new format spool record addresses. It should be used only once per MAS.

The **$D SPOOL** command can be used to determine if a volume is in large data set format or not by noticing the number of digits displayed for TRKRANGE, as shown in Figure 5-1. Four digits implies old format and eight digits implies new format.

Figure 5-1 shows a display of volume SB0X02, which is still using MTTR, the old address format, and a display of volume SBOXFC, which is using MTTtrs, the new format spool address.

```
$DSPOOL(SBOX02),UNITDATA
$HASP893 VOLUME(SBOX02)
$HASP893 VOLUME(SBOX02)  UNITDATA=(EXTENT=02,
$HASP893                 TRKRANGE=(00A5,1814),RECMAX=12,
$HASP893                 TRKPERCYL=15)

$DSPOOL(SBOXFC),UNITDATA
$HASP893 VOLUME(SBOXFC)
$HASP893 VOLUME(SBOXFC)  UNITDATA=(EXTENT=00,
$HASP893                 TRKRANGE=(00000001,000244F5),
$HASP893                 BASETRAK=0000001D,RECMAX=12,
$HASP893                 TRKPERCYL=15)
```

Figure 5-1 Response from $DSPOOL command.

The **$DSPOOLDEF** command can be used to verify if the spool is enabled to use large spool data sets (LARGEDS = ALLOWED/ALWAYS) or not (LARGEDS=FAIL).

Figure 5-2 shows a display of the **$DSPOOLDEF** command for an image which is not enabled to use large spool data sets (LARGE=FAIL) and another image which is enabled (LARGE=ALLOWED).

```
$DSPOOLDEF
$HASP844 SPOOLDEF  BUFSIZE=3992,
$HASP844           DSNAME=SYS1.JES2.SC63.HASPACE,
$HASP844           FENCE=(ACTIVE=NO,VOLUMES=1),
$HASP844           GCRATE=NORMAL,LASTSVAL=(2003.046,
$HASP844           15:44:39),LARGEDS=FAIL,SPOOLNUM=32,
$HASP844           TGSIZE=30,TGSPACE=(MAX=81440,
$HASP844           DEFINED=52045,ACTIVE=52045,
$HASP844           PERCENT=46.1177,FREE=28043,WARN=80),
$HASP844           TRKCELL=6,VOLUME=SBOX0

$DSPOOLDEF
$HASP844 SPOOLDEF  BUFSIZE=3992,DSNAME=SYS1.JESCPACE,
$HASP844           FENCE=(ACTIVE=NO,VOLUMES=1),
$HASP844           GCRATE=NORMAL,LASTSVAL=(2005.124,
$HASP844           21:33:03),LARGEDS=ALLOWED,
$HASP844           SPOOLNUM=32,TGSIZE=30,
$HASP844           TGSPACE=(MAX=81440,DEFINED=49575,
$HASP844           ACTIVE=49575,PERCENT=0.0161,
$HASP844           FREE=49567,WARN=80),TRKCELL=6,
$HASP844           VOLUME=SBOXF
```

Figure 5-2 Display showing the LARGEDS parameter

5.1.4 New spool messages

Message HASP720, shown in Figure 5-3, has been added and it occurs when a JES2 member pre-z/OS V1R7 tries to warm start on an MAS where the SPOOLDEF initialization statement has been defined with LARGEDS=ALLOWED or ALWAYS.

```
$HASP720 WARM START DENIED - LARGE SPOOL DATA SET FORMAT HAS
BEEN ACTIVATED IN THE MAS.  THIS RELEASE DOES NOT SUPPORT
LARGE SPOOL DATA SET FORMAT.
```

Figure 5-3 Message when a pre JES2 z/OS V1R7 joins a MAS using LARGEDS

5.1.5 Migration and coexistence considerations

The relative track external has been removed in z/OS V1R7 (RELADDR= on SPOOLDEF). All new spool volumes that are started on z/OS V1R7 use relative track addressing (even if it is not required).

To set LARGEDS=ALLOWED or ALWAYS, the checkpoint must be large enough to increase the size of the TGR data area and *all* members must be at the z/OS V1R7 level or higher.

Attention: Back-level systems will *not* start if LARGEDS=ALLOWED or ALWAYS was *ever* specified. This is because allocation IOTs with MTTTTs exist on spool and pre-z/OS V1R7 systems cannot process these IOTs.

Using user exits

Exits or applications that read or write the checkpoint directly may be impacted by the changes in this support. In particular, an application that converts the MTTR to BBCCHHR for use in CCWs needs to be updated. Checking code in HASPNUC can help you convert any code you may have that does this. The routine to check in HASPNUC is $EXCP.

5.1.6 Implementing large spool volume support

The following steps are the preferred way to migrate to large spool data sets. This procedure minimizes the risk to the system and provides a reasonable backout plan:

1. On a test system, test applications that access spool by setting LARGEDS=ALWAYS and starting a spool volume.
2. Migrate to z/OS V1R7 JES2 on *all* MAS members.
3. Wait for z/OS V1R7 JES2 to stabilize (no need to fall back).
4. Change JES2 parmlib in SPOOLDEF statement and include LARGEDS=ALLOWED.
5. Alter SPOOLDEF dynamically using the **$TSPOOLDEF,LARGEDS=ALLOWED** command. Figure 5-4 shows the **$TSPOOLDEF** command and its results.

```
$TSPOOLDEF,LARGEDS=ALLOWED
$HASP844 SPOOLDEF  BUFSIZE=3992,DSNAME=SYS1.JESCPACE,
$HASP844           FENCE=(ACTIVE=NO,VOLUMES=2),
$HASP844           GCRATE=NORMAL,LASTSVAL=(2005.125,
$HASP844           12:46:27),LARGEDS=ALLOWED,SPOOLNUM=32,
$HASP844           TGSIZE=30,TGSPACE=(MAX=81440,
$HASP844           DEFINED=49575,ACTIVE=49575,
$HASP844           PERCENT=0.0161,FREE=49567,WARN=80),
$HASP844           TRKCELL=6,VOLUME=SBOXF
```

Figure 5-4 LARGEDS=ALLOWED activation

6. Stabilize with the new format of data areas.

   As soon as JES2 starts to use LARGEDS=ALLOWED (or ALWAYS) and a job with 3 byte MTT in the allocation IOT (version 0 IOT) needs spool space from that volume, the allocation IOT will be converted to a version 1 IOT and all new TGBs will be in MTTTT format.

   So, it is recommended that installations first set LARGEDS=ALLOWED and allow the system to run in that state for some time before a large spool data set is actually started.

   This allows many of the old format IOTs to be processed before the new format spool addresses are used.

   You can consider using spool affinity to limit jobs using new spool.

7. Start a large spool volume.

   Figure 5-5 shows an example to create a large spool data set.

```
//ALLOCJ2 JOB (999,POK),'CLAUDIA TOMAS',CLASS=A,MSGCLASS=T,
// NOTIFY=TOMAS,TIME=1440,REGION=4M
//*-------------------------------------------------------------------*
//* DOC: CREATE LARGE SPOOL DATA SET                                  -*
//*-------------------------------------------------------------------*
//ALLOC1   EXEC PGM=IEFBR14
//SPOOL1    DD DISP=(,KEEP),DSN=SYS1.JESCPACE,UNIT=3390,
//          VOL=SER=SBOXFC,SPACE=(CYL,(10015,0,0)),DSNTYPE=LARGE
```

Figure 5-5 JCL model to create a large spool data set

8. Once stabilized, drain old spool volumes and migrate to new larger spool data sets.

   Clear spool affinities if it have been used for testing earlier.

5.1.7 Fallback

Once LARGEDS is set to ALLOWED or ALWAYS, JES2 starts creating spool control blocks that back-level members do not know how to process. Because of this, once LARGEDS is set to ALLOWED or ALWAYS, back-level JES2s cannot ever join this MAS (unless a cold start is done).

5.2 Internal reader enhancements

The INTRDR RDINUM= initialization statement is no longer needed. Internal reader processing now occurs in the allocating address space, and there is no need to pre-allocate data areas.

The **$D RDI** command is retained to display information on allocated internal readers. In addition, a **$T RDI** command is allowed to activate tracing for an allocated internal reader.

The **$D PCE** command is not needed since there are no internal reader PCEs.

5.3 Checkpoint enhancements

With the JES2 checkpoint, any corruption of the JES2 checkpoint could potentially cause a severe outage (possibly even a cold start). Over the years, code has been added to deal with corruption of the JQEs and JOEs and to recover as much as possible. Additional validation has been added in z/OS V1R7 to detect and correct problems with the DAS CTENT, minimizing the amount of damage that can occur if this area is corrupted.

The improvements have been made in z/OS V1R7 logic so JES2 can be able to recover from a checkpoint corruption without impacting the system. Additional changes have been made in order to avoid checkpoint data corruption such as attempting to use the same spool volume configuration in more than one MAS in a sysplex, which can cause a multi-system outage.

5.3.1 Checkpoint recovery

In order to control its data, checkpoint has control blocks for each spool volume. These control blocks are called Direct Access Spool Data Set ($DAS). In previous releases if an installation started JES2 with part or all of the $DAS unknowingly corrupted, the corresponding spool volumes would be unusable and data would be lost. A cold start could be necessary to make JES2 operational.

In z/OS V1R7, enhancements have been made to avoid a failure caused by overlays on $DAS control block. Now, on any warm or hot start, each $DAS and appropriate chains are verified for potential errors. If errors are found, the $DAS and its chains will be corrected using saved data (stored in a new control block called $RECY).

If a rebuild was necessary, the operator will be notified via the new $HASP896 message.

In IPCS, new function has been added to do $DAS versus $RECY analysis when formatting either the $DAS or $RECY. An error message will be displayed when a comparison is not equal, showing the data being compared.

5.3.2 Preventing checkpoint corruption

Problems have occurred over the years where an installation tries to use the same spool volume configuration for more than one MAS in a sysplex. In previous releases, the second MAS using the spool volume configuration could possibly write over the original MAS's in-use data, leading to multi-system outage.

Whenever a spool volume is allocated, an ENQ will be obtained. If another MAS has already successfully allocated the spool volume (meaning it has obtained the ENQ), then the $HASP443 message will be issued with a new reason code (RC=09, DSNAME AND VOLUME ALREADY OWNED BY ANOTHER MAS).

If an installation tries to use the same spool volume configuration for more than one MAS in a sysplex, the second MAS using the spool volume configuration could possibly write over the original MAS's in-use data.

5.3.3 New and changed messages

Message HASP896 has been added to inform the operator that JES2 has detected an error on $DAS structures and that a rebuild has been done to fix it.

Message HASP443 has been changed to inform the operator when a spool volume could not be started because it is in use by another MAS in the sysplex.

5.4 JES2 health monitor display

The current JES2 health monitor (or the JES2 monitor) only has an MVS command interface. Since it can produce hundreds of lines of output for a single command, it is not the best way to gather data to study to look for problems. This is especially true of the history information it maintains. What is needed is an API to pass the data to a GUI for a better presentation.

The job information SSI (function code 71) is expanded in this release to allow the return of JES2 monitor information to an application. Information returned is the same as what can be obtained via monitor commands. There are some additional fields that are returned to complete the data that is available to applications.

5.5 Migrations to JES2 V1R7

Consider the following migrations to JES2 V1R7 from an older release of JES2:

- From JES2 OS/390 V2R10 or earlier:
  - Migrate to more recent spool-compatible release first (z/OS V1R5) to avoid cold start.
  - $ACTIVATE,LEVEL=z2 on that release; there is no $ACTIVATE support in z/OS V1R7.
- From JES2 z/OS V1R2:
  - $ACTIVATE,LEVEL=z2 is required to avoid cold start.
  - No MAS coexistence (all member warm start).
- From JES2 z/OS V1R4 and V1R5:
  - $ACTIVATE,LEVEL=z2 is required to coexist with z/OS V1R7.
  - APAR OA08145 needed on all members; it includes toleration for:
    - Long SYSIN records
    - Local node name changes
    - Persistent NJE connections
    - $HASP549 message changes
  - APARs OA11953 and OA12472 fix errors in APAR OA08145

5.5.1 Installing JES2 V1R7

Some significant changes have been made to the packaging of JES2. The the load library is now a PDSE. This was done to take advantage of the RMODE(SPLIT) binder option to load

most of the HASJES20 load module above the 16M line. The new library is SYS1.SHASLNKE.

New JES2 distribution library

The distribution library is moved from SYS1.SHASLINK (PDS) to SYS1.SHASLNKE (PDSE). This allows HASJES20 to be a split load module. Most JES2 modules are now loaded above the 16M line. A few modules had to stay below16M.

Assembling JES2 modules

If you do your own assembly and linkedit of JES2, JES2 modules should be assembled with the XOBJECT and LIST(133) parameters. JES2 should be linkedited into a PDSE with the RMODE(SPLIT) option.

6

WLM enhancements

In this chapter we describe WLM enhancements introduced in z/OS V1R7. The following improvements are discussed:

- WLM and RMF changes for subcapacity pricing support. This change may result in lower software bills for subcapacity customers that run with z/OS guests.
- Dynamic processor speed changes.
- Server-specific WLM load balancing services.

6.1 WLM enhancements for subcapacity pricing support

Subcapacity pricing support consists of configuring your system so that a subcapacity-eligible product runs on less than the entire capacity of CPC. z/Series customers will be billed accordingly for that product. The capacity of a central processor complex (CPC) is represented in terms of millions of service units (MSUs). Subcapacity pricing is determined by the LPAR utilization capacity associated with the logical partitions (LPARs) in which the subcapacity-eligible product runs.

Note: To be eligible for subcapacity pricing the installation must have z/OS systems running in LPAR mode.

6.1.1 Subcapacity Reporting Tool (SCRT)

The Subcapacity Reporting Tool (SCRT) is a stand-alone reporting tool that allows qualifying zSeries customers to take advantage of subcapacity pricing. SCRT post-processes selected fields in SMF type 70 and 89 records in order to produce an SCRT report that customers eligible for subcapacity pricing are required to submit to IBM on a monthly basis.

The report shows the peak rolling four-hour average value (in MSUs) of each subcapacity-eligible product running on a zSeries CPC. Subcapacity products are charged based on the rolling four hour average utilization of the LPARs in which the subcapacity products execute. The subcapacity report examines, for each hour in the reporting period:

- The rolling four-hour average utilization, by LPAR.
- Which eligible products were active in each LPAR.

SCRT then cross-references LPAR utilization and product execution by LPAR to determine the maximum concurrent LPAR rolling four-hour average utilization — the highest combined utilization of LPARs where each product executes during the reporting period.

SCRT Web site

SCRT is available as a download from the zSeries pricing Web site:

`http://www.ibm.com./zseries/swprice`

This Web site contains SCRT code and the manual - *Using the Subcapacity Report Tool SCRT*, SG24-6522.

SCRT gathers information from SMF registers that have been written by WLM and RMF and produces a report with the four hour rolling average. Any other program that produces a similar report, not only SCRT, will take advantage of the WLM and RMF Subcapacity Pricing support enhancements:

- z/OS guests
- Dedicated and shared CPs with WAIT COMPLETION=YES

6.1.2 z/OS guests and SCRT

Until z/OS V1R6, z/OS systems running as a z/OS guest (under a z/VM system, for example) had no support for the four hour rolling average. Product MSUs for a subcapacity product were based on the maximum capacity of the LPAR in which the z/OS guest ran.

WLM and RMF code has been changed in z/OS V1R7 to support z/VM guests. Now, the SCRT tool or any other similar program is able to produce reports with actual four hour rolling averages for subcapacity products running under a z/OS guest.

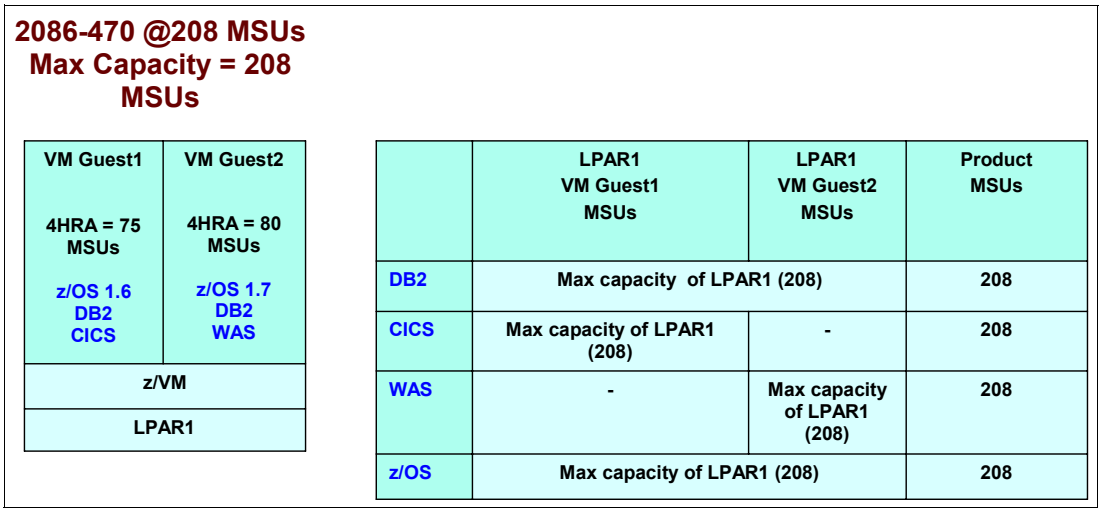
This change can result in lower software bills for subcapacity customers that run with z/OS guests since they will pay based on the *actual peak* rolling four-hour average value, rather than on the *maximum capacity* of the LPAR in which the z/OS guest runs.

Attention: *All* z/OS guest systems on a given machine *must* be running z/OS V1R7 at all times during the reporting period to obtain the benefits of the z/OS guest support.

6.1.3 Sample configuration with mixed z/OS guests

Figure 6-1 shows a configuration in which there is a single LPAR (LPAR1) with two z/OS guests (z/VM Guest1 and z/VM Guest2), with a mix of z/OS levels running in these z/VM guests (z/VM Guest1 is running z/OS V1R6 and z/VM Guest2 is running z/OS V1R7).

In this example, the installation will not be able to obtain the benefits of z/OS guest support because there is at least one z/OS guest running a pre z/OS V1R7 level (z/VM Guest1). The product MSUs for the products running in these z/OS guests are based on the maximum capacity of LPAR1, in which both z/VM Guest1 and z/VM Guest2 ran, which is 208 MSUs.



	LPAR1 VM Guest1 MSUs	LPAR1 VM Guest2 MSUs	Product MSUs
DB2	Max capacity of LPAR1 (208)		208
CICS	Max capacity of LPAR1 (208)	-	208
WAS	-	Max capacity of LPAR1 (208)	208
z/OS	Max capacity of LPAR1 (208)		208

Figure 6-1 Single LPAR with mix of z/OS Guests at z/OS V1R6 and z/OS V1R7

6.1.4 Sample configuration with all z/OS V1R7 guests

Figure 6-2 on page 74 shows a configuration of a single LPAR with *all* z/OS guests at the V1R7 level. This example shows the results with the new WLM/RMF and SCRT z/OS guests enhancements.

This example shows that the product MSUs for the subcapacity products DB2, CICS, WAS, and z/OS are based on LPAR four hour rolling average (4HRA) utilization rather than on maximum capacity of the LPAR in which the z/OS guest ran.

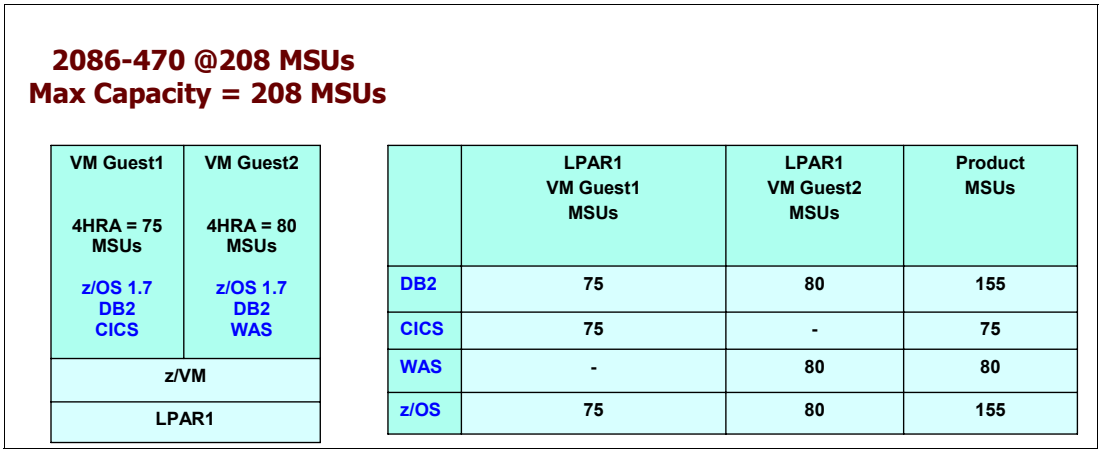
The product MSUs for DB2 and z/OS is 155 MSUs, which is the sum of:

- 4HRA of 75 MSUs, since DB2 and z/OS ran in z/VM Guest1.
- 4HRA of 80 MSUs, since DB2 and z/OS ran in z/VM Guest2.

CICS ran only in z/VM Guest, therefore the product MSUs for CICS is based on the 4HRA for z/VM Guest1 or 75 MSUs.

WAS ran only on z/VM Guest 2, therefore the product MSUs for WAS is based on the 4HRA for z/VM Guest2 or 80 MSUs.

As we can see, in the first case (Figure 6-1 on page 73) the site would have to pay for 832 MSUs and in the second case (Figure 6-2) the site would have to pay for 465 MSUs.



	LPAR1 VM Guest1 MSUs	LPAR1 VM Guest2 MSUs	Product MSUs
DB2	75	80	155
CICS	75	-	75
WAS	-	80	80
z/OS	75	80	155

Figure 6-2 Single LPAR with ALL z/OS Guests z/OS V1R7

6.1.5 CPs and wait completion=yes

Pre-z/OS V1R7 releases calculate the four hour rolling average using the total dispatch time of all logical CPUs of the running LPAR. The dispatch time depends on the LPAR configuration as follows:

- Dedicated CPUs.

  The dispatch time includes the total wait time for all logical CPUs of the LPAR.
- Shared CPUs with wait completion = YES.

  The dispatch time includes the total wait time for all logical CPUs of the LPAR.
- Shared CPUs with wait completion = NO.

  The dispatch time includes no wait time for all logical CPUs of the LPAR.

Now, z/OS V1R7 provides a four hour rolling average that reflects the actual CPU time consumed without the wait time, independent of the LPAR configuration.

Note: This enhancement, however, is transparent to SCRT users as SCRT tool V5.1 and V6.1 have already provided this support into their own code. z/OS system code has been changed just to properly place processing in WLM and RMF components.

6.1.6 Dependencies

The following conditions are prerequisites to obtain the benefits of the z/OS guest support:

- All z/OS guest systems on a given machine must be running z/OS V1R7 at all times during the reporting period.
- The installation must use SCRT V11 or higher.

6.1.7 Migration and coexistence

Considerations for controlling capping if defined capacity is used by an installation is as follows:

- For shared CPs with WAIT COMPLETION = YES, the defined capacity control is now based on the MVS busy time and is no longer based on the dispatch time.
- There are no considerations for configurations that have dedicated CPs or shared CPs with WAIT COMPLETION = NO, since defined capacity controls are not available for dedicated CPs and no WLM or RMF changes were required to remove wait time for systems running with WAIT COMPLETION = NO.

Using defined capacity

Installations that make use of defined capacity and have configurations involving shared CPs with WAIT COMPLETION = YES should consider making adjustments to their defined capacity used since defined capacity is now based on MVS busy time and is no longer based on dispatch time.

If your installation uses any program that accesses WLM control structures to produce a report like SCRT does, then you must be aware of the changes described in Table 6-1 and alter your programs as appropriate.

Table 6-1 Control structures and their fields that have been changed in z/OS V1R7.

Control structure	Field	Modification
IRARCT	RCTLACS	No longer includes CPU wait time for systems running in LPAR mode.
IRARCT	RCTFLAG1_LACS_RCU	This field is always set on as of z/OS V1R7, meaning that RCTLACS no longer includes wait time and provides four hour rolling average for z/OS guests run in LPAR mode.
IRALPDAT	LPDatAvgImgService	This field is a copy of RCTLACS and no longer includes CPU wait time for systems running in LPAR mode, as well.
IRALPDAT	LPDatVer	A value of 2 as of z/OS V1R7 indicates the design changes: The LPDatDefCapData section in IRALPDAT is now available for z/OS guests run in LPAR mode, which contains LPDatAvgImgService. CPU wait time.

SMF changes

To maintain consistency, SMF record type 70 has been changed also. These are the modifications on SMF record type 70 subtype:

- SMF70LAC no longer includes CPU wait time for systems running in LPAR mode.
- SMF70RCU, which is bit 3 of byte SMF70STF, is always set on as of z/OS V1R7. This means that SMF70LAC no longer includes CPU wait time and it now provides 4 hour rolling average for z/OS guests run in LPAR mode.

For more information about SMF record type 70 changes, refer to *z/OS MVS System Management Facilities*, SA22-7630.

6.2 Dynamic processor speed changes

Prior to z/OS V1R7, the performance of a system could be changed by changing the number of processors. Now, z/OS V1R7 provides a new way of dynamically changing the performance of a system by changing the speed of the processors. This is called support of dynamic processor speed changes.

6.2.1 Hardware dependencies

The hardware currently supporting this new function is the z890 processor. Support of dynamic processor speed changes is part of z/OS V1R7 code. For z/OS V1R4 and above this function is provided via APAR OA07510.

6.2.2 Migration and coexistence

New support should be installed on *all* z/OS LPARs in order to have a *dynamic* processor speed change. Otherwise, IPLing the system will set the proper timing and performance values as well.

6.2.3 Implementing dynamic speed changes

z/OS does not have a user interface to support processor speed changes. All the installation needs is to apply a new LICCC on the support element.

As soon as the new processor speed is activated, the hardware signals this event to the operating system where the supervisor passes this event to SRM. SRM then checks whether the processor speed value has changed. If so, then all timing and performance related numbers are adjusted for the new speed and the WLM policy is reactivated.

Once the policy is reactivated a new message is issued, as follows:

```
IWM063I WLM POLICY WAS REFRESHED DUE TO A PROCESSOR SPEED CHANGE
```

Changing the processor speed can be done while the system is running and no IPL is required. In order to make this happen the hardware and microcode must support this. Currently, only the z890 supports dynamic speed changes.

Note: Processor speed changes are only done for regular 390 processors. Other types of processors like zAAPs are not affected and will always run with full speed.

z/VM 5.1 supports dynamic support processors speed changes and passes this event to the guest operating system. SRM does not distinguish if a processor speed change happened on native hardware or under z/VM. The same code is executed.

6.3 Server-specific WLM load balancing

Workload balancing using pre-z/OS V1R7 routing services took only the behavior of the whole system into account.

Server-specific WLM load balancing provides a more granular view of how well a server is doing in a sysplex (as observed by WLM) in making its load balancing recommendations by taking the Service Level Agreement (SLA) in the WLM policy into account.

6.3.1 WLM routing services pre-z/OS V1R7

The *sysplex routing services* allow work associated with a server to be distributed across a sysplex. They are intended for use by clients and servers.

A client is any application or product in the network that requires a service. The service could be a request for data, a program to be run, or access to a database or application.

In terms of the sysplex routing services, a client is any program routing work to a server. A server is any subsystem address space that provides a service on a MVS image.

The sysplex routing services provide two main functions:

- IWMSRSRG macro lets a caller register as a server.
- IWMSRSRS macro provides a caller with a list of registered servers and the number of requests that should be routed to each server.

Assume, for example, that you have a Web application. On each system of your sysplex you have one Web server that is able to handle the incoming requests to one externalized IP address. Your routing manager wants to distribute the incoming requests among the Web servers in a balanced way. That means the percentage of the requests being sent to one Web server should be equivalent to its ability to handle those requests at the given point of time – taking into account the system utilization and the servers behavior.

The WLM routing services give you a recommendation for this distribution by returning a weight for each server that you have registered for routing.

The Sysplex Distributor of Communication Server, DB2, and other router programs exploit these services.

Figure 6-3 shows an example of sysplex routing.

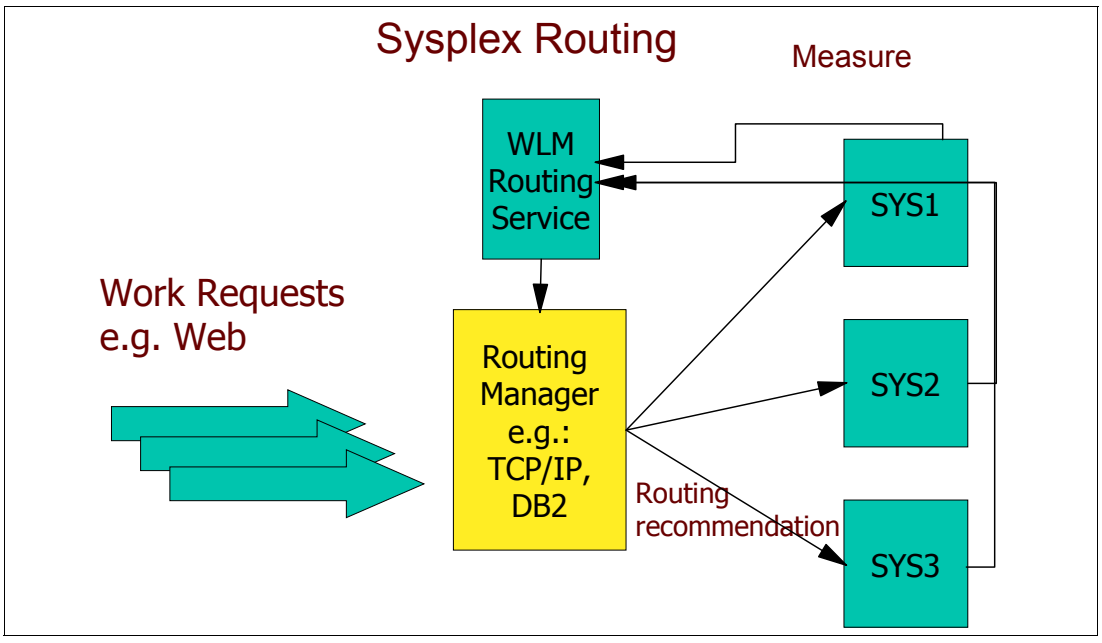


Figure 6-3 Sysplex routing overview

IWMSRSRG macro lets a caller register as a server

Before being able to use the service for routing recommendations, IWMSRSRS, you have to register the servers through the IWMSRSRG service. You group servers during registration by associating them with an identifier called LOCATION.

Figure 6-4 gives an example of how the servers are registered using IWMSRSRG. The list of servers registered on the different systems of a sysplex is communicated between the systems every 10 seconds.

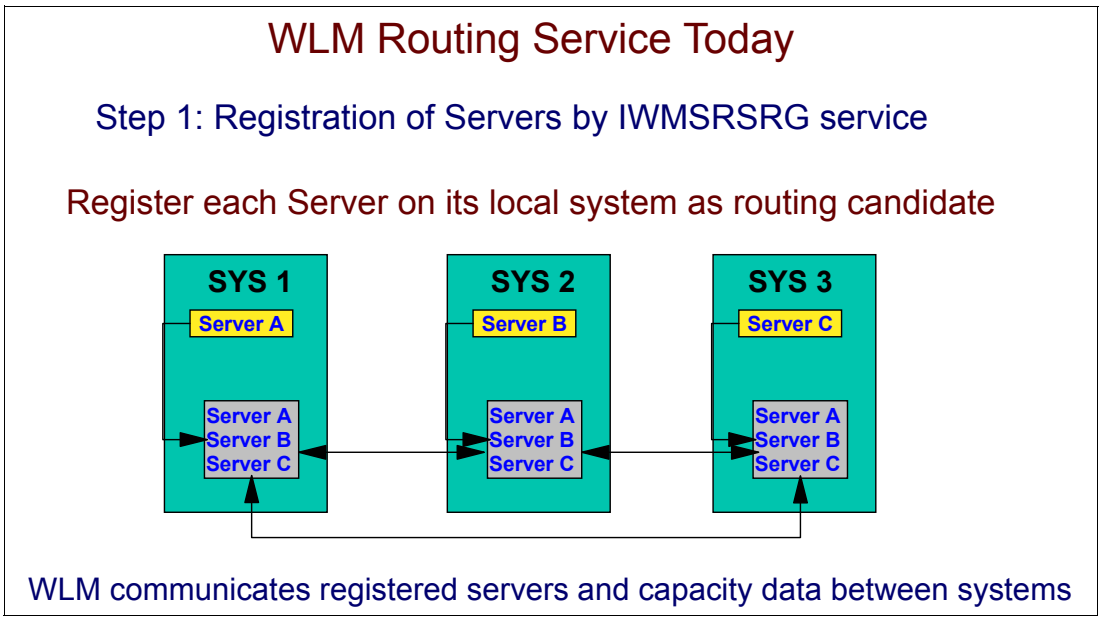


Figure 6-4 Registration of servers by IWMSRSRG macro

Ask WLM for routing recommendations by IWMSRSRS service

To get a routing recommendation, you can call the IWMSRSRS service with the LOCATION parameter on any system in the sysplex.

It returns recommendations, called weights, which are numbers between 1 and 64 for each server that was registered under the LOCATION ID.

Then you would use those weights to distribute the incoming requests among the servers according to the size of their weights.

Figure 6-5 on page 79 gives an idea of how it works.

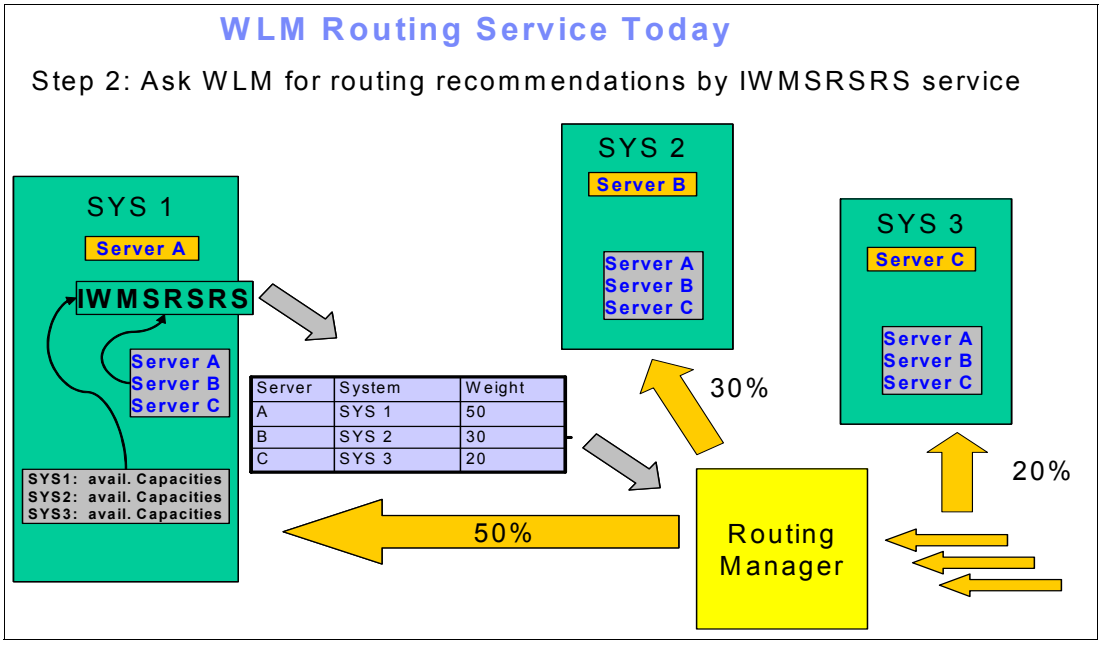


Figure 6-5 IWMSRSRS provides a caller with a list of registered servers and the number of requests that should be routed to each server

How to calculate the weights for the servers

WLM selects an importance level that consumes more than 5% of system CPU capacity at least at one system (in Service Units - SUs).

Then calculate the system weight for each system using the following formula:

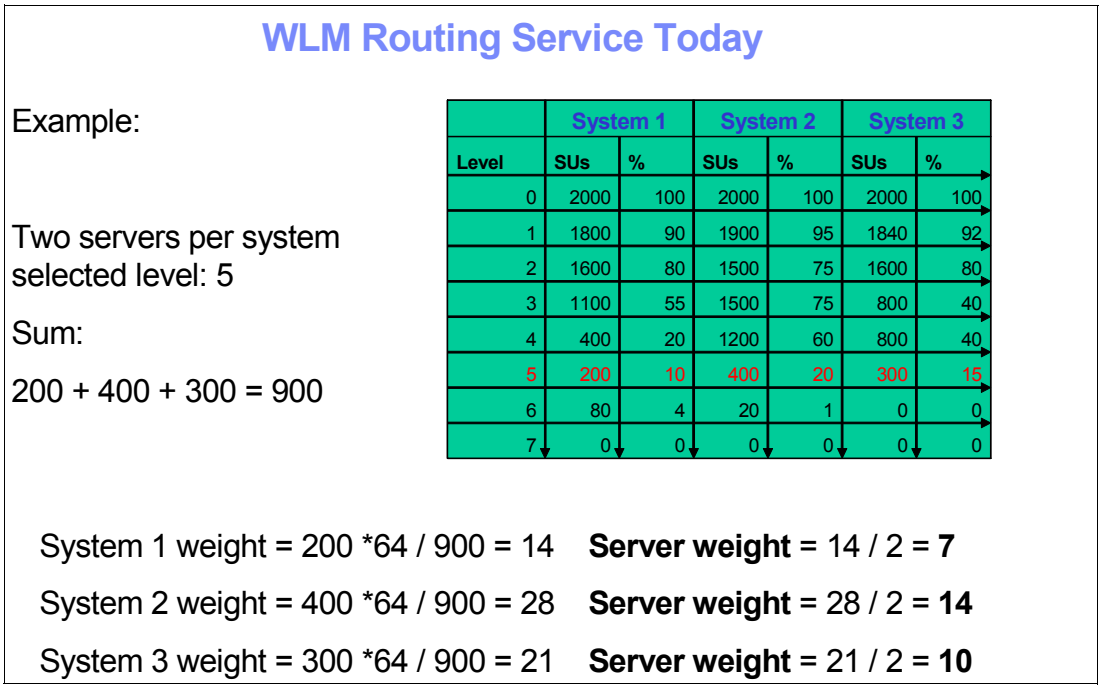
System weight = (SUs for this system at selected level * 64) / total SUs for all systems at selected level

The importance level corresponds to the importance values between 1 and 5 that each workload is associated with through the WLM classification with the active WLM service policy.

Additionally, we have these importance levels:

- 0 for system related work
- 6 for discretionary work
- 7 for free, unused capacity

Calculate server weights by dividing system weight by the number of servers on system. Figure 6-6 on page 80 shows how the weights are calculated by WLM.



	System 1		System 2		System 3	
Level	SUs	%	SUs	%	SUs	%
0	2000	100	2000	100	2000	100
1	1800	90	1900	95	1840	92
2	1600	80	1500	75	1600	80
3	1100	55	1500	75	800	40
4	400	20	1200	60	800	40
5	200	10	400	20	300	15
6	80	4	20	1	0	0
7	0	0	0	0	0	0

Figure 6-6 Example of how to calculate the server weights

To assign weights to the three systems, WLM scans the CPU consumption table from the bottom up, looking for a level where at least one system has 5% or greater cumulative CPU consumption. In this case, the level 5 row is used.

The sum of all SUs used by importance level 5 or lower (called *displaceable capacity*) is 200+400+300 = 900.

By dividing each system's displaceable capacity of level 5 by this sum, we get the system weights.

The resulting server weights are the result of dividing each system weight by the number of registered servers on that system.

Weakness of WLM routing service pre z/OS V1R7

The problem of pre-z/OS V1R7 WLM routing service is that the routing recommendation, the weight, for a server is only based on the available capacity of the system (LPAR). It does not take the specific information about the performance-related behavior of the servers itself into account.

This could lead to unexpected results. For example, consider a server running on a system that is fully loaded with low importance work, but the server has high importance. In this case the server would be able to displace other work and handle more requests than the routing recommendation specified to send there.

It might also happen, that the server runs on a low utilized system, but does not have enough needed resources to run much work. (For instance, a DB2 server could run out of DBATs.) But due to the low system load it is recommended to send much more work to this server.

6.3.2 New WLM routing services

A new routing service IWM4SRSC and a new function for the old routing service IWMSRSRS have been implemented in z/OS V1R7 in order to solve the routing services problems detected in previous releases. These new changes are both system and server specific.

The new function SPECIFIC for the IWMSRSRS service is called in the same way as the old functions SELECT and QUERY. It requires having the servers registered in advance (through the registration service IWMSRSRG). It can be called on any system and takes all registered servers on all systems into account. DB2 is the first exploiter of this new function.

The other new service, IWM4SRSC, does not require registration of the servers. But it can only return a recommendation for one server per call, so this routine has to be called for each server that a recommendation is wanted for, on the system that server is running on. Communication Server is the first exploiter of this new function (Sysplex Distributor for TCP/IP).

New routing service IWM4SRSC

IWM4SRSC has the STOKEN parameter to identify the server address space that the callers want information about. This new service gives the weight parameter as output. The weight parameter is a number between 1 and 64; it indicates the relative performance behavior of the server so that it can be compared to other servers.

The weight calculation takes into account two factors, scaled by 64:

- Performance Indicator (PI) factor

  PI gives an indication of how well this server, respective of the work that is related to this server, is achieving its goals as defined in the active WLM policy.

- The importance factor

  This is a measurement of how much CPU capacity is displaceable by work of the server‘s importance, respective the work that is related to this server.

New function code SPECIFIC in routing service IWMSRSRS

The weight calculation for the new function code, SPECIFIC, in routing service is a product of three factors:

- System utilization factor:

  This is the same as the resulting system weight for the old SELECT function in pre z/OS V1R7 releases.

- PI factor:

  This is calculated in the same way as the PI factor of the IWS4SRSC.

- Queue time ratio:

  If the server owns independent enclaves, this is the ratio of queue time to elapsed time of those enclaves.

  The queue time is the time between the time given in the ARRIVALTIME parameter of the enclave create service and the start time of the enclave.

  The queue times and the total elapsed times of the enclaves owned by the server are stored. The average ratio of the queue times versus the elapsed times are calculated to get the queue time ratio of the server.

Note: If two or more servers are registered on the same system, the weight is divided by the number of those servers. This is the same as for the old SELECT function for that service, to avoid overloading a system that has many servers.

Migration and coexistence considerations

Following are some migration and coexistence considerations to use the new WLM routing services:

- IWM4SRSC can be used on any system running z/OS V1R7.
- IWMSRSRS with the function SPECIFIC can be called when all servers registered under the given LOCATION run on systems with z/OS V1R7.

  The services can be called by assembler macros by any application program.
- Current exploiters are Communication Server for z/OS V1R7 and DB2 V.8.
- The old functions of the IWMSRSRS routing service still can be used and are not modified. The old functions are `Function = SELECT` and `Function = QUERY`.

Usage and invocation

New routing services of WLM can be invoked by:

- Macro services
- IWM4SRSC, IWMSRSRS with function=SPECIFIC

Figure 6-7 shows some examples of how these new routing services are invoked.

```
IWM4SRSC STOKEN=STKN,
   WEIGHT=WGHT,
   RETCODE=RC,RSNCODE=RSN

IWMSRSRS SYSINFO_BLOCK=DATA,EXTENDED_DATA=YES,
   ANSLEN=SIZE,ENTRY_COUNT=E,QUERYLEN=Q,LOCATION=LOC,
   FUNCTION=SPECIFIC,RETCODE=RC,RSNCODE=RSN
```

Figure 6-7 Examples of how to invoke the new routing services.

See z/OS *MVS Programming: Workload Management Services*, SA22-7619, for more details about the new routings services.

7

SMP/E V3R4 enhancements

This chapter provides an overview of the SMP/E V3R4 enhancements that were introduced in z/OS V1R7, in particular the following:

- Service acquisition and Internet delivery
- SMP/E **RECEIVE** command extensions
- ORDER management panel
- New command generation panel
- Configuration and setup

7.1 Service acquisition and Internet delivery

Prior to z/OS V1R7, ShopzSeries was the standard application for ordering PTF service for the z/OS platform, with an option for Internet delivery. In ShopzSeries there are many individual tasks to perform, with several opportunities for error. In addition, the process is a series of manual steps with little opportunity for automation of these steps.

With SMP/E V3R4, a single step provides service ordering and delivery capability that can be used as needed for PTF requests. In addition, using a job scheduler, you can automate service ordering and delivery on whatever frequency is desired, thus providing a self-service subscription capability.

To make this work, you must have the following:

- Access to IBM's service repository.

  The local z/OS system must be able to access a remote server that is connected to IBM's PTF service repository.
- A user identity and certificate provided by ShopzSeries.

You must specify the order type as follows:

- Corrective service
  - PTFs specified by name.
  - PTFs to resolve specified APARs.
- Preventive service
  - PTFs to resolve HIPER problems or PTFs in error (PE).
  - Recommended PTFs, that is, PTFs identified with an RSUnnnn sourceid, and PTFs to resolve HIPERs and PEs. The software inventory file determines what PTFs are applicable to the local z/OS system environment and which are already applied.

7.1.1 Order transaction overview

The Automated Delivery Server is the remote IBM server that manages access to IBM's service repository. To access the server, users must register and obtain a user identity. An x.509 client certificate will be generated at ShopzSeries and will contain unique identifying information. This client certificate must be stored on z/OS in a security product database like z/OS Security Server RACF. SMP/E will read the certificate from the security product database and use it to gain access to the IBM Automated Delivery Server.

The flow for a typical SMP/E order transaction is as follows:

- Build a software inventory.
- Extract the named client certificate from the specified keyring.
- Submit the order request to the server. Client/server communication uses HTTP 1.0 protocol with SSL.
- The server accepts the order request.
- SMP/E polls the server periodically for status of the order.
- When the order is fulfilled the server responds with package download information: FTP server, uid, pw, package SHA-1 hash.
- SMP/E uses existing FROMNETWORK infrastructure to automatically download (pull) the package into the SMPNTS using FTP.

- The contents of the package are expanded and received into the global zone and SMPPTS.

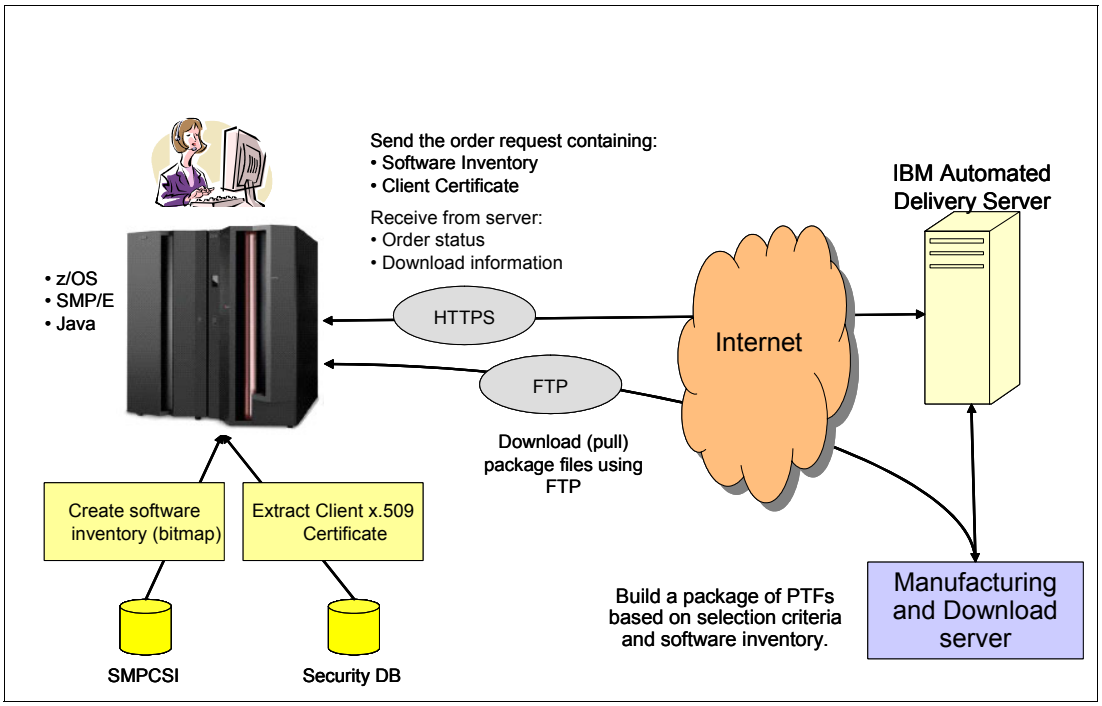


Figure 7-1 SMP/E automated order and delivery through Internet

7.2 SMP/E RECEIVE command extensions

To provide SMP/E with the ability to put orders with selection criteria, and download the PTFs, the **RECEIVE** command was extended, as follows:

ORDER Indicates an order is to be processed.

- **CONTENT** Indicates the desired PTF or HOLDDATA content for the order.
  - **ALL** All available PTFs that are applicable to the specified target zones.
  - **PTFS** Specifies one or more PTFs to be ordered.
  - **APARS** Specifies one or more APARs.
  - **RECOMMENDED** All available PTFs identified with an RSU sourceid, or resolve HIPER or PE problems.
  - **CRITICAL** All available PTFs that resolve HIPER or PE problems.
  - **HOLDDATA** Only HOLDDATA is to be ordered.
- **FORTGTZONES** Defines the scope for the order, which target zones will be used for the software inventory.
- **WAIT** How long SMP/E should wait for the order to be ready for download.
- **TRANSFERONLY** Receive processing should stop after the package files have been downloaded into the SMPNTS directory.

ORDERSERVER The DD name of the ORDERSERVER data set, which contains the information necessary for **RECEIVE ORDER** command processing to identify the IBM Automated Delivery Server as well as the client certificate to be used, as shown in Figure 7-3.

```
SET  BOUNDARY(GLOBAL).
RECEIVE ORDER(
               CONTENT(
                       ALL |
                       APARS(sysmodid list) |
                       PTFS(sysmodid list) |
                       RECOMMENDED |
                       CRITICAL |
                       HOLDDATA
                      )
               FORTGTZONES(zone list)
               WAIT(minutes | NOLIMIT)
               TRANSFERONLY
               CLIENT(DDCLIENT)
               ORDERSERVER(DDSERVER)
             ).
```

Figure 7-2 Extensions to RECEIVE command

ORDERSERVER parameters

```
//DDSERVER  DD *
  <ORDERSERVER
     url="server url"
     keyring="keyring name"
     certificate="certificate label" >
  </ORDERSERVER>
/*
```

Figure 7-3 ORDERSERVER parameters

CLIENT The DD name of the CLIENT data set, which contains information to describe the local z/OS system environment (Figure 7-4).

```
//DDCLIENT  DD *
  <CLIENT debug="NO" retry="n"
          classpath="path"
          javadebugoptions="options" >
     <FIREWALL>
        <SERVER host="name" port="number"
                user="userid" pw="password">
        </SERVER>
        <FIRECMD> command
        </FIRECMD>
     </FIREWALL>
     <HTTPPROXY host="name" port="number"
                user="userid" pw="password">
     </HTTPPROXY>
     <HTTPSOCKSPROXY host="name" port="number"
                     user="userid" pw="password">
     </HTTPSOCKSPROXY>
  </CLIENT>
/*
```

Figure 7-4 CLIENT parameters

Once an order has been submitted to the IBM server, the **RECEIVE ORDER** will wait for the server to manufacture a package to satisfy the request. If the package is not ready for download within the time allowed, then SMP/E will stop processing. The order is described by an order entry in the global zone and remains in a “pending” state. The package for such an order can be retrieved later by SMP/E using **RECEIVE ORDER PENDING**(Figure 7-5).

```
RECEIVE ORDER(
              PENDING(ordername)
              WAIT(minutes | NOLIMIT)
              CLIENT(ddname)
              TRANSFERONLY
             ).
```

Figure 7-5 RECEIVE ORDER PENDING

PENDING Specifies the name of an existing order entry whose package has not yet been downloaded

Orders will be downloaded to SMPNTS specified in JCL. The **RECEIVE FROMNTS** command can be used to process an order that was already downloaded using the **TRANSFERONLY** operand.

```
RECEIVE FROMNTS(
                  packageid |
                  ORDER(ordername)
                ).
```

Figure 7-6 RECEIVE FROMNTS

7.3 ORDER management panel

The SMP/E Primary Option Menu is shown in Figure 7-7 on page 88 with a new Option 7.

```
---------------------- SMP/E PRIMARY OPTION MENU ---------------- SMP/E 34.07
===>


    0  SETTINGS           - Configure settings for the SMP/E dialogs
    1  ADMINISTRATION     - Administer the SMPCSI contents
    2  SYSMOD MANAGEMENT  - Receive SYSMODs and HOLDDATA
                            and install SYSMODs
    3  QUERY              - Display SMPCSI information
    4  COMMAND GENERATION - Generate SMP/E commands
    5  RECEIVE            - Receive SYSMODs, HOLDDATA and
                            support information
    6  MIGRATION ASSISTANT- Generate Planning and Migration Reports
    7  ORDER MANAGEMENT   - Manage ORDER entries in the global zone

    D  DESCRIBE           - An overview of the dialogs
    T  TUTORIAL           - Details on using the dialogs
    W  WHAT IS NEW        - What is New in SMP/E
```

Figure 7-7 New option at main menu

The new order management dialog was created to manage order entries in the global zone. Order entries are created by using the **RECEIVE ORDER** command to request orders of HOLDDATA and PTFs from an IBM server.

```
-------------------------------- ORDER Entries ------------- Row 1 to 13 of 13
===>                                                          SCROLL ===> PAGE

Commands: FIND -Find a string

Actions:  S -Select, D -Delete

  Entry                                 Order          Download
  Name      Status      Content    Date and Time    Date and Time
  --------  ----------  --------  ---------------  ---------------
  ORD00001  DOWNLOADED  HOLDDATA  05.048 15:30:42  05.048 15:40:58
  ORD00002  DOWNLOADED  CRITICAL  05.049 08:23:47  05.049 08:29:03
  ORD00004  DOWNLOADED  PTFS      05.052 11:52:55  05.053 14:22:01
  ORD00005  DOWNLOADED  APARS     05.053 14:45:36  05.053 14:50:52
  ORD00006  DOWNLOADED  HOLDDATA  05.054 10:28:33  05.054 11:26:50
  ORD00007  DOWNLOADED  RSU0501   05.054 10:34:06  05.054 17:56:03
  ORD00009  PENDING     CRITICAL  05.056 15:47:56
  ORD00010  DOWNLOADED  PTFS      05.059 16:20:04  05.059 16:33:45
  ORD00011  PENDING     HOLDDATA  05.063 10:32:38
  ORD00012  PENDING     PTFS      05.068 17:30:12
  ORD00013  DOWNLOADED  PTFS      05.068 17:46:57  05.069 13:25:47
****************************** Bottom of data ********************************
```

Figure 7-8 Order entries dialog

The new dialog can be used to see the status of all orders, as well as the details, and to delete order entries.

Note: The package associated with an order entry is not affected when the entry is deleted from the global zone.

7.4 New command generation panel

```
 COMMAND GENERATION - RECEIVE SELECTION
===>

Select one of the following:

   1 Receive from data sets on tape or DASD
   2 Download and receive a package from an FTP server (RECEIVE FROMNETWORK)
   3 Receive an existing package in the SMPNTS directory (RECEIVE FROMNTS)
   4 Order and receive HOLDDATA or PTFs from a server (RECEIVE ORDER)

To return to the previous panel, enter END .
```

Figure 7-9 New option at Command Generation dialog

A new option was created in the Generate Command dialog for the **RECEIVE ORDER** command.

```
                COMMAND GENERATION - RECEIVE ORDER
===>
                                                                      More:     +
Enter the RECEIVE ORDER options:
  ORDER TYPE    ===> NEW          Create a NEW order or download a PENDING
                                  order?
  For a NEW order:
    CONTENT     ===> RECOMMENDED  Content for a new order.  ALL, APARS,
                                  CRITICAL, HOLDDATA, PTFS, or RECOMMENDED
    FORTGTZONES ===> YES          Specify target zones for the inventory?
                                  YES or NO
  For a PENDING order:
    ORDER NAME  ===>              ORDER entry name of the pending order.

  WAIT          ===> NOLIMIT      Minutes to wait for order completion.
                                  0 - 1440 minutes or NOLIMIT
  TRANSFERONLY  ===> YES  Stop RECEIVE ORDER processing after package is
                          stored in the SMPNTS?  YES or NO
  DELETEPKG     ===> NO   Delete the package from SMPNTS when done?  YES or NO

Enter the ORDERSERVER file information:
  DDNAME        ===> ORDSRVR   DDNAME for the ORDERSERVER file
   -or-
  DATA SET NAME ===> HLQ.SERVER.CONFIG
  VOLUME SERIAL ===>           Volser for allocation
  UNIT          ===>           Unit type

Enter the CLIENT file information:
  DDNAME        ===> SMPCLNT   DDNAME for the CLIENT file
   -or-
  DATA SET NAME ===> HLQ.CLIENT.CONFIG
  VOLUME SERIAL ===>           Volser for allocation
  UNIT          ===>           Unit type

Enter the SMPNTS directory name (if not defined by a DDDEF entry):
===> '/u/soares/orders/'
```

Figure 7-10 New command generation panel

The new panel to generate a **RECEIVE ORDER** command is shown at Figure 7-10.

In Figure 7-11 there is an example of an order for recommended service to the MVSTGT1 target zone. The order will be only transferred from the server to the SMPNTS path especified. The statement WAIT(NOLIMIT) means that there is no time-out for this order. Once the transfer is completed, processing terminates and the order can be processed later with the **RECEIVE FROMNTS** command.

```
//S1       EXEC PGM=GIMSMP,
//         PARM='PROCESS=WAIT',
//         DYNAMNBR=120
//*
//* NOTE:      THIS JCL CREATED BY THE COMMAND GENERATION DIALOGS.
//*
//*            SMP ZONE-RELATED FILES ARE DYNAMICALLY ALLOCATED,
//*            THIS INCLUDES THE SMPPTS, SMPLOG, AND SMPTLIB DATA SETS,
//*            IF APPLICABLE.
//*
//* SMP FILES
//*
//SMPCSI   DD DISP=SHR,DSN=HLQ.GLOBAL.CSI
//*
//*
//SMPCNTL  DD *
  SET    BOUNDARY (GLOBAL)
                  .
  RECEIVE
       ORDER (
       ORDERSERVER(
           ORDSRVR
                 )
       CONTENT (
         RECOMMENDED
               )
       FORTGTZONES (
           MVSTGT1
               )
       CLIENT(SMPCLNT)
       WAIT(NOLIMIT)
       TRANSFERONLY
             )
                .
//*********************************************************************
//*      ADDITIONAL JCL FOR THE RECEIVE COMMAND.
//*
//*********************************************************************
//ORDSRVR DD DISP=(SHR,KEEP),
//            DSN=HLQ.SERVER.CONFIG
//SMPCLNT DD DISP=(SHR,KEEP),
//            DSN=HLQ.CLIENT.CONFIG
//SMPNTS   DD PATHDISP=KEEP,
//          PATH='/u/soares/orders/'
```

Figure 7-11 An example of a JCL created by SMP/E

7.5 Configuration and setup

Before using the **RECEIVE ORDER** command, there are various configurations and setup tasks that must be performed:

- Establish access to the Java runtime and SMP/E application classes.
- Obtain a user certificate at ShopzSeries.
- Set up a security product database (that is, RACF).

These tasks are documented in detail in *SMP/E User's Guide,* SA22-7773.

Prior releases of SMP/E use ICSF to compute SHA-1 hash values. This is done during **RECEIVE FROMNETWORK** command processing, as well as when using the **GIMZIP**, **GIMUNZIP**, and **GIMGTPKG** service routines. For a number of reasons, not all user z/OS systems have ICSF enabled. Therefore, these SMP/E functions are not usable by all users on all z/OS systems.

The Java MessageDigest class provides an alternate method for computing SHA-1 hash values. SMP/E was extended to use the Java MessageDigest class to compute SHA-1 hash values if SMP/E determines ICSF is not active or configured. This allows the **RECEIVE FROMNETWORK** command to be used on z/OS systems that cannot use ICSF.

8

Health Checker for z/OS

The objective of IBM Health Checker for z/OS is to identify potential problems before they impact system availability or, in the worst case, cause outages. It checks the current active z/OS and sysplex settings and definitions for a system and compares the values to those suggested by IBM or defined by you. It is not meant to be a diagnostic or monitoring tool, but rather a continuously running preventative that finds deviations from best practices. IBM Health Checker for z/OS produces output in the form of detailed messages to let you know of both potential problems and suggested actions to take. Note that these messages do not mean that IBM Health Checker for z/OS has found problems that you need to report to IBM! IBM Health Checker for z/OS output messages simply inform you of potential problems so that you can take action.

This chapter contains information about the new z/OS component:

- Health Checker for z/OS
- SDSF support for the Health Checker

8.1 Health Checker for z/OS introduction

Health Checker for z/OS is a tool developed to address component configuration and setup errors commonly made by installations. The goal of this tool is to avoid outages by identifying potential problems before they impact availability of your installation.

This tool became a new component incorporated in z/OS V1R7, but it is also available as a download from the Web for use in older z/OS versions.

Health Checker for z/OS checks the current active z/OS sysplex settings and definitions for a system and compares the values to those suggested by IBM or defined by you. It is not meant to be a diagnostic or monitoring tool, but rather a continuously running preventative that finds deviations from best practices.

8.1.1 Health Checker for z/OS and the prototype

IBM Health Checker for z/OS had a limited release as a prototype. This reference to the prototype is to distinguish it from the current IBM Health Checker for z/OS product available with z/OS V1R7. The prototype was available as a downloadable package that included sample JCL for a batch invocation. The prototype was developed to address the situation that 15 to 20% of multi-system outages are attributed to installation setup and configuration definitions.

The current IBM Health Checker for z/OS is an integrated part of z/OS V1R7 and is also available as a Web deliverable. The Web deliverable is functionally identical to the integrated version and should not be confused with the prototype. It can be used with z/OS V1R4 up to z/OS V1R6, and is valid for z/OS.e as well. The site for the download is:

```
http://www.ibm.com/servers/eserver/zseries/zos/downloads/
```

Migration from the prototype

If you have the prototype installed already, you should check the guidelines to migrate to the new Health Checker for z/OS in *IBM Health Checker for z/OS User's Guide*, SA22-7994.

Differences from the prototype

There are differences between the Health Checker for z/OS and the prototype; these differences are summarized in Table 8-1.

Table 8-1 Differences between the prototype and z/OS V1R7 functions

	Prototype	Health Checker for z/OS
Releases supported	OS/390 V2R10 - z/OS V1R4	► z/OS V1R7 and higher: integrated ► z/OS V1R4, V1R5, V1R6: supported by Web deliverable
Invocation	Batch job	Started task in its own address space
Check delivery	Included with base function	It is integrated, additions by component PTFs
Check management and user overrides	USERPARM members	► SDSF interface ► **MODIFY** command ► Macros ► HZSPRMxx parmlib member with support for symbolic and policy statements

	Prototype	Health Checker for z/OS
Create user defined groups of checks	No	Yes
Output	SYSOUT data set	► **DISPLAY** command ► WTO and other messages reporting ► Log stream ► SDSF ► HZSPRINT utility

Check name changes

Several check names have changed since they were released in the prototype. Table 8-2 lists the old and new check names.

Table 8-2 Updated check names in z/OS V1R7

Prototype check name	IBM Health Checker for z/OS check name
EMCS_hardcopy	CNZ_EMCS_Hardcopy_Mscope
GRS_SyncRsv	GRS_SYNCHRES
SYSCONS_MSCOPE	CNZ_Syscons_Mscope
SYSCONS_ROUTCODES	CNZ_Syscons_Routcode
SYSCONS_PDMODE	CNZ_Syscons_PD_Mode
SYSCONS_MASTER	CNZ_Syscons_Master
Console_Master	CNZ_Console_MasterAuth_Cmdsys
Console_MSCOPE_and_Routcodes	CNZ_Console_Mscope_and_Routcode
AMRF_And_MPF_Consistent	CNZ_AMRF_Eventual_Action_Msgs
Console_routcode_11	CNZ_Console_Routcode_11

8.1.2 Health Checker for z/OS component support

Health Checker for z/OS runs in its own address space as shown in Figure 8-1 on page 96. The primary support of the address space is to provide check routines. With z/OS V1R7, Health Checker for z/OS has the following changes and enhancements:

- It is available with its own FMID.
- Current and future checks are shipped with the individual components.
- Additional checks can be added in the service stream.
- Checks may be provided by vendors and 3rd parties.
- SDSF support is provided.

8.1.3 Health checks

A *check* is actually a program or routine that identifies potential problems before they impact availability or, in some worst cases, cause outages. A check is owned, delivered, and supported by the component, element, or product that writes it, and processing is as follows:

- You can update or override some check values using either SDSF or statements in the

HZSPRMxx parmlib member or the **MODIFY** command. You might want to apply these installation updates or overrides if some check values are not suitable for your environment or configuration.

As shown in Figure 8-1, you can override some check values in the following ways:

- Statements in the HZSPRMxx parmlib member

  Make permanent overrides by creating policies in the HZSPRMxx SYS1.PARMLIB member.

- The **MODIFY** command
- SDSF interactive command panel (Table 8-5 on page 108 identifies when SDSF can be used.)

- Check output is created when a check issues its output as WTOs and other messages, which you can view using:
  - SDSF
  - The HZSPRINT utility
  - A log stream that collects a history of check output

  If a check finds a deviation from best practices or a potential problem, it issues a WTO message. We call these WTO messages *exceptions*. Check exception messages include not only a description of the potential problem found, including the severity, but also information on what to do to fix the potential problem.

Save the check results using System Logger

IBM Health Checker for z/OS retains only the check results from the last iteration of a check in the message buffer. If you want to retain a historical record of check results, which is a good idea, you must define and connect to a log stream. When you have a log stream connected, the system writes check results to the log stream every time a check completes.

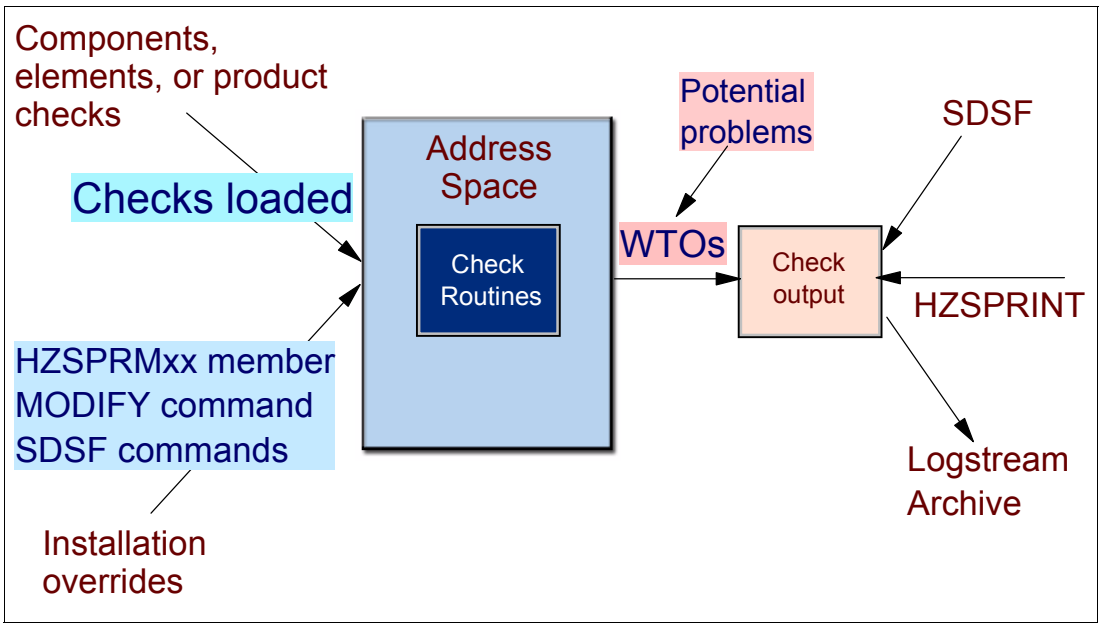


Figure 8-1 Health Checker for z/OS runs in its own address space

8.1.4 Check values used for comparison

The values used by the checks for comparison to installation settings are learned from past experience and IBM recommendations. These comparison checks come from a variety of sources including product documentation, z/OS system test, and z/OS service.

Web sites available for additional information

- Parallel Sysplex availability checklist at:

  `http://www.ibm.com/servers/eserver/zseries/pso/`

- ITSO Redbooks at:

  `http://www.redbooks.ibm.com/`

- zSeries platform test report at:

  `http://www.ibm.com/servers/eserver/zseries/zos/integtst/`

- Washington System Center flashes at:

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

Migration information

For migration to a 64-bit environment, see whitepaper WP100269 “z/OS Performance: Managing Processor Storage in a 64-bit environment,” and the Washington System Center Flash 10086, “Software Capacity Planning: Migration to 64 bit Mode.”

Other documentation

Some very good information about Health Checker can be found in the following product manuals:

- *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- *z/OS MVS Planning: Global Resource Serialization*, SA22-7600
- *z/OS MVS Planning: Operations*, SA22-7601
- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- *z/OS Security Server RACF Command Language Reference*, SA22-7687
- *z/OS Security Server RACF Security Administrator’s Guide*, SA22-7683
- *z/OS UNIX System Services Planning*, GA22-7800

8.1.5 How Health Checker for z/OS can identify problems

IBM Health Checker for z/OS output messages simply inform you of potential problems so that you can take action on your installation. The goal is to know where you are running a risk. Then, you can be aware and take preventive actions.

The following situations are examples of how IBM Health Checker for z/OS can help by identifying potential problems:

- Changes in defaults or configuration values that occur dynamically over the life of an IPL. Checks that look for changes in these values should run periodically to keep the installation aware of changes.
- Threshold levels approaching the upper limits, especially those that might occur gradually.
- Single points of failure in a configuration.
- Unhealthy combinations of configurations or values that an installation might not think to check.

- Configuration abnormalities in what was believed to be a stable system.
- Unexpected values on a system. Investigation reveals changes had been correctly made to that system, but not replicated on other systems.
- Default configurations that were never optimized for performance.
- Outdated settings that do not support all current applications.
- Mismatched naming conventions that can lead to an outage.

Tip: Combine Health Checker for z/OS messages with an automation product, like System Automation. This is the one way to have your installation running as defined by your installation, and to have actions going on without human intervention. This way, only a part of the messages will need your action, and you can focus on the high severity exceptions.

8.1.6 Enhanced Preventive Service Planning Tool

IBM will be adding more checks to IBM Health Checker for z/OS periodically, both integrated into z/OS and as APARs. To identify checks that have been provided in PTFs, use the Enhanced Preventive Service Planning Tool, available at the following Web site:

```
http://techsupport.services.ibm.com/390/psp_main.html
```

You can identify checks by specifying the keyword HCHECKER/K.

Using the tool

To use the IBM Enhanced Preventive Service Planning Tool, do the following:

- Download the host program to your workstation. You only need to do this once.
- Select the PSP buckets you are interested in.
- Download the bucket extract file, which contains a list of APAR and FMID pairs for each bucket you selected.
- Upload the host file and the bucket extract file to your z/OS system.
- Run the host program with the bucket extract file against your SMP/E data. This will produce a report similar to a REPORT ERRSYSMODS that tells you which of the associated fixes have not been received or applied on a particular target zone.

8.2 Health Checker for z/OS processing

As illustrated in Figure 8-2, Health Checker for z/OS functions in the following way:

1. Check values provided by components.

   Each check includes a set of pre-defined values, such as:

   – Interval, or how often the check will run

   – Severity of the check, which influences how check output is issued

   – Routing and descriptor codes for the check

   You can update or override some check values using either SDSF or statements in the HZSPRMxx parmlib member or the **MODIFY** command.

2. Check output.

   A check issues its output as WTOs and other messages, which you can view using SDSF, the HZSPRINT utility, or a log stream that collects a history of check output. If a check

finds a deviation from best practices or a potential problems, it issues a WTO message known as an exception. Check exception messages include not only a description of the potential problem found, including the severity, but also information on what to do to fix the potential problem.

3. Resolve check exceptions.

   To get the best results from IBM Health Checker for z/OS, you should let it run continuously on your system so that you will know when your system has changed dynamically from best practice values. When you get an exception, you should resolve it using the information in the check exception message or overriding check values, so that you do not receive the same exceptions over and over. You can use either SDSF, the HZSPRMxx parmlib member, or the IBM Health Checker for z/OS MODIFY (F hzsproc) command to manage checks.

4. If you solve an exception, changing a product setting or system control, it is a good policy to rerun the checks related with this action, to guarantee the problem identified was fixed.

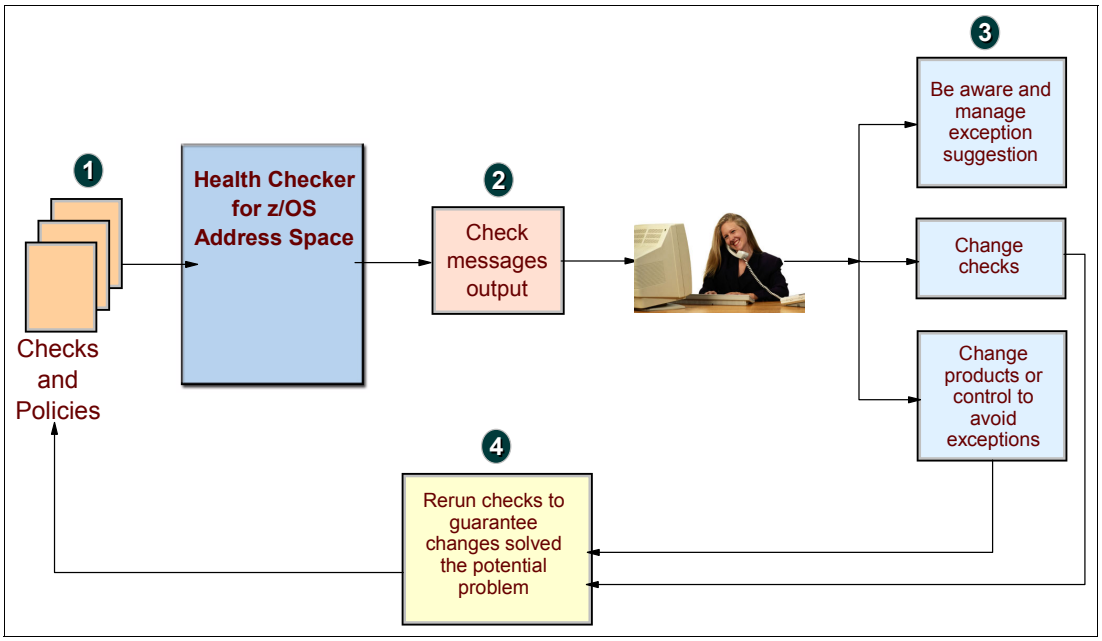


Figure 8-2 Flow of the Health Checker for z/OS

8.3 Installation of Health Checker for z/OS

The following items must be used during installation and use of Health Checker for z/OS:

SYS1.SAMPLIB(HZSALLCP) A sample job to format a data set to track persistent data (one for every system).

SYS1.SAMPLIB(HZSPROC) The sample procedure to start IBM Health Checker for z/OS. HZSPROC must be a USS superuser.

SYS1.SAMPLIB(HZSPRINT) The sample job to copy the message buffers for one or more checks.

SYS1.SAMPLIB(HZSMSGNJ) The sample job to format a message table for a new check. If logger support is desired, the log stream must be defined, and the first 3 characters of the log stream name must be HZS.

SYS1.PARMLIB(HZSPRMxx) Used to define the Health Checker for z/OS policy statement overrides and the System Logger command. The System Logger can be used to enable log stream processing to the specified log stream.

See *IBM Health Checker for z/OS User's Guide,* SA22-7994 for installation details.

8.3.1 Security definitions

Both IBM Health Checker for z/OS and users looking at check output need access to resources. You must create security definitions to control access and maintain security for these resources.

Note: You must set up security for IBM Health Checker for z/OS the same way you would for any other started task.

Steps in setting up security

You need a user ID for IBM Health Checker for z/OS with superuser authority to make sure IBM Health Checker for z/OS works correctly. The following example is a guideline.

1. Create a user ID for IBM Health Checker for z/OS with superuser authority (UID(0)) and connect this superuser user ID to a group (such as OMVSGRP). For example:

```
ADDUSER HZSUSER
        OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
        NOPASSWORD
ADDGROUP OMVSGRP OMVS(GID(46))
CONNECT HZSUSER GROUP(OMVSGRP)
```

2. Associate the new user ID with the IBM Health Checker for z/OS started task HZSPROC. An example follows:

```
SETROPTS GENERIC(STARTED)
RDEFINE  STARTED HZSPROC.* STDATA(USER(HZSUSER) GROUP(OMVSGRP))
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
SETROPTS RACLIST(STARTED) REFRESH
```

3. Give the IBM Health Checker for z/OS started task superuser User ID access to the HZSPDATA data set on each system where you run IBM Health Checker for z/OS. For example, you might specify the following:

```
ADDSD 'SYS1.PRODSYS.HZSPDATA' UACC(NONE)
PERMIT SYS1.PRODSYS.HZSPDATA CLASS(DATASET) ID(HZSUSER) ACCESS(UPDATE)
```

4. Give the IBM Health Checker for z/OS started task superuser User ID READ access to the HZSPRMxx parmlib members. For example, you might specify the following:

```
ADDSD 'SYS1.PARMLIB' UACC(NONE)
PERMIT SYS1.PARMLIB CLASS(DATASET) ID(HZSUSER) ACCESS(READ)
```

5. If using a log stream, you must define UPDATE access for the IBM Health Checker for z/OS started task superuser User ID to each RESOURCE(log_stream_name) CLASS(LOGSTRM). IBM Health Checker for z/OS connects directly to the defined log stream or streams. For example, you might specify the following:

```
RDEFINE  LOGSTRM logstreamname UACC(NONE)
PERMIT   HZS.logstreamname CLASS(LOGSTRM) ID(HZSUSER) ACCESS(UPDATE)
SETROPTS CLASSACT(LOGSTRM)
SETROPTS RACLIST(LOGSTRM) REFRESH
```

Security definitions for the HZSPRINT utility

If you will be using HZSPRINT to view check output, authorize HZSPRINT users to QUERY and MESSAGES services with RACF. Define a profile for the users who need access to check information through HZSPRINT. In the following examples, HZSUSER is either a user ID or a group ID that you are giving access to HZSPRINT output.

- If you use HZSPRINT to look at one check at a time, define an EXEC as follows:

  ```
  //EXEC PGM=HZSPRINT,PARM='CHECK(IBMRACF,RACF_GRS_RNL)'
  ```

  Then you must define access to one of the following sets of resources:

  ```
  HZS.sysname.owner.QUERY
  HZS.sysname.owner.MESSAGES
  ```

  or

  ```
  HZS.sysname.owner.name.QUERY
  HZS.sysname.owner.name.MESSAGES
  ```

  For example, you might define the following:

  ```
  RDEFINE XFACILIT HZS.SYS1.IBMRACF.RACF_GRS_RNL.QUERY UACC(NONE)
  PERMIT HZS.SYS1.IBMRACF.RACF_GRS_RNL.QUERY CLASS(XFACILIT) ID(HZSPRINTU)
  ACC(READ)
  RDEFINE XFACILIT HZS.SYS1.IBMRACF.RACF_GRS_RNL.MESSAGES UACC(NONE)
  PERMIT HZS.SYS1.IBMRACF.RACF_GRS_RNL.MESSAGES CLASS(XFACILIT) ID(HZSPRINTU)
           ACC(READ)
  SETROPTS CLASSACT(XFACILIT)
  SETROPTS RACLIST(XFACILIT) REFRESH
  ```

- If you use HZSPRINT with a wildcard character for the check owner name to look at multiple checks, as follows:

  ```
  //EXEC PGM=HZSPRINT,PARM='CHECK(*,GRS_Mode)'
  ```

  You will need to define access to the following set of resources:

  ```
  HZS.sysname.QUERY
  HZS.sysname.owner.MESSAGES
  ```

  or

  ```
  HZS.sysname.owner.name.MESSAGES
  ```

  For example, you might define the following:

  ```
  RDEFINE  XFACILIT HZS.sysname.QUERY UACC(NONE)
  PERMIT   HZS.sysname.QUERY CLASS(XFACILIT) ID(HZSPRINTU) ACCESS(READ)
  RDEFINE  XFACILIT HZS.sysname.owner.MESSAGES UACC(NONE)
  PERMIT   HZS.sysname.owner.MESSAGES CLASS(XFACILIT) ID(HZSPRINTU) ACC(READ)
  SETROPTS CLASSACT(XFACILIT)
  SETROPTS RACLIST(XFACILIT) REFRESH
  ```

- If you use HZSPRINT with a wildcard character for the check name to look at multiple checks, as follows:

  ```
  //EXEC PGM=HZSPRINT,PARM='CHECK(IBMRACF,*)'
  ```

  You will need to define access to the following set of resources:

  ```
  HZS.sysname.owner.QUERY
  HZS.sysname.owner.MESSAGES
  ```

  or

  ```
  HZS.sysname.owner.name.MESSAGES
  ```

For example, you might define the following:

```
RDEFINE  XFACILIT HZS.SYS1.IBMRACF.QUERY UACC(NONE)
PERMIT   HZS.SYS1.IBMRACF.QUERY CLASS(XFACILIT) ID(HZSPRINTU) ACC(READ)
RDEFINE  XFACILIT HZS.SYS1.IBMRACF.owner.MESSAGES UACC(NONE)
PERMIT   HZS.SYS1.IBMRACF.owner.MESSAGES CLASS(XFACILIT) ID(HZSPRINTU)
ACC(READ)
SETROPTS CLASSACT(XFACILIT)
SETROPTS RACLIST(XFACILIT) REFRESH
```

- If you use HZSPRINT to print IBM Health Checker for z/OS log stream data as follows:

  ```
  //EXEC PGM=HZSPRINT,PARM='LOGSTREAM(logstreamname)'
  ```

  or

  ```
  //EXEC PGM=HZSPRINT,PARM='LOGSTREAM(logstreamname),CHECK(owner,name)'
  ```

  or

  ```
  //EXEC
  PGM=HZSPRINT,PARM='LOGSTREAM(logstreamname),CHECK(owner,name),EXCEPTIONS'
  ```

  Then you must define a profile for the log stream and assign READ access to users accessing the log stream through HZSPRINT. For example, you might do this as follows:

  ```
  RDEFINE  FACILITY log_stream_data_set_name UACC(NONE)
  PERMIT   log_stream_data_set_name(FACILITY) ID(HZSPRINTU) ACCESS(READ)
  SETROPTS CLASSACT(FACILITY)
  SETROPTS RACLIST(FACILITY) REFRESH
  ```

Summary of security settings

Table 8-3 contains a summary of the security settings for the HZSPRINT utility.

Table 8-3 Summary of security settings for the HZSPRINT utility on the EXEC statement

PGM=HZSPRINT,PARM=	Resource	Class	Access
'CHECK(owner,name)'	HZS.sysname.owner.QUERY HZS.sysname.owner.MESSAGES or HZS.sysname.owner.name.QUERY HZS.sysname.owner.name.MESSAGES	XFACILIT	READ
'CHECK(*,name)'	HZS.sysname.QUERY HZS.sysname.owner.MESSAGES or HZS.sysname.owner.name.MESSAGES	XFACILIT	READ
'CHECK(owner,*)'	HZS.sysname.owner.QUERY HZS.sysname.owner.MESSAGES or HZS.sysname.owner.name.MESSAGES	XFACILIT	READ

Security definitions in a multilevel system environment

If your system is a multilevel system environment and you are using multilevel security labels to control access to resources, you must assign a multilevel security label of SYSLOW to the IBM Health Checker for z/OS superuser User ID, HZSUSER, which you defined previously. That means that any data object that the check touches must have a SECLABEL that would pass the mandatory access check for the type of operation that is being performed.

The following example enables the SECLABEL class and assigns a multilevel security label of SYSLOW:

```
SETROPTS CLASSACT(SECLABEL) RACLIST(SECLABEL)
ALTUSER hcsuperid SECLABEL(SYSLOW)
```

8.3.2 Starting Health Checker for z/OS

The HZSPROC should be started when the system is started, and it always must be activated in order to keep looking for potential problems in the installation.

To start Health Checker for z/OS, issue one of the following commands:

```
s hzsproc
s hzsproc,HZSPRM=xx
```

or

```
s hzsproc,HZSPRM=(x1,...,xn)
```

Checks are automatically run at a scheduled interval, or on a scheduled date, but it is possible to restart checking using the SDSF interface.

Health Checker for z/OS should not be stopped except in an emergency situation. In this case, issue the following command:

```
f hzsproc,stop
```

8.3.3 Specifying the HZSPRMxx members you want the system to use

HZSPRMxx parmlib members can be specified when starting the IBM Health Checker for z/OS or dynamically to direct the system to process the corresponding parmlib member:

- To specify HZSPRMxx members dynamically, use one of the following **MODIFY** commands:

  ```
  f hzsproc,SET,PARMLIB=(x1,x2,...xn)
  f hzsproc,ADD,PARMLIB=(x1,x2,...xn)
  f hzsproc,REPLACE,PARMLIB=(x1,x2,...xn
  ```

  where *xn* is the two digit suffix of an HZSPRMxx parmlib member.

- To specify the HZSPRMxx members at startup time, specify the two digit suffix of an HZSPRMxx member in one of the commands shown in the previous section.

In this example, hzsproc is the name of the IBM Health Checker for z/OS procedure.

IBM Health Checker for z/OS procedure

The IBM Health Checker for z/OS procedure must contain the following:

```
//HZSPROC JOB JESLOG=SUPPRESS
//HZSPROC PROC HZSPRM='00'
//HZSSTEP EXEC PGM=HZSINIT,REGION=0K,TIME=NOLIMIT,
// PARM='SET PARMLIB=&HZSPRM'
//HZSPDATA DD DSN=SYS1.&SYSNAME..HZSPDATA,DISP=OLD
// PEND
// EXEC HZSPROC
```

The value for PARM= must resolve to SET PARMLIB=(x1,...,xn). You can also use an `ADD` or `REPLACE` command in place of `SET` because the command is issued during the initialization phase.

8.4 User interface to manage checks

You can manage checks by making dynamic or temporary changes to current checks. You make these decisions by deactivating, adding, or temporarily updating check values, using the following ways for administrators to interact with Health Checker for z/OS:

- SDSF panels
  - CK command, see “Using SDSF panels” on page 104
- **MODIFY** command
  - Temporary check changes, see “Health Checker for z/OS commands via MODIFY command” on page 109
- HZSPRMxx parmlib member
  - Permanent check changes, see “HZSPRMxx parmlib member and policies” on page 111

8.4.1 Using SDSF panels

SDSF has a new **CK** command where you can access new panels to work and control Health Checker for z/OS.

SDSF provides an easy to use interface. You can change Health Checker for z/OS behavior from SDSF screens, and you can manage the check’s status and results as well.

Figure 8-6 on page 106 and Figure on page 107 show the new SDSF panel display after issuing the **CK** command.

Checks on the CK panel in SDSF

You can protect the checks from IBM Health Checker for z/OS that are displayed on the CK panel by providing RACF authorization checks for access to use the action characters, shown in Figure 8-6 on page 106 and overtype capability. This is done by defining resource names in the XFACILIT class, as shown in Table 8-4 on page 104.

Table 8-4 Authority required to use action characters and overtypes

Action or overtype	Function	Resource name	Class	Access
A action character	Activate	HZS.sysname.owner.name.ACTIVATE	XFACILIT	UPDATE
D action characters	Display	HZS.sysname.owner.name.QUERY	XFACILIT	READ
E action character	Refresh	HZS.sysname.owner.name.REFRESH	XFACILIT	CONTROL
H action character	Deactivate	HZS.sysname.owner.name.DEACTIVATE	XFACILIT	UPDATE
P action characters	Delete	HZS.sysname.owner.name.DELETE	XFACILIT	CONTROL
R action character	Run	HZS.sysname.owner.name.RUN	XFACILIT	UPDATE
S and X action characters	Browse, Print	HZS.sysname.owner.name.MESSAGES	XFACILIT	READ
U action character and overtypeable fields	Remove	HZS.sysname.owner.name.UPDATE	XFACILIT	UPDATE

Messages on an authorization failure

Figure 8-3 shows the message displayed when a user without authorization attempts to use the ACTION characters, shown in Figure 8-6 on page 106 on the SDSF panel display. In this example the user used the D action character on check CNZ_AMRF_EVENTUAL_ACTION_MSGS.

```
ICH408I USER(ROGERS  ) GROUP(SYS1    ) NAME(ROGERS             )
  HZS.SC70.IBMCNZ.CNZ_AMRF_EVENTUAL_ACTION_MSGS.QUERY
  CL(XFACILIT)
  INSUFFICIENT ACCESS AUTHORITY
  FROM HZS.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

Figure 8-3 RACF authorization failure messages for user using the D action character

Example of protecting checks

To protect all checks and permit a user to control the checks, you can define generic profiles as follows:

```
RDEFINE XFACILIT HZS.** UACC(NONE)
PERMIT  HZS.** CLASS(XFACILIT) ID(userid or groupid) ACCESS(CONTROL)
SETROPTS RACLIST(XFACILIT) REFRESH
```

After the authorization is made, the user uses the D action character and receives the display shown in Figure 8-4.

```
SDSF HEALTH CHECKER DISPLAY  SC70                        COMMAND ISSUED
COMMAND INPUT ===>                                            SCROLL ===> PAGE
RESPONSE=SC70
 HZS0200I 16.47.50 CHECK SUMMARY     508
 CHECK OWNER      CHECK NAME                       STATE STATUS
 IBMCNZ           CNZ_AMRF_EVENTUAL_ACTION_MSGS    AE   SUCCESSFUL
   A - ACTIVE          I - INACTIVE
   E - ENABLED         D - DISABLED
   G - GLOBAL CHECK    + - CHECK ERROR MESSAGES ISSUED
```

Figure 8-4 Display using D action character

Columns on the CK display

Figure 8-5 shows some of the columns on the CK display. There 42 columns in total. You must use PF11 to begin displaying the columns not shown on the main display panel.

```
Column    Title      Width  Description
Name      (Displayed)
OWNER     CheckOwner  16  Check owner
STATE     State       18  Check state
STATUS    Status      18  Check status
RESULT    Result       6  Result code from the last invocation of the check
DIAG1     Diag1        8  Diagnostic data from check, word 1
DIAG2     Diag2        8  Diagnostic data from check, word 2 U
DIAGFROM  DiagFrom     8  Source of the diagnostic data, words 1 and 2: ABEND,HCHECKER or CHECKRTN
GLOBAL    Global       6  Indicator of whether the check is global
GLOBALSY  GlobalSys    9  Name of the system where the global check is running
EXCOUNT   ExcCount     8  Number of exceptions detected by this check on the last iteration
COUNT     RunCount     8  Number of times the check has been invoked
FAIL      Fail         4  Number of times the check failed
SEVERITY  Severity     8  Severity level of the check  (HIGH,MEDIUM,LOW,NONE)
```

Figure 8-5 Columns that appear on the CK display

Figure 8-6 shows the display of all the health checks that can been seen when you issue the SDSF **CK** command.

```
 Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF HEALTH CHECKER DISPLAY  SC70                      LINE 1-23 (54)
 COMMAND INPUT ===>                                        SCROLL ===> PAGE
 ACTION=//-Block,=-Repeat,+-Extend,A-Activate,D-Display,E-Refresh,H-Deactivate,
 ACTION=P-Delete,R-Run,S-Browse,U-RemoveCat,X-Print
 NP   NAME                           CheckOwner       State              Status
      CNZ_AMRF_EVENTUAL_ACTION_MSGS  IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      CNZ_CONSOLE_MASTERAUTH_CMDSYS  IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      CNZ_CONSOLE_MSCOPE_AND_ROUTCODE IBMCNZ          ACTIVE(ENABLED) EXCEPTION-LOW
      CNZ_CONSOLE_ROUTCODE_11        IBMCNZ           ACTIVE(ENABLED) EXCEPTION-LOW
      CNZ_EMCS_HARDCOPY_MSCOPE       IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      CNZ_EMCS_INACTIVE_CONSOLES     IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      CNZ_SYSCONS_MASTER             IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      CNZ_SYSCONS_MSCOPE             IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      CNZ_SYSCONS_PD_MODE            IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      CNZ_SYSCONS_ROUTCODE           IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      CNZ_TASK_TABLE                 IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL
      GRS_CONVERT_RESERVES           IBMGRS           ACTIVE(ENABLED) EXCEPTION-LOW
      GRS_EXIT_PERFORMANCE           IBMGRS           ACTIVE(ENABLED) EXCEPTION-LOW
      GRS_MODE                       IBMGRS           ACTIVE(ENABLED)    SUCCESSFUL
      GRS_SYNCHRES                   IBMGRS           ACTIVE(ENABLED)    SUCCESSFUL
      RACF_GRS_RNL                   IBMRACF          ACTIVE(ENABLED)    SUCCESSFUL
      RACF_SENSITIVE_RESOURCES       IBMRACF          ACTIVE(ENABLED) EXCEPTION-HIGH
      RED_CHECK_LOGON                REDBOOK          ACTIVE(ENABLED)    SUCCESSFUL
      RRS_DUROFFLOADSIZE             IBMRRS           ACTIVE(ENABLED)    SUCCESSFUL
      RRS_MUROFFLOADSIZE             IBMRRS           ACTIVE(ENABLED)    SUCCESSFUL
      RRS_RMDATALOGDUPLEXMODE        IBMRRS           ACTIVE(ENABLED) EXCEPTION-MED
      RRS_RMDOFFLOADSIZE             IBMRRS           ACTIVE(ENABLED)    SUCCESSFUL
      RRS_RSTOFFLOADSIZE             IBMRRS           ACTIVE(ENABLED)    SUCCESSFUL
      RSM_AFQ                        IBMRSM           ACTIVE(ENABLED)    SUCCESSFUL
      RSM_HVSHARE                    IBMRSM           ACTIVE(ENABLED)    SUCCESSFUL
```

Figure 8-6 SDSF new panel display using the CK command

```
NP    NAME                            CheckOwner      State             Status
      RSM_MAXCADS                    IBMRSM          ACTIVE(ENABLED)   SUCCESSFUL
      RSM_MEMLIMIT                    IBMRSM          ACTIVE(ENABLED) EXCEPTION-LOW
      RSM_REAL                        IBMRSM          ACTIVE(ENABLED)   SUCCESSFUL
      RSM_RSU                         IBMRSM          ACTIVE(ENABLED)   SUCCESSFUL
      SDUMP_AUTO_ALLOCATION           IBMSDUMP        ACTIVE(ENABLED)   SUCCESSFUL
      SDUMP_AVAILABLE                 IBMSDUMP        ACTIVE(ENABLED)   SUCCESSFUL
      USS_AUTOMOUNT_DELAY             IBMUSS          ACTIVE(ENABLED)   SUCCESSFUL
      USS_FILESYS_CONFIG              IBMUSS          ACTIVE(ENABLED) EXCEPTION-HIGH
      USS_MAXSOCKETS_MAXFILEPROC      IBMUSS          ACTIVE(ENABLED) EXCEPTION-LOW
      VSM_CSA_CHANGE                  IBMVSM          ACTIVE(ENABLED)   SUCCESSFUL
      VSM_CSA_LIMIT                   IBMVSM          ACTIVE(ENABLED)   SUCCESSFUL
      VSM_CSA_THRESHOLD               IBMVSM          ACTIVE(ENABLED)   SUCCESSFUL
      VSM_PVT_LIMIT                   IBMVSM          ACTIVE(ENABLED)   SUCCESSFUL
      VSM_SQA_LIMIT                   IBMVSM          ACTIVE(ENABLED)   SUCCESSFUL
      VSM_SQA_THRESHOLD               IBMVSM          ACTIVE(ENABLED) EXCEPTION-MED
      XCF_CDS_SEPARATION              IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_CF_CONNECTIVITY             IBMXCF          ACTIVE(ENABLED) EXCEPTION-MED
      XCF_CLEANUP_VALUE               IBMXCF          ACTIVE(ENABLED) EXCEPTION-MED
      XCF_DEFAULT_MAXMSG              IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_FDI                         IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_MAXMSG_NUMBUF_RATIO         IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_SFM_ACTIVE                  IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_SIG_CONNECTIVITY            IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_SIG_PATH_SEPARATION         IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_SIG_STR_SIZE                IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_SYSPLEX_CDS_CAPACITY        IBMXCF          ACTIVE(ENABLED) EXCEPTION-MED
      XCF_TCLASS_CLASSLEN             IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_TCLASS_CONNECTIVITY         IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
      XCF_TCLASS_HAS_UNDESIG          IBMXCF          ACTIVE(ENABLED)   SUCCESSFUL
```

Figure 8-6 (Continued) SDSF new panel display using CK command

SDSF commands to customize panels

You can customize how your main panel is displayed by choosing which columns to show first.

The command **SET ACTION** permits you to see all available commands in the top panel:

```
SET ACTION (ON|LONG|SHORT|OFF|?)
```

To display the action character, use the following command:

```
SET ACTION ON
```

There are many columns with information about each check. You can choose what to see in the first screen displayed. The command **ARRANGE** allows you to place columns in the order that you want, and select the width of a column. The **ARRANGE** command can be used as follows:

```
ARRANGE from column After|Before to column
ARRANGE from-column First|Last|width
```

Figure 8-7 on page 108 shows the SDSF **CK** command panel after the **ARRANGE** command has changed the order of the columns. The two columns that are now displayed are:

Result Result code from the last invocation of the check

RunCount Number of times the check has been invoked

```
Display  Filter  View  Print  Options  Help
 ------------------------------------------------------------------------------
 SDSF HEALTH CHECKER DISPLAY  SC70                      LINE 1-23 (53)
 COMMAND INPUT ===>                                        SCROLL ===> CSR
 ACTION=//-Block,=-Repeat,+-Extend,A-Activate,D-Display,E-Refresh,H-Deactivate,
 ACTION=P-Delete,R-Run,S-Browse,U-RemoveCat,X-Print
 NP   NAME                            CheckOwner Status          Result RunCount
      CNZ_AMRF_EVENTUAL_ACTION_MSGS   IBMCNZ     SUCCESSFUL           0        2
      CNZ_CONSOLE_MASTERAUTH_CMDSYS   IBMCNZ     SUCCESSFUL           0        5
      CNZ_CONSOLE_MSCOPE_AND_ROUTCODE IBMCNZ     EXCEPTION-LOW        4        5
      CNZ_CONSOLE_ROUTCODE_11         IBMCNZ     EXCEPTION-LOW        4        5
      CNZ_EMCS_HARDCOPY_MSCOPE        IBMCNZ     SUCCESSFUL           0        5
      CNZ_EMCS_INACTIVE_CONSOLES      IBMCNZ     SUCCESSFUL           0        5
      CNZ_SYSCONS_MASTER              IBMCNZ     SUCCESSFUL           0        5
      CNZ_SYSCONS_MSCOPE              IBMCNZ     SUCCESSFUL           0        5
      CNZ_SYSCONS_PD_MODE             IBMCNZ     SUCCESSFUL           0       97
      CNZ_SYSCONS_ROUTCODE            IBMCNZ     SUCCESSFUL           0        5
      CNZ_TASK_TABLE                  IBMCNZ     SUCCESSFUL           0      386
      GRS_CONVERT_RESERVES            IBMGRS     EXCEPTION-LOW        4        1
      GRS_EXIT_PERFORMANCE            IBMGRS     EXCEPTION-LOW        4        5
      GRS_MODE                        IBMGRS     SUCCESSFUL           0        1
      GRS_SYNCHRES                    IBMGRS     SUCCESSFUL           0       97
```

Figure 8-7 CK display panel with the order of the columns changed

SDSF and the MODIFY command

SDSF uses a subset of the **MODIFY** command. This means that some commands can be used from SDSF, and others cannot. Table 8-5 shows differences between SDSF and the **MODIFY** command.

It is possible to change some values dynamically by changing them directly on the SDSF screen. For example, it is possible to change the check's severity from LOW to MEDIUM, just by overtyping on the SDSF screen if you are authorized. When you do this, SDSF issues a **MODIFY** command in background for you, changing the severity to MEDIUM.

Table 8-5 Issuing commands with SDSF or the MODIFY command

Functions	**SDSF**	**MODIFY command: f hzsproc *parameters***
► Change check states ► Add new checks ► Delete checks ► Display check information ► Run checks ► Change interval to run each check ► Update or override check values in use ► Categorize checks	Yes	Yes
Stop IBM Health Checker for z/OS address space	No	Yes
Connect to IBM Health Checker for z/OS log stream	No	Yes
Request processing of HZSPRMxx parmlib members	No	Yes
Policy statement support	No	Yes - See Note

Note: Policy statements are allowed in the **MODIFY** command, but we recommend this only for test purposes. When you need permanent changes, or permanent overrides, you must use the HZSPRMxx parmlib member to create policies with the changes you want. See “Specifying the HZSPRMxx members you want the system to use” on page 103 for details.

SDSF compatibility and requirements

Every release of z/OS SDSF requires the level of the BCP that it ships with.

In the case of an installation running on a sysplex, you may need WebSphere MQ. Without the MQ support, SDSF shows checks only for the system you are logged on to. Sysplex-wide data requires WebSphere MQ on each system.

8.4.2 Health Checker for z/OS commands via MODIFY command

All Health Checker for z/OS MODIFY command subcommands are listed in Table 8-6. These subcommands can be issued using the **f hzsproc** command or they can be used as statements in the HZSPRMxx parmlib member.

MODIFY command

You can update or override some check values using the **MODIFY** command. These are called installation updates. The commands are useful for making dynamic, temporary changes to checks. You might do this if some check values are not suitable for your environment or configuration. The **MODIFY** command has the parameter options shown in Table 8-6. The format of the command follows:

```
F HZSPROC,parameters
```

Table 8-6 MODIFY command subcommands

MODIFY parameters	Purpose of the parameters
ACTIVATE	Sets the check state to active.
ADD/ADDREPLACE/ REMOVE POLICY	Adds, replaces, or removes a policy statement.
ADDNEW	Adds new checks to Health Check.
ADD/REPLACE/SET	Used it to indicate which parmlib members’ suffixes are going to be in use with Health Checker.
DEACTIVATE	Disables running of specified check.
DELETE	Deletes the specified check from Health Check. Once it is deleted you can bring it back only by refresh processing.
DELETE,FORCE	Deletes a check that is running.
DISPLAY	Shows information about the check.
DISPLAY,CHECKS	Finds the check owner and check name.
LOGGER	Connects to a pre-defined log stream.
REFRESH	Deletes the check, than performs the ADDNEW function, adding the check to Health Checker again.
RUN	Run a check immediately.
STOP	Stops the Health Checker.

MODIFY parameters	Purpose of the parameters
UPDATE	Allows a temporary update to the current default or override values for a specific check. The new value is in effect until the next refresh for the specified check.

Command example

Figure 8-8 shows use of the **MODIFY** command to display check settings.

```
F hzsproc,DISPLAY,CHECKS(check_owner,check_name),
   [SUMMARY|DETAIL]
   [,ANY|,NOTDELETED|,DELETED]
   [,POLICYEXCEPTIONS][,EXCEPTIONS]
```

Figure 8-8 Command to display check settings

Figure 8-9 is an example of the details about a specific check.

```
F HZSPROC,DISPLAY,CHECKS,CHECK=(IBMGRS,GRS_MODE),DETAIL
HZS0201I 10.15.56 CHECK DETAIL     670
 CHECK(IBMGRS,GRS_MODE)
  STATE: ACTIVE(ENABLED)     GLOBAL  STATUS: SUCCESSFUL
  EXITRTN: ISGHCADC
  LAST RAN: 05/06/2005 09:54    NEXT SCHEDULED: (NOT SCHEDULED)
  INTERVAL: ONETIME   SEVERITY: LOW       WTOTYPE: INFORMATIONAL
  SYSTEM DESCCODE: 12
DEFAULT PARAMETERS:       STAR
  REASON FOR CHECK:  GRS should run in STAR mode to improve
                     performance.
  MODIFIED BY: MODIFY COMMAND
  CATEGORIES: GRS
  DEFAULT DATE: 20050105                       DEBUG MODE: OFF
```

Figure 8-9 MODIFY command example

Temporary check changes

Temporary check changes are useful when you need to test a modification before making it permanent, or when you just need to change a check for a specific situation. A temporary change is in effect until the first check refresh or system IPL. When you want a change to be permanent you must use the HZSPRMxx parmlib member.

Use the SDSF panel or the following **MODIFY** command to make temporary check changes:

```
f hzsproc,UPDATE,filters,action
```

In this example, *filters* can be:

```
CHECK=(check_owner,check_name)
EXITRTN=exit routine
CATEGORY=([{ONLY|ANY|EVERY|EXCEPT},][category1[,...,categoryn]])
```

Filters specify which check or checks you wish to take an action against. You can specify wildcard characters * and ? for filters. An asterisk (*) represents any string having a length of zero or more characters. A question mark (?) represents a position that may contain any single character.

Also in this example, the update *action* can be:

```
[,ACTIVE|INACTIVE]
[,ADDCAT=(cat1,...,cat16)]
[,DATE=date]
[,DEBUG={OFF|ON}]
[,DESCCODE=(desccode1,...,desccoden)]
[,{INTERVAL=ONETIME|INTERVAL=hhh:mm}]
[,PARM=parameter,REASON=reason,DATE=date]
[,REASON=reason]
[,REPCAT=(cat1[,cat2[,...,cat16]])]
[,REMCAT=(cat1[,cat2[,...,cat16]])]
[,ROUTCODE=(routcode1,...,routcoden)]
[,SEVERITY={HIGH|MEDIUM|LOW|NONE}]
[,WTOTYPE={CRITICAL|EVENTUAL|INFORMATIONAL|HARDCOPY|NONE}]
```

8.4.3 HZSPRMxx parmlib member and policies

The Health Checker for z/OS processes and applies policy information from the HZSPRMxx parmlib members in use every time checks are refreshed, added, or when there is an IPL. The system applies these policy statements in the order they occur in the HZSPRMxx parmlib members. Define an HZSPRMxx parmlib member to define and modify policies, and to enable log stream processing.

Defining a policy

Make permanent changes by creating policies in the HZSPRMxx parmlib member. The IBM Health Checker for z/OS policy is a tool to let you manage checks by applying permanent changes to your checks. The IBM Health Checker for z/OS policy simply consists of the following:

- A set of update statements in the HZSPRMxx parmlib member or members currently in use for a system.
- The information in your IBM Health Checker for z/OS policy is applied to all existing checks and to any new checks you add.
- IBM Health Checker for z/OS processes policy information every time checks are refreshed, added, or when there is an IPL.
- The policy is the place to put any check changes you want to make permanent and to have applied to any checks you add in the future.
- You can have one IBM Health Checker for z/OS policy per system.
- Use the non-policy statements to test changing values.

HZSPRMxx parmlib member policy

A policy allows you to make permanent overrides or changes in the Health Checker for z/OS behavior. Before creating a policy, test your changes using the following **MODIFY** command:

```
f hzsproc,UPDATE,filters,action
```

When you are sure about the changes you want, change your HZSPRMxx parmlib member by adding a new policy. Including other non-policy statements in your HZSPRMxx parmlib member will be ineffective, because the parmlib member specified in the HZSPROC procedure is processed before any checks are added or the Health Checker for z/OS begins running.

8.4.4 Policy statements

Use the policy statement in the HZSPRMxx parmlib member to establish permanent overrides to existing checks. Policy statements are as follows:

- ADD POLICY
- ADDREPLACE POLICY
- REMOVE POLICY

They are applied immediately, and are applied again whenever a check is added or refreshed. We recommend that you use policy statements only in an HZSPRMxx parmlib member.

All policy statements are named for easy reference, and each one has a date and reason also. So when a check is updated, a policy with an older date may not be applied, forcing a review of policy statements when a check is updated.

You can use a policy statement to apply any kind of update command to change checks or to permanently delete checks. You can change or create a policy, changing HZSPRMxx or creating a new HZSPRMxx parmlib member and concatenating it.

We recommend that you use the POLICY statement in HZSPRMxx parmlib member. Use the following policy statement to create, replace, or remove a policy:

```
{ADD | ADDREPLACE}
POLICY STATEMENTNAME(name) UPDATE(filters) [UPDATE options] REASON(reason) DATE(date)
POLICY STMT(name) DELETE(filters) REASON(reason) DATE(date)

REMOVE POLICY STATEMENT(name)
```

In this example, the *filters* can be:

```
CHECK=(check_owner,check_name)
EXITRTN=exit routine CATEGORY=([{ONLY|ANY|EVERY|EXCEPT},][category1[,...,categoryn]])
```

Policy filters

Filters specify which check or checks you wish to take an action against. You can specify wildcard characters * and ? for filters, as follows:

Asterisk (*) Represents any string having a length of zero or more characters.

Question mark (?) Represents a position which may contain any single character.

Filter parameters are as follows:

- CHECK is a required filter, except for the **`D CHECKS,filters`** command.
- EXITRTN=exit routine: EXITRTN specifies the HZSADDCHECK exit routine that added checks to IBM Health Checker for z/OS.
- CATEGORY=([{ONLY|ANY|EVERY|EXCEPT},][category1[,...,categoryn]]). Filter checks are by user-defined categories. The CATEGORY filters can be one of the following:

  ONLY These checks are in every one of the specified categories, and have only as many categories as are specified. For example, a check assigned to three categories would not match if the CATEGORY=ONLY statement on this **`MODIFY`** command specified two categories.

  ANY These checks are in any of the specified categories.

  EVERY These checks are in every specified category.

EXCEPT Checks that are not in any of the specified categories.

Update Options

The syntax of the update options you can use to update check values is as follows:

```
[ACTIVE|INACTIVE]
[ADDCAT=(cat1,...,cat16)]
[DATE=date]
[DEBUG={OFF|ON}]
[DESCCODE=(desccode1,...,desccoden)]
[{INTERVAL=ONETIME|INTERVAL=hhh:mm}]
[PARM=parameter,REASON=reason,DATE=date]
[REASON=reason]
[REPCAT=(cat1[,cat2[,...,cat16]])]
[REMCAT=(cat1[,cat2[,...,cat16]])]
[ROUTCODE=(routcode1,...,routcoden)]
[SEVERITY={HIGH|MEDIUM|LOW|NONE}]
[WTOTYPE={CRITICAL|EVENTUAL|INFORMATIONAL|HARDCOPY|NONE}]
```

In the update filters, identify checks you want to change.

Policy statement example

In the following example, we create a policy statement called policy1 to change the GRS_SYNCHRES check's running interval from 1 hour to 30 minutes:

```
ADD POLICY STMT(POLICY1) UPDATE CHECK(IBMGRS,GRS_SYNCHRES)
INTERVAL(00:30) REASON('CHANGING INTERVAL FROM 1H TO 30M')
DATE(20050510)
```

MODIFY command to add a policy example

You can use this statement as a test proposal with the **MODIFY** command also. You can specify filters and the update option shown in Figure 8-10. The same options were described in the HZSPRMxx parmlib member in the previous examples.

```
F hzsproc,{ADD | ADDREPLACE},POLICY,STMT=stmntname,UPDATE,filters
[,update options],REASON=reason,DATE=date
```

Figure 8-10 MODIFY policy command

Policy statement to create policy1 example

In the following example, a policy statement called policy1 is created to change the GRS_SYNCHRES check's running interval from 1 hour to 30 minutes:

```
ADD POLICY STMT(POLICY1) UPDATE CHECK(IBMGRS,GRS_SYNCHRES)INTERVAL(00:30)
REASON('CHANGING INTERVAL FROM 1H TO 30M') DATE(20050510)
```

Using the statement on the parmlib member makes it permanent. If you issue after a **MODIFY UPDATE** command (**F hcproc,UPDATE**), the updated value is valid until the next check refresh or system IPL.

It is possible to use the **MODIFY UPDATE** command directly on the parmlib member also, but this update is lost when the first check refresh is done.

Tip: For tests or temporary changes, use SDSF panels and the **MODIFY** command very carefully. For permanent changes, the best practice is to use the POLICY statement in the HZSPRMxx parmlib member.

ADDREPLACE POLICY example

These same values will be applied to all checks owned by IBMGRS, every time they are refreshed or added. This means that all new and existing IBMGRS checks will be set to HIGH severity until the system is IPLed or IBM Health Checker for z/OS is restarted. At IPL time, you loose your policy updates unless you have updated HZSPRMxx parmlib member with the new (or changed) policy P2.

```
ADDREPLACE POLICY STMT(p2) UPDATE CHECK(ibmgrs,*) SEVERITY(high) REASON('change policy')
DATE(20050901)
```

Figure 8-11 Another policy example

Use the command as an example of an HZSPRMxx parmlib member.

Syntax for HZSPRM*xx* parmlib members

The syntax for HZSPRMxx parmlib members and the **MODIFY** command are similar. You can use the same parameters in both the HZSPRMxx parmlib member and the **F hzsproc,parameters** command, but there are differences. To specify parameters in an HZSPRMxx parmlib member, consider the following:

- Use parentheses where an equal sign is used in the **MODIFY** command.
- Separate parameters with blanks instead of commas.

8.4.5 Categories to manage and display information

Health Checker for z/OS offers you a new resource, called *category*, to control your checks. When you have many checks, you can use categories to make it easier to manage or display information. Use the ADDCAT, REPCAT, and REMCAT parameters, as follows:

ADDCAT Lets you add the specified check to a category

REPCAT Lets you replace a category for a check

REMCAT Lets you remove a check from a category

Note: All categories are user-defined. IBM does not define any categories for checks.

The following examples show how you can use categories in the HZSPRMxx parmlib member and in the **MODIFY** command to manage checks.

Category filters

Use the CATEGORY filter to filter actions against checks by category. For example, you might put checks into categories such as shift and offshift, global, or exception. Then you can perform actions such as activate, deactivate, or run a group of checks with one command.

It is very easy to create or change a check's category using the SDSF **CK** panel. However, this kind of change is temporary, as discussed previously.

Command example

In the example in Figure 8-12, we grouped GRS_MODE and GRS_SYNCHRES checks into a GRS category. The easiest way to create your own categories is on the SDSF panel, but if you want a category to be permanent, use the policy statement in the HZSPRMxx parmlib member instead**.** Now, it is possible to display details for this group of checks defined by the category.

```
F HZSPROC,DISPLAY,CHECKS,CATEGORY=(GRS),DETAIL
HZS0201I 10.45.01 CHECK DETAIL     725
 CHECK(IBMGRS,GRS_SYNCHRES)
 STATE: ACTIVE(ENABLED)              STATUS: SUCCESSFUL
 EXITRTN: ISGHCADC
 LAST RAN: 05/09/2005 09:54    NEXT SCHEDULED: 05/09/2005 10:54
 INTERVAL: 1:00    SEVERITY: LOW       WTOTYPE: INFORMATIONAL
 SYSTEM DESCCODE: 12
 THERE ARE NO PARAMETERS FOR THIS CHECK
 REASON FOR CHECK:  GRS synchronous RESERVE processing should be
                    enabled to avoid deadlock conditions.
 MODIFIED BY: MODIFY COMMAND
 CATEGORIES: GRS
 DEFAULT DATE: 20050105                        DEBUG MODE: OFF

 CHECK(IBMGRS,GRS_MODE)
 STATE: ACTIVE(ENABLED)     GLOBAL  STATUS: SUCCESSFUL
 EXITRTN: ISGHCADC
 LAST RAN: 05/06/2005 09:54    NEXT SCHEDULED: (NOT SCHEDULED)
 INTERVAL: ONETIME   SEVERITY: LOW       WTOTYPE: INFORMATIONAL
 SYSTEM DESCCODE: 12
 DEFAULT PARAMETERS:       STAR
 REASON FOR CHECK:  GRS should run in STAR mode to improve
                    performance.
MODIFIED BY: MODIFY COMMAND
 CATEGORIES: GRS
DEFAULT DATE: 20050105                        DEBUG MODE: OFF
```

Figure 8-12 Command displaying a user defined category

8.5 Managing Health Checker's alerts

Each check can be defined by the user in up to 16 categories. This allows a user to manipulate a group of checks.

The HZSPRINT utility is provided to write check message buffers to a dataset or sysout. Use SDSF or HZSPRINT to look at the check output from any checks that generate exceptions.

It is also possible to keep message buffers in a log stream. Log stream processing allows check results to be archived. HZSPRINT can be used to print message buffers that have been archived to a log stream.

Once Health Checker for z/OS is running, you must see the messages to take the corrective action based on the message buffers.

It is possible to rerun the checks as necessary to verify the corrective action.

8.5.1 HZSPRINT utility

The HZSPRINT utility allows you to look at check output. HZSPRINT writes the current message buffer for the target checks to SYSOUT for one check, multiple checks, or all checks. The following information assumes that you have already set up security (see the SYS1.SAMPLIB JCL for the HZSPRINT utility as shown in Figure 8-13.

```
//HZSPRINT EXEC PGM=HZSPRNT,TIME=1440,REGION=0M,
// PARM=('CHECK(check_owner,check_name)')
//* PARM=('CHECK(check_owner,check_name)',
//* 'EXCEPTIONS')
//* PARM=('LOGSTREAM(logstreamname)')
//* PARM=('LOGSTREAM(logstreamname)',
//* 'CHECK(owner,name)')
//* PARM=('LOGSTREAM(logstreamname)','EXCEPTIONS',
//* 'CHECK(owner,name)')
//* PARM=('LOGSTREAM(logstreamname)','EXCEPTIONS')
//* PARM=('LOGSTREAM(logstreamname)','SYSNAME(sysname)')
//* PARM=('LOGSTREAM(logstreamname)','SYSNAME(sysname)',
//* 'CHECK(owner,name)')
//* PARM=('LOGSTREAM(logstreamname)','EXCEPTIONS',
//* 'SYSNAME(sysname)',
//* 'CHECK(owner,name)')
//* PARM=('LOGSTREAM(logstreamname)','EXCEPTIONS',
//* 'SYSNAME(sysname)')
//SYSOUT DD SYSOUT=A,DCB=(LRECL=256)
```

Figure 8-13 HZSPRINT sample JCL example

HZSPRINT parameters

Following are the HZSPRINT parameters:

CHECK(check_owner,check_name) Identifies checks you are interested.You can also use wildcard characters * and ? in both the check owner and check name fields to get output from multiple checks.

EXCEPTIONS Optional parameter that permits you to limit the output in SYSOUT to messages from checks that wrote at least one check exception message. For example, to see the output of all checks that found exceptions, use the following:

```
// PARM='CHECK(*,*),EXCEPTIONS'
```

LOGGER(log_stream_name) Optional parameter for the LOGGER that specifies that you want to print the specified log stream.

SYSNAME(system_name) Optional parameter that lets you limit the output in SYSOUT to output from checks running on the specified system. The *system_name* is the name of a system where the checks were executed. You can only specify the SYSNAME parameter with LOGSTREAM. You can use wildcard characters * and ? in the system_name field to specify that you want check output from multiple systems. The default for SYSNAME is SYSNAME(*), which will give you output for specified checks from all the systems in the sysplex.

If you want to allocate a data set for HZSPRINT output:

- The data set must be:
  - Fixed block
  - Logical record length of 256
- Add the name of the output data set allocated to the HZSPRINT JCL. For example:

  ```
  //SYSOUT DD DISP=SHR,DSNAME=D10.HCHECKER.REPORT.FEB2505,DCB=(LRECL=25
  ```

- Note that the first character of each line of HZSPRINT output is a carriage control character. The example of HZSPRINT output shown in Figure 8-14 includes output for all checks with exceptions.

```
HZSU001I IBM Health Checker for z/OS Check Messages
Filter: CHECK(*,*)
Filter: Only checks with exception(s)6)

********************************************************************************
* Start: CHECK(IBMUSS,USS_MAXSOCKETS_MAXFILEPROC)
********************************************************************************
CHECK(IBMUSS,USS_MAXSOCKETS_MAXFILEPROC)
START TIME: 03/30/2005 11:31:06.593289
CHECK DATE: 20040808 CHECK SEVERITY: LOW
CHECK PARM: 64000,64000

BPXH003I z/OS UNIX System Services is configured using OMVS=(00) which correspond to the
BPXPRMxx suffixes. The IBMUSS specification for IBM Health Checker for z/OS
USS_MAXSOCKETS_MAXFILEPROC is 64000,64000. . . .


END TIME: 03/30/2005 11:31:08.457023 STATUS: EXCEPTION-LOW
********************************************************************************
End: CHECK(IBMUSS,USS_MAXSOCKETS_MAXFILEPROC)
********************************************************************************
```

Figure 8-14 Partial output from HZSPRINT

8.5.2 Use LOGGER to keep historical data

Use the LOGGER parameter to connect to and use a pre-defined log stream whenever a check generates output.

Activate the logger by using the following statement, at HZSPRMxx parmlib member:

```
LOGGER=ON,LOGSTREAMNAME=HZSlogstream
```

IBM Health Checker for z/OS retains only the check results from the last iteration of a check in the message buffer. If you want to retain a historical record of check results, which is a good idea, you must define and connect to a log stream. When you have a log stream connected, the system writes check results to the log stream every time a check completes.

Use the following procedures to set up, use, and disable the logger.

1. Plan for setting up log streams, including allocation of coupling facility and DASD space. Careful planning of DASD and coupling facility space is important because if the log stream fills up, no additional data will be written to it and data will be lost. See the “Planning for system logger applications” section of *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Keep in mind the following:

- Define either:
  - One log stream for each system
  - One log stream for multiple systems to use
- HZS must be the first letters of log stream names you define. For example, you might define a log stream name of HZSLOG1.
- System logger requires at least a base sysplex configuration in your installation.
- System logger requires SMS to be active, in at least a null configuration, even if you do not use SMS to manage your volumes and data sets. See the “Set up the SMS environment for DASD data sets section” of *z/OS MVS Setting Up a Sysplex*, SA22-7625

2. Set up security for log streams, as discussed in 8.3.1, “Security definitions” on page 100.
3. Enable log streams in one of the following ways:
   - Use the **MODIFY** command: `f hzsproc,logger=on,logstreamname=logstreamname`
   - Use the `LOGGER` parameter in the HZSPRMxx parmlib member: `LOGGER(ON) LOGSTREAMNAME(logstreamname)`
4. To disable a log stream, issue the following **MODIFY** command:

   ```
   f hzsproc,logger=off
   ```

Figure 8-15 shows a sample coupling facility log stream definition defined in the LOGR policy using the administrative data utility, IXCMIAPU.

```
DEFINE LOGSTREAM NAME(HZS.HEALTH.CHECKER.HISTORY) DESCRIPTION(HEALTH_CHECK_RPT)
STRUCTNAME(HZS_HEALTHCHKLOG)
STG_DUPLEX(NO)
LS_DATACLAS(NO_LS_DATACLAS)
LS_MGMTCLAS(NO_LS_MGMTCLAS)
LS_STORCLAS(NO_LS_STORCLAS)
LS_SIZE(4096)
AUTODELETE(YES)
RETPD(7)
HLQ(NO_HLQ)
HIGHOFFLOAD(80)
LOWOFFLOAD(0)
DIAG(YES)
```

Figure 8-15 Log stream definition

Defining the log stream

You must also define the structure for use by IBM Health Checker for z/OS log stream in the LOGR couple data set. the For more information on the LOGR couple data set, see “Add information about log streams and coupling facility structures to the LOGR policy” section of *z/OS MVS Setting Up a Sysplex*, SA22-7625.

For a coupling facility log stream, you must also add the log structure to the CFRM policy. Figure 8-16 shows a sample coupling facility structure definition for a IBM Health Checker for z/OS log stream structure.

```
STRUCTURE NAME(HZS_HEALTHCHKLOG)
SIZE(4096M)
PREFLIST(cfname,cfname)
```

Figure 8-16 Log stream structure definition

8.6 Products that already have checks defined

Checks are supplied with IBM Health Checker for z/OS. In the future, more checks to IBM Health Checker for z/OS will be made periodically, both integrated into z/OS and as APARs. To identify checks that have been provided in PTFs, use the Enhanced Preventive Service Planning Tool, available at the following Web site:

`http://techsupport.services.ibm.com/390/psp_main.html`

You can identify checks by specifying the keyword `HCHECKER/K`. See “Enhanced Preventive Service Planning Tool” on page 98 for more information.

The following products have checks available:

- Consoles
- GRS
- RACF
- RRS
- RSM™
- UNIX System Services
- VSM™
- Cross system coupling facility (XCF)
- Others: APF libraries, LINKLST, LINKLIB,APFLIST

8.7 Using Health Checker for z/OS

During normal operation, the HZSPROC must be started when a system is started. Most checks are automatically run on a scheduled interval, which you can define and modify as needed, using the SDSF panel or the **`MODIFY`** command.

While Health Checker for z/OS is running it sends status messages for the checks done. The levels of messages indicate whether corrective action is necessary or if you just need to be aware of some information:

- Exception messages
  - Includes WTO summary text
  - Entire message written to message buffer
  - WTO issued based on check severity/WTOTYPE override
- HI severity message - HZS0003E - issued as immediate action message
- MED severity message - HZS0002E - issued as eventual action message
- LOW severity message - HZS0001I, issued as informational message
- None, HZS0004I, issued as hardcopy only message

Look at the check output from any checks that generate exceptions, using SDSF, the HZSPRINT utility, or issuing a **MODIFY** command. The messages indicate potential problems on the installation, and guide you with corrective information. Take the corrective actions based on the messages, then rerun the checks as necessary to verify the success of the corrective actions.

Use POLICY statements when you want to disable or delete checks that are not applicable anymore. You can change the default parameters as necessary for your installation.

8.7.1 Where to find information to create your own checks

The Health Checker for z/OS is a component of MVS that checks the system environment, looking for places an installation is deviating from best practices or where there might be configuration problems. IBM provides a set of check routines in IBM Health Checker for z/OS, but vendors, consultants, and system programmers can add other routines.

You can write a check for the Health Checker for z/OS that analyzes a configuration for the following:

- Changes in recommended values that occur dynamically over the life of an IPL. Checks that look for changes in these values should run periodically to keep the installation aware of changes.
- Threshold levels approaching the upper limits, especially those that might occur gradually or insidiously.
- Single points of failure in a configuration.
- Unhealthy combinations of configurations or values that an installation might not think to check.

When your are planning your checks, keep in mind that a check should only check for one thing. This will make it much easier for the installation to resolve exceptions that the check finds, and override defaults.

You can use information about the last outage conditions and problems in your installation as an insight to develop your own checks. Use your own experience, along with IBM and third-party product documentation and best practices, to create new checks.

Each check has a corresponding message table.

SGML messages

Messages are written in SGML-based language, as follows:

- A standard message structure is required.
- All messages should be self-contained.
- Avoid required additional references.
- Are shipped as an assembler CSECT.

You can find more information on how to create your own checks on *IBM Health Checker for z/OS User's Guide,* SA22-7994.

8.8 Checks overview

This section summarizes some checks included in Health Checker for zOS and z/OS V1R7 in the following components:

- UNIX System Servicesl
- Global resource serialization (GRS)
- RACF
- Consoles
- SVC DUMP (SDUMP)

Additional checks can be write by software vendors and supplied with products. Refer to *IBM Health Checker for z/OS User's Guide*, SA22-7994 for details on how to modify a check and add additional checks.

8.8.1 UNIX System Services

Three checks are defined for UNIX Systems Services, as follows:

- USS_FILESYS_CONFIG

  This check looks for an incorrect configuration of the UNIX Systems Services file systems in a shared HFS environment, which can cause outages and performance problems. This check evaluates the file system configuration, which includes the following:

  - AUTOMOVE setup verification
  - zFS for a multilevel security (MLS) configuration, except in z/OS V1R4
  - Mode of the root, system specific, and version HFS.

- USS_AUTOMOUNT_DELAY

  Low automount delay times in a sysplex can cause the system to hang, continually trying to unmount file systems and failing. This check verifies the automount policy of a system in a sysplex for low automount delay times.

- USS_MAXSOCKETS_MAXFILEPROC

  If the MAXSOCKETS or MAXFILEPROC keywords specified in the BPXPRMxx parmlib member are set too low, the system can run out of usable sockets or file descriptors, respectively. This check compares the internal MAXSOCKETS and MAXFILEPROC values to the defaults or the override values if they exist. An exception message is issued if either is too low.

8.8.2 Global resource serialization (GRS)

The following checks are made for global resource serialization:

- GRS_MODE

  If the environment is running in a sysplex mode, a STAR configuration is recommended because it provides better availability, real storage consumption, processing capacity, and response time. This check verifies the mode specified for a GRS complex and encourages the use of a GRS STAR configuration.

- GRS_SYNCHRES

  Integrity problems can happen if GRS SYNCHRES processing is not enabled. Prior to z/OS V1R6, GRS was shipped with a default of SYNCHRES(NO). Enabling GRS

synchronous reserve processing can prevent deadlock conditions. This check verifies if GRS synchronous reserve processing is enabled.

- GRS_CONVERT_RESERVES

  This check verifies if GRS's generic recommendations for hardware RESERVE conversion to global ENQs are being followed. Converting RESERVEs helps avoid the following conditions:

  - Deadlocks and interlocks
  - False contention between jobs for the same volume
  - The possibility that one system might monopolize a shared device
  - The data integrity exposure that occurs as a result of a system reset while a reserve is in effect

- GRS_EXIT_PERFORMANCE

  GRS has a set of installation exits. Over time, new exits have been introduced which provide better performance and function. For migration and compatibility reasons, the older exits are allowed to persist. The GRS_EXIT_PERFORMANCE check insures that the known optimum choices are made.

8.8.3 RACF

The following health checks are implemented for RACF:

- RACF_GRS_RNL

  This check is to see if any of the RACF ENQ names are on a GRS resource name exclusion list, which changes the scope of the RACF ENQ. This check detects a situation which can result in the corruption of the RACF database.

- RACF_SENSITIVE_RESOURCES

  This check looks at the current APF data sets and the RACF database data sets and flags those that are improperly protected. Specifically, it flags those that:

  - Are not found on the indicated volume
  - Are protected by a profile with:
    - A UACC greater than READ for APF data sets or NONE for the RACF DB
    - An ID(*) specification greater than READ for APF data sets or NONE for the RACF DB
  - Have WARNING mode set
  - Have no RACF profile and PROTECTALL(FAIL) not in effect
  - A specified user ID has UPDATE for APF data sets or READ for the RACF DB or has a greater access specified

8.8.4 Consoles

The following checks are made for the console support in z/OS:

- CNZ_EMCS_Inactive_Consoles

  This is the only check that is global, and should only be run on one system in a sysplex. The rest of the checks are local, and should execute on each system it is added to.

- CNZ_AMRF_Eventual_Action_Msgs

  Checks that eventual action messages are not retained if the action message retention facility (AMRF) is active.

- CNZ_Console_MasterAuth_Cmdsys

  Checks that there is an active console with MASTER authority that has command association to this system.

- CNZ_Console_Mscope_And_Routcode

  Checks that each MCS/SMCS/EMCS console is not defined with multi-system message scopes and receiving all routing codes (or all except routing code 11).

- CNZ_Console_Routcode_11

  Ensures that no MCS or SMCS console is receiving ROUTCODE 11 messages.

- CNZ_EMCS_Hardcopy_Mscope

  Checks to see that each EMCS console defined with a multi-system message scope is not receiving the hardcopy message set.

- CNZ_EMCS_Inactive_Consoles

  Ensures that there are not an excessive number of inactive EMCS consoles. If the EMCS console is no longer needed, the EMCS console removal service (IEARELEC) can be used to remove the EMCS console definition. The number of inactive EMCS consoles in use in a sysplex can affect the time it takes for a system to join a sysplex.

- CNZ_Syscons_Master

  Ensures that the system console has MASTER authority.

- CNZ_Syscons_Mscope

  Ensures that the system console has a single-system message scope and to avoid flooding the system console.

- CNZ_Syscons_PD_Mode

  Ensures that the system console is not in problem determination (PD) mode.

- CNZ_Syscons_Routcode

  Ensures that the system console is receiving the minimum set of routing codes (1, 2, and 10).

- CNZ_Task_Table

  Reports the status of important tasks that run in the CONSOLE address space. Using the report generated from this check, installations can determine if there are real or potential problems with specific functions of the Consoles component.

8.8.5 SVC DUMP (SDUMP)

The following checks are made for SDUMP support:

- SDUMP_AUTO_ALLOCATION

  Checks to see whether automatic allocation of SVC dump data sets is enabled. Automatic allocation of dump data sets efficiently writes the dump from virtual storage to DASD.

- SDUMP_AVAILABLE

  Ensures that SDUMP is enabled to collect SVC Dumps. The SDUMP setup should ensure a dump can be generated when system problems occur.

8.9 Planning your own checks

As of z/OS V1R7 you have the opportunity to write your own checks. An IBM Health Checker for z/OS check consists of three parts. These parts and the order in which they should be developed are the following:

1. **The check routine load module**

   Your check routine gathers all the information and compares current values with best practices or looks for configuration problems and issues messages with the result of the check. When the check routine runs, IBM Health Checker for z/OS passes the check routine a a check-specific copy of the HZSPQE data area.

2. **The message table load module**

   Your message table contains all the messages your check wants to write. You must use the HZSFMSG macro to issue the messages from your check routine.

3. **An HZSADDCHECK exit routine load module**

   This routine contains all the default values of your check (time interval, parameter, and so forth) specified with the HZSADDCK macro. This authorized routine runs in the IBM Health Checker for z/OS address space, called by the IBM Health Checker for z/OS dynamic exit, HZSADDCHECK.

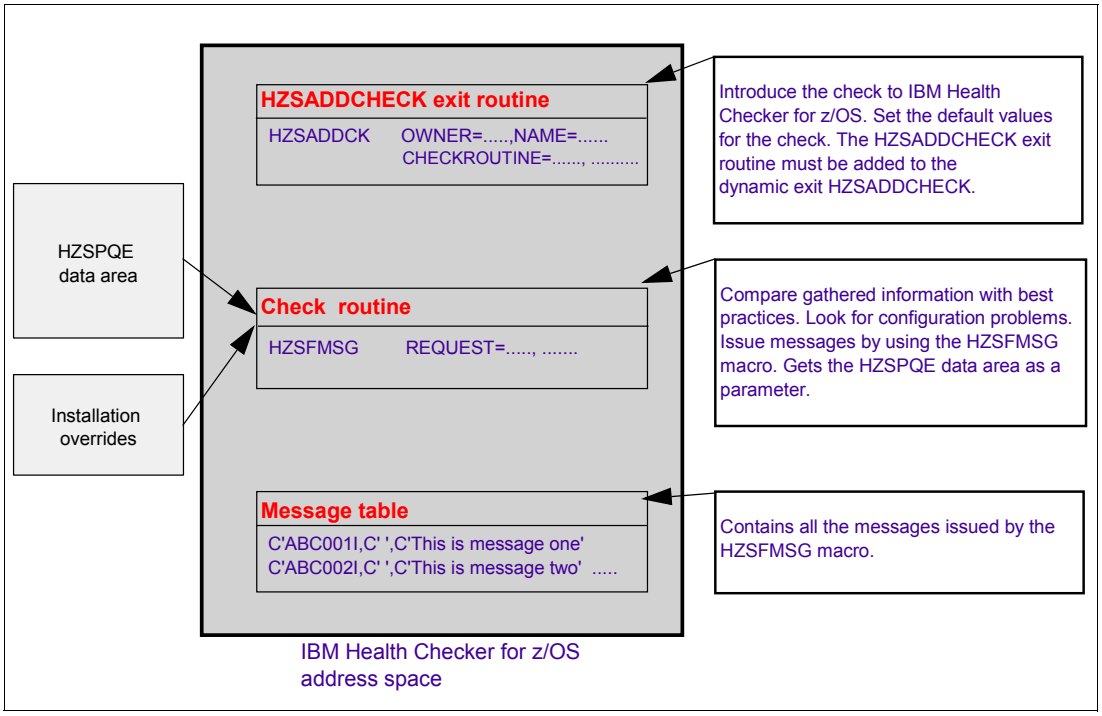


Figure 8-17 The parts of a check for the IBM Health Checker for z/OS

8.10 Developing checks for IBM Health Checker for z/OS

A check is only as good as its implementation. Therefore, we recommend the following considerations *before* you begin developing your check:

- **Keep the check simple**

  When you are planning your checks, keep in mind that a check should only check for one thing. This will make it easier for the installation to resolve exceptions that the check finds and override defaults.

- **Resources**

  Release obtained resources properly. There is no end-of-task cleanup processing for the checks on a regular basis.

- **No disruptive practices in your check routines**

  We recommend that checks read but not update system-specific data such as system control blocks. Try to avoid I/O intensive operations within your check routine.

- **Recovery**

  Your check routine should be designed to handle abends. If a check abends on three consecutive iterations, and the error percolates, the system renders the check inactive until the check is refreshed or parameters for the check are changed.

- **Names**

  Consider the names you want to use for the check routine, the messages, the message table, and the HZSADDCHECK exit routine. It is very helpful to see the components that belong to one another. We recommend that you use a component or product prefix so that you can easily identify where a check comes from. In addition, using the prefix ensures that all the checks for a particular component or product will be grouped together in an SDSF check display, if supported on your system. For example, the product ABC has its check components beginning with ABC.

- **Time interval**

  Which time interval fits best to your check? Check routines can execute with a timer interval or only one time. The minimum time interval is one minute.

- **Messages**

  Check routines should communicate their check results by issuing messages. Which messages are practical for your needs? If a check runs successfully and finds no exceptions to the best practices, the check routine should issue a message to report this and summarize what the check looked for. Furthermore, can you track the messages with another product (such as System Automation) to make sure further activities take place.

- **Parameters**

  Parameters are a good opportunity to keep your check routine flexible. Consider whether or not parameters are useful in your installation. A good example for parameters are default values that can be changed, so you do not need to change your code. You only change the parameter with the following command:

  ```
  f hzsproc,update,check=(checkowner,checkname),parm=xxx
  ```

  However, your check routine must handle this situation. Furthermore, it is a good practice to stop the check itself if a problem was found in the parameter passed to the check routine. The check routine can issue an HZSFMSG stop request (REQUEST=STOP) to stop the check from running.

- **Multilevel security environments**

  In a multilevel security environment each user and each data object is assigned a sensitivity label called a *SECLABEL*. The IBM Health Checker for z/OS address space must be assigned a SECLABEL. The guideline is to assign this address space a SECLABEL of SYSLOW. The check developer must ensure that the check reads and writes data at the SYSLOW level of security.

8.10.1 Write the check routine

The check routine runs in the IBM Health Checker for z/OS address space; the load module must reside in an APF-authorized library, must be reentrant, and should use the following programming considerations:

State	Supervisor, Task mode
Cross memory mode	PASN=SASN=HASN
AMODE	31
ASC mode	Primary
Key	The system will choose a key for a check and use it for all function code calls to the check routine. The key will match the key in field TCBPKF.
Interrupt status	Enabled for I/O and external interrupts.
Locks	No locks held.

When a check routine receives control the contents of the registers are as follows:

Register 0	Points to the 4K dynamic work area.
Register 1	Points to a parameter list containing the address of the Process Queue Element data area (PQE) and the address of the 4K dynamic work area. The PQE will be mapped by the HZSPQE macro.
Register 13	Points to a 144 byte save area.
Register 14	Return address.
Register 15	Address of the check routine.

Note: A maximum of 20 checks can run concurrently. If this limit is reached, the system waits for one check to complete before starting the next one.

The Process Queue Element data area (HZSPQE)

The PQE data area is passed as a parameter to the check routine each time the check routine is called. It contains all the information the check routine needs. It contains in the section PQEChkParms both the defaults defined in the HZSADDCHECK exit routine and any installation overrides to those defaults (from the HZSPRMxx parmlib member or from the **`modify`** command). It also contains a 2K user area (PQEChkWork field) in which you can store check-related information. However, this area will be cleared after the check is deleted (PQE_Function_Code_Delete).

The 2K user work area

Use the 2K user work area to store information across check iterations. This area is saved for the life of the check, and cleared to zeroes after a check is deleted, which is the first step of refresh processing.

The 4K dynamic area

Use the 4K dynamic area to store information only for *one* iteration of the check.

Note: It is recommended that you use the 2K user work area and the 4K dynamic work area instead of passing parameters to or obtaining resources for your check routine. If you obtain resources for your check routine, the storage must be either:

- Obtained and freed in the same function code processing.
- Owned by the jobstep-task.

Function codes for the check routine

An IBM Health Checker for z/OS check routine can be invoked with four different function codes. The field PQE_Function_Code in HZSPQE indicates why the check was called. Your check routine must be able to handle these different situations. Table 8-7 gives an overview:

Table 8-7 Summary of function codes

Function	What the check should do	When is it invoked?
INIT	Validate the environment is suitable for the check. If it is not, issue the HZSFMSG REQUEST=STOP macro to stop the check. If you obtain additional storage for the check, obtain it in INIT processing and obtain it in jobstep-task owned storage. (You cannot assume that each function code runs under the same task.)	► Refresh (for example, issued from SDSF panels) ► When a check transitions to the active enabled state
CHECK	1. Check whether the installation has overridden the default parameter (if any exist) for the check: – Check to see if the PQE_LookatParm bit is set on, indicating either that this is the first iteration of the check, or that the installation has changed the check parameters since the last iteration. – If the bit is on, validate the parameters in the PQE_UserParmArea of the HZSPQE data area. If the check finds bad installation parameters, it should issue the HZSFMSG REQUEST=STOP macro to stop the check. 2. Check processing. 3. Report results using the HZSFMSG macro to issue messages.	► After INIT ► At CHECK interval time
CLEANUP	Clean up anything that you want cleaned up between check iterations. For example, clean up anything that you are not cleaning up in Check processing, or that must be cleaned up if Check processing abends. If you obtained resources during check function processing, check the PQE_CurrentTaskOwned. If the bit is on, the system has already cleaned up the resources for you.	► Refresh (for example, issued from SDSF panels) ► At CHECK interval time
DELETE	Cleanup any storage obtained during INIT or CHECK processing.	► Delete (for example, issued from SDSF panels) ► When the check transitions out of the active enabled state ► When the IBM Health Checker for z/OSaddress space is stopped

Issuing messages in your check routine

To issue a message from your check routine you must use the HZSFMSG macro, shown in Figure 8-18, and you must provide a message table. Your check routine should communicate both unsuccessful and successful checks to the best practices. All checks should issue their messages consistently so that IBM Health Checker for z/OS users get a consistent look and feel no matter what checks they use. You can issue the following kinds of messages:

- Defined messages in the message table (HZSFMSG REQUEST=CHECKMSG).
- IBM Health Checker for z/OS messages (HZSFMSG REQUEST=HZSMSG).
- IBM Health Checker for z/OS messages that indicate the check is stopped (HZSFMSG REQUEST=STOP).

There are 4 message types you can issue:

- Exception messages
- Information messages
- Report messages
- Debug messages

See also “Create the message table for your check” on page 132.

```
HZSFMSG
,REQUEST=request
,MGBADDR=mgbaddr
,REASON=reason
,DIAG=diag
,RETCODE=retcode
,RSNCODE=rsncode
,PLISTVER=plistver
,MF=macroform
```

Figure 8-18 The HZSFMSG macro

The parameters on the HZSFMSG macro are as follows:

- REQUEST=CHECKMSG
  REQUEST=HZSMSG
  REQUEST=STOP

  A required parameter that identifies the source of the message text.

  - **CHECKMSG** indicates that the message text is provided in the message table identified by the MSGTBL parameter of the HZSADDCK macro when the check was added.
  - **HZSMSG** indicates that the message text is provided by IBM Health Checker for z/OS.
  - **STOP** indicates that the system is to stop calling this check. The message text is provided by IBM Health Checker for z/OS.
- MGBADDR=*mgbaddr*

  When REQUEST=CHECKMSG is specified, a required input parameter that contains the address of the MGB control block used to describe the message request. The MGB identifies which message in the check’s message table is requested and describes optional dynamic variables to be used in that message. The HZSMGB macro maps the MGB (structure name HZSMGB). The HZSMGB structure contains the message identifier and an array of text values (each entry consisting of an address and a length). The text

values are inserted into the text based on the positional sequence of the message variable, <mv>, in the sgml source.

Each variable-length text value is limited to 256 characters.

To code: Specify the RS-type address, or address in register (2)-(12), of a pointer field.

- REASON=ERROR

  When REQUEST=HZSMSG is specified, a required parameter that indicates the type of situation being reported.

  Indicates that the message is being issued because of an error. The system is to issue message HZS1003I. This message is also recorded in the check's message buffer. The state of the check is changed to error. The check remains active.

- DIAG=*diag*

  When REQUEST=HZSMSG is specified, a required input parameter, which is displayed as hex data in message output to provide diagnostic information for the failure that is being reported. There is no pre-defined format for this data; it may well be internal component diagnostic data.

  To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

- REASON=BADPARM
  REASON=ERROR
  REASON=ENVNA

  When REQUEST=STOP is specified, a required parameter that indicates the type of situation being reported.

  - **BADPARM** indicates that the parameters are not valid. The system is to issue message HZS1001E. This message is also recorded in the check's message buffer. The state of the check is changed to parameter error. The check remains disabled until the PARMS are changed, presumably to address the error.
  - **ERROR** indicates that the message is being issued because of an error. The system is to issue message HZS1002I. The state of the check is changed to error. The check is disabled. The check will not be called again until the check is refreshed.
  - **ENVNA** indicates that the check is not applicable in the current system environment. Message HZS1003I is written as hardcopy-only and is also written to the check's message buffer. The state of the check is changed to not applicable. The check is disabled. The check will not be called again until the reason for the condition is resolved and the check is refreshed (or its parameter is changed).

- DIAG=*diag*

  When REASON=ERROR and REQUEST=STOP are specified, a required input parameter, which is displayed as hex data in message output to provide diagnostic information for the failure that is being reported. There is no pre-defined format for this data; it may well be internal component diagnostic data.

  To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

- RETCODE=*retcode*

  An optional output parameter into which the return code is to be copied from GPR 15.

  To code: Specify the RS-type address of a fullword field, or register (2)-(12).

- RSNCODE=*rsncode*

  An optional output parameter into which the reason code is to be copied from GPR 0.

  To code: Specify the RS-type address of a fullword field, or register (2)-(12).

- PLISTVER=IMPLIED_VERSION
  PLISTVER=MAX
  PLISTVER=1

  An optional input parameter that specifies the version of the macro. PLISTVER determines which parameter list the system generates. PLISTVER is an optional input parameter on all forms of the macro, including the list form. When using PLISTVER, specify it on all macro forms used for a request and with the same value on all of the macro forms. The values are:

  - **IMPLIED_VERSION** is the lowest version that allows all parameters specified on the request to be processed. If you omit the PLISTVER parameter, IMPLIED_VERSION is the default.
  - **MAX** If you want the parameter list to be the largest size currently possible. This size might grow from release to release and affect the amount of storage that your program needs. If you can tolerate the size change, IBM recommends that you always specify PLISTVER=MAX on the list form of the macro. Specifying MAX ensures that the list-form parameter list is always long enough to hold all the parameters you might specify on the execute form, when both are assembled with the same level of the system. In this way, MAX ensures that the parameter list does not overwrite nearby storage.
  - **1** If you use the currently available parameters.

- MF=S
  MF=(L,*list addr*)
  MF=(L,*list addr,attr*)
  MF=(L,*list addr*,0D)
  MF=(E,*list addr*)
  MF=(E,*list addr,*COMPLETE)

  An optional input parameter that specifies the macro form.

  - **MF=S** Specifies the standard form of the macro, which builds an inline parameter list and generates the macro invocation to transfer control to the service. MF=S is the default.
  - **MF=L** Specifies the list form of the macro. Use the list form together with the execute form of the macro for applications that require reentrant code. The list form defines an area of storage that the execute form uses to store the parameters. Only the PLISTVER parameter can be coded with the list form of the macro.
  - **MF=E** Specifies the execute form of the macro. Use the execute form together with the list form of the macro for applications that require reentrant code. The execute form of the macro stores the parameters into the storage area defined by the list form, and generates the macro invocation to transfer control to the service.
  - ***list addr*** The name of a storage area to contain the parameters. For MF=S and MF=E, this can be an RS-type address or an address in register (1)-(12).
  - ***attr*** An optional 1- to 60-character input string that you use to force boundary alignment of the parameter list. Use a value of 0F to force the parameter list to a word boundary, or 0D to force the parameter list to a doubleword boundary. If you do not code attr, the system provides a value of 0D.
  - **COMPLETE** Specifies that the system is to check for required parameters and supply defaults for omitted optional parameters.

Note: If a message issued with the HZSFMSG macro is incorrect (for instance, the message is not in the message table), the system issues an abend X'290' and creates a logrec error record. See *z/OS System Codes,* SA22-7626 for more details.

Example of an HZSFMSG macro call

Figure 8-19 is an example of an HZSFMSG macro iteration.

```
.
.
.
   LA       R0,WHZSMGB
   ST       R0,PHZSMGB
   HZSFMSG  REQUEST=CHECKMSG,MGBADDR=PHZSMGB,MF=(E,WHZSFMSG)
.
.
.
*********************************************************************
*        WORK AREA DSECT
*********************************************************************
STATWORK DSECT ,
SAVEAREA DS   18F                  SAVE AREA
         HZSFMSG MF=(L,WHZSFMSG)
PHZSMGB  DS   F'0'                 Pointer
WHZSMGB  DS   CL(HZSMGB_LEN)       control block for HZSFMSG macro
```

Figure 8-19 Example of an HZSFMSG macro call

The HZSMGB control block

You specify in the MGBADDR parameter of the HZSFMSG macro a pointer that contains the address of the HZSMGB control block. This control block is used to specify which message of your message table you want to issue. You also specify the values for the variables to be inserted. Figure 8-20 gives an overview.

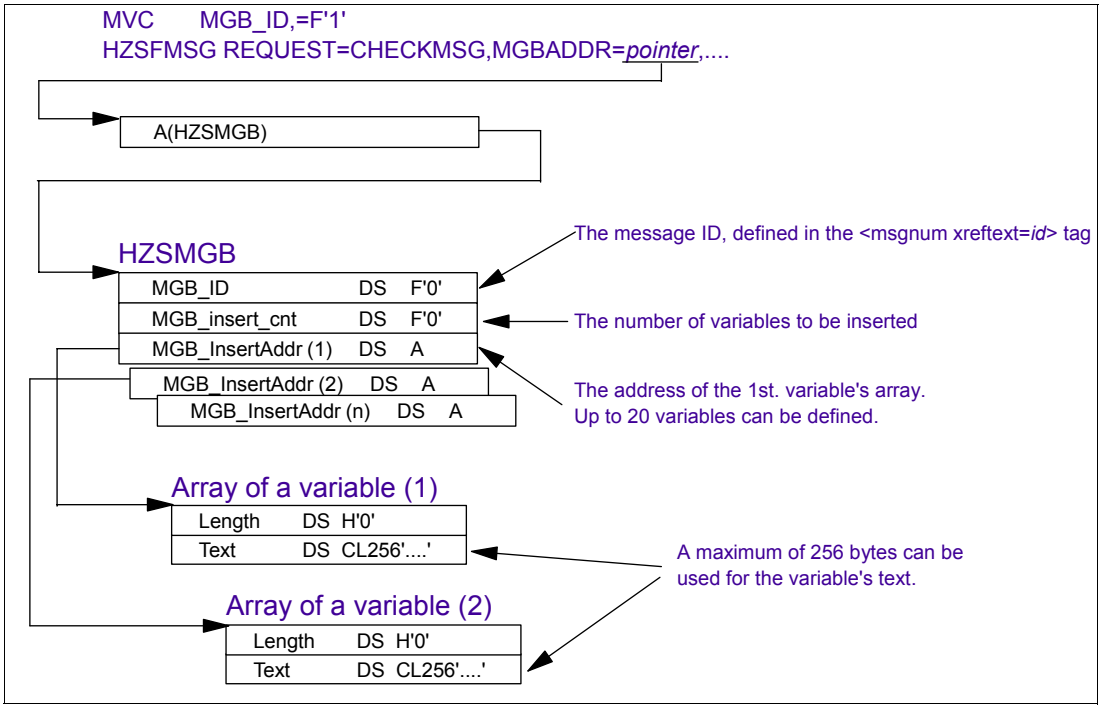


Figure 8-20 The HZSMGB control block

See also “Using variables in your messages” on page 138.

8.10.2 Create the message table for your check

Check messages are the output of your check routine. They communicate the results uncovered by a check to the user. The messages should include both the results of the check and recommendations for action to take in response to the result.

To code messages for your check, you must do the following:

1. Create a message table.
   - Create a message input data set that contains both message text and explanations for the check.
   - Create a setup data set that is customized for your check. The setup data set contains entries for symbols that you create for use in your messages.
   - Generate the messages into an assembler CSECT by using the HZSMSGEN REXX exec.
   - Compile and link the message CSECT to create the message table load module.
2. Define the message variables for check messages. The variables will resolve at run time as the message is issued.

Creating the message input data set and the setup data set

In the message input data set you provide all the messages your check routine wants to issue. You define both the message text and the explanation for each message. The setup data set contains all your variables and the replacement texts you want to provide. You must code all your messages with tags. Once you have created the message input data set and the setup data set, you must use the REXX utility HZSMSGEN to generate the assembler

CSECT of the message table. You can use the HZSMSGNJ member in SYS1.SAMPLIB, which invokes the REXX utility HZSMSGEN.

Figure 8-21 shows the process of creating a message table.

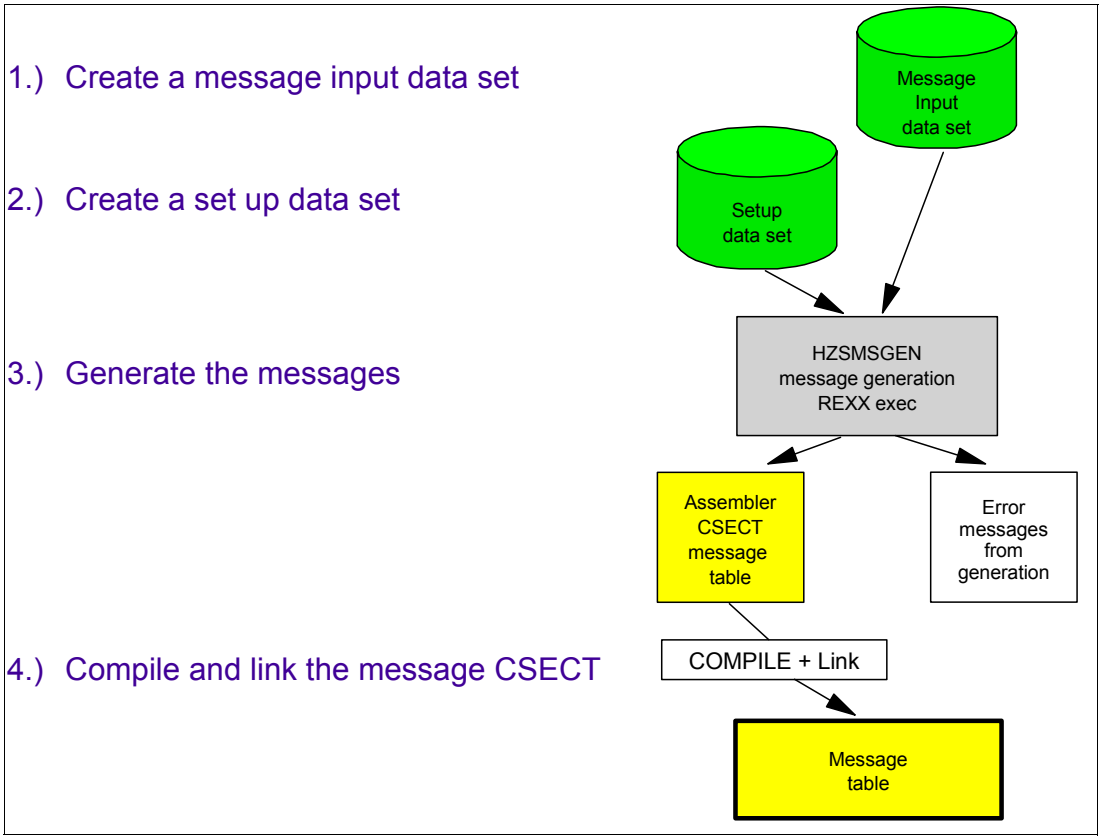


Figure 8-21 Process of creating a message table

Check routine messages

A check routine can issue four types of messages:

- Exception messages
- Information messages
- Report messages
- Debug messages

Table 8-8 gives an overview of the message types and their meanings.

Table 8-8 The message types for IBM Health Checker for z/OS

Message type **<msg class=*msgtype*>**	**Description**
Exception messages ► <msg class=exception> ► <msg class=exception_nocriteria> ► <msg class=exception_checkcriteria	An exception message notifies the installation that action is required because a check routine found a deviation from a best practices. For an exception message, a WTO including both message text and explanation. Therefore the messages must include: ► Results of the check ► Recommendation actions ► Details and reference information to help user take action.
Information messages <msg class=information>	A check can issue information messages either as a one line non-exception check result or as the first line for a longer report. If it the first line of a report, the report can be documented under the information message number.
Report messages <msg class=report>	A check issues report messages as single lines of data to the message buffer under an information message. A report contains non-exception check results.
Debug messages <msg class=debug>	A check issues debug messages when the check is in debug mode to aid in testing and diagnosis.

Example of an exception message

Figure 8-22 is an example of an exception message from a user-written check routine.

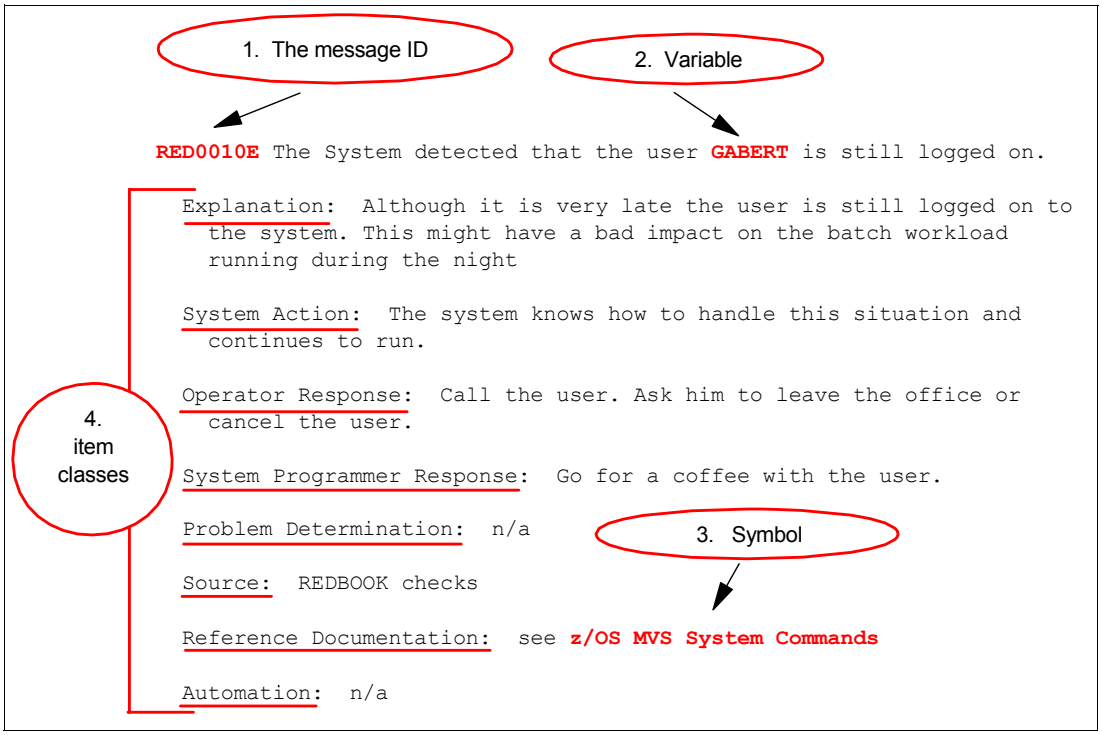


Figure 8-22 Example of an exception message output

Example of exception message coding

Figure 8-23 shows how we coded the message. As shown, we coded an exception message, indicated by:

```
<msg class=exception>
```

The following is an explanation of the most important options:

1. The message ID is coded with the tag:

   ```
   <msgnum xreftext=001>RED0010E</msgnum>
   ```

   The parameter `xreftext=001` is the message ID in the message table. If we want to issue this message in our check routine with the HZSFMSG macro, we specify this value in the HZSMGB control block we pass to the macro as a parameter.

2. To make the message flexible we used a variable in the message text:

   ```
   <mv xreftext=OUTLEN(8)>userid</mv>
   ```

   The parameter `xreftext=OUTLEN(8)` specifies the length of the variable and userid is the variable's name. In the check routine we specified the replacement text of the variable in the HZSMGB control block we pass to HZSFMSG macro.

3. The symbol `&book1` must be defined as an entity in the setup data set. It is a fixed value and cannot be modified by the check routine.

4. As you can see, there are different message item classes (such as explanation, sysact, oresp), indicated by the tag:

   ```
   <msgitem class=class>
   ```

:

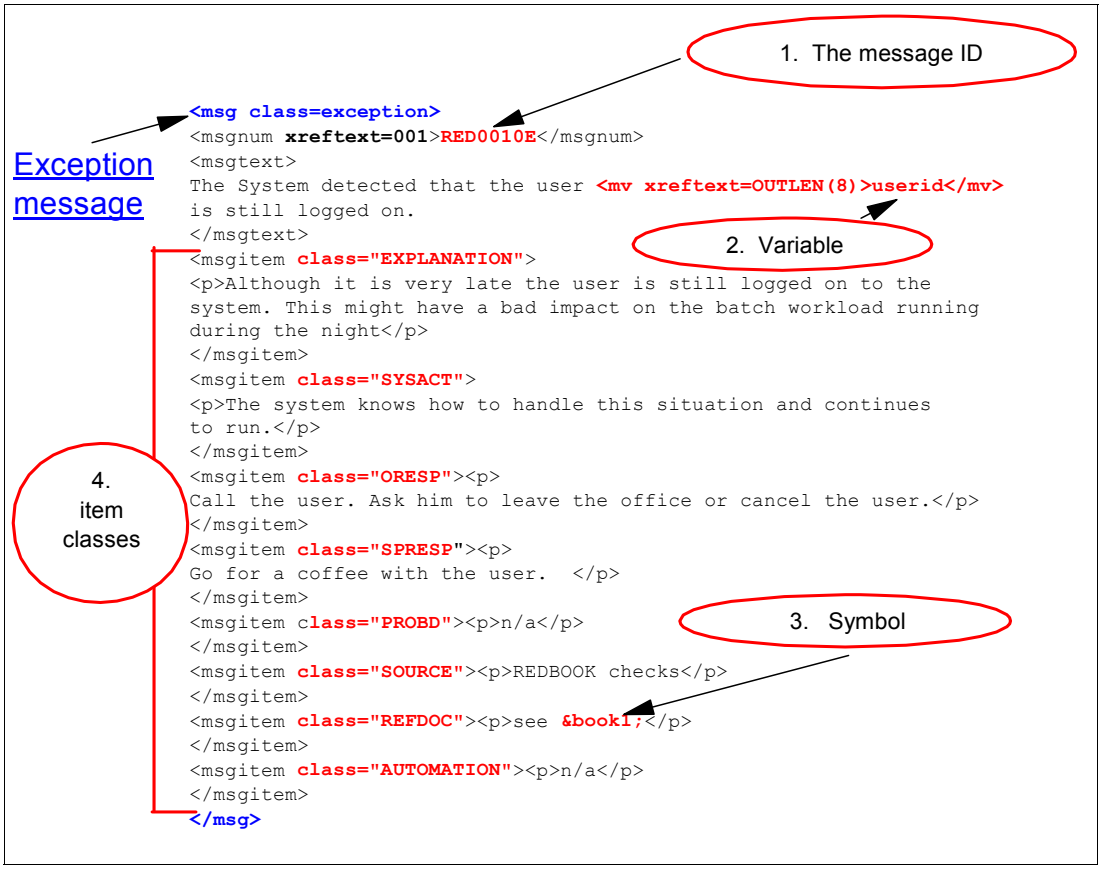


Figure 8-23 Example of an exception message source code in the input data set

Figure 8-24 shows the setup data set to resolve the symbols.

:

```
//HZSSDSN  DD  DATA
<!--                                                                  -->
<!-- HZSSDSN is used by the check writer to define symbol             -->
<!-- definitions that are resovled during message generation          -->
<!--                                                                  -->
<!ENTITY book1 "z/OS MVS System Commands">
```

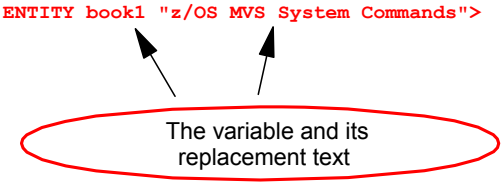


Figure 8-24 Example of a setup data set

Example of an information message

Figure 8-25 is an example of an information message.

```
RED0100I The user GABERT1 is not logged on to the system.
```

Figure 8-25 Example of an information message output

The source code for this message is shown in Figure 8-26. As you can see, we coded an information message, indicated by:

```
<msg class=information>
```

You do not need to specify the length of the variable in information messages.

For more details about the message input data set, the tags used in the message input data set, and the setup data set, see *z/OS IBM Health Checker for z/OS User's Guide,* SA22-7994.

Figure 8-26 Example of an information message source code in the input data set

Using symbols in your messages

You can use symbols in your messages. Symbols are resolved to text when you generate the input data set with the REXX utility HZSMSGEN. There are several types of symbols that you can use:

- Predefined system symbols set by IBM Health Checker for z/OS.
- Symbols defined in the HZSADDCHECK exit routine or HZSPRMxx parmlib member.
- Symbols defined in the setup data set.

Symbols are specified in the input data set as follows:

```
&symbolname;
```

The symbol is delimited by an ampersand (&) and a semicolon. Symbols defined in the setup data set are defined as entities, as shown in Figure 8-27 on page 138.

```
<!--                                                                  -->
<!-- HZSSDSN is used by the check writer to define symbol             -->
<!-- definitions that are resolved during message generation          -->
<!--                                                                  -->
<!ENTITY book1 "z/OS MVS System Commands">
<!ENTITY book2 "z/OS MVS Initialization and Tuning Guide">
<!ENTITY book3 "z/OS MVS Initialization and Tuning Reference">
.
.
.
```

Figure 8-27 Symbols in the setup data set

Predefined system symbols

Figure 8-28 identifies the predefined system symbols you can use in your messages. These symbols are set by IBM Health Checker for z/OS.

Predefined symbol	Maximum number of characters	Symbol resolves to
&hzs;	38	IBM Health Checker for z/OS and Sysplex
&hzsproc;	8	The name of the start up procedure for IBM Health Checker for z/OS and Sysplex
&hzssysname;	8	System name
&hzssysplex;	8	Sysplex name
&hzsreason;	256	User or component reason from HZSPRMxx parmlib member or the HZSADDCHECK exit routine.
&hzsexitrtn;	8	The name of the HZSADDCHECK exit routine.
&hzsrsnsource;	16	Resolves to 'Installation' or &hzsowner; to indicate whether the defaults from the HZSADDCHECK exit routine or user overrides are in effect
&hzssev;	6	Resolves to the severity set defined at runtime from either the HZSPRMxx parmlib member or the HZSADDCHECK exit routine (HI I MEDIUM I LO)
&hzsparms;	126	Active check parameters
&hzsckname;	32	The check name, as defined in the HZSADDCHECK exit routine
&hzsowner;	16	The check owner, which is the component or subsystem as defined in the HZSADDCHECK exit routine. For example, &owner; might resolve to OEM, IBMGRS, IBMRSM, IBMXCF, or IBMUSS.
&hzsdate;	10	The current system date in for form `dd mmm yyyy` :
&hzsgmttime;	16	The current system GMT time is displayed in the form: `mm/dd/yyyy hh:mm:ss.tttttt`
&hzslocaltime;	16	The current system time is adjusted to local time and displayed in the form : `mm/dd/yyyy hh:mm:ss.tttttt`
&rbl;		The ever useful blank line. &rbl; must be the only value on a line.
>		Greater than symbol, >
<		Less than symbol, <

Figure 8-28 Predefined system symbols

Using variables in your messages

You can use dynamic variables for your messages. Dynamic variables are resolved at runtime. You must define the value for each variable in the HZSMGB control block which

contains an array of pointers to variables. To use dynamic variables you must do the following:

- Use the <mv>...</mv> tag to define a variable in the message input data set.
- Provide the pointer to the variable's text in the HZSMGB control block in your check routine.

The maximum number of variables that can be defined for one HZSFMSG iteration is 20.

See also "The HZSMGB control block" on page 131.

8.10.3 The HZSADDCHECK exit routine

You describe the information about your check in a simple authorized HZSADDCHECK exit routine. This routine runs in the IBM Health Checker for z/OS address space. Using the HZSADDCK macro, your HZSADDCHECK exit routine specifies the default values for the check, such as the check routine name, message table name, the check interval and the reason for the check.You can say this exit routine is an introduction of your check to IBM Health Checker for z/OS. The exit routine must be reentrant.

The HZSADDCK macro

To pass all the information about your check you must use the HZSADDCK macro within your HZSADDCHECK exit routine.

```
HZSADDCK
   ,CHECKOWNER=checkowner
   ,OWNER=owner
   ,CHECKNAME=checkname
   ,NAME=name
   ,CHECKROUTINE=checkroutine
   ,ENTRYCODE=entrycode
   ,EXITRTN=exitrtn
   ,MSGTBL=msgtbl
   ,DATE=date
   ,REASON=reason
   ,REASONLEN=reasonlen
   ,PARMS=parms
   ,PARMSLEN=parmslen
   ,LOCAL
   ,GLOBAL
   ,ACTIVE
   ,INACTIVE
   ,SEVERITY=severity
   ,INTERVAL=interval
   ,HOURS=hours
   ,MINUTES=minutes
   ,USS=uss
   ,RETCODE=retcode
   ,RSNCODE=rsncode
   ,PLISTVER=plistver
   ,MF=macroform
```

Figure 8-29 The HZSADDCK macro

The parameters on the HZSADDCK macro are as follows:

- CHECKOWNER=*checkowner*
  OWNER=owner

  A required input parameter that specifies the owner of the check being added. The check owner and check name identify the check. IBM recommends that you use your company name followed by the short component name (for example, IBMGRS) as the owner. Upper and lower case alphabetic characters (A-Z, a-z), numerics (0-9), national characters (@,$,#) and the underscore (_) are allowed. Lower case alphabetic characters are folded to upper case and are treated as equivalent to their corresponding upper case value.

  To code: Specify the RS-type address, or address in register (2)-(12), of a 16-character field.

- CHECKNAME=*checkname*
  NAME=*name*

  A required input parameter that specifies the name of the check being added. IBM recommends using the naming convention of a short component reference followed by a descriptive title (for example, GRS_MODE). Upper and lower case alphabetic characters (A-Z, a-z), numerics (0-9), national characters (@,$,#) and the underscore (_) are allowed. Lower case alphabetic characters are folded to upper case and are treated as equivalent to their corresponding upper case value.

  To code: Specify the RS-type address, or address in register (2)-(12), of a 32-character field.

- CHECKROUTINE=chec*kroutine*

  A required input parameter that specifies the module name of the check. The system gives control to the entry point of this module to run the check. The check routine module must be in an APF-authorized library.

  To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

- ENTRYCODE=en*trycode*

  An optional input parameter that specifies a unique check entry value when the same check routine will be accessed by multiple checks. This value is passed to the check routine in the field Pqe_EntryCode.

  To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

- EXITRTN=*exitrtn*

  A required input parameter that specifies the name of the exit routine that invoked this HZSADDCK request.

  To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

- MSGTBL=*msgtbl*

  A required input parameter that specifies the module name of the message table that will be used when generating messages for the check. The message table must be built using the HZSMSGEN REXX exec. The message table module must be in an APF-authorized library.

  To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

- DATE=*date*

  A required input parameter, date (its format is YYYYMMDD) that indicates when the default values for the check were defined. When two HZSADDCK requests are received

with the same check owner and check name, the request with the latest date will be honored. When the date provided on a matching POLICY UPDATE or POLICY DELETE statement is older than this date, that policy statement is not applied to this check.

To code: Specify the RS-type address, or address in register (2)-(12), of an 8-character field.

- REASON=*reason*

  A required input parameter that indicates what the check routine validates. The text is limited to 126 characters.

  To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

- REASONLEN=*reasonlen*

  A required input parameter, length of the Reason text. It must be in the range 1 through 126.

  To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field.

- PARMS=*parms*
  PARMS=NO_PARMS

  An optional input parameter that specifies the default parameters for the check. The length of the parameter string is specified by the PARMSLEN parameter. Alphanumeric or national characters separated by commas are the standard form of expressing check parameters. IBM recommends that each parameter be of the form “keyword(value)” and that multiple parameters be separated from each other by a comma. An example of a parameter string following that protocol is “MAXLEN(8),MINLEN(1)”. Although the parameters are not checked when the check is added, the check routine itself will likely do so. The default is NO_PARMS.

  To code: Specify the RS-type address, or address in register (2)-(12), of a character field.

- PARMSLEN=*parmslen*

  When PARMS=parms is specified, a required input parameter, length of the default parameters for each check. The length must be in the range 0 through 256.

  To code: Specify the RS-type address, or address in register (2)-(12), of a fullword field. parmslen must be in the range 0 through 256.

- LOCAL

  An optional input parameter that indicates the check should run on this system.

  To code: Specify a value.

- GLOBAL

  An optional input parameter that indicates the check should run on only one system in a sysplex. The system on which the check runs is designated as the global system for that check. Serialization for the global check is accomplished via exclusive ownership of SCOPE=SYSTEMS ENQ with QNAME SYSZHZS and RNAME checkowner.checkname.

  To code: Specify a value.

- ACTIVE

  An optional input parameter that indicates the check should run when it is added to the system.

  To code: Specify a value.

- INACTIVE

  An optional input parameter that indicates the check should not run until the state is changed to active.

To code: Specify a value.

- SEVERITY=LOW
  SEVERITY=MED
  SEVERITY=HI

  A required parameter that indicates the severity assigned to the check.

  - **LOW** indicates that this is a low-severity check. When a low-severity check detects an exception, an informational WTO is issued.
  - **MED** indicates that this is a medium-severity check. When a medium-severity check detects an exception, an eventual action WTO is issued.
  - **HI** indicates that this is a high-severity check. When a high-severity check detects an exception, a critical eventual action WTO is issued.

- INTERVAL=ONETIME
  INTERVAL=TIMER

  A required parameter that specifies the time interval for the next running of the check.

  - **ONETIME** indicates that the check should run once. It will not be rescheduled.
  - **TIMER** indicates that a timer is used to reschedule the check. The number of hours is combined with the number of minutes to determine how long after the completion of the check routine's running the next running of the check routine should occur. When both the hours and minutes values are zero, the system treats this as if INTERVAL=ONETIME had been specified.

- HOURS=*hours*
  HOURS=0

  When INTERVAL=TIMER is specified, an optional input parameter that specifies the number of hours. It must be in the range 0 through 999. The default is 0.

  To code: Specify the RS-type address of a halfword field. hours must be in the range 0 through 999.

- MINUTES**=***minutes*
  MINUTES=0

  When INTERVAL=TIMER is specified, an optional input parameter that specifies the number of minutes. It must be in the range 0 through 59. The default is 0.

  To code: Specify the RS-type address of a halfword field. Minutes must be in the range 0 through 99.

- USS=NO
  USS=YES

  An optional parameter that indicates whether the check uses UNIX System Services. This information is used when UNIX System Services itself is shut down, at which time IBM Health Checker for z/OS will wait for the completion of the running of any check that has indicated it uses UNIX System Services before allowing the UNIX System Services shutdown to complete. Also, when UNIX System Services are not available, checks that have indicated they use those services are not run. Thus, indicating "YES" if the check actually does not use UNIX System Services could delay USS shutdown and would result in the check's not being run when those services are not available. The default is USS=NO.

  - **NO** indicates the check does not use UNIX System Services.
  - **YES** indicates the check does use UNIX System Services.

- RETCODE=*retcode*

  An optional output parameter into which the return code is to be copied from GPR 15.

  To code: Specify the RS-type address of a fullword field, or register (2)-(12).

- RSNCODE=*rsncode*

  An optional output parameter into which the reason code is to be copied from GPR 0.

  To code: Specify the RS-type address of a fullword field, or register (2)-(12).

- PLISTVER=IMPLIED_VERSION
  PLISTVER=MAX
  PLISTVER=0

  An optional input parameter that specifies the version of the macro. PLISTVER determines which parameter list the system generates. PLISTVER is an optional input parameter on all forms of the macro, including the list form. When using PLISTVER, specify it on all macro forms used for a request and with the same value on all of the macro forms. The values are:

  – **IMPLIED_VERSION** which is the lowest version that allows all parameters specified on the request to be processed. If you omit the PLISTVER parameter, IMPLIED_VERSION is the default.

  – **MAX** if you want the parameter list to be the largest size currently possible. This size might grow from release to release and affect the amount of storage that your program needs. If you can tolerate the size change, IBM recommends that you always specify PLISTVER=MAX on the list form of the macro. Specifying MAX ensures that the list-form parameter list is always long enough to hold all the parameters you might specify on the execute form, when both are assembled with the same level of the system. In this way, MAX ensures that the parameter list does not overwrite nearby storage.

  – **0** if you use the currently available parameters.

  MF=S
  MF=(L,*list addr)*
  MF=(L,*list addr,attr)*
  MF=(L,*list addr,*0D*)*
  MF=(E,*list addr)*
  MF=(E,*list addr,*COMPLETE*)*

  – **MF=S** specifies the standard form of the macro, which builds an inline parameter list and generates the macro invocation to transfer control to the service. MF=S is the default.

  – **MF=L** specifies the list form of the macro. Use the list form together with the execute form of the macro for applications that require reentrant code. The list form defines an area of storage that the execute form uses to store the parameters. Only the PLISTVER parameter may be coded with the list form of the macro.

  – **MF=E** specifies the execute form of the macro. Use the execute form together with the list form of the macro for applications that require reentrant code. The execute form of the macro stores the parameters into the storage area defined by the list form, and generates the macro invocation to transfer control to the service.

  – ***list addr*** is the name of a storage area to contain the parameters. For MF=S and MF=E, this can be an RS-type address or an address in register (1)-(12).

  – ***attr*** is an optional 1- to 60-character input string that you use to force boundary alignment of the parameter list. Use a value of 0F to force the parameter list to a word

boundary, or 0D to force the parameter list to a doubleword boundary. If you do not code attr, the system provides a value of 0D.

– **COMPLETE** specifies that the system is to check for required parameters and supply defaults for omitted optional parameters.

Figure 8-30 is an example of an HZSADDCK iteration.

```
.
.
.
HZSADDCK OWNER==CL16'REDBOOK',                                  X
      NAME==CL32'RED_CHECK_LOGON',                             X
      CHECKROUTINE==CL8'REDCHK01',                             X
      MSGTBL==CL8'REDMSG1',                                    X
      EXITRTN==CL8'REDAC001',                                  X
      DATE==CL8'20050518',                                     X
      REASON==CL126'Playing with z/OS V1R7',                   X
      REASONLEN==F'126',                                       X
      SEVERITY=HI,                                             X
      INTERVAL=TIMER,MINUTES==H'5',                            X
      PARMS==CL8'GABERT',PARMSLEN=8,                           X
      MF=(E,WHZSADDCK)
.
.
.
*********************************************************************
*        Work area DSECT
*********************************************************************
STATWORK DSECT ,
SAVEAREA DS   18F                    Save area
         HZSADDCK MF=(L,WHZSADDCK)
STATWLEN EQU  *-STATWORK             Length of local storage area
```

Figure 8-30 Example of an HZSADDCK macro call

For more information about the HZSADDCK macro and a detailed description about the parameters see *z/OS IBM Health Checker for z/OS User's Guide,* SA22-7994.

Adding your HZSADDCHECK exit routine

After you write your exit routine you must link-edit it reentrant into an APF-authorized link library. Then you must add your exit routine to the HZSADDCHECK exit and have the system call the exit to run the exit routines which adds your check to IBM Health Checker for z/OS.

While you are testing your check you can use operator commands to do this:

1. Issue the **setprog** command.

   ```
   SETPROG EXIT,ADD,EXITNAME=HZSADDCHECK,MODNAME=loadmodule
   ```

2. Issue the **modify** command to add your check routine to IBM Health Checker for z/OS.

   ```
   F hzsproc,ADDNEW
   ```

   If your HZSADDCHECK exit routine ended successfully you will see your check in SDSF on the CK panel.

Once you have finished your testing you can use the PROGxx EXIT statement to define your HZSADDCHECK exit routine to the HZSADDCHECK dynamic exit during IPL. When IBM

Health Checker for z/OS comes up, it activates all checks, so you do not need to code for activating your checks.

8.10.4 Example of a IBM Health Checker for z/OS check

Appendix B provides sample code of an IBM Health Checker for z/OS check.

8.11 Using SDSF to manage checks

There is a new panel in SDSF for Health Checker (CK). It shows information from IBM Health Checker for z/OS and Sysplex, which tests for key z/OS and sysplex values at specified intervals.

Enter **CK** from the SDSF command line to open the Health Checker panel, as shown in Figure 8-31. Like all SDSF primary displays, **CK** can also be accessed from a pull-down list when SDSF is running as an ISPF dialog. To display the IBM Health Checker for z/OS CK action characters, use the **SET ACTION SHORT** or **SET ACTION LONG** command.

```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF HEALTH CHECKER DISPLAY  SC70                      LINE 1-26 (54)
COMMAND INPUT ===>                                        SCROLL ===> CSR
NP   NAME                              CheckOwner       State              Status
     CNZ_AMRF_EVENTUAL_ACTION_MSGS     IBMCNZ           ACTIVE(ENABLED)    SUCCES
     CNZ_CONSOLE_MASTERAUTH_CMDSYS     IBMCNZ           ACTIVE(ENABLED)    SUCCES
     CNZ_CONSOLE_MSCOPE_AND_ROUTCODE IBMCNZ           ACTIVE(ENABLED)    EXCEPT
     CNZ_CONSOLE_ROUTCODE_11           IBMCNZ           ACTIVE(ENABLED)    EXCEPT
     CNZ_EMCS_HARDCOPY_MSCOPE          IBMCNZ           ACTIVE(ENABLED)    SUCCES
     CNZ_EMCS_INACTIVE_CONSOLES        IBMCNZ           ACTIVE(ENABLED)    SUCCES
     CNZ_SYSCONS_MASTER                IBMCNZ           ACTIVE(ENABLED)    SUCCES
     CNZ_SYSCONS_MSCOPE                IBMCNZ           ACTIVE(ENABLED)    SUCCES
     CNZ_SYSCONS_PD_MODE               IBMCNZ           ACTIVE(ENABLED)    SUCCES
     CNZ_SYSCONS_ROUTCODE              IBMCNZ           ACTIVE(ENABLED)    SUCCES
     CNZ_TASK_TABLE                    IBMCNZ           ACTIVE(ENABLED)    SUCCES
     GRS_CONVERT_RESERVES              IBMGRS           ACTIVE(ENABLED)    EXCEPT
     GRS_EXIT_PERFORMANCE              IBMGRS           ACTIVE(ENABLED)    EXCEPT
```

Figure 8-31 SDSF Health Checker display using the CK command

You can also:

- Browse a check using the **S** action character as shown in Figure 8-32 on page 146. When you are running SDSF under ISPF, you can also use the **SB** or **SE** action characters to browse the output with ISPF browse or edit.
- Limit the checks shown with the **FILTER** and **S** commands. For example, **S GRS*** would show all checks that start with GRS. Reset the checks shown by typing **S** without parameters.

You can display sysplex-wide checks using SDSF's server and WebSphere MQ. Without the MQ support, SDSF shows checks for the system you are logged on to.

To view action characters, fields, commands, and other things about Health Checker use the online help for command **CK**.

```
 Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY CNZ_EMCS_HARDCOPY_MSCOPE      LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
******************************** TOP OF DATA ********************************
CHECK(IBMCNZ,CNZ_EMCS_HARDCOPY_MSCOPE)
START TIME: 05/24/2005 09:54:26.795999
CHECK DATE: 20040816  CHECK SEVERITY: MEDIUM

CNZHS0006I There are no EMCS consoles with a multi-system message scope
that are receiving the hardcopy message set.

END TIME: 05/24/2005 09:54:26.796840  STATUS: SUCCESSFUL
```

Figure 8-32 Viewing a check

9

z/OS UNIX for z/OS V1R7

This chapter describes enhancements introduced in z/OS V1R7 UNIX System Services, specifically the following:

- Logical file system (LFS) support for HFS to zFS migration
- pax enhancements for migration from HFS to zFS
- Dynamic service activation
- ISHELL enhancements
- Miscellaneous enhancements
- Mounting file systems with SET OMVS

9.1 LFS support for HFS to zFS migration

The logical file system (LFS), shown in Figure 9-1, plays a large role in the conversion of HFS to zFS data sets. The migration to zFS file systems needs to be well planned since it will take significant effort to migrate all of the data sets from HFS file systems to zFS file systems. No capability exists to convert a file system in place to zFS or to use an HFS data set with zFS. From this, it follows that file systems that are used in a read/write mode will have to be made unavailable while being migrated to zFS and that two data sets will exist at least during the migration.

The unavailability of the data for read/write must be planned for, as well as ensuring the availability of adequate space for two file systems that are about the same size.

Given that there are now two data sets, it is likely to want to have different naming standards for the zFS data sets and their HFS data set counterparts or you may prefer to keep the names just as they are. In general, if the zFS data set is named to be the same as the HFS data set, no changes are needed in this area.

z/OS V1R7 provides a migration utility that is invoked via a REXX exec called BPXWH2Z. This is a tool that may help you migrate your HFS file systems to zFS file systems. It can migrate one file system at a time, or build a list and do many. This utility must run from ISPF to set up the file system migrations, but the actual migration work can be run in UNIX background as well as foreground.

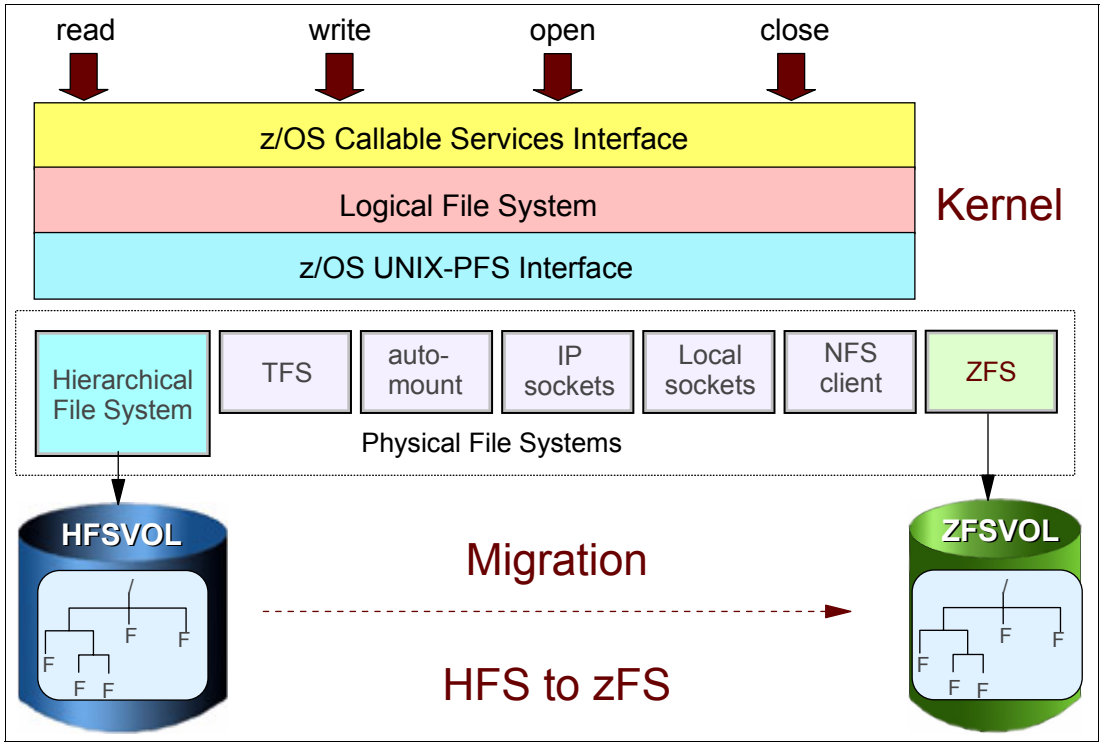


Figure 9-1 z/OS UNIX communication between LFS and PFS

9.1.1 Migrations steps for HFS to zFS

The following steps are recommended to migrate an HFS file system to zFS. They can be done manually, or the migration tool can be used since it will perform all of the following steps.

- If the HFS file system is mounted, either unmount it or switch it to read-only mode.
  - The migration tool switches it to R/O.
- If the file system is not mounted, make a directory for it and mount it in R/O mode.
  - The migration tool creates a temporary directory in /tmp for the mountpoint and deletes it when the migration is complete.
- Create a temporary directory for the zFS file system.
  - The migration tool creates a temporary directory in /tmp for the mountpoint and deletes it when the migration is complete.
- Check the allocation attributes of the HFS file system.
- Define a new zFS file system with the appropriate allocation attributes.
  - The tool defaults these to the same attributes as the HFS file system if its utilization is below about 75 percent; otherwise, it adds about 10 percent (below 90 percent) or 20 percent (above 90 percent). It is difficult to determine the exact size necessary for a zFS file system to contain all of the data within an HFS file system. Depending on the number of files, directories, ACLs, symlinks, and file sizes, zFS can consume either more or less space than HFS. zFS also has a log in the aggregate. For general purpose file systems, it appears that they consume about the same amount of space.
- Mount the zFS file system on its temporary mountpoint (a pre-existing zFS cannot already be mounted).
- To use the pax utility to copy the HFS contents to the zFS, enter the shell, change directory to the HFS mountpoint, and run the pax utility. The **pax** command will look something like:

```
pax -rw -X -E . /tmp/zfsmountpoint
```

- If the HFS contains active mountpoints, do a **mkdir** for each of these in the zFS file system and set directory attributes as required.
- Set attributes for the zFS root based on the HFS root that was copied.

Note: This must include all extended attributes, ACLs, owner, group, code page, mode bits, or SECLABEL.

- Unmount the zFS file system.
- Unmount the HFS file system.

Note: If it contains active mountpoints, those file systems and any hierarchy below them must be unmounted first.

- Remove any temporary directories used as mountpoints.
- Rename the data sets as appropriate and ensure all mount scripts and policies have been updated as needed.
  - The migration tool does not modify or search for any type of mount scripts or policies.
- If the HFS file system was originally mounted, mount the zFS file system in that same location, along with any hierarchy that also had to be unmounted to unmount the HFS.

9.1.2 Migration of the root

If you have not migrated the root file system during a ServerPac install, you can use the migration tool to copy the root from HFS to zFS, as follows:

- Set attributes for the zFS root based on the HFS root that was copied.

  Note: This must include all extended attributes, ACLs, owner, group, code page, mode bits, and SECLABELs.

- Unmount the zFS file system.
- Unmount the HFS file system.

  Note: If it contains active mountpoints, those file systems and any hierarchy below them must be unmounted first.

- Remove any temporary directories used as mountpoints.
- Rename the data sets as appropriate and ensure all mount scripts and policies have been updated as needed.
  - The migration tool does not modify or search for any type of mount scripts or policies.
- If the HFS file system was originally mounted, mount the zFS file system in that same location, along with any hierarchy that also had to be unmounted to unmount that HFS.

9.1.3 Migration and coexistence considerations

Currently, OMVS initialization suspends until the recalls are complete for HFS data sets and mounts complete asynchronously to the processing for zFS data sets. Parmlib mounts in z/OS V1R7 is now changed to suspend for zFS mounts and fail HFS mounts for migrated data sets.

If multi-file system aggregates are in use, it is necessary to ensure that there are not any HFS data sets cataloged with the same name as any file system in a multi-file system aggregate. The mount direction locates the HFS data set in the catalog and directs or redirects the mount to HFS.

Non-parmlib zFS mounts are done asynchronously. There is an accommodation in automount for HFS so that an HSM migrated data set is recalled prior to entering the mount flow.

File system names

If the zFS data set names are to be the same as the HFS data set names, it is likely that scripts or policies that are used to mount file systems would not need to be changed. Automount polices that specify allocany or allocuser will continue to allocate file systems based on the file type specified. So, policies will need to change if new file systems are to be zFS file systems.

Since this is a migration scenario, mount processing first checks for the ZFS file system type along with the assumption that if that data set exists, the migration has been done. In order to revert back to using HFS, that data set must be renamed or removed.

Mount processing (HFS or zFS)

HFS and zFS are now generic file system types. They can be used for either HFS or ZFS. Mount processing first searches for a data set matching the file system name, as follows:

- If the data set is not an HFS data set and zFS has been started, the file system type is changed to ZFS and the mount proceeds to zFS.
- If the data set is not found, the mount proceeds to zFS (assuming zFS is started). If zFS is not started, the type is not changed and the mount proceeds to HFS.

A new place holder, ///, is created and represents the string HFS or ZFS as appropriate if the file system type is HFS, as shown in the following mount statement and mount command:

```
MOUNT    FILESYSTEM('ZOSR17.MAN.///')
   TYPE(HFS)
   MODE(READ)
   MOUNTPOINT('/usr/man')
```

Mount processing first substitutes ZFS for the place holder, then checks if the data set exists. If it does not, HFS is used in the placeholder and again a check is run to determine if the data set exists. The mount is directed either to zFS or HFS depending on which data set was found.

9.1.4 Using BPXWH2Z migration tool

If you use the BPXWH2Z migration tool with no arguments, it prompts for an HFS file system name. You can also specify one or more file system names as an argument. The names can be simple names or patterns that you might use for the ISPF data set list panel. When you invoke BPXWH2Z, the ISPF panel shown in Figure 9-2 on page 153 is displayed.

Using BPXWH2Z

From the ISPF Option panel, specify **BPXWH2Z**. When you specify **BPXWH2Z**, you can also specify option flags. The option flags are preceded by a -, and then followed immediately by the flags. The available flags and their meanings are:

-v Additional messages are issued at various points in processing.

-c Summary information goes to a file when background (**bg**) is selected once you are in the application. If this is not specified, summary information goes to the console.

Command line examples from ISPF Option 6 are:

```
bpxwh2z -v
bpxwh2z -c
```

In additional to the option flags, you can specify a command argument that is a character substitution string with the **BPXWH2Z** command.

You can preallocate the zFS file system or specify a character substitution string. This causes the zFS data set to not be renamed by the migration tool. To rename the zFS file system use a substitution string as the first argument on the command line as follows:

```
bpxwh2z /fromstring/tostring/ datasetname
```

- For example, if your data set is OMVS.DB2.HFS and you want your zFS data set to be named OMVS.DB2.ZFS, you can specify the following command argument:

  ```
  bpxwh2z /hfs/zfs/ omvs.db2.hfs
  ```

  Be careful; all strings matching the *from string* in the data set name are replaced.

BPXWH2Z panels

By default, the panels are primed such that your HFS data set is renamed with a .SAV suffix and the zFS data set is renamed to the original HFS name. The documentation about these panels is available on line by pressing PF1.

Use the **End** or **Cancel** commands to exit from this tool. No changes or allocations will have been done.

This migration process makes use of /tmp. The file system containing /tmp should not be migrated with this utility. In general, if you are using HFS as your /tmp, it would be best to start with a clean zFS. Also, you should not migrate your /dev file system. Start with a clean zFS. If you are migrating your root file system you should do this in the foreground. As part of migration, all file systems will be unmounted, including /tmp and /dev, which will keep you from seeing the status of the migration if run in background.

This tool can display many different messages. These messages are documented only in the on-line help file.

To begin the migrations, enter either the **FG** or **BG** command to run the migration in foreground or background, respectively.

Data set migration considerations

Consider the following options when migrating data sets from HFS to zFS:

- After an HFS file system is migrated to zFS, the HFS data set will be renamed to the name in the “Save HFS as” field, and the zFS data set will be renamed to the old HFS data set name. If you do not want data sets to be renamed, set this field to the HFS data set name or clear the field and the tool will set it to the HFS name.
- If your zFS file system is not pre allocated, it is allocated based on the attributes for the HFS data set. You will have an opportunity to alter any of these attributes.
  - The size is set at the same size as your HFS if current utilization is below about 75 percent. If your HFS is multivolume you must preallocate your zFS in order to have similar attributes and multivolume.
- This utility best handles file systems that are not currently mounted or file systems that are mounted but don't have other file systems mounted below them.
  - If the file system is mounted, it is unmounted and the new zFS put in its place. If the file system does contain active mount points, this attempt to unmount that subtree to replace the HFS with the zFS and put back everything it needed to unmount.
  - Mounted file systems are switched to read-only during the migration. They should not be in use for update when you are doing this migration. The file system table display will indicate file systems that are mounted.

9.1.5 Data set migration examples

To migrate an HFS data set to zFS, type in the data set name as shown in Figure 9-2 on page 153. In this example, the HFS file system to be migrated is mounted.

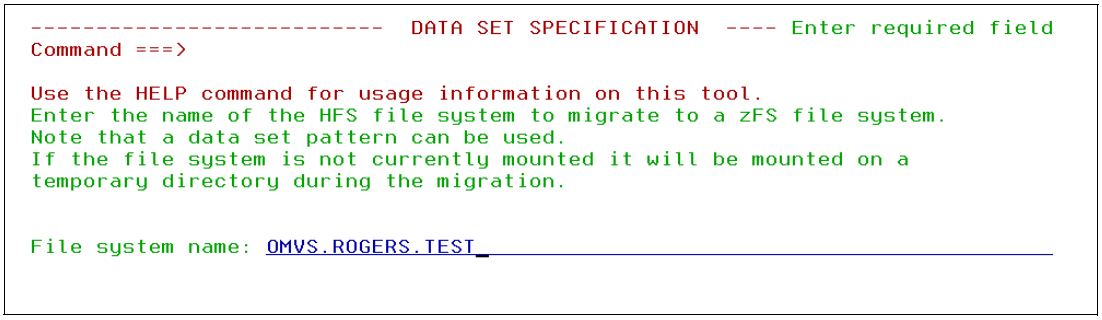


Figure 9-2 ISPF migration tool primary panel

Press Enter; the screen shown in Figure 9-3 is displayed. As shown in the figure, you can change the SMS classes and volume information for the creation of the zFS data set.

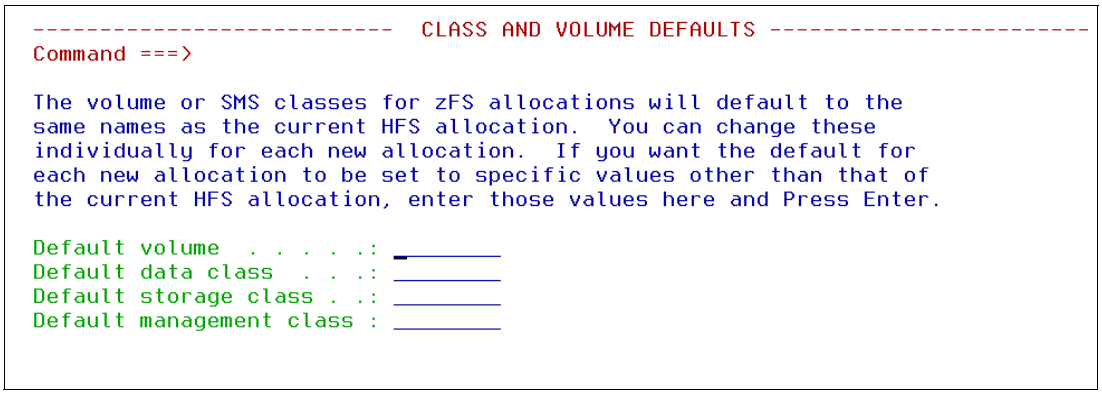


Figure 9-3 Panel to change data set attributes when creating the zFS data set

Press Enter; the screen shown in Figure 9-4 on page 154 is displayed. Here you see the defaults for saving the HFS and the initial zFS data sets. The next step is to create the new zFS data set. There are two options for doing this to begin the migrations: enter the command **FG** or **BG** to run the migration in either the foreground or background.

Note: Depending on the size of the HFS data set, the copy process may take a long time. If the migration is run in the background, the tool keeps a standard output log and also a summary log. The prefix for the pathnames is displayed.

Use the **End** or **Cancel** command to exit from this tool if no changes or allocations have been done. Note that the copy process may take a long time. If this is run in the background the tool will keep a standard output log and also a summary log. The prefix for the pathnames will be displayed.

Use of /tmp

This migration process makes use of /tmp. Make sure the following considerations are evaluated:

- The file system containing /tmp should not be migrated with this utility. In general, if you are using HFS as your /tmp, it would be best to start with a clean zFS.
- You should not migrate your /dev file system. Start with a clean zFS.

- If you are migrating your root file system, you should do this in the foreground. As part of migration all file systems will be unmounted, including /tmp and /dev, which will keep you from seeing the status of the migration if run in background.

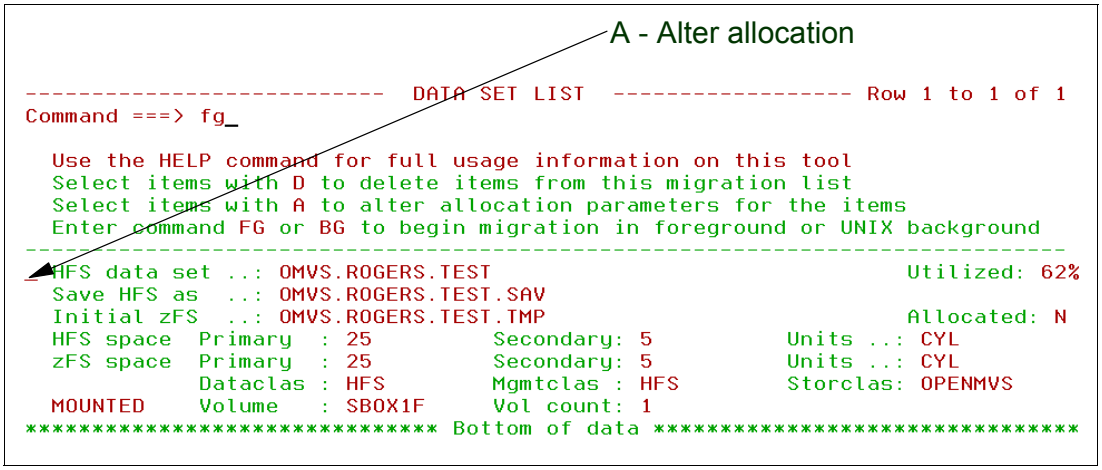


Figure 9-4 Create the zFS data set in the foreground

The following messages are seen on the screen after the creation of the zFS data set:

```
 Migrating OMVS.ROGERS.TEST
creating zFS OMVS.ROGERS.TEST.TMP
copying OMVS.ROGERS.TEST to OMVS.ROGERS.TEST.TMP Blocks to copy: 2832
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IDC0531I ENTRY OMVS.ROGERS.TEST.TMP ALTERED
IDC0531I ENTRY OMVS.ROGERS.TEST.TMP.DATA ALTERED
mount /u/rogers/tmp OMVS.ROGERS.TEST ZFS 1
***
```

Note: If the HFS file system was mounted, it will be unmounted and the new zFS put in its place. If the file system does contain active mount points, this will attempt to unmount that subtree to replace the HFS with the zFS and put back everything it needed to unmount. Mounted file systems are switched to read-only during the migration. They should not be in use for update when you are doing this migration.

Altering the allocation parameters

As shown in Figure 9-4, if you use the action character A, you can alter the space requirements for the creation of the zFS data set. The new allocation changed the zFS primary space to 35 from 25. The following messages are then seen on the screen after the creation of the zFS data set when, as in this example, the HFS file system was not mounted when migration was started:

```
Migrating OMVS.ROGERS.TEST
 creating zFS OMVS.ROGERS.TEST.TMP
 copying OMVS.ROGERS.TEST to OMVS.ROGERS.TEST.TMP Blocks to copy: 2832
 IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
 IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
 IDC0531I ENTRY OMVS.ROGERS.TEST.TMP ALTERED
 IDC0531I ENTRY OMVS.ROGERS.TEST.TMP.DATA ALTERED
 ***
```

The data set display from ISPF Option 3.4 shows the following, where you can see that the zFS data set after migration (OMVS.ROGERS.TEST.DATA) is larger than the original HFS data set:

```
OMVS.ROGERS.TEST
OMVS.ROGERS.TEST.DATA                    525    ?    1  3390
OMVS.ROGERS.TEST.SAV                     375   62    1  3390
```

Specifying a data set list for migration

You can specify a list of data sets to be migrated. After getting the list of data sets to migrate, this utility obtains data set information on each data set. The list is examined and determines that it cannot migrate data sets if they meet any of the following conditions:

- They do not exist.
- They are HSM-migrated.
- They are not HFS data sets.

Data set list example

In this example, the HFS data sets are:

```
OMVS.ROGERS.TEST1
OMVS.ROGERS.TEST2
OMVS.ROGERS.TEST3
OMVS.ROGERS.TEST4
OMVS.ROGERS.TEST5
OMVS.ROGERS.TEST6
OMVS.ROGERS.TEST7
```

To specify this list to the migration tool, see Figure 9-5.

```
---------------------------  DATA SET SPECIFICATION  ---------------------------
Command ===>

Use the HELP command for usage information on this tool.
Enter the name of the HFS file system to migrate to a zFS file system.
Note that a data set pattern can be used.
If the file system is not currently mounted it will be mounted on a
temporary directory during the migration.


File system name: omvs.rogers.test*_____________________________________________
```

Figure 9-5 Migrating a list of data sets

The resulting data set list is presented in a table, shown in Figure 9-6 on page 156. Use your normal ISPF up and down to scroll through the list. Each table entry shows the following:

- The data set names that are used
- Current HFS space utilization and space allocation
- Allocation parameters that are used to create the zFS file system

In Figure 9-6 on page 156, only the first 3 data sets of the 7 data sets are shown. At the top of the list of data sets are the three options to continue the migration. They are:

- Select items with D to delete items from this migration list.
- Select items with A to alter allocation parameters for the items.
- Enter command **FG** or **BG** to begin migration in foreground or UNIX background.

```
-------------------------  DATA SET LIST----------------------------- Row 1 to 3 of 8
 Command ===> fg
Use the HELP command for full usage information on this tool
   Select items with D to delete items from this migration list
   Select items with A to alter allocation parameters for the items
   Enter command FG or BG to begin migration in foreground or UNIX background
 ------------------------------------------------------------------------------
 _ HFS data set ..: OMVS.ROGERS.TEST                              Utilized: 62%
   Save HFS as  ..: OMVS.ROGERS.TEST.SAV
   Initial zFS  ..: OMVS.ROGERS.TEST.TMP                          Allocated: N
   HFS space  Primary  : 25         Secondary: 5          Units ..: CYL
   zFS space  Primary  : 25         Secondary: 5          Units ..: CYL
              Dataclas : HFS        Mgmtclas : HFS        Storclas: OPENMVS
   MOUNTED    Volume   : SBOX1F     Vol count: 1
 ------------------------------------------------------------------------------
 _ HFS data set ..: OMVS.ROGERS.TEST1                             Utilized: 62%
   Save HFS as  ..: OMVS.ROGERS.TEST1.SAV
   Initial zFS  ..: OMVS.ROGERS.TEST1.TMP                         Allocated: N
   HFS space  Primary  : 25         Secondary: 5          Units ..: CYL
   zFS space  Primary  : 25         Secondary: 5          Units ..: CYL
              Dataclas : HFS        Mgmtclas : HFS        Storclas: OPENMVS
              Volume   : SBOX1E     Vol count: 1
 ------------------------------------------------------------------------------
 _ HFS data set ..: OMVS.ROGERS.TEST2                             Utilized: 62%
   Save HFS as  ..: OMVS.ROGERS.TEST2.SAV
   Initial zFS  ..: OMVS.ROGERS.TEST2.TMP                         Allocated: N
   HFS space  Primary  : 25         Secondary: 5          Units ..: CYL
   zFS space  Primary  : 25         Secondary: 5          Units ..: CYL
  Dataclas : HFS        Mgmtclas : HFS        Storclas: OPENMVS
             Volume   : SBOX1H     Vol count: 1
------------------------------------------------------------------------------
_ HFS data set ..: OMVS.ROGERS.TEST3                             Utilized: 62%
  Save HFS as  ..: OMVS.ROGERS.TEST3.SAV
  Initial zFS  ..: OMVS.ROGERS.TEST3.TMP                         Allocated: N
  HFS space  Primary  : 25         Secondary: 5          Units ..: CYL
  zFS space  Primary  : 25         Secondary: 5          Units ..: CYL
             Dataclas : HFS        Mgmtclas : HFS        Storclas: OPENMVS
             Volume   : SBOX1D     Vol count: 1
------------------------------------------------------------------------------
```

Figure 9-6 List of data sets to be migrated

Using the **fg** option for migration, shown in Figure 9-6, the migration begins; the resulting messages displayed on the screen are shown in Figure 9-7 on page 157.

```
Migrating OMVS.ROGERS.TEST
creating zFS OMVS.ROGERS.TEST.TMP
copying OMVS.ROGERS.TEST to OMVS.ROGERS.TEST.TMP Blocks to copy: 2832
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IDC0531I ENTRY OMVS.ROGERS.TEST.TMP ALTERED
IDC0531I ENTRY OMVS.ROGERS.TEST.TMP.DATA ALTERED
mount /u/rogers/tmp OMVS.ROGERS.TEST ZFS 0
Migrating OMVS.ROGERS.TEST1
creating zFS OMVS.ROGERS.TEST1.TMP
copying OMVS.ROGERS.TEST1 to OMVS.ROGERS.TEST1.TMP Blocks to copy: 2832
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IDC0531I ENTRY OMVS.ROGERS.TEST1.TMP ALTERED
IDC0531I ENTRY OMVS.ROGERS.TEST1.TMP.DATA ALTERED
Migrating OMVS.ROGERS.TEST2
creating zFS OMVS.ROGERS.TEST2.TMP
copying OMVS.ROGERS.TEST2 to OMVS.ROGERS.TEST2.TMP Blocks to copy: 2832
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IDC0531I ENTRY OMVS.ROGERS.TEST2.TMP ALTERED
IDC0531I ENTRY OMVS.ROGERS.TEST2.TMP.DATA ALTERED
Migrating OMVS.ROGERS.TEST3
creating zFS OMVS.ROGERS.TEST3.TMP
copying OMVS.ROGERS.TEST3 to OMVS.ROGERS.TEST3.TMP Blocks to copy: 2832
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IGD01009I MC ACS GETS CONTROL &ACSENVIR=RENAME
IDC0531I ENTRY OMVS.ROGERS.TEST3.TMP ALTERED
IDC0531I ENTRY OMVS.ROGERS.TEST3.TMP.DATA ALTERED
-------------------------
***
```

Figure 9-7 Messages issued by the migration tool during the migration

Preallocated zFS data sets

In this list of data sets, you can change the allocation attributes for any of the data sets. Since the migration tool allows the preallocation of the zFS data set, this is shown in the following example of a data set list entry in Figure 9-8.

The allocation attributes for this new zFS file system can be changed. If the file system is already allocated, enter `Y` for preallocated (replacing the `N`) and the name of the data set as the temp name. The attributes will not be used since you have preallocated the zFS data set.

```
 File system name . . . OMVS.HFS.ROGERS
 Save HFS as  . . . . . OMVS.HFS.ROGERS.SAV
 Temp name for zFS. . . OMVS.HFS.ROGERS.TMP
 Preallocated zFs . . . N

 Primary allocation . . 120
 Secondary allocation   0
 Allocation units . . . CYL        (CYL or TRK)
 Data class . . . . . . HFS
 Management class . . . IMS
 Storage class  . . . . STANDARD
 Volume . . . . . . . . SBOX1F
```

Figure 9-8 Data set entry for a preallocated zFS data set

9.1.6 Migration summary considerations

Decide whether to have a different naming standard for your zFS data sets than for your HFS data sets. You may decide to keep the names as they are. If the zFS data set names are to be the same as the HFS data set names, it is likely that scripts or policies that cause it to be mounted would not need to change.

If you decide to have a new naming convention, then you must change mount policies and mount scripts (including from the BPXPRMxx parmlib) to mount the correct file system name. You will have to look at your automount policies very carefully. You may need to modify the generic entry, migrate all file systems at one time, or add specific entries for each migrated file system.

Two data sets will exist during the migration. Plan for the availability of space for two file systems of about the same size. During the migration, the data will not be available for read or write. To prevent loss of updates during this time, change the mount mode to read only or do the migration when updates are not being done.

Automount policy migration example

Consider a generic entry that contains a file system parameter similar to the following:

```
OMVS.HFS
```

You can change it to use a pattern that will replace the HFS string with ZFS when appropriate:

```
OMVS.///
```

zFS file systems

zFS file systems that will span volumes must be preallocated prior to invoking the tool.

zFS file systems that are greater than 4 GB (about 4825 cylinders of a 3390) must be defined with a data class that includes extended addressability. zFS file systems cannot be striped.

9.2 Pax enhancements for migration from HFS to zFS

New features have been added to pax to make copying data for migration from HFS data sets to zFS data sets easier.

zFS is now the preferred file system, and continued use of HFS is discouraged. Migration needs to be well planned since it will take significant effort to migrate all data from HFS file systems to zFS file systems.

Pax, used in copy mode, is the tool of choice for these data migrations. With z/OS V1R7, changes have been made to pax to do this. Pax was modified to do the following:

- Keep sparse files sparse when they are copied. (Prior to V1R7, pax causes those files to be expanded).
- Pax must report errors reading a source file (for instance, I/O errors or corrupt file system) and then continue to the next file to be copied (Prior to V1R7, pax reports the error and ends).
- To assist migrations if the source contains active mountpoints, pax will do **mkdir** for each of these in the target file system and set directory attributes as required.

9.2.1 Pax utility and sparse files

Pax writes target files as s*parse files* when it is working in copy mode. Sparse files are a way of preserving disk space when storing files that contain large sections of data composed only of zeros. When a file is sparse, these large sections containing zeros are not stored on disk, but when the file is read, the file system returns zeros for those sections. After pax reads and places data to be copied to the target in a write buffer, pax then scans the write buffer in 4k increments (the HFS blocksize) for an increment containing all zeros. When pax encounters an increment of all zeros if will seek 4k ahead instead of writing the zeros. This causes the target file to be sparse. Pax does this regardless of whether the source file was sparse. This reduces the storage used on the target system and should otherwise be transparent to end users. There is an option to turn this feature off, so that pax does not scan files for blocks of zeros and pax writes all bytes read to the target.

Read errors

Currently when pax encounters an error when reading a source file while in copy mode, pax prints an error message and exits. With z/OS V1R7, there is now an option flag to cause pax to continue processing when a read error occurs, after pax has printed an error message to stderr. A non-zero value is returned when pax exits.

The current pax `-X` option causes pax to write only those files that are on the same device as the parent directory. Invoking this option causes active mountpoints to be ignored when pax is in copy mode. With z/OS V1R7, a new pax option causes an empty directory to be created on the target for these mountpoints.

pax command new options

The four new option flags for pax in copy mode only are: **-C**, **-D**, **-M**, and **-W**. The syntax of the **pax** command follows:

```
pax -r -w [-CdDiklLMnquvXE] [-p string(W) ...] [-s substitute ...] source destination
```

Where:

-C Causes pax to continue after encountering an error on the source file system. Pax will print an error message and return a non-zero value after the command ends. Errors on the target file system (for example, out of space or write errors) still cause the **pax** command to end as it always has. This option is only for pax copy mode.

-D Files will not be created sparse in the target directory tree. Sparse files are those which do not use real disk storage for pages of file data that contain only zeros. This saves on disk space. When those files are opened and read the file system returns zeros for those portions of the files that do not have real disk storage. The new default for pax is to copy all files as sparse, whether or not the original file was sparse, if sparse files are supported on the target file system. This option is only for pax copy mode.

-M Create empty directories within the target directory tree for each active mountpoint encountered within the source directory tree. Pax identifies mountpoints by checking if a subdirectory in the source tree is on the same device as the parent current directory. This behavior is like the current pax -X option (write out only those files and directories that are on the same device as their parent directory) except instead of skipping the subdirectory entirely a corresponding empty directory is created in the target directory tree. Any contents in the subdirectory on the source directory tree are ignored. This option is only for pax copy mode.

-p W Preserves user-requested audit attributes and auditor-requested audit attributes.

-s -s substitute modifies pathnames using a substitution command substitute. This is similar to the substitution command of the ed text editor. The full option has the form:

`-s#bregexp/string/[gp]`

- `bregexp` is a basic regular expression and `string` is a string that pax is to insert in place of matches for the regular expression. `String` can contain an ampersand (&), standing for the string matching bregexp, or \1, \2, and so on (with the meanings defined in regexp), for subexpression matching. The `#` is used as the delimeter character separating `bregexp` and `string`. You can use any non-null character instead. There cannot be any space between -s and the delimeter character.
- Normally, -s replaces only the first match for bregexp. A `g` following the string replaces all matches in the line. A `p` following the string prints all successful substitutions on the standard error stream. pax displays a substitution in the format: oldname >> newname.

  There may be more than one `-s` option on the command line. In this case, pax tries the substitutions in the order given. Pax stops trying to make these substitutions as soon as it makes its first successful substitution. If the null string replaces a filename, pax ignores that filename on both input and output.

```
pax [-cdEnqvz] [-cdnqvzE][-f archive] [-o type] [-s substitute] ... [pattern ...]

pax -r [-cdEiknuvz] [-cdiknuvzE][-f archive] [-o options ...] [-p string ...] [-s
substitute ...] [-V volpat] [pattern ...]

pax -w [-dEiLqtuvXz] [-diLqtuvXzE] [-b blocksize] [[-a] [-f archive]] [-o options ...]
[-ssubstitute ...] [-V volpat] [-x format] [pathname ...]

pax -r -w [-CdDiklLMnquvXE] [-diklLnquvXECDM] [-p string ...] [-s substitute ...]
[pathname ...] directory
```

Figure 9-9 pax syntax examples

9.3 Dynamic service activation for z/OS UNIX

Currently, with z/OS UNIX it is not possible to activate maintenance without taking a system outage. z/OS V1R7 provides the ability to activate maintenance on a running system in a non-disruptive manner that is consistent with current maintenance procedures.

Dynamic service activation allows you to dynamically activate and deactivate service items (PTFs, ++APARS, ++Usermods) that affect the UNIX System Service component modules without having to re-IPL. This will minimize the number of both planned and unplanned outages and will lead to a higher level of system availability.

In order to be prepared to exploit dynamic service activation, you have to stay current on z/OS UNIX System Services maintenance. Doing so will make it more likely that any given service item can be activated dynamically because the running system will be at a high enough level to accept the service item.

Note: Although this feature can be used to activate most z/OS UNIX System Services component PTFs, it is not intended to be used as a way to activate a large set of maintenance for preventive purposes. On a periodic or need basis, you have to determine the PTFs that you would be interested in activating dynamically for corrective purposes. These PTFS would likely be of the highest severity and impact to your system.

The PTFs that are capable of being activated dynamically are identified by ++HOLD REASON(DYNACT) data inside them. You have to follow the instructions included with this hold data in order to properly activate the PTF.

9.3.1 Exploiting dynamic service activation

This capability is primarily intended to allow an installation to activate corrective service to avoid unplanned re-IPLs of your systems. Additionally, this capability can be used to activate a temporary patch that can be used in gathering additional documentation for a recurring system problem. Although this capability can be used to activate preventive service on an ongoing basis, it is not intended for this purpose as a replacement for the regular application of service that does require a re-IPL.

To exploit this feature do the following:

- Define SERV_LPALIB and SERV_LINKLIB parameters.

  On the SERV_LPALIB and SERV_LINKLIB parameters in the BPXPRMxx parmlib member, specify the LPALIB and LINKLIB target libraries for z/OS UNIX modules where you install service with SMP/E and you intend to activate it dynamically. They are normally the same system clone data sets where you would SMP/E install your regular service to be used on the next IPL, for example:

  ```
  SERV_LPALIB('dsname','volser')
  SERV_LINKLIB('dsname','volser')
  ```

  These parameters can be set up at IPL time or at restart time of OMVS, or dynamically via the use of the **SET OMVS=** or SETOMVS command.

- The following are the console commands to do the activate and deactivate:
  - **F OMVS,ACTIVATE=SERVICE** activates maintenance at any time after z/OS UNIX System Services Initialization during the operation of a running system.
  - **F OMVS,DEACTIVATE=SERVICE** deactivates service that is no longer required or desired.
  - **D OMVS,ACTIVATE=SERVICE** invokes the BPXEKDA interface to retrieve dynamic service activation information.
  - **D OMVS,O** invokes C/C++ API get_system_settings() to retrieve new parmlib settings.

9.3.2 Activate service items

To activate the service items dynamically after applying maintenance, do the following:

1. In the BPXPRMxx parmlib member, define the target service activation libraries on `SERV_LPALIB` and `SERV_LINKLIB` parameters.

   In our example, these 2 parameters point to the SMPE target libraries. We put them in a separate BPXPRMSS member and activate it by using the SETOMVS RESET=SS command.

2. Issue: **F OMVS,ACTIVATE=SERVICE**

This will dynamically activate the service from the target libraries specified in SERV_LPALIB and SERV_LINKLIB parameters.

The new service will only be activated if the system is at a high enough level to accept the service and if all of the service items are found to be complete. Error messages will be displayed indicating which service items are incomplete or cannot be activated due to the level of the current system. Even when an **F OMVS,SHUTDOWN** has been done to shut down z/OS UNIX System Services, this command can be issued to activate service against the component. Additionally, any service activated prior to an **F OMVS,SHUTDOWN** command stays in effect when z/OS UNIX System Services is restarted.

You are prompted as follows to validate that the listed service items are the ones that are intended to be activated. This message insures that these are the service items that you intended to activate. Additionally, the amount of storage that will be consumed in both ECSA (Nnnnnn bytes) and in the OMVS address space private area (Mmmmm bytes) to perform the activation is identified in this message.

```
BPXM061I THE FOLLOWING SERVICE ITEMS WILL BE ACTIVATED:
OA09999  OA08888
ECSA STORAGE BYTES:  24576 AND OMVS PRIVATE STORAGE BYTES:  12468
WILL BE CONSUMED FOR THIS ACTIVATION.
*06 BPXM061D REPLY "Y" TO CONTINUE. ANY OTHER REPLY ENDS THE COMMAND.
```

9.3.3 Deactivate service items

Issue an **F OMVS,DEACTIVATE=SERVICE** to back off the last set of service items that were activated dynamically.

Only the service items that were previously activated with the **F OMVS,ACTIVATE=SERVICE** are backed off. This function is intended to be used to remove maintenance that was put on for temporary purposes or to remove maintenance that was previously activated and is causing problems on the system**.** When the command completes, the service items are deactivated, but the modules that were part of the service items remain in storage because of potential latent usage of the modules.

You will be prompted to verify the deactivation.

```
BPXM063I THE FOLLOWING SERVICE ITEMS WILL BE DEACTIVATED:
          OA09999  OA08888
*07 BPXM063D REPLY "Y" TO CONTINUE. ANY OTHER REPLY ENDS THE COMMAND.
```

9.3.4 Display activated service items

To display all dynamically activated service items, issue a **D OMVS,ACTIVATE=SERVICE** command. The service items are displayed in sets based on when they were activated, from the most recently activated set of service items to the oldest set of service items. The most recently activated set of service items is shown at the top of the display to indicate it is the highest level of service active on the system, as shown in the example. The display also indicates the amount of ECSA and OMVS address space private storage that is consumed for all of the activated service items.

```
BPXO059I 08.51.42 DISPLAY OMVS 284

OMVS     000E ACTIVE          OMVS=(6D)
DYNAMIC SERVICE ACTIVATION REPORT
 SET  #2:
  LINKLIB=SYS1.DYNLIB.PVT                    VOL=BPXLK1
  LPALIB=SYS1.DYNLIB.LPA                     VOL=BPXLK1
  OA02001  OA02002  OA02003  OA02004
SET  #1:
 LINKLIB=SYS2.DYNLIB.PVT                     VOL=BPXLK1
 LPALIB=SYS1.DYNLIB.LPA                      VOL=BPXLK1
 OA01001  OA01002  OA01003
ECSA STORAGE:  68496        OMVS  STORAGE:  268248
```

Figure 9-10 Display activated service

Display the Parmlib settings

Issue command **D OMVS,O** to display all z/OS UNIX System Services parmlib settings, which includes the settings for the new SERV_LPALIB and SERV_LINKLIB parameters.

```
BPXO043I 12.34.23 DISPLAY OMVS 209
OMVS     000E ACTIVE          OMVS=(7F)
CURRENT UNIX CONFIGURATION SETTINGS:

MAXPROCSYS      =        256    MAXPROCUSER     =         16
.
.
.
SWA                       =  BELOW
SERV_LINKLIB     =  SYS1.DYNSERV.LINKLIB   BPXLK1
SERV_LPALIB      =  SYS1.DYNSERV.LPALIB    BPXLK1
```

Figure 9-11 Display parmlib settings

BPXEKDA

The BPXEKDA assembler language API interface, which returns information similar to the **D OMVS** command for assembler language callers, will return new dynamic service activation data and parmlib settings.

```
ODMVDYNACT DSECT
ODMVDYNACTLINKLIB DS CL44         LinkLib library name
ODMVDYNACTLINKLIBVOL DS CL6       LinkLib volume
ODMVDYNACTLPALIB DS CL44          Lpalib library name
ODMVDYNACTLPAVOL DS CL6           Lpalib volume
ODMVDYNACTITEMCNT DS F            Number of Service Items in the activation
*                                 instance
ODMVDYNACTNEXTPTR DS A            Address of Next OdmvDynAct
ODMVDYNACTITEMPTR DS A            Address of 1st OdmvDynActItem
ODMVDYNACT_LEN EQU *-ODMVDYNACT
```

Figure 9-12 BPXEKDA output

The C/C++ API get_system_settings()

The C/C++ API get_system_settings() returns a structure __Optn that contains most of the current z/OS UNIX parmlib settings. It is updated to return the data for the new SERV_LPALIB and SERV_LINKLIB parameters.

```
Unsigned short _OptnServLpaLibLen     length of service LPA library data set name
unsigned short _OptnServLinkLibLen;   length of service LINK library data set name
unsigned short _OptnServLpaLibVolLen; length of service LPA library volume serial number
unsigned short _OptnServLinkLibVolLen;length of service LINK library volume serial
number
char _OptnServLpaLibMVS[44];         service LPA library data set name (not null
terminated)
char _OptnServLinkLibMVS[44];        service LINK library data set name (not null
terminated)
char _OptnServLpaLibVolMVS[6];       service LPA library volume serial number (not null
terminated)
char _OptnServLinkLibVolMVS[6];       service LINK library volume serial number (not
null terminated)
```

Figure 9-13 Output from API call from C/C++ program

9.4 ISHELL enhancements

New options have been added to improve the ISHELL functionality in this release. The ISHELL is a user interface to perform many tasks, especially those related to file systems and files. In z/OS V1R7, the new ISHELL enhancements do the following:

- Enable you to specify logical or real path on the directory list options
- Improve ISHELL entry messages when the user cannot access ISHELL
- Allow specification of file attributes when creating a new file
- Enable directory reference list
- Preserve file format and CCSID on copy
- Support a refresh command on the file list
- Allow you to add a group list panel similar to the user list panel
- Capture and show zFS errors when trying to create a zFS file system
- Do not exit execute dialog until execute main panel is dismissed
- Do not save last pathname in profile until ISHELL exit

9.4.1 Option to specify logical or real path on the file list

Select this option to display pathnames as you enter or select them rather than have any symbolic links in the pathnames expanded on the displays.

It is possible to not normalize the path to display pathnames as you enter or select them rather than have any symbolic links in the pathnames expanded on the displays.

As shown in Figure 9-14 on page 165, this option is selected from the main panel pull-down choice, with **Options** → **Directory list** as the selection.

```
                  Directory List Options

Select options and fields to be displayed with /

   File type   ( 4 columns)
   Permissions: octal ( 4 columns)
   Permissions: rwx (10 columns)
   Change time (16 columns)
   Owner       ( 9 columns)
   File size   (10 columns)

   View/change sort options...
   View/change file name highlighting...
   Verbose directory list panel
   Null Enter refreshes list
   Stop processing multiple selections after a message
/  Do not normalize the selected path to the real path
```

Figure 9-14 Do not normalize the selected path to the real path option

9.4.2 Improve ISHELL entry messages

When a user is not able to access an ISHELL session, new entry messages will no longer be displayed, the ISHELL will terminate, and an ISPF message is displayed. This will delete the Create HFS panel when there is a dub failure.

9.4.3 Specification of file attributes when creating a new file

There is a new entry in the Create a New File panel that enables you to specify file attributes when creating a new file, as shown in Figure 9-15.

```
                          Create a New File

Pathname:
                                                              More:     +
  /u/rogers/tst100
  ____________________________________________________________________
  ____________________________________________________________________
  ____________________________________________________________________

Permissions  . . 777  (3 digits, each 0-7)

File type                          File source for regular file
2   1.  Directory                  1   1.  Edit...
    2.  Regular file                   2.  Copy file...
    3.  FIFO                           3.  Copy data set...
    4.  Symbolic link...
    5.  Hard link...               View and set attributes  . .  Y
```

Overtype N to Y

Figure 9-15 View and set attributes entry

Select the option with `Y` or / to view the file attributes after the file is created. When you hit the Enter key Figure 9-16 on page 166 is displayed.

```
  Edit  Help

           Display File Attributes

Pathname : /u/rogers/tst100
                                        More:     +
File type . . . . . : Regular file
Permissions . . . . . : 0 ---------
Access control list . : 0
File size . . . . . : 0
File owner  . . . . . : HAIMO(0)
Group owner . . . . . : SYS1(2)
Last modified . . . . : 2005-08-16 13:10:50
Last changed  . . . . : 2005-08-16 13:10:50
Last accessed . . . . : 2005-08-16 13:10:50
Created . . . . . : 2005-08-16 13:10:50
Link count  . . . . . : 1
 F1=Help         F3=Exit         F4=Name
 F7=Backward     F8=Forward     F12=Cancel
```

Figure 9-16 Display File Attributes panel

From this panel, place the cursor under the Edit field and press Enter; the screen is Figure 9-17 on page 167 is displayed. From this panel, you can specify options **1** through **8** to specify the file attributes for the new file that is being created.

```
 Edit  Help

    1. Mode fields...
    2. Owning user...
    3. Owning group...
    4. User auditing...                More:     +
    5. Auditor auditing...
    6. File format...
    7. Extended attributes...
    8. Access contol list...
    *. Directory default ACL...
   *0. File default ACL...
                                          13:10:50
 Last changed  . . . . : 2005-08-16 13:10:50
 Last accessed . . . . : 2005-08-16 13:10:50
 Created . . . . . . . : 2005-08-16 13:10:50
 Link count  . . . . . : 1
  F1=Help          F3=Exit          F4=Name
  F7=Backward      F8=Forward      F12=Cancel
```

Figure 9-17 Panel to begin to set the file attributes for a new file

9.4.4 Enable directory reference list

The directory reference list is a list of recently viewed directories. When a directory is selected for viewing, that directory is typically added to the top of the reference list. If the name is already in the list, the older entry is removed. When you view the reference list you will see the most recently viewed directory at the top.

Enter **REF ON** on the command line of the main panel to enable the directory reference list. You can also specify this using the main panel options, and on the pull-down list choose a new option **5 Advanced**, as shown in Figure 9-18.

```
                    Advanced Options

Select options

_  Bypass delete confirmations
_  Bypass exit confirmation
_  No auto-skip on action panels
_  Always start initial panel with current directory
/  Enable directory reference list

Command line position:
1   1.  Top
    2.  Bottom
    3.  Inherit
```

Figure 9-18 Enable directory reference list option

Using directory reference list

You can now enter **REF** on the command line on the main panel to view the directory reference list, as shown in Figure 9-19.

Either enter **S** or **/** to select only one entry. The list can also be viewed by selecting **Reference list** from the tools pull-down.

If you use this frequently, it may be easier to put the **REF** command on a PF key. Just enter **KEYS** on the command line to access the ISPF keylist dialog. In a similar fashion, you may want to assign the **LEFT**, **RIGHT**, and **EXPAND** commands to PF keys if they are not already assigned. These default to F10, F11, and F4, respectively.

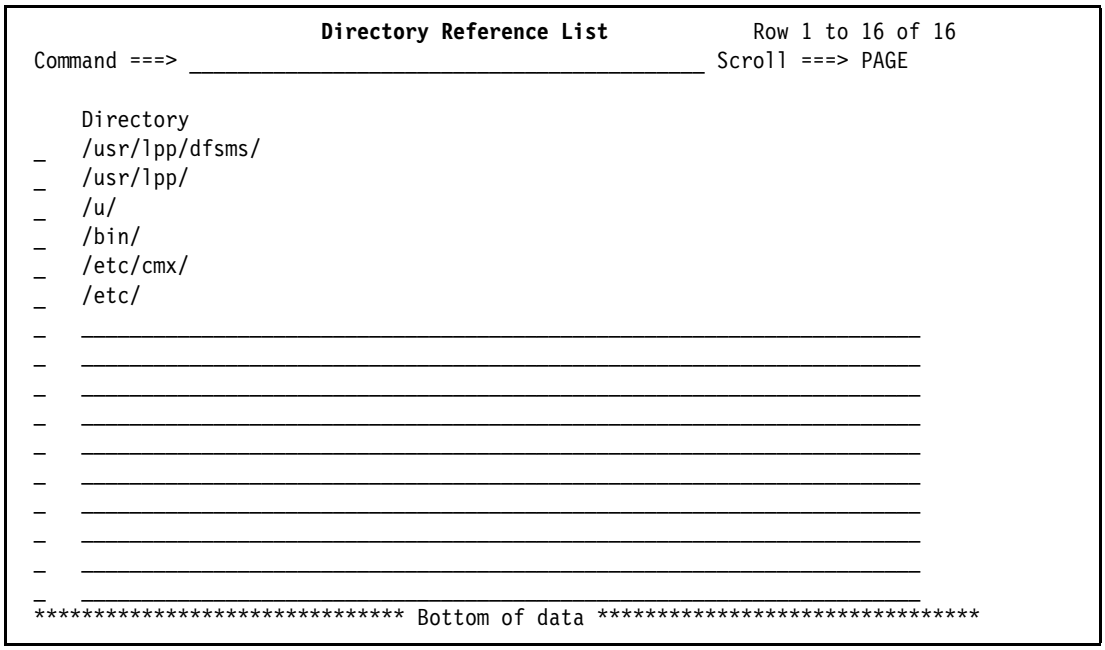


Figure 9-19 Directory Reference List

Note: Enter **Help** on the Directory Reference List to get more information about use of the **REF** option.

9.4.5 Preserve file format and CCSID on copy

This is done unconditionally and prior to the data movement so conversions resulting from the coded character set identifier (CCSID) setting occur. CCSID identifies the coded character set identifier to be implicitly set for the untagged file. CCSID is specified as a 2 byte decimal value from 0 to 65535. It is only used when TEXT is specified. Both the Mount and Root statements can use the Tag statement. TAG(NOTEXT,0) is the Default.

9.4.6 Support a refresh command on the file list

REFR and **REFRESH** commands will refresh the directory list. An option already exists to set the behavior of a null Enter to do this. Select **Options** → **Directory list** as shown in Figure 9-20 on page 169.

```
                  Directory List Options

Select options and fields to be displayed with /

   File type   ( 4 columns)
   Permissions: octal ( 4 columns)
   Permissions: rwx (10 columns)
   Change time (16 columns)
   Owner       ( 9 columns)
   File size   (10 columns)

   View/change sort options...
   View/change file name highlighting...
   Verbose directory list panel
/  Null Enter refreshes list
   Stop processing multiple selections after a message
   Do not normalize the selected path to the real path
```

Figure 9-20 Null Enter refreshes list option

9.4.7 Add a group list panel similar to the user list panel

The *group list* is a list of all groups and the GID for each, as shown in Figure 9-22 on page 170. The GID field is blank if there is no GID for the group. By default the list is sorted by group name. The group list choice is available from the Setup pull-down list, by selecting option **8 Group list**, as shown in Figure 9-21.

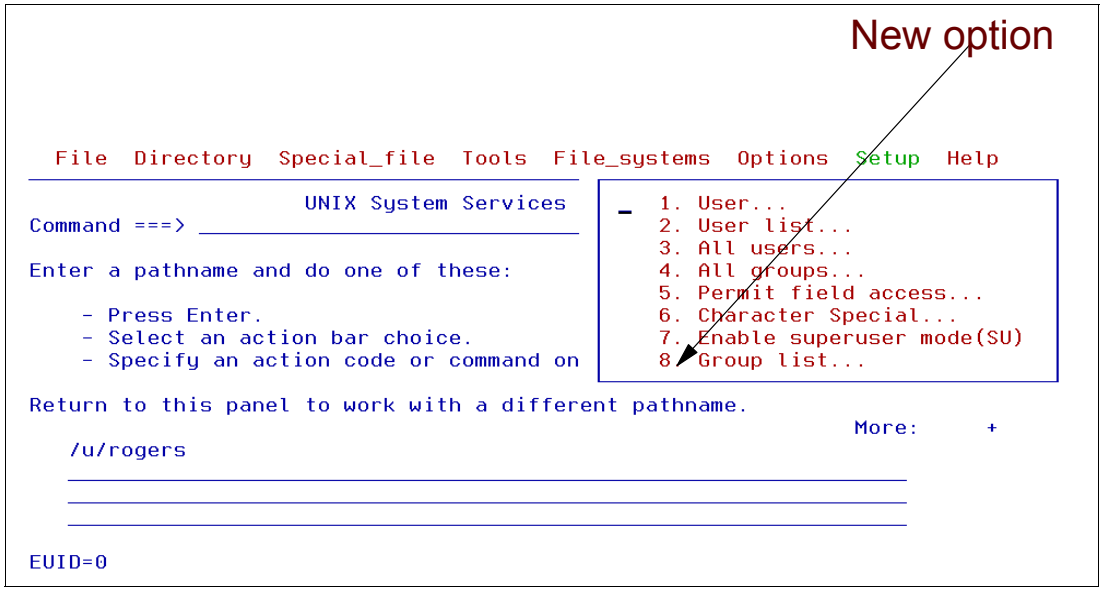


Figure 9-21 Group list option to list groups and GIDs

When you select option 8, the panel shown in Figure 9-22 on page 170 is displayed.

```
    File  Help
_______________________________________________________________________________
                              Group List                  Row 1 to 24 of 249
Command ===> _____________________________________________________________

Group     GID
@PL       9876790
ACFNCP    9876791
ADSM      9876792
AMS       9876793
ANO       9876794
AOF       9876795
AOPADMIN  101
AOPOPER   100
APL2      9876796
```

Figure 9-22 Group List display panel

Note: Some choices on the Setup pull-down list require superuser authority or the "special" attribute for full function, or both.

You can sort the list by GID by selecting **File** to access the pull-down list, which includes sort options as shown in Figure 9-23.

```
_  File  Help
  _  1. Sort name
     2. Sort GID
     3. Print
     4. Exit
                              Group List                  Row 1 to 24 of 251

ACFNCP    9876791
ADSM      9876792
AMS       9876793
ANO       9876794
AOF       9876795
AOPADMIN  101
AOPOPER   100
APL2      9876796
```

Figure 9-23 Sort GID option

9.5 Miscellaneous enhancements

In this section we describe improvements introduced in these other utilities and commands:

- OEDIT improvements
- TSO utility
- BPXWDYN interface
- Mount wait option

9.5.1 OEDIT improvements

OEDIT opens files in R/W mode. If a file has authorized extended attributes set, this open causes these attributes to be reset — even if the user does not save changes. If any of the extended attributes are set (shared lib, apf, prog ctl), a confirmation panel is displayed before

proceeding. You can use the **OEDIT** command in the ISPF editor to create or edit text in a hierarchical file. The following changes are introduced in z/OS V1R7:

- A higher maximum width for editing is included (up to 32752).
- A warning is issued if extended attributes are set on a file being edited before the **OEDIT** causes them to get reset, as shown in Figure 9-24.

```
------------------- EDIT - EXTENDED ATTRIBUTES -------------------------------
Command ===>

The file you are about to edit has one or more authorized extended attributes
set.  If you continue these attributes may be reset.
Press Enter to continue or END (usually PF3) to stop.

Path:
/u/USER1/tstO509a
```

Figure 9-24 Edit Extended Attributes warning panel

9.5.2 TSO utility

A user can allocate SYSTSPRT with the TSOOUT environment variable. When TSOOUT is specified, the shell command post processing routine is not called to display and delete its usual output data set.

Example:

```
export tsoout=''alloc path('/dev/tty') pathopts(owronly)
```

The format of the TSOOUT variable is in BPXWDYN format without a ddname.

9.5.3 BPXWDYN interface

This is a text interface to dynamic allocation and dynamic output services. It can be invoked as a REXX function or called with a single string parameter using a standard MVS variable length parameter list.

The enhancements in BPXWDYN are:

- SVC99 info retrieval capability is added to enable the user to determine the DD names, data set names, and path names for current allocations. Allocation attributes are not supported at this time.
- Keys are added for tape processing: position, label, retpd, trtch.
- Resetting of S99NOMNT is allowed.

9.5.4 Mount wait option

A wait option is provided so the mount will wait for async mounts to complete. Syntax of this optional flag is **-w** ***n*** (*n* is wait time in seconds). If n is specified as 0, the wait will be indefinite.

Example:

```
mount -w 5 -f HFS.TEST.MOUNT /u/User1/tst1
```

The **-w** option flag for the mount utility is tolerated on any form of the mount command and is ignored if no wait needs to be done.

The complete syntax of the **mount** command is in *z/OS V1R7.0 UNIX System Services Command Reference*, SA22-7802.

9.6 Mounting file systems with SET OMVS

Installations can now change their mount configuration from the console using a BPXPRMxx parmlib member.

Mount from the console invoked by the **SET OMVS=xx** console command, specifying a BPXPRMxx parmlib member previously created with the desired MOUNT, FILESYSTYPE, SUBFILESYSTYPE, NETWORK or ROOT statements included.

This feature could be used when a large number of MOUNTs need to be done from the console, or if a significant PFS configuration change is needed.

During the processing for this MOUNT:

- Any MOUNT that duplicates an existing file system in both FILESYSTEM name and MOUNTPOINT is silently ignored.
- Standard mount error messages are written to the console for any other failure.
- Successful mounts will also generate a console message.
- Any FILESYSTYPE or SUBFILESYSTYPE statement that duplicates an existing PFS will generate an informational message. No messages are written to the console or hardcopy log for successful processing.
- Any NETWORK statement that duplicates an existing address family is accepted as an update to the MAXSOCKETS value.

Note that during system startup, successful mount messages only go to the hard copy log, but due to the interactive nature of **SET OMVS** they will also be written to the console.

FILESYSTYPE, SUBFILESYSTYPE, and NETWORK statements are currently supported by the **SETOMVS RESET=(xx)** command and this capability is now included for **SET OMVS**.

After a **SET OMVS=xx** with FILESYSTYPE and SUBFILESYSTYPE statements is successfully processed, the **D OMVS,PFS** command can be used to show the new PFSs.

9.6.1 Example of MOUNT with SET OMVS

We prepared an example using BPXPRMxx to mount HFS with **SET OMVS=xx**, in different situations. It has four mount commands as shown in Figure 9-25 on page 173.

```
/*-------------------------------------------------------------------*/
/* MOUNTs TO BE USED WITH SET OMVS=XX COMMAND - TEST                  */
/*-------------------------------------------------------------------*/
/*--- NOT MOUNTED -------------------------------------------------*/
MOUNT FILESYSTEM('HFS.TEST.MOUNT')
      MOUNTPOINT('/TEST/TSTMOUNT')
      NOAUTOMOVE
      TYPE(HFS)  MODE(RDWR)

/*--- ALREADY MOUNTED ---------------------------------------------*/
MOUNT FILESYSTEM('HFS.TEST.MOUNT2')
      MOUNTPOINT('/TEST/TSTMOUNTY')
      NOAUTOMOVE
      TYPE(HFS)  MODE(RDWR)

/*--- WITHOUT MOUNTPOINT ------------------------------------------*/
MOUNT FILESYSTEM('HFS.TEST.MOUNT3')
      MOUNTPOINT('/TEST/TSTMOUNTW')
      NOAUTOMOVE
      TYPE(HFS)  MODE(RDWR)

/*--- WITHOUT HFS DATASET -----------------------------------------*/
MOUNT FILESYSTEM('HFS.TEST.MOUNT4')
      MOUNTPOINT('/TEST/TSTMOUNTZ')
      NOAUTOMOVE
      TYPE(HFS)  MODE(RDWR)
```

Figure 9-25 BPXPRMxx only with mount commands

After the **SET OMVS=tt** is issued for member BPXPRMGG, the result is shown in Figure 9-26. There are four mounts showing different mount messages for each one.

```
SET OMVS=tt
IEE252I MEMBER BPXPRMTT FOUND IN SYS1.PARMLIB
BPXO032I THE SET OMVS COMMAND WAS SUCCESSFUL.

BPXF013I FILE SYSTEM HFS.TEST.MOUNT 815
WAS SUCCESSFULLY MOUNTED.

BPXF237I FILE SYSTEM HFS.TEST.MOUNT2 816
WAS ALREADY MOUNTED ON PATHNAME
/TEST/TSTMOUNTY.

BPXF008I FILE SYSTEM HFS.TEST.MOUNT3 817
WAS NOT MOUNTED.
THE MOUNT POINT SPECIFIED IN BPXPRMTT DOES NOT EXIST.

BPXF002I FILE SYSTEM HFS.TEST.MOUNT4 WAS 818
NOT MOUNTED.  RETURN CODE = 00000081, REASON CODE = EF096055
```

Figure 9-26 Result of mount with SET OMVS

9.6.2 Displaying mount failures

Failures from prior MOUNT or MOVE file system commands, of any form, will be remembered and the pertinent information is available for display or application retrieval.

This includes, for instance, mounts issued from TSO, ISHELL, those from BPXPRMxx during system startup, and sysplex mounts.

Also, file system new owner failures (**Move** commands) that occur during setomvs, chmount, shutdown, and member gone event processing will be saved.

To check move and mount file system failures use the following display commands:

- **D OMVS,MF** displays the last 10 (or less), as shown in Figure 9-27.
- **D OMVS,MF=A** or **D OMVS,MF=ALL** displays up to 50 failures, as shown in Figure 9-28.
- **D OMVS,MF=P** or **D OMVS,MF=PURGE** deletes the failure information log, as shown in Figure 9-29 on page 175.

```
D OMVS,MF
BPXO058I 13.26.31 DISPLAY OMVS 972
OMVS     0011 ACTIVE          OMVS=(TT)
LAST PURGE: TIME=08.49.43  DATE=2005/05/11
SHORT LIST OF FAILURES:
TIME=09.36.41  DATE=2005/05/13        MOUNT RC=0081  RSN=EF096055
  NAME=HFS.TEST.MOUNT4
  TYPE=ZFS
  PATH=/TEST/TSTMOUNTZ
  PLIB=BPXPRMTT
TIME=09.36.40  DATE=2005/05/13        MOUNT RC=0081  RSN=0503005C
  NAME=HFS.TEST.MOUNT3
  TYPE=HFS
  PATH=/TEST/TSTMOUNTW
  PLIB=BPXPRMTT
TIME=10.04.47  DATE=2005/05/12        MOUNT RC=0081  RSN=EF096055
  NAME=OMVS.TST1.HFS
  TYPE=ZFS
  PATH=/u/tst1
(...up to 10 failures)
```

Figure 9-27 Short list of mount failures

If you want to list more failures, use **D OMVS,MF=ALL**, which displays up to 50 entries. For other failures not listed, you can view the hard copy log.

```
D OMVS,MF=A
BPXO058I 13.26.37 DISPLAY OMVS 974
OMVS     0011 ACTIVE          OMVS=(TT)
LAST PURGE: TIME=08.49.43  DATE=2005/05/11
ENTIRE LIST OF FAILURES:
TIME=09.36.41  DATE=2005/05/13        MOUNT RC=0081  RSN=EF096055
  NAME=HFS.TEST.MOUNT4
  TYPE=ZFS
  PATH=/TEST/TSTMOUNTZ
  PLIB=BPXPRMTT
(...up to 50 failures)
```

Figure 9-28 List of mount failures

```
D OMVS,MF=P
BPXO058I 13.30.25 DISPLAY OMVS 010
OMVS     0011 ACTIVE          OMVS=(TT)
PURGE COMPLETE: TIME=13.30.25  DATE=2005/05/13

D OMVS,MF
BPXO058I 13.42.45 DISPLAY OMVS 041
OMVS     0011 ACTIVE          OMVS=(TT)
LAST PURGE: TIME=13.30.25  DATE=2005/05/13
NO MOUNT OR MOVE FAILURES TO DISPLAY
```

Figure 9-29 Purge failure information log

10

DFSMS enhancements

DFSMS is a key component of z/OS that automatically manages data from creation to expiration. DFSMS consists of five elements: DFSMSdfp, DFSMSdss, DFSMShsm, DFSMSrmm, and DFSMStvs. DFSMS provides allocation control for availability and performance, backup and recovery, disaster recovery services, space management, tape management, reporting, and simulation for performance and configuration tuning.

This chapter describes the z/OS V1R7 DFSMS changes and enhancements and covers the following topics:

- DFSMSdfp enhancements
- DFSMSdss enhancements
- DFSMShsm enhancements
- DFSMSrmm enhancements

10.1 DFSMSdfp enhancements

The DFSMSdfp functional component of DFSMS provides the storage, program, data, and device management functions of z/OS. The storage management subsystem (SMS) component of DFSMSdfp is fundamental to providing these functions.

z/OS V1R7 DFSMSdfp has been enhanced to support:

- VSAM RLS 64-bit virtual support
- Large format data sets
- VSAM extent constraint relief
- SMS enhancements as follows:
  - Change SMS status of volumes
  - New volume selection messages and traces
  - New ACS environment - allocation test
  - Performance improvement for VARY command
- Removal of STEPCAT/JOBCAT support
- Removal of ISAM support
- DEFINE PAGESPACE with a CATALOG parameter
- Device support address space (DEVMAN)
- New DEVSERV QLIB command
- Catalog improvements as follows:
  - Default space parameter for VVDS implicit define
  - CATALOG service task lockup
  - CATALOG autotuning support
  - REPRO MERGECAT enhancements
- XRC Plus
- OAM enhancements as follows:
  - Immediate recall to DB2
  - Enhanced MOVEVOL utility
  - New parameter TAPEDISPATCHERDELAY
  - New parameter CLEAROLDLOC
  - New exit routine return to MVS scratch

10.1.1 VSAM RLS 64-bit virtual support

Currently all VSAM RLS (record level sharing) index and data control interval buffers and most of the RLS control blocks reside in the 31-bit addressable SMSVSAM data space. This can be up 2 GB in size. Users with a high rate of transactions could encounter virtual storage constraints with this design.

With z/OS V1R7, you can specify that VSAM RLS is to use 64-bit addressable virtual storage (above the 2-gigabyte bar) for data buffers. Doing so can help you avoid possible buffer

space constraints and can potentially improve performance for your high volume transaction applications. Use of this feature is optional and is enabled by the following:

- At the system level, a new option in IGDSMSxx to define the size of the above the bar buffer pool
- At the data set level, a new RLS DATACLAS attribute

RLS buffers above the bar

To use RLS buffers above the 2 GB bar for a particular data set, first the overall size of the system's RLS buffer pool must be set and the data set must have the new RLS "buffers above the bar" attribute. If you choose not to exploit this function, VSAM RLS continues to use buffers that reside below the 2 GB bar.

In addition, there is a new RLS performance option that can be set in the IGDSMSxx parmlib member to specify the amount of real storage that can be permanently fixed for RLS buffers. This applies regardless of whether the buffers are above or below the 2GB bar, and storage will be fixed on a first come, first served basis.

New RLS options in IGDSMSxx

The RLSABOVETHEBARMAXPOOLSIZE parameter specifies the total size in MBs of the RLS buffer pool that is to be above the bar. Valid values are between 500 and 2000000 and can apply to either all systems or a list of specific systems.

You can specify a different value for each system, or one value for the entire sysplex, as follows:

```
RLSABOVETHEBARMAXPOOLSIZE(ALL,512)
RLSABOVETHEBARMAXPOOLSIZE(SYS1,512;SYS2,1024)
```

The default value of 0 disables the option, which means that there would be no RLS buffers above the 2 GB bar.

The RLSFIXEDPOOLSIZE parameter specifies the total amount of real storage in MBs that can be permanently fixed (pinned) for use by RLS buffers. This value applies regardless of whether the storage is above or below the bar and may be adjusted by VSAM RLS internally to not exceed 80% of available real storage.

You can specify a different value for each system, or one value for the entire sysplex. For example:

```
RLSFIXEDPOOLSIZE(ALL,768)
RLSFIXEDPOOLSIZE(SYS1,768;SYS2,1024;SYS3,0)
```

The data in SMF record type 42, subtype 19 can be used to choose a suitable amount of storage to be fixed. For example, an excessive number of buffer misses might indicate the need for more real storage to be fixed for the use of VSAM RLS data buffers.

The default value of 0 disables the option, which means that there would be no storage fixed permanently for RLS buffers.

Commands to set the RLS options

The **SETSMS** and **SET SMS=xx** commands can be used to specify the values for the options:

```
RLSABOVETHEBARMAXPOOLSIZE
RLSFIXEDPOOLSIZE
```

Examples are provided in Figure 10-1, Figure 10-2, and Figure 10-3 on page 180.

```
SETSMS RLSABOVETHEBARMAXPOOLSIZE(SC70,500)

D SMS,OPTIONS
IGD002I 14:19:53 DISPLAY SMS 275
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
....
....
RlsAboveTheBarMaxPoolSize = 500
RlsFixedPoolSize = 0
```

Figure 10-1 Using the SETSMS command to set the rlsabovethebarmaxpoolsize

```
SET SMS=99
IEE252I MEMBER IGDSMS99 FOUND IN SYS1.PARMLIB
IGD031I SMS PARAMETERS 659
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
....
....
RlsAboveTheBarMaxPoolSize = 512
RlsFixedPoolSize = 10
```

Figure 10-2 Using the SET SMS=xx command to set the RLS fixed pool size

```
SETSMS RLSFIXEDPOOLSIZE(0)

IGD029I REJECT SETSMS COMMAND 114
ERROR IS PREVIOUS COMMAND WITH KEYWORD
RlsFixedPoolSize IS STILL IN PROCESS.
```

Figure 10-3 pending RLSFixedPoolSize command

Note: If a data set is already opened with RLS on a system, then any changes to RlsAboveTheBarMaxPoolSize and RlsFixedPoolSize do not take effect on that system until the SMSVSAM address space is recycled.

If there have been no data sets opened in RLS mode on the system, then any changes to the two keywords will take effect when an RLS data set is opened for the first time.

New RLS DATACLAS attribute

For a data set to be eligible to have its RLS buffers above the 2GB bar, the data set must have a DATACLAS definition that includes the new attribute RLS ABOVE THE BAR set to YES.

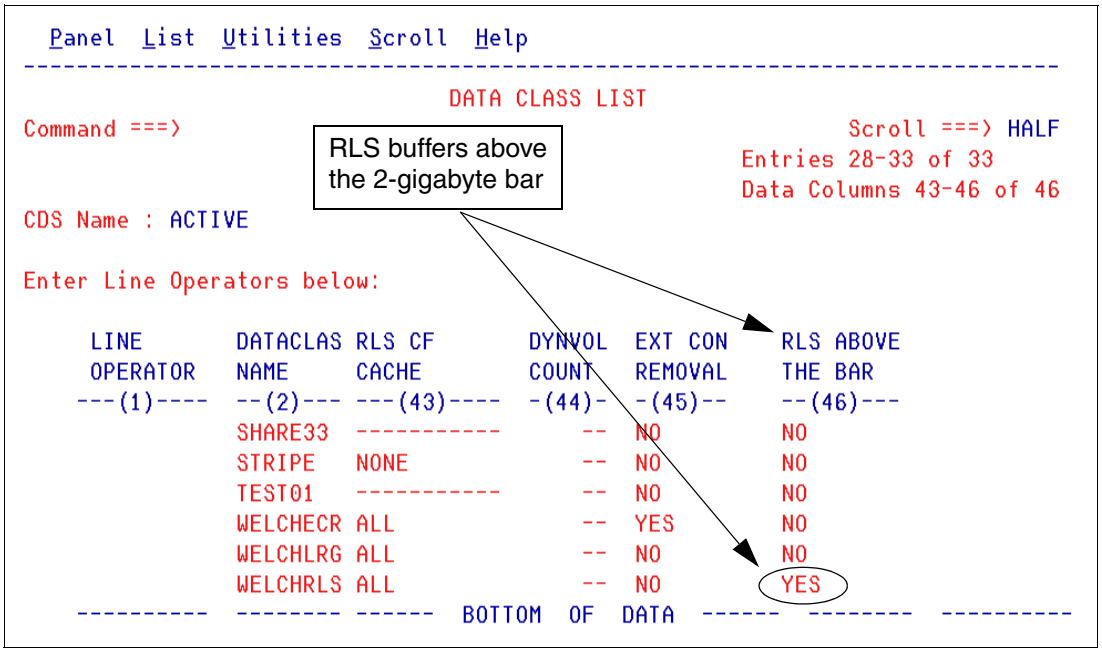


Figure 10-4 New RLS above the bar DATACLAS attribute

Migration and coexistence considerations

The following migration and coexistence considerations apply:

- Decide whether to enable 64-bit data buffers for VSAM RLS processing.
- Review RLS buffer statistics and decide whether to fix an amount of real storage for RLS buffers.

There are no coexistence considerations for pre-z/OS V1R7 systems except for the sharing of the IGDSMSxx parmlib member. The management of the VSAM RLS buffers is independent of the operation of other systems in the sysplex.

SMF record type 42 is changed in z/OS V1R7 to include information about usage of buffers above the bar for the following sub-types:

- Sub-type 19 will contain statistics for buffers above as well as below the bar.
- Sub-type 18 will contain statistics by CF cache storage classes.
- Sub-type 16 will contain new fields that will indicate whether or not the data set is enabled above the 2 GB bar and whether or not it is actually being used.

10.1.2 Large format data sets

The example in Figure 10-5 shows what happens currently when you try to allocate a data set that exceeds the 65535 track limit.

```
   IEF344I WELCHA ALLOC DD1 - ALLOCATION FAILED DUE TO DATA FACILITY SYSTEM
   ERROR
   IGD17051I ALLOCATION FAILED FOR DATA SET
   WELCH.LARGE.DATASET
, PRIMARY SPACE EXCEEDS 65535 TRKS
```

Figure 10-5 Unable to allocate more than 65535 tracks

Allocate a large data set

To make better use of disk storage devices, z/OS V1R7 now supports a new data set type of large format data sets, which are physical sequential data sets with the ability to grow beyond the previous size limit of 65535 (x'FFFF') tracks per volume. Large format data sets will reduce the need to use multiple volumes for single data sets, especially very large data sets like spool data sets, dumps, logs, and traces.

Large format data sets are physical sequential data sets, with generally the same characteristics as other non-extended format sequential data sets but with the capability to grow beyond the previous size limit. Unlike extended-format data sets, which also support greater than 65535 tracks per volume, large format data sets are compatible with programs that use EXCP and don't need to be SMS-managed.

Large format data sets are allocated using a new DSNTYPE value of LARGE.

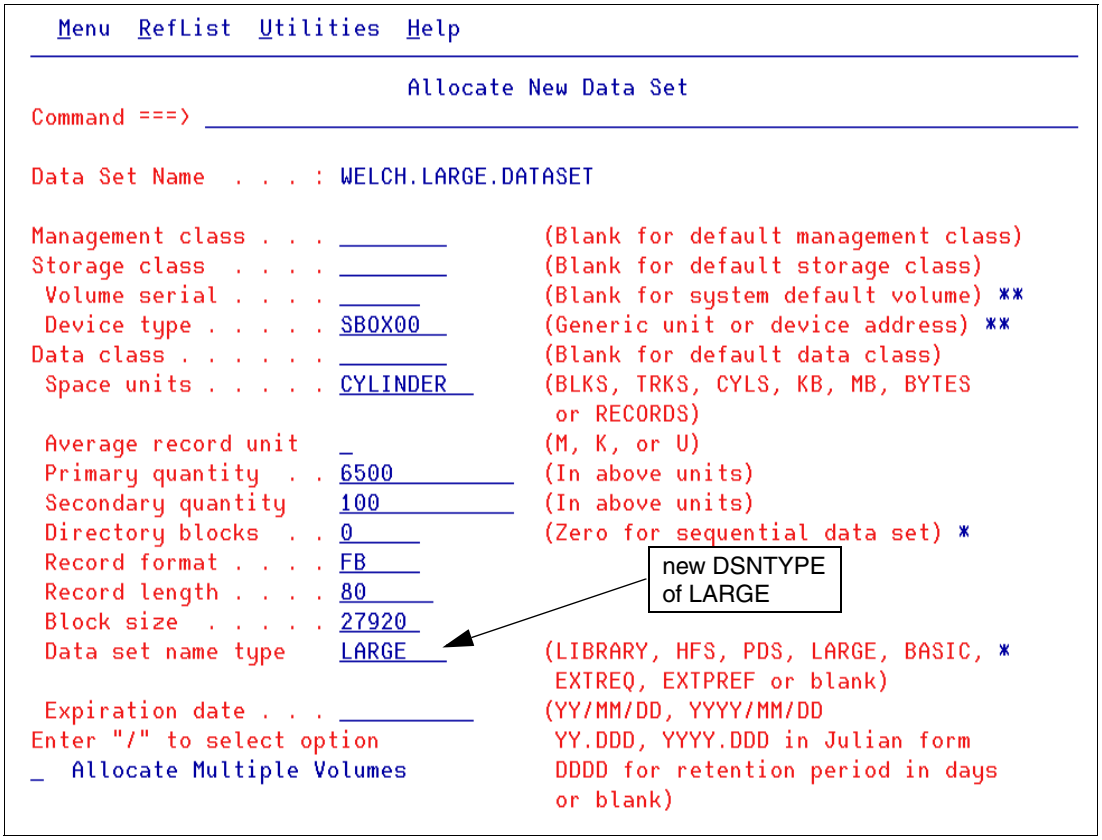


Figure 10-6 Allocating a large format data set

The new DSNTYPE attribute of LARGE can be assigned via any of the standard ways of allocating a new data set, as follows:

- A DD statement in JCL
- TSO/E ALLOCATE command
- IDCAMS allocate
- Dynamic allocation (SVC 99)

Alternatively you can make use of a DATACLAS definition that has the LARGE attribute specified for the data set name type, as shown in Figure 10-7 on page 183.

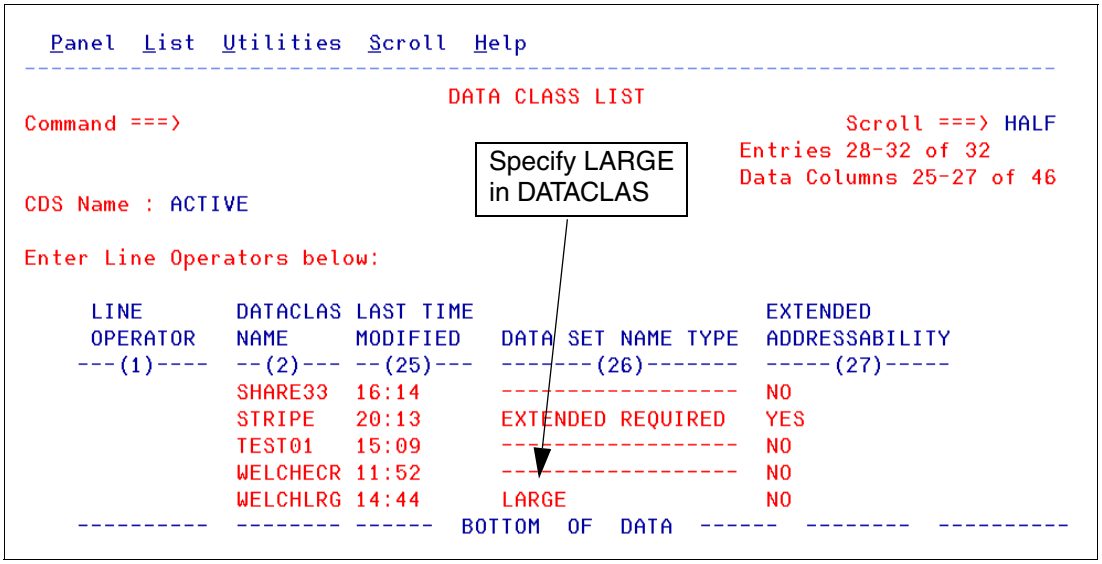


Figure 10-7 Using LARGE in a DATACLAS definition

The sample JCL in Figure 10-8 shows how to allocate a large format data set using a suitable DATACLAS name.

```
//ALLOC EXEC PGM=IEFBR14
//DD1  DD DSN=WELCH.LARGE.DATASET,DISP=(NEW,CATLG),
//  VOL=SER=SBOX00,UNIT=SYSDA,SPACE=(CYL,6000),DATACLAS=WELCHLRG
```

Figure 10-8 JCL to define a large data set using DATACLAS

New IGDSMSxx parameter BLOCKTOKENSIZE

There is a new SMS parameter BLOCKTOKENSIZE that can be specified in IGDSMSxx in PARMLIB. This parameter is used to control the ability of programs to use large format data sets. The default value of BLOCKTOKENSIZE(REQUIRE) means that most existing programs, except standard system utilities, will fail with a 213 abend.

The error message in Figure 10-9 shows what will happen when a user program tries to write to a large format sequential data set with the SMS default option of BLOCKTOKENSIZE(REQUIRE) in effect.

```
IEC143I 213-15,IFG0196J,WELCHA,TESTIT,LARGE,8405,SBOX00,WELCH.LARGE.DATASET
IEA995I SYMPTOM DUMP OUTPUT  104
SYSTEM COMPLETION CODE=213  REASON CODE=00000015
```

Figure 10-9 213 abend for user program with BLOCKTOKENSIZE(REQUIRE)

Changing the option to NOREQUIRE can be done by specifying BLOCKTOKENSIZE(NOREQUIRE) in your IGDSMSxx parmlib member and issuing the MVS command **SET SMS=xx** to dynamically change the SMS options.

The program that failed with the 213 abend now executes to completion and is able to write records to a large format data set.

Controlling use of large data sets

If you wish to prevent existing user programs from being able to access large format data sets, use the default option of REQUIRE and force programs to be changed to specify BLOCKTOKENSIZE=LARGE on the DCBE macro. Note that this will prevent programs written in high level languages such as COBOL from being able to work with large format data sets (unless you are opening the data set for input and it has not exceeded the 65535 track limit).

Note: If you wish to use the default setting of BLOCKTOKENSIZE(REQUIRE), then it is recommended that you specify this in your IGDSMSxx members in PARMLIB. This ensures that the option is explicitly set to the value that you want.

The fragment of code shown in Figure 10-10 illustrates the use of a DCBE macro to allow the program to use large format data sets. This means that this program will now work regardless of the setting of the SMS BLOCKTOKENSIZE option.

```
          OPEN  (SEQIN,(INPUT),LARGE,(OUTPUT))

          ....
          ....

          CLOSE (SEQIN,,LARGE)

LARGE     DCB   DDNAME=LARGE,DSORG=PS,MACRF=PL,LRECL=80,                X
                DCBE=L_DCBE
L_DCBE    DCBE  BLOCKTOKENSIZE=LARGE
```

Figure 10-10 Specify BLOCKTOKENSIZE on DCBE macro to allow use with large format data sets

User program considerations

Some user programs may require other changes to work correctly with large format data sets and this is why the BLOCKTOKESIZE(REQUIRE) option is provided. For example, the format-1 DSCB is changed so that DS1LSTAR is extended (non-contiguously) by the addition of DS1TTTHI to provide a high order byte. This provides room for a 3-byte field to address up to x'FFFFFF' tracks instead of the previous x'FFFF' tracks.

In Figure 10-11 you can see the dump of a format-1 DSCB for a large format data set with more than 65535 tracks used. DS1FLAG1 is set to indicate that this is a large format data set and if you are using DS1LSTAR then you must allow for the high order byte.

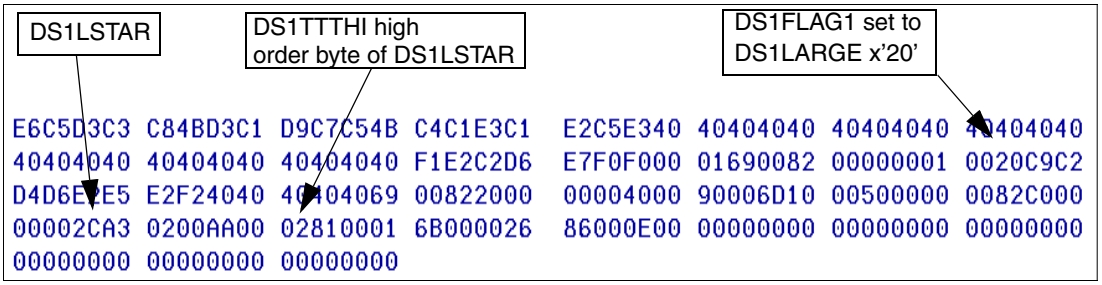


Figure 10-11 Format-1 DSCB for a large format data set with tracks used greater than 65535

Using large format data sets

z/OS V1R7 supports large format data sets in many areas, including:

- Standalone dump.
- IPCS.
- JES2 and JES3 spool data sets.
- ISPF.
- DFSMShsm journal.
- DFSMShsm support for migration/recall, backup/restore and ABACKUP/ARECOVER.
- DFSMSdss support for COPY, DUMP/RESTORE, RELEASE, PRINT and DEFRAG.
- DFSMSrmm journal and temporary work data sets.
- SMS, DFSORT, IDCAMS REPRO, IEBGENER, IEBDG, and IEBCOPY all support large format data sets and are unaffected by the setting of BLOCKTOKENSIZE.

z/OS V1R7 components that do not support large format data sets include:

- VIO data sets.
- System dump data sets.
- The BDAM access method.
- TSO/E EXECIO, XMIT and RECEIVE commands.
- High level languages do not support large format data sets when the default SMS option of BLOCKTOKENSIZE(REQUIRE) is in effect.

Migration and coexistence considerations

Decide on the approach you want to take with the new IGDSMSxx option of BLOCKTOKENSIZE and set it in your PARMLIB member.

Check with the vendors of your ISV products to determine if they will work correctly with large format data sets.

User programs that need to make use of large format data sets will need to be examined to see if they make use of interfaces that have changed, specifically:

- DS1LSTAR or DS1EXT1 (first extent) fields in the format 1 DSCB
- Track conversion routines
- The DEBNMTRK field in the DEB
- The BSAM NOTE or POINT Macros on DASD

For a more detailed discussion on this subject see “Using large format data sets” in *DFSMS: Using the New Functions,* SC26-7473. If BLOCKTOKENSIZE(NOREQUIRE) is in effect, then generally you can expect programs that use QSAM or use BSAM without NOTE or POINT to work unchanged.

Lower level systems will not be able to use large format data sets, and an abend will be issued unless the data set is not more than 65535 tracks in size and you are opening it for input only.

Toleration maintenance

Toleration PTFs will be required for lower level systems so that the appropriate action is taken when opening a data set with the new attribute. These are listed in Table 10-1 on page 186.

Table 10-1 Toleration PTFs for lower level systems

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08515	UA15869	UA15870	UA15871
OA08517	UA15869	UA15870	UA15871

New DSNTYPE values

In addition to the new DSNTYPE of LARGE that was discussed previously, it is now possible with z/OS V1R7 to specify the following values:

- LARGE
- BASIC
- EXTREQ (extended required)
- EXTPREF (extended preferred)

With the exception of LARGE, these do not represent new data set types; however, they give the ability to create existing data set types by specifying the appropriate value for DSNTYPE. Previously it was only possible to create an extended format data set indirectly by using a dataclas definition or by using the LIKE= keyword on a DD statement. An extended format data set can now be defined explicitly by specifying the new values of EXTREQ or EXTPREF via the DSNTYPE keyword.

```
//ALLOC EXEC PGM=IEFBR14
//DD1  DD DSN=WELCH.TEST.EXTEND.FORMAT,DISP=(NEW,CATLG),
//  SPACE=(CYL,100),DSNTYPE=EXTREQ,STORCLAS=STANDARD,
//  DCB=(DSORG=PS,RECFM=FB,LRECL=80)
```

Figure 10-12 Allocating an extended format data set by using new DSNTYPE values

A basic format data set is a new term for a sequential data set that is neither extended format or large format. Specifying DSNTYPE=BASIC could be used, for example, to force a data set type of basic if you are unsure what attributes may be assigned by DATACLAS.

The new DSNTYPE values of LARGE, BASIC, EXTREQ, and EXTPREF can be assigned via any of the standard ways of allocating a new data set, namely:

- The DD statement
- TSO/E ALLOCATE command
- IDCAMS ALLOCATE
- Dynamic allocation (SVC 99)

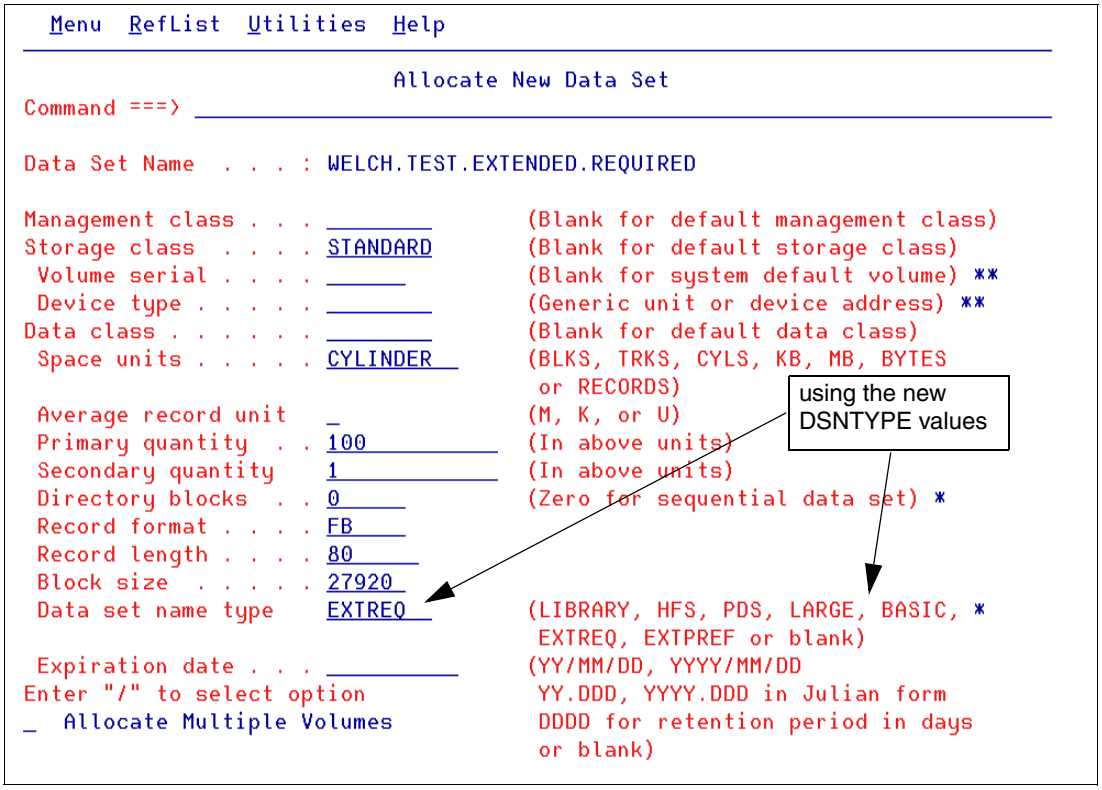


Figure 10-13 Using the new DSNTYPE values in ISPF to allocate a data set

ACS variable &DSNTYPE

The read-only variable &DSNTYPE available to the SMS ACS routines can now take the additional values of:

- LARGE
- BASIC
- EXR (extended required)
- EXC (extended preferred)

10.1.3 VSAM extent constraint relief

Currently a VSAM data set is restricted to a maximum of 255 extents per component for non-striped data sets and 255 extents per stripe for striped data sets, and with a maximum of 123 extents on any one volume. With z/OS V1R7 the 255 extent limit is removed for system-managed VSAM data sets, but note that the 123 extents per volume remains. This allows for a theoretical maximum of 7257 extents (59 volumes with 123 extents per volume).

Support for this feature is enabled via a new field Extent Constraint Removal in the dataclas definitions.

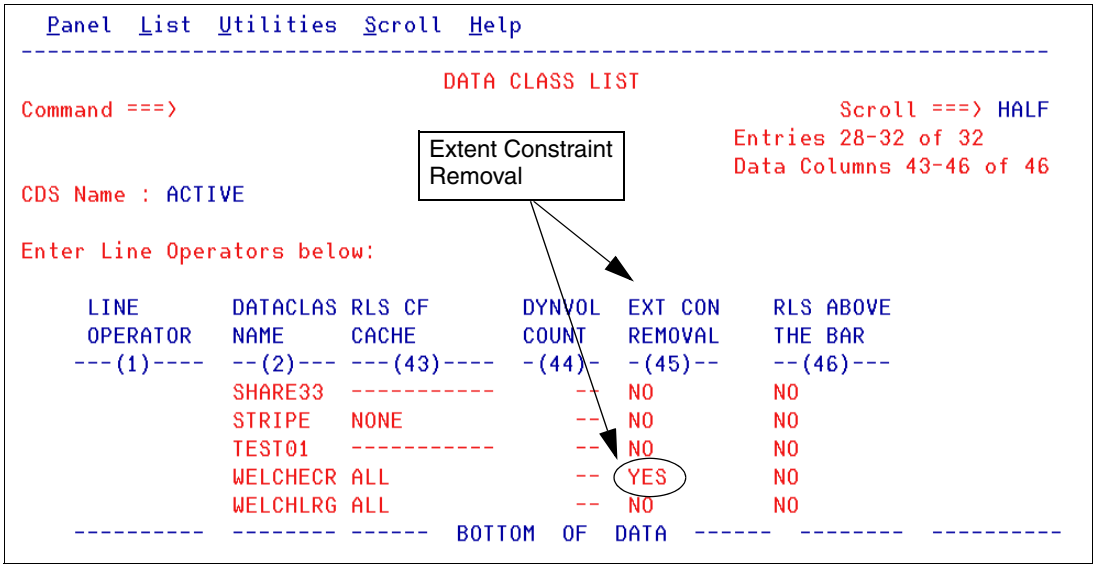


Figure 10-14 Extent Constraint Removal attribute on the DATACLAS definition

```
DEFINE CLUSTER (NAME(WELCH.KSDS.TESTX) -
               VOL   (SBOX1A,SBOX1B,SBOX1C) -
               DATACLAS(WELCHECR)    /* Extent Constraint Removal */ -
               STORCLAS(GSPACE) -
               KEYS  (4 0) -
               RECSZ (4096 4096) -
               TRACKS(1 1) -
               SHAREOPTIONS (2,3))
```

Figure 10-15 Defining a VSAM data set that can have more than 255 extents

Note: It can be difficult to observe a test case for this new feature because VSAM will automatically consolidate adjacent extents as it expands. This feature was introduced in z/OS V1R5.

In Figure 10-16 on page 189 you can see an example of a LISTCAT of a VSAM data set that has expanded beyond 255 extents.

```
SDSF OUTPUT DISPLAY LISTCAT JOB25431  DSID   102 LINE 28      COLUMNS 55- 134
COMMAND INPUT ===>                                          SCROLL ===> CSR
00.000
(NULL)
--(NO)



--4096     BUFSPACE-----------53760     CISIZE-------------26624
--4096     EXCPEXIT----------(NULL)     CI/CA------------------2
0ERASE     INDEXED       NOWRITECHK     NOIMBED       NOREPLICAT


-----0     EXCPS---------------1252
-----0     EXTENTS-------------262
-----0     SYSTEM-TIMESTAMP:
-----0          X'BCF81B306B8F33CA'
-----0

204416
204416



-26624     HI-A-RBA---------6549504     EXTENT-NUMBER--------123
            TIME: 16:20:32        05/12/05     PAGE      2
-----2     HI-U-RBA---------6549504     EXTENT-TYPE--------X'00'
-----1
```

More than 255 extents

maximum 123 extents per volume

Figure 10-16 LISTCAT of a VSAM data set with more than 255 extents

Migration and coexistence considerations

The default value for extent constraint removal is NO.

A VSAM data set defined with extent constraint removal should not be used if the data set may be shared with a pre-V1R7 system because lower level systems will not be able to access a data set that has expanded beyond 255 extents. The following example shows what happens when a job on a z/OS 1.6 system attempts to open a VSAM data set with more than 255 extents:

```
IEC161I 071-026,WELCHA,REPRO,IN,,,WELCH.KSDS.TESTX,
IEC161I WELCH.KSDS.TESTX.DATA,UCAT.VSBOX01
```

Toleration maintenance

Table 10-2 Toleration PTFs for lower level systems

APAR	Component	HDZ11G0	HDZ11H0	HDZ11J0
OA07128	VSAM	UA14882	UA14883	UA14884
OA07129	RLS	UA16223	UA16224	UA16225
OA08286	DFSMSdss	UA16796	UA16797	UA16798

Note: The DFSMSdss APAR OA08286 is required for both Extent Constraint Removal and for large format data sets.

&MAXSIZE variable

Users of the variable &MAXSIZE in ACS routines should be aware that the value of this variable now depends on whether or not the data set has been defined with or without extent constrain relief. For example, the maximum size of a VSAM data set with extent constraint removal set to YES is:

```
&MAXSIZE = Primary + (Secondary * Volume count * 123)
```

SMF record type 64

Users of the extent information section in the SMF record type 64 records should be aware that this section has been modified to only record data about the last extent.

10.1.4 SMS enhancements

This section describes the SMS enhancements that are implemented in the DFSMSdss component of DFSMS.

Change SMS status of volumes

In current releases, you cannot change the status of a volume from NOTCON (not connected) to any other status, for example ENABLE or QUIESCE, without updating the SCDS and activating the new configuration.

z/OS V1R7 provides an enhancement to allow you to use the **V SMS,VOLUME** command to alter the volume status from NOTCON to any of the other status. Note that to make this change permanent you will still need to update the SCDS.

New volume selection messages and traces

In z/OS V1R7 the following new functions are added for SMS-managed data sets to assist the user in performing problem diagnosis on volume selection with minimum assistance from IBM support personnel:

- Providing summarized and detailed analysis messages on request
- Externalizing DADSM failure reasons and diagnostic codes in summarized analysis messages
- Externalizing volume selection analysis data in SMS trace
- Adding new trace data for SMS-managed and non-SMS-managed VSAM allocations.

SMS volume selection messages

Currently SMS issues diagnostic messages when volume selection fails for a data set allocation. These messages provide diagnostic information by summarizing the number of volumes rejected for each failure reason during volume selection. An example is shown in Figure 10-17.

```
IGD17207I VOLUME SELECTION HAS FAILED - THERE ARE NO ACCESSIBLE
VOLUMES FOR DATA SET WELCH.TEST.LARGE
IGD17277I THERE ARE (9) CANDIDATE VOLUMES OF WHICH (0) ARE ENABLED OR QUIESCED
IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE FIRST 1
WERE ELIGIBLE FOR VOLUME SELECTION.
THE CANDIDATE STORAGE GROUPS WERE:DJLVSAM1
IGD17279I 9 VOLUMES WERE REJECTED BECAUSE THEY WERE NOT ONLINE
IGD17279I 9 VOLUMES WERE REJECTED BECAUSE THE UCB WAS NOT AVAILABLE
```

Figure 10-17 Existing messages for volume selection failure

In many cases, however, these messages are not sufficient to determine why an allocation has failed, and in some cases you may require information as to how a particular data set was successfully allocated. In z/OS V1R7 you can obtain additional detail for both successful and unsuccessful allocations.

IGDSMSxx parmlib member

There are new parameters in the IGDSMSxx parmlib member that control the issuance of summarized and detailed volume selection analysis messages when creating a new SMS-managed data set or extending an SMS-managed data set to a new volume.

```
VOLSELMSG({ON | OFF},{0 | nnnn | ALL})
```

The meanings of the parameters are as follows:

- ON | OFF controls whether or not you get the SMS volume selection analysis messages. The default is OFF.
- 0 | nnnn | ALL
  - 0 - only summarized messages will be produced. This is the default.
  - nnnn - number of volumes to be included in detailed analysis messages.
  - ALL - indicates all volumes will be included in detailed analysis messages

If you want to specify VOLSELMSG ON and request detailed volume analysis messages, then you will also need to specify at least one of the scope limiting parameters, as follows:

- JOBNAME
- ASID
- STEPNAME
- DSNAME

The parameters can also be specified by using the **SETSMS** command, as in the example in Figure 10-18.

```
SETSMS VOLSELMSG(ON,ALL),TYPE(ALL),JOBNAME(WELCHX)
IEE712I SETSMS   PROCESSING COMPLETE
```

Figure 10-18 Example of using the SETSMS command to set the VOLSELMSG options

The current setting can be displayed by either of the following commands:

```
D SMS,OPTIONS
D SMS,VOLSELMSG
```

An example of the **D SMS,VOLSELMSG** command is shown in Figure 10-19.

```
D SMS,VOLSELMSG
IGD002I 16:19:49 DISPLAY SMS 592
VOLSELMSG = (ON,ALL)    TYPE = ALL    JOBNAME = WELCHX
    ASID = *  STEPNAME = *
    DSNAME = *
TRACE    = ON     SIZE = 128K     TYPE = ALL
  JOBNAME = WELCHX      ASID = *
```

Figure 10-19 Command example to display VOLSELMSG options

With the new SMS option VOLSELMSG enabled, new messages are produced that contain additional information that can be useful in solving SMS allocation problems.

Message example 1

The example in Figure 10-20 shows the new summarized messages for an allocation failure. The setting of the VOLSELMSG options that applies is included in message IGD17386I and for this example specifies the following:

- VOLSELMSG(ON,0)
- TYPE(ERROR)
- The limiting keywords JOBNAME, ASID, STEPNAME, and DSNAME are not set.

This means that the new summarized messages are produced for any allocation errors.

```
IGD17385I =====SUMMARIZED ANALYSIS MESSAGES ON DEFINING DATA SET WELCH.TEST.LARGE =====
IGD17386I VOLSELMSG(ON,0) TYPE(ERROR) JOBNAME(*) ASID(*)
STEPNAME(* ) DSNAME(* )
IGD17387I DS_TYPE(NONVSAM) SC(DJLVSAM) DC() GS(Y) SPACE(830KB) BESTFIT(N) STRIPING(N)
IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE FIRST 1
WERE ELIGIBLE FOR VOLUME SELECTION.
THE CANDIDATE STORAGE GROUPS WERE:DJLVSAM1
IGD17279I 9 VOLUMES WERE REJECTED BECAUSE THEY WERE NOT ONLINE
IGD17279I 9 VOLUMES WERE REJECTED BECAUSE THE UCB WAS NOT AVAILABLE
```

Figure 10-20 Sample summarized allocation messages for a failure

Note: The new information that is supplied is in message IGD17387I, as follows:

```
IGD17387I DS_TYPE(NONVSAM) SC(DJLVSAM) DC() GS(Y) SPACE(830KB) BESTFIT(N) STRIPING(N)
```

Message example 2

The next example, in Figure 10-21 on page 193, shows the new summarized messages for a successful allocation. Note that the setting of the VOLSELMSG options this time is:

- VOLSELMSG(ON,0)
- TYPE(ALL)
- JOBNAME(WELCHX) and DSNAME(WELCH.SPECIAL)

At least one of the limiting keyword is required when TYPE(ALL) is specified. This means that the new summarized messages will be produced for both successful and unsuccessful allocations for any jobs that match the specified criteria. In this case the job name and data set name must match the values specified in the limiting keywords. Any one or more of the four limiting parameters can be used to control the scope of the production of these messages.

You must be careful that you do not generate an excessive number of messages.

```
IGD17385I =====SUMMARIZED ANALYSIS MESSAGES ON DEFINING DATA SET WELCH.SPECIAL ====
IGD17386I VOLSELMSG(ON,0) TYPE(ALL) JOBNAME(WELCHX ) ASID(*)
STEPNAME(* ) DSNAME(WELCH.SPECIAL )
IGD17387I DS_TYPE(NONVSAM) SC(DB8B) DC(WELCHLRG) GS(N) SPACE(830KB) BESTFIT(N)
STRIPING(N)
IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE FIRST 1
WERE ELIGIBLE FOR VOLUME SELECTION.
THE CANDIDATE STORAGE GROUPS WERE:DB8B
```

Figure 10-21 Sample summarized allocation messages for a successful allocation

Message examples 3 and 4

As mentioned previously, you can also obtain additional information at the volume level. In the next two examples, Figure 10-22 and Figure 10-23, the setting is VOLSELMSG(ON,ALL), which means that information about all of the volumes that were used for volume selection will be included in the detailed analysis messages.

```
IGD17385I =====SUMMARIZED ANALYSIS MESSAGES ON DEFINING DATA SET WELCH.TEST.LARGE =====
IGD17386I VOLSELMSG(ON,ALL) TYPE(ALL) JOBNAME(WELCHX ) ASID(*)
STEPNAME(* ) DSNAME(* )
IGD17387I DS_TYPE(NONVSAM) SC(DB8B) DC(WELCHLRG) GS(N) SPACE(830KB) BESTFIT(N)
STRIPING(N)
IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE FIRST 1
WERE ELIGIBLE FOR VOLUME SELECTION.
THE CANDIDATE STORAGE GROUPS WERE:DB8B
IGD17385I =====DETAILED ANALYSIS MESSAGES ON DEFINING DATA SET WELCH.TEST.LARGE =====
IGD17388I ==POOL SG DB8B
IGD17389I MHL125(S,3868001D) MHL126(N,3868001D)
```

Volume selected

Volume not selected

Figure 10-22 Successful allocation with detailed volume information

```
IGD17385I =====SUMMARIZED ANALYSIS MESSAGES ON DEFINING DATA SET WELCH.TEST.LARGE =====
IGD17386I VOLSELMSG(ON,ALL) TYPE(ERROR) JOBNAME(* ) ASID(*)  095
STEPNAME(* ) DSNAME(* )
IGD17387I DS_TYPE(NONVSAM) SC(DB8B) DC(WELCHLRG) GS(N) SPACE(6640312KB) BESTFIT(N)
STRIPING(N)
IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE FIRST 1  097
WERE ELIGIBLE FOR VOLUME SELECTION.
THE CANDIDATE STORAGE GROUPS WERE:DB8B
IGD17279I 2 VOLUMES WERE REJECTED BECAUSE THEY DID NOT HAVE SUFFICIENT SPACE (041A041D)
IGD17385I =====DETAILED ANALYSIS MESSAGES ON DEFINING DATA SET WELCH.TEST.LARGE =====
IGD17388I ==POOL SG DB8B
IGD17389I MHL125(R,0868001D,00020000,041A041D) MHL126(R,0868001D,00020000,041A041D)
```

Volume rejected

insufficient space

DADSM diagnostic code

Figure 10-23 Unsuccessful allocation with detailed volume information

Message IGD17398I

The format of the IGD17389I message is:

```
volser(x,preference_flags,failure_flags,diagnostic_data)
```

In this example, x can be:

- S for volume selected
- R for volume rejected
- N for volume not used

For a full explanation of the IGD17389I message refer to the manual *z/OS MVS System Messages Volume 8 (IEF-IGD),* SA22-7638.

Additional DADSM failure reason codes

Currently SMS provides only 2 distinct failure reason codes if a volume is rejected by DADSM:

```
IGD17279I nn VOLUMES WERE REJECTED BECAUSE OF DADSM FAILURE
IGD17279I nn VOLUMES WERE REJECTED BECAUSE THEY DID NOT HAVE SUFFICIENT SPACE
```

With z/OS V1R7 six new DADSM failure reason codes have been added, as follows:

```
IGD17279I nn VOLUMES WERE REJECTED BECAUSE...
   OF DUPLICATE DATA SET NAME (diagnostic code)
   OF NO ROOM IN VTOC OR INDEX (diagnostic code)
   OF PERMANENT I/O OR CVAF ERROR (diagnostic code)
   THEY WERE REJECTED BY INSTALLATION EXIT (diagnostic code)
   THEY WERE NOT INITIALIZED (diagnostic code)
   OF EOF MARK WRITE FAILED (diagnostic code)
```

Each DADSM failure reason may have different diagnostic codes. An example is shown in Figure 10-24.

```
IGD17279I 23 VOLUMES WERE REJECTED BECAUSE THE SMS VOLUME STATUS WAS DISABLED
IGD17279I 1 VOLUMES WERE REJECTED BECAUSE OF DUPLICATE DATA SET NAME (041C0416)
IGD17219I UNABLE TO CONTINUE DEFINE OF DATA SET
WELCH.TEST
```

Figure 10-24 Example of a new DADSM error reason code for a SMS allocation

Volume selection trace data

When you take an SMS trace with the VTOCC or VTOCA option, then the trace facility will now create a trace entry containing an IGDVSUIB (volume selection user information block) in the private area of the SMS address space for each data set being traced. IGDVSUIB will contain key information used in SMS volume selection, which can be used to perform problem diagnosis on volume selection problems. The options have the following meanings:

- VTOCA - Add a volume to a data set.
- VTOCC - Create a new data set.

Figure 10-25 on page 195 shows an example of a display of the SMS trace options.

```
D SMS,TRACE
IGD002I 12:27:42 DISPLAY SMS 218
TRACE    = ON      SIZE = 128K     TYPE = ALL
  JOBNAME = WELCHX       ASID = *
  TRACING EVENTS:
    MODULE = ON    SMSSJF = ON    SMSSSI = ON    ACSINT = ON
    OPCMD  = ON    CONFC  = ON    CDSC   = ON    CONFS  = ON
    MSG    = ON    ERR    = ON    CONFR  = ON    CONFA  = ON
    ACSPRO = ON    IDAX   = ON    DISP   = ON    CATG   = ON
    VOLREF = ON    SCHEDP = ON    SCHEDS = ON    VTOCL  = ON
    VTOCD  = ON    VTOCR  = ON    VTOCC  = ON    VTOCA  = ON
    RCD    = ON    DCF    = ON    DPN    = ON    TVR    = ON
    DSTACK = ON    UAFF   = ON
VOLSELMSG = (ON,ALL)     TYPE = ALL     JOBNAME = WELCHX
    ASID = *  STEPNAME = *
    DSNAME = *
```

SMS trace options

Figure 10-25 Display SMS trace options

```
IPCS OUTPUT STREAM ----------------------------------------- Line 95 Cols 1 78
Command ===>                                                  SCROLL ===> CSR
            +000004B0: 00000000 00000000                     *........*

 7F72DE40 RTRN06258D34 BD08791D50D95F4A 000009C0 E01 H-006E P-006E S-006E 8004
          Trace Events         : VTOCC  VTOCA
          Recording Module     : IGDVTSC2
          Recording Procedure  : BUILD_SG_MSG
          Control Block Name   : IGDVSUIB
          Control Block Data   :
           +00000000: C9C7C4E5 E2E4C9C2 00010000 00000958   *IGDVSUIB........*
           +00000010: 00000026 E6C5D3C3 C84BE3C5 E2E34040   *....WELCH.TEST  *
           +00000020: 40404040 40404040 40404040 40404040   *                *
           +00000030: 40404040 40404040 40404040 40404040   *                *
           +00000040: C7E2D7C1 C3C54040 0006D4C3 C4C2F2F2   *GSPACE  ..MCDB22*
           +00000050: 00000037 00000000 00000000 C0000000   *.......{....{...*
           +00000060: 00000000 00000000 00000000 00000000   *................*
           +00000070: D6D7C5D5 D4E5E200 01000000 E2C2D6E7   *OPENMVS.....SBOX*
           +00000080: F1C300D5 00001FB8 00001F37 00001A75   *1C.N.....8...7...*
           +00000090: 3868001D 00000000 00000000 00000000   *................*
           +000000A0: 00000000 00000000 00000000 D6D7C5D5   *............OPEN*
           +000000B0: D4E5E200 01000000 E2C2D6E7 F1C600D5   *MVS.....SBOX1F.N*
```

Figure 10-26 IPCS SMSTRACE showing new volume selection data

For further information see *DFSMSdfp Diagnosis,* GY27-7618.

New trace data for SMS-managed and non-SMS-managed VSAM allocations

This enhancement will add a new trace entry after the return from SVC 26 and will dump out additional critical data during the define of an SMS-managed or non-SMS-managed VSAM data set such as:

- Volume lists
- UCB lists
- Space information

Figure 10-27 on page 196 shows new trace data for VSAM allocations.

```
IPCS OUTPUT STREAM ---------------------------------------- Line 152 Cols 1 78
Command ===>                                                SCROLL ===> CSR
           +000000100 000180                                *.......*

 7F71F7A0 RTRN062596ED BD089C0E75BCC4AA 000000A0 N05 H-006E P-006E S-006E 8004
          Trace Events        : VTOCC
          Recording Module    : IGDVTSCT
          Recording Procedure : AFTER_SVC26_CALL
          Control Block Name  : DATA VOLLIST
          Control Block Data  :
           +00000000: 00100001 E2C2D6E7 F1C10003 E2C2D6E7   *....SBOX1A..SBOX*
           +00000010: F1C2                                  *1B*

 7F71F840 RTRN062596EE BD089C0E75BD95AA 00000080 N06 H-006E P-006E S-006E 8004
          Trace Events        : VTOCC
          Recording Module    : IGDVTSCT
          Recording Procedure : AFTER_SVC26_CALL
          Control Block Name  : DATA UCBLIST
          Control Block Data  :
           +00002021E 2388021E 91A8                         *.....h..jy*
```

Figure 10-27 New trace data for VSAM allocations

New ACS environment - allocation test

ACS services can be invoked by various callers representing different environments, for example ALLOC, RECALL and RECOVER. Frequently ACS services need to know who the caller is since processing may be different for the different environments. Some ISV products invoke ACS services directly with no specific environment, causing unpredictable results.

With z/OS V1R7 a new value ALLOCTST has been provided for the ACS environment variable &ACSENVIR. This has been added to enable your ACS routines to distinguish the new Allocation Test environment from the environment that represents the systems allocation environment.

Performance improvement for VARY command

SMS maintains a table where z/OS and SMS volume status is maintained. This enhancement streamlines the process by which the table is updated with the z/OS status of the volume.

Table 10-3 Support for VARY command enhancement added via APARs

APAR	HDZ11F0	HDZ11G0	HDZ11H0	HDZ11J0
OA03641		UA06829	UA06830	
OA05546	UA07370			
OA06234	UA09643	UA09644	UA09645	
OA07033	UA09643	UA09644	UA09645	
OA07446	UA10496	UA10497	UA10498	UA10499
OA08259	UA12556	UA12557	UA12558	UA10559

10.1.5 Removal of STEPCAT/JOBCAT support

IBM has previously indicated that support for the use of STEPCAT and JOBCAT DD statements was to be withdrawn. As a first step, in z/OS V1R5 usage of STEPCAT/JOBCAT is disabled by default and had to be enabled by a catalog modify command.

In z/OS V1R7 usage of JOBCAT/STEPCAT is not supported at all. The removal of this support has allowed improvements in catalog performance.

Any job that attempts to use a STEPCAT or JOBCAT DD statement will fail with this message:

```
IEFC034I JOBCAT OR STEPCAT NOT PERMITTED
```

If you are still using STEPCAT and JOBCAT DD statements, for example during the build of a master catalog for a new system, then there are documented alternate processes that do not require using STEPCAT or JOBCAT, as follows:

- Use system-specific aliases to direct the catalog entries to a specific catalog.
- Use the CATALOG statement on IDCAMS DEFINE statements.

Note: Current usage of STEPCAT and JOBCAT requires that the UCB of the device containing the catalog is in 24-bit addressable storage. Elimination of STEPCAT and JOBCAT will allow these devices to be redefined with LOCANY=YES in the hardware definition.

10.1.6 Removal of ISAM support

Data sets with an organization of ISAM are not supported on a z/OS V1R7 system. Any job that attempts to create a data set with a DSORG of IS on a z/OS V1R7 system will fail with the messages shown in Figure 10-28.

```
IEF344I WELCHA STEP1 ISAMOUT - ALLOCATION FAILED DUE TO DATA FACILITY SYSTEM ERROR
IGD17039I ALLOCATION FAILED FOR DATA SET
WELCH.TEST.ISAM , THE SYSTEM NO LONGER
SUPPORTS CREATION OF INDEXED SEQUENTIAL DATA SETS
```

Figure 10-28 Unable to create ISAM data sets on z/OS V1R7

Any job that attempts to access an ISAM data set on a z/OS V1R7 fails with the messages shown in Figure 10-29.

```
IEC134I WELCHA,TEST,ISAMIN,WELCH.TEST.ISAM Z/OS AS OF V1R7 DOES NOT ALLOW OPENING OF
INDEXED SEQUENTIAL DATA SETS
IEC143I 213-1C,IFG0194A,WELCHA,TEST,ISAMIN,8405,SBOX00,WELCH.TEST.ISAM
```

Figure 10-29 Unable to open an ISAM data set on z/OS V1R7

For systems on earlier levels of z/OS and on current maintenance, any usage of ISAM data sets is flagged with the message shown in Figure 10-30.

```
IEC134I WELCHA,TEST,ISAMIN,WELCH.TEST.ISAM WARNING: IBM INTENDS TO DROP ISAM
SUPPORT IN A FUTURE RELEASE
```

Figure 10-30 Current usage of ISAM on pre-z/OS V1R7 systems is flagged

Note: All ISAM data sets must be converted to VSAM before migrating to z/OS V1R7.

10.1.7 DEFINE PAGESPACE with a CATALOG parameter

At present, an attempt to create a new page data set in a catalog other than the running system results in the problem described by OW32120. In z/OS V1R7, to prevent this situation occurring, IDCAMS now returns without allocating the page data set and with issues message IDC3171I, unless the RECATALOG parameter is specified.

```
IDC3171I CATALOG PARAMETER NOT ALLOWED ON DEFINE PAGESPACE UNLESS RECATALOG SPECIFIED
```

Figure 10-31 CATALOG parameter not allowed for DEFINE PAGESPACE

In situations where you need to define a new page data set in another system's master catalog, there are several different procedures that can be used to accomplish this. Figure 10-32 illustrates use of the recatalog method. Note that in this case the CATALOG parameter is only allowed with RECATALOG.

```
//NEWPAGE  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DEFINE  PAGESPACE (NAME(SYS1.SYSX.LOCAL4) -
   VOLUME(SBOXFD) CYLINDERS(3330))
/*
//RECAT     EXEC  PGM=IDCAMS
//SYSPRINT  DD    SYSOUT=*
//SYSIN     DD  *
 DEFINE PAGESPACE  (NAME(SYS1.SYSX.LOCAL4) -
   VOLUME(SBOXFD) RECATALOG) CATALOG(CATALOG.SYSX)
/*
```

Figure 10-32 JCL to define a new page data set in a different catalog

10.1.8 Device support address space (DEVMAN)

There is a new system started task DEVMAN that is started at IPL time and provides a data space for collecting component trace data for CVAF and DADSM events. This new component trace replaces GTF tracing for CVAF events.

Any CVAF and DADSM initiated dumps will include the CTRACE data space for diagnostics. In order to capture CVAF trace data, tracing must be enabled:

```
   TRACE CT,ON,COMP=SYSDMO
   *579 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.

    R 579,OPTIONS=(CVAF1,DADSM1),END
```

Figure 10-33 Turn CVAF trace on

To terminate tracing, issue the command shown in Figure 10-34.

```
TRACE CT,ON,COMP=SYSDMO
*580 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.

 R 580,OPTIONS=(CVAF0,DADSM0),END
```

Figure 10-34 Stop CVAF trace

The modify commands for the DEVMAN started task are documented in the JCL in SYS1.IBM.PROCLIB.

```
F DEVMAN,DUMP=DUMP1
 IEA794I SVC DUMP HAS CAPTURED: 710
 DUMPID=005 REQUESTED BY JOB (DEVMAN  )
 DUMP TITLE=COMPONENT=DEVICE MANAGER,COMPID=DF133,ISSUER=DMOVS00
            1,JOBNAME=DEVMAN
```

IPCS is used to format and display the collected CTRACE data.

```
IPCS OUTPUT STREAM ----------------------------------------- Line 0 Cols 1 78
Command ===> _                                               SCROLL ===> CSR
 ******************************** TOP OF DATA ***********************************


  COMPONENT TRACE FULL FORMAT
  COMP(SYSDMO)
  **** 05/09/2005

  SYSNAME   MNEMONIC  ENTRY ID    TIME STAMP     DESCRIPTION
  -------   --------  --------  ---------------  -------------

  SC70      TRDATA    40000004  10:35:27.169077  DELETE INDEX

     +0000  MODULE... ICVIXD00  LOCATION. DELETE INDEX
     +0010  SEQNO.... 00000000  CPUID.... 0041      ASID..... 006A
     +001C  TCB...... 007E26B0  DATA@.... 7F6C7414  DATASIZE. 0064
     +004C  USER..... (TRDATA)
     +006C  CVAFDSN.. WELCH.TEST.ALLOCATE
     +0098  CVAFVOL.. SBOXFD    CVAFCCHR. 00030000  04
     +00A3  CVAFRBA.. 00000000
  SC70      TRDATA    40000002  10:35:27.172693  WRITE VIR TO VTOC INDEX

     +0000  MODULE... ICVCMI00  LOCATION. WRITE VIR TO VTOC INDEX
     +0010  SEQNO.... 00000001  CPUID.... 0041      ASID..... 006A
     +001C  TCB...... 007E26B0  DATA@.... 7F6C7A1C  DATASIZE. 0064
     +004C  USER..... (TRDATA)
     +006C  CVAFDSN.. WELCH.TEST.ALLOCATE
     +0098  CVAFVOL.. SBOXFD    CVAFCCHR. 00000000  00
     +00A3  CVAFRBA.. 00000000
```

Figure 10-35 IPCS formatted CVAF CTRACE records

10.1.9 New DEVSERV QLIB command

The **DEVSERV** command has been enhanced to display tape library information. Use the QLIB option of the **DEVSERV** command to:

- Request a list of tape library subsystems that are defined to the host. Libraries are listed by serial number (library-ID).

- Request a list of devices within a library. Devices are listed by device number and the library port for each device is displayed.
- Validate the connection status of devices in a library, for example, devices that are connected to the host.
- Delete an improperly defined library control block in preparation for an IODF activate.
- Issue a diagnostic state save order to a library when requested by the IBM Support Center.

```
DS QL,LIST
IEE459I 11.31.04 DEVSERV QLIB 465
The following are defined in the   ACTIVE configuration:
*10435

DS QL,10435
IEE459I 11.32.04 DEVSERV QLIB 467
The following are defined in the   ACTIVE configuration:
LIBID  PORTID       DEVICES
10435  03           0B90* 0B91* 0B92* 0B93*
```

Figure 10-36 Sample DEVSERV QLIB displays

For the syntax of the DEVSERV QLIB command and further examples, see *z/OS V1.R7 MVS System Commands,* SA27-7627.

10.1.10 Catalog improvements

The catalog improvements in DFSMSdfp are described in this section.

VVDS default space parameter

A VVDS is defined implicitly when the first VSAM data set is allocated on a non-SMS-managed volume, or any data set is allocated on an SMS-managed volume, and there is no VVDS currently on the volume. The current default size is TRACKS(10,10) which may be too small for sites that use custom 3390 volumes (larger than a 3390-9). These volumes can have a large number of data sets that will require entries in the VVDS.

The system default can now be set using the new VVDSSPACE keyword of the **F CATALOG** command, as follows:

```
F CATALOG,VVDSSPACE(primary,secondary)
```

For example:

```
F CATALOG,VVDSSPACE(30,15)
```

This will set the default VVDS space to 30 tracks primary and 15 tracks secondary.

The current system defaults can be displayed with the REPORT option of the **F CATALOG** command.

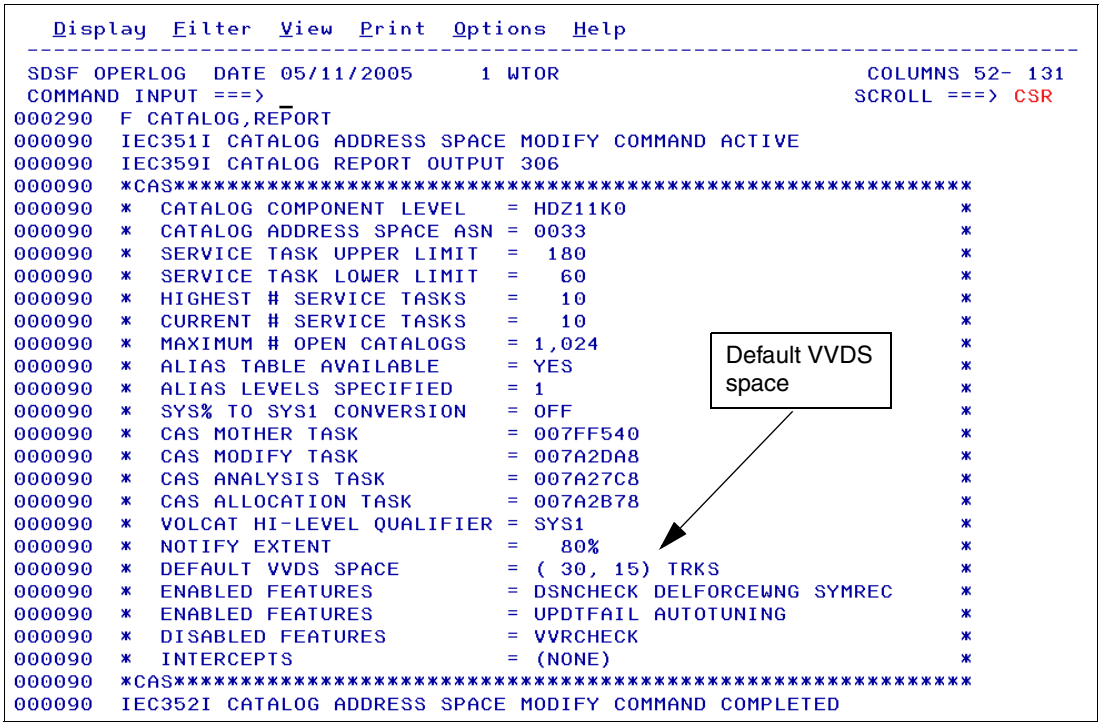


Figure 10-37 CATALOG REPORT shows current default values for VVDS space

Note: Any new site default is retained across a catalog restart but is *not* preserved across an IPL.

CATALOG service task lockup

The CATALOG address space has a maximum number of service tasks (the default is 180) that are available to process catalog requests. Sometimes an address space may use a large number of these service tasks, which can affect the performance of other work in the z/OS system or use all of the available service tasks and prevent further activity. The current setting of this value can be seen in the output from a **CATALOG REPORT** command. For example, see Figure 10-37.

To assist with this situation a new message IEC392I has been introduced to identify the top three holders of the CATALOG service tasks whenever the task maximum is reached.

```
IEC392I JOB jobname HOLDS xxx OF THE yyy MAXIMUM SERVICE TASKS
```

CATALOG autotuning support

The performance of a catalog is dependent on the following three parameters:

- Number of strings
- Number of data buffers
- Number of index buffers

The new catalog autotuning enhancement automatically tunes some default parameter values to improve performance, temporarily modifying the number of data and index buffers and VSAM strings for the catalog on the current system. When the tuning occurs, message IEC391I displays the new tuned values as shown in Figure 10-38 on page 202.

```
IEC391I CATALOG UCAT.TCIODF HAS BEEN AUTOTUNED TO:
IEC391I BUFNI:    10 BUFND:    16 STRNO:     8
```

Figure 10-38 Example of CATALOG autotuning

If the values are increased for a particular catalog, they will be remembered if the catalog is closed and then referenced again. Catalog performance is monitored every 10 minutes.

Note: The tuned values are temporary in the sense that the actual catalog entry will not reflect the tuned values unless you make them permanent by using the **IDCAMS ALTER** command to update the catalog definition.

Use of the new catalog autotuning feature can be controlled by using the catalog **ENABLE** and **DISABLE** command. By default, catalog autotuning is enabled. To disable this feature you would issue the command:

```
F CATALOG,DISABLE(AUTOTUNING)
```

The current setting of this value can be seen in the output from a **CATALOG REPORT** command in the section showing enabled and disabled features. See the sample REPORT output in Figure 10-39.

```
  Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF OPERLOG  DATE 05/11/2005     1 WTOR                      COLUMNS 52- 131
 COMMAND INPUT ===>                                           SCROLL ===> CSR
000290  F CATALOG,REPORT
000090  IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
000090  IEC359I CATALOG REPORT OUTPUT 306
000090  *CAS*****************************************************************
000090  *  CATALOG COMPONENT LEVEL    = HDZ11K0                              *
000090  *  CATALOG ADDRESS SPACE ASN = 0033                                  *
000090  *  SERVICE TASK UPPER LIMIT   =  180                                 *
000090  *  SERVICE TASK LOWER LIMIT   =   60                                 *
000090  *  HIGHEST # SERVICE TASKS    =   10                                 *
000090  *  CURRENT # SERVICE TASKS    =   10                                 *
000090  *  MAXIMUM # OPEN CATALOGS    = 1,024                                *
000090  *  ALIAS TABLE AVAILABLE      = YES                                  *
000090  *  ALIAS LEVELS SPECIFIED     = 1                                    *
000090  *  SYS% TO SYS1 CONVERSION    = OFF                                  *
000090  *  CAS MOTHER TASK            = 007FF540                             *
000090  *  CAS MODIFY TASK            = 007A2DA8                             *
000090  *  CAS ANALYSIS TASK          = 007A27C8                             *
000090  *  CAS ALLOCATION TASK        = 007A2B78                             *
000090  *  VOLCAT HI-LEVEL QUALIFIER = SYS1                                  *
000090  *  NOTIFY EXTENT              =   80%                                *
000090  *  DEFAULT VVDS SPACE         = ( 30, 15) TRKS                       *
000090  *  ENABLED FEATURES           = DSNCHECK DELFORCEWNG SYMREC          *
000090  *  ENABLED FEATURES           = UPDTFAIL AUTOTUNING                  *
000090  *  DISABLED FEATURES          = VVRCHECK                             *
000090  *  INTERCEPTS                 = (NONE)                               *
000090  *CAS*****************************************************************
000090  IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

CATALOG AUTOTUNING

Figure 10-39 CATALOG AUTOTUNING enabled

REPRO MERGECAT enhancements

The IDCAMS REPRO MERGECAT function has been enhanced to provide the capability to copy a range of records from one user catalog to another. This, for example, gives the ability to mend a broken catalog by enabling you to copy from one specific key to another specific key just before where the break occurred and then recover data beginning after the break.

```
 REPRO  MERGECAT INDATASET(UCAT.WELCH1) -
        OUTDATASET(UCAT.WELCH2)          -
        FROMKEY(WELCHA.C.*) TOKEY(WELCHA.E.*)
IDC01460I THE NUMBER OF ENTRIES MERGED WAS 6
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

Figure 10-40 Example REPRO MERGECAT with FROMKEY/TOKEY

Performance considerations

Currently REPRO MERGECAT processing uses CATALOG management requests to perform a DELETE NOSCRATCH and DEFINE RECATALOG for every data set. This can be a lengthy and expensive process for large catalogs. With z/OS V1R7 the MERGECAT processing performs updates directly, giving a performance improvement with a reduction in:

- VVDS I/O
- ENQ/DEQ activity
- CPU consumption

10.1.11 XRC Plus

XRC Plus is an enhancement to XRC that provides a solution to the problem of how to mirror logdata from the System Logger for a disaster recovery system. The current solution requires using mirrored disk staging data sets, which forces a wait for a synchronous I/O to take place to the staging data set even if you are also using Coupling Facility structures. This can mean an unacceptable delay to your applications, which will wait for the logdata to be written to both the CF structure and the disk staging data set.

The solution to this problem allows the System Logger to use the Coupling Facility and write asynchronously to the disk staging data sets. XRC Plus provides the capability to mirror the volume that contains the system logger staging data sets and maintain consistency. By allowing the system logger to use the coupling facility, performance will be improved.

The new LOGPLUS keyword for the **XADDPAIR** command specifies that the primary volume of the volume pair is to be explicitly written to by the z/OS System Logger. When using LOGPLUS, you can specify a single pair or a single pair and a utility pair, but the utility pair must be specified last. A unique storage control session number will be assigned to the primary volume. LOGPLUS is available only in DFSMS z/OS V1R7 and later, and is mutually exclusive with the SCSESSION keyword.

```
XADDPAIR LOGXPLUS VOLUME(DRXRC1 DRXRC2 XRCP1C XRCUTL) ERRORLEVEL(SESSION)
LOGPLUS
```

10.1.12 OAM enhancements

In previous releases, OAM allowed for movement of objects within the storage hierarchy; however, this movement did not take place until the OAM storage management component (OSMC) cycle ran for the storage group where the object resides. When objects have been written to removable media, the response time for reads is 30-60 seconds instead of milliseconds. In many cases, when an object has been referenced, the chances that it will be referenced again are very high. These subsequent references can be much faster if the object has already been recalled to a DB2 table residing on DASD.

Immediate recall to DB2

z/OS V1R7 supports the immediate recall to DB2 of objects that currently reside on removable media, for a specified number of days. This allows subsequent requests to read the objects to be satisfied from DB2 DASD rather than another read from the tape or optical volume where the object resides. OSMC will restore the object to its original location after the specified number of days have passed.

Immediate recall to DB2 is activated implicitly via parmlib parameters or explicitly via the OSREQ macro interface. When a full or partial object is retrieved from optical or tape, a full copy of that object is written to DB2 DASD. The object's location field is updated with a new location value of R to indicate that the object has been recalled to DB2.

The object's pending action date will be set to the current date plus the number of days the object is to be recalled for. Once the pending action date has elapsed, OSMC will delete the object from DB2 DASD and restore the object directory to point back to the original optical or tape location.

For each OSREQ RETRIEVE request successfully processed for an object that currently resides on optical or tape media, OAM will determine whether or not a recall to DB2 DASD is required, in one of the following ways:

- Explicitly via the RECALL keyword on the OSREQ RETRIEVE
- Implicitly via SETOSMC keywords in the CBROAMxx parmlib member

If the recall is required, then at the same time the OSR component of OAM is servicing the object retrieval, OAM initiates a request to the OSMC component to write a full copy of the object to DB2 DASD. Recalls are processed asynchronously from the actual RETRIEVE request in order to avoid impact to RETRIEVE response time.

SETOSMC statement in CBROAMxx parmlib member

Following are new keywords in the SETOSMC parameter in parmlib:

- RECALLOPTICAL(nnn)
  - Number of days an object is recalled to DB2 DASD from optical prior to OSMC restoring the object back to its original location
  - 0-255 (0 indicates that the recall is for the current day only)
  - Global or at the storage group level
  - Pending action data set to today's date plus the number of days specified
  - Applies to implicit recalls only
- RECALLTAPE(nnn)
  - Number of days an object is recalled to DB2 DASD from tape prior to OSMC restoring the object back to its original location.
  - 0-255 (0 indicates that the recall is for the current day only)
  - Global or at the storage group level
  - Pending action data set to today's date plus the number of days specified
  - Applies to implicit recalls only
- RECALLALL(nnn)
  - Number of days an object is recalled to DB2 DASD from tape or optical prior to OSMC restoring the object back to its original location
  - 0-255 (0 indicates that the recall is for the current day only)

  - Global or at the storage group level
  - Pending action data set to today's date plus the number of days specified
  - Applies to implicit recalls only
- RECALLNONE
  - Indicates that when objects residing on optical or tape are retrieved, they will not be recalled to DB2 DASD
  - Global or at the storage group level
  - Applies to implicit recalls only
- RECALLOFF(mode)
  - (ON | OFF)
  - ON means that explicit and implicit recalls are disabled
  - OFF means that explicit and implicit recalls are enabled
- MAXRECALLTASKS(nn)
  - Maximum number of RECALL tasks that can be run concurrently
  - 0-255 (Default is 0)
  - Explicit and implicit RECALLS are disabled if MAXRECALLTASKS=0
  - Specified at the global level only
  - Applies to explicit and implicit recalls

These values can also be set dynamically by using the modify **OAM UPDATE** command, as follows:

```
F OAM,UPDATE,SETOSMC,ALL,RECALLA,2
F OAM,UPDATE,SETOSMC,MYGRP,RECALLF,ON
```

For further information see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426.

Migration considerations

To use the new immediate recall function you must specify the new SETOSMC parameters in the CBFOAMxx parmlib member to specify the RECALL parameters to be used when an object is retrieved.

Applications can make use of this new function, regardless of the PARMLIB setting, by explicitly coding the RECALL parameter on the OSREQ RETRIEVE requests.

If the new function is not going to be used then no changes are necessary.

Users of the OSREQ macro should note that it has two new parameters and has increased in size from 96 bytes to 120 bytes.

SMF record type 85 is updated to provide information on immediate recall.

Toleration maintenance is required for lower level systems to coexist in an OMAplex with a z/OS V1R7 system. The APAR is OA08230.

Enhanced MOVEVOL utility

The MOVEVOL utility is enhanced to accommodate OAM scratch volumes. This allows MOVEVOL to be used with the DELETE option to remove scratch volumes from OAM's inventory.

New parameter TAPEDISPATCHERDELAY

A new TAPEDISPATCHERDELAY parameter is added to the SETOAM keyword in the CBROAMxx parmlib member to delay processing of certain requests and minimize demounting and remounting tapes.

- TAPEDISPATCHERDELAY(nn)
  - Specifies the number of seconds that the OAM tape dispatcher is to delay processing of certain requests in order to minimize demounting and remounting the same tape volumes in certain applications
  - Global level only
  - Valid values for nn are 1 to 60 seconds

The OAM tape dispatcher will delay processing of a unit of work for a specific period of time, when *all* of the following conditions are true:

- A read request for an object on a currently mounted tape volume has just been completed.
- There is no request for the currently mounted tape volume waiting to be processed on the OAM tape dispatcher queue.
- The OAM tape dispatcher has found a request for another tape volume and is about to dispatch this unit of work.
- A nonzero tape dispatcher delay value has been specified with the TAPEDISPATCHERDELAY keyword on the SETOAM statement in the CBROAMxx parmlib member.

If all of the previous conditions are true, the OAM tape dispatcher delays the dispatching of this selected unit of work (for the number of seconds specified by the installation) expecting that another read request for the currently mounted tape volume will arrive within this delay interval. The OAM tape dispatcher will delay dispatching of the selected unit of work for up to the number of seconds specified with the TAPEDISPATCHERDELAY keyword on the SETOAM statement in the CBROAMxx PARMLIB member.

- If another read request for the currently mounted tape volume arrives within the delay interval, that unit of work will be dispatched immediately upon arrival.
- If no read request for the currently mounted volume arrives within the delay interval, another request for a different tape volume is dispatched.

The value specified with the TAPEDISPATCHERDELAY value can be used to circumvent a performance problem experienced by applications that serially send down multiple read requests for data on a given tape volume.

New parameter CLEAROLDLOC

When an object is moved from an optical or tape volume to DB2 DASD during an OSMC cycle, the volser and sector location, or blockid, is retained in the object directory. This is because the transition to DB2 is usually temporary and the object will later move back to optical or tape. While an optical or tape volser is associated with an object, that volser cannot be expired, even if the object is currently residing in DB2.

z/OS V1R7 adds a new CLEAROLDLOC keyword to instruct OAM to clear the original volser and sector location or blockid in the object directory for a given object when that object is moved by OSMC to DB2 DASD. This new keyword will be most useful to installations that do not normally transition objects back to tape or optical volumes once they have moved to DB2 DASD.

- CLEAROLDLOC(mode)
  - OPT would cause the previous volser and sector location values in the object directory to be cleared when an object that resides on optical is transitioned to DB2 during OSMC processing.
  - TAPE would cause the previous volser and blockid values in the object directory to be cleared when an object that resides on tape is transitioned to DB2 during OSMC processing.
  - BOTH would cause the previous volser and sector location or blockid values in the object directory to be cleared when an object that resides on either optical or tape is transitioned to DB2 during OSMC processing.
  - NONE would cause the previous volser and sector location or blockid values in the object directory to be left unchanged when an object that resides on optical or tape is transitioned to DB2 during OSMC processing.

Old location values are only cleared when an object transitions from optical or tape to DB2 DASD during an OSMC cycle; old location values will not be cleared when an object is recalled to DB2 DASD even if CLEAROLDLOC is active.

CLEAROLDLOC can also be set dynamically by using the modify **OAM UPDATE** command. For example:

```
F OAM,UPDATE,SETOSMC,ALL,CLEAROLD,BOTH
```

New exit routine return to MVS scratch

A new dynamic exit routine CBRUXTVS_EXIT can be used to notify the tape management system that all knowledge of a given tape volume has been removed from OAM's tape volume inventory. This is a *notification only* exit in that OAM does not change it's tape volume expiration processing regardless of the return code supplied by the user exit.

This notification exit is patterned after the DFSMSrmm EDGTVEXT exit and the DFSMShsm ARCTVEXT exit.

The exit is invoked after OAM issues the CBR2165I message indicating that OAM has removed the tape volume from the OAM inventory and returned it back to the MVS scratch pool.

10.2 DFSMSdss enhancements

DFSMSdss (Data Set Services) provides data movement, copy, backup, and space management functions for data sets. In z/OS V1R7 the following enhancement has been implemented to DFSMSdss:

- Support for large data sets

10.2.1 Support for large format data sets

DFSMSdss is enhanced to support the new large format sequential data sets that were discussed previously.

With z/OS V1R7, DFSMSdss supports the basic functions of the following components for the new large sequential data sets:

- COPY
- DUMP/RESTORE

- RELEASE
- PRINT
- DEFRAG

DFSMSdss also supports the use of large sequential data sets for the following:

- Output from DUMP
- Input to RESTORE
- Input or output for COPYDUMP

10.2.2 Using ADRDSSU with large data sets

Figure 10-41 shows a JCL example of how to use ADRDSSU with a large format output data set.

```
//A    EXEC PGM=ADRDSSU
//DASD      DD UNIT=SYSDA,VOL=SER=Z17RB1,DISP=SHR
//LARGE     DD DSN=WELCH.LARGE.DUMP,DISP=(NEW,CATLG),
//          STORCLAS=STANDARD,DSNTYPE=LARGE,
//          SPACE=(CYL,(9000,1000),RLSE)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  DUMP INDD(DASD) OUTDD(LARGE) ALLDATA(*) ALLEXCP
/*
```

Figure 10-41 Example usage of ADRDSSU with a large format output data set

Figure 10-42 displays, from the data list panel of Option **3.4**, an output for a dump data set indicating the size of the large data set.

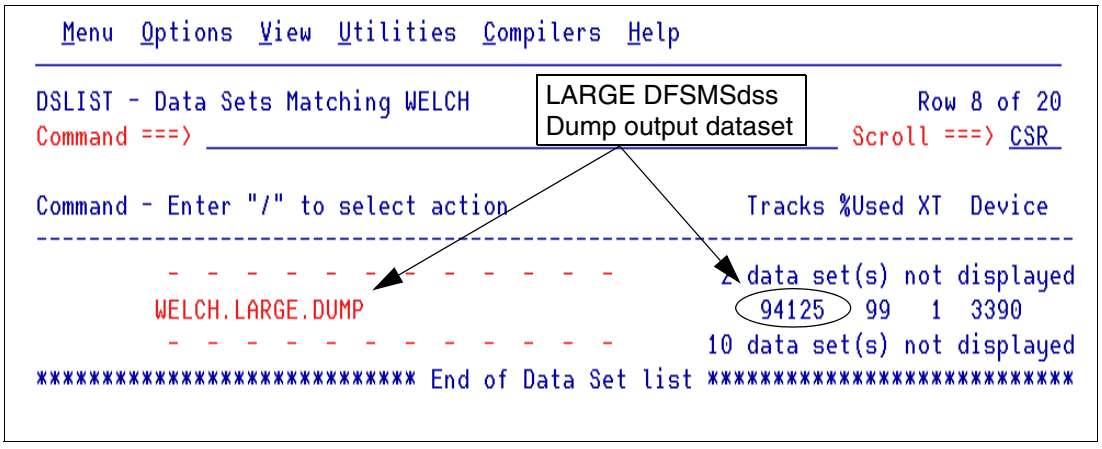


Figure 10-42 A large output data set from DFSMSdss

The example output in Figure 10-43 on page 209 shows a scenario where a large data set is restored from backup with a preallocated basic (or non-large) data set. DFSMSdss will delete and reallocate a large data set of sufficient size.

```
DATA SET WELCH.TEST.DATASET PREALLOCATED, ON VOLUME(S): SBOX1D
DATA SET WELCH.TEST.DATASET CONSISTS OF 00015000 TARGET TRACKS AND 00075000 SOURCE
TRACKS
DATA SET WELCH.TEST.DATASET HAS BEEN DELETED
DATA SET WELCH.TEST.DATASET WILL BE SCRATCHED FROM SBOX1D BECAUSE OF UNMATCHED SIZE. IT
WILL BE REALLOCATED
DATA SET WELCH.TEST.DATASET HAS BEEN ALLOCATED USING STORCLAS STANDARD, NO DATACLAS, AND
MGMTCLAS MCDB22
DATA SET WELCH.TEST.DATASET WAS RESTORED
THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
 WELCH.TEST.DATASET
```

Figure 10-43 Example RESTORE of a large data set

The DFSMSdss logical and physical data set RELEASE function is enhanced to recognize a large sequential data set and correctly take into account the extra byte of used track information in DS1TTTHI. After RELEASE processing, a large sequential data set will still be a large sequential data set — even if its used space is within the 65535 track limit.

DFSMSdss DEFRAG processing already supports large sequential data sets because it deals only with extents as described in the VTOC format-1 and format-3 DSCBs, and not the last used track of any type of data set.

Migration/coexistence considerations

PTFs will be available for earlier systems to prevent DFSMSdss from attempting to process large format data sets on systems which are not capable of processing them correctly. Message ADR778E with reason code 17 will be issued when a large sequential data set is encountered on these systems.

The error message in Figure 10-44 shows what happens on a prior level of DFSMSdss when you try to process a large format data set.

```
ADR778E (001)-DTDSC(01), DATA SET WELCH.LARGE.DATASET WAS NOT SELECTED
BECAUSE
THE DATA SET TYPE IS NOT SUPPORTED IN THIS RELEASE,17
```

Figure 10-44 unable to process large data sets on pre z/OS V1R7 systems

Table 10-4 Coexistence PTFs for DFSMSdss for large format data sets

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08286	UA16796	UA16797	UA16798

10.3 DFSMShsm enhancements

DFSMShsm (Hierarchical Storage Manager) provides backup, recovery, migration, and space management functions.

z/OS V1R7 DFSMShsm has been enhanced to provide:

- Support for large format data sets
- Fast subsequent migration improvements
- Extended TTOC (tape table of contents)

- Removal of ABARS requirement for INCLUDE statement
- Cancellation of individual HSM tasks
- Using wildcards with HMIGRATE
- Saving LRECL of migrated data sets in the MCD
- New recycle processing option for connected sets

10.3.1 Support for large format data sets

DFSMShsm is enhanced to support the new large sequential data sets that are discussed in “Large format data sets” on page 181.

DFSMShsm supports the processing of large format data sets in typical operations involving:

- Migration and recall
- Backup and recovery
- ABACKUP and ARECOVER

DFSMShsm supports large format data sets just as it does traditional sequential data sets.

DFSMShsm also supports the use of a large format data set for the HSM journal data set. A larger journal data set can allow more DFSMShsm activity to take place between journal backups and helps to minimize occurrences of journal full conditions.

For large format data sets, a coexistence PTF is required so that lower level releases of DFSMShsm will fail migration/recall, backup/recover, or ABACKUP/ARECOVER of large format data sets.

Table 10-5 Coexistence PTFs for HSM support of large format data sets

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08865	UA16954	UA16955	UA16956

10.3.2 Fast subsequent migration improvements

FSM (Fast Subsequent Migration) is the process that allows data sets that have been recalled from ML2 tape to be subsequently reconnected to the ML2 tape when the data set again becomes eligible for migration, and if the data set has not been modified since being recalled.

However, systems which have implemented TMM (Tape Mount Management) have found that they have a large number of data sets on ML2 tapes that are not eligible for FSM even though they were not modified after being recalled. The reason for this is that most TMM candidate data sets have a data management class that specifies that the data sets do not require backup, and since the data sets have not been backed up, the change bit in the DSCB does not get turned off by the backup process. FSM currently has checks that prevent data sets from being reconnected if the change bit is still on. If on, DFSMShsm considers that the data set had been modified after being recalled and should not be reconnected to the original ML2 tape.

To increase the number of migrated data sets eligible for fast subsequent migration, this function has been changed to use new indicators for reconnection eligibility. This enhanced implementation can increase your use of fast subsequent migration in the following ways:

1. It allows you to use fast subsequent migration even if you use a product other than DFSMShsm to back up your data sets.

2. It allows DFSMShsm to reconnect data sets originally migrated to ML2 tape without a valid backup copy (such as TMM data). In previous releases, these data sets were not eligible for reconnection.

Migration/coexistence considerations

Installations using Tape Mount Management should see an increased number of reconnections for recalled TMM data sets.

Coexistence PTFs are provided for lower level systems. This will allow data sets recalled on a pre z/OS V1R7 system to be eligible for reconnection using the new criteria if the migration is performed on a z/OS V1R7 system.

Data sets recalled prior to installation of the coexistence PTFs will be eligible for reconnection under the original criteria.

Table 10-6 Coexistence PTFs for HSM Fast Subsequent Migration support

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08848	UA16944	UA16945	UA16946
OA09096	UA16860	UA16861	UA16862
OA08858	UA16849	UA16850	UA16851

10.3.3 Extended TTOC (tape table of contents)

Currently DFSMShsm can write a maximum of 330,000 data set entries per tape volume. This limits the number of data sets that an installation can put on a tape and can prevent full utilization of high capacity tapes.

To enable better use of new high capacity tape volumes, z/OS V1R7 DFSMShsm can write more than one million data sets to a migration tape or backup tape.

To enable support for the extended TTOC requires that the following conditions exist:

- OCDS (offline control data set) has a maximum record size of 6144.
- **SETSYS EXTENDEDTTOC(Y)** is issued.

This allows DFSMShsm to write 106 data set entries per TTOC record for a maximum of 1,060,000 data sets per tape volume. Note that this only affects newly created TTOC entries.

Migration and coexistence considerations

Once all sharing systems are at z/OS V1R7, a sample migration plan is:

- Shut down all of the DFSMShsm subsystems in the HSMplex that share the OCDS.
- Back up the OCDS.
- ALTER its RECORDSIZE to 6144.
- Start the DFSMShsm subsystems.
- Issue SETSYS EXTENDEDTTOC(Y).

For extended TTOCs in a sysplex, a coexistence PTF is required on lower level systems. During DFSMShsm startup, if a lower level DFSMShsm finds that the OCDS record size is 6144 bytes, then tape operations on that DFSMShsm will be inhibited and an ARC0130I message will be issued with new return code 19.

It is suggested that you do not use extended TTOCs in a sysplex until all systems are upgraded to z/OS V1R7 to avoid inhibiting tape operations on any systems that are not at the z/OS V1R7 level.

Table 10-7 Coexistence PTFs for Extended TTOC

APAR	HDZ11G0	HDZ11H0	HDZ11J0
OA08863	UA16947	UA16948	UA16949

10.3.4 Removal of ABARS requirement for INCLUDE statement

ABARS currently requires that there is at least one data set in the INCLUDE list when backing up an aggregate group. Many users would like aggregates that contain only ALLOCATE and ACCOMPANY lists of data sets; these can be used to define a catalog and populate it with disk and tape data sets as well as define necessary GDG base definitions.

With this change, ABARS processing no longer requires you to specify an INCLUDE statement in the data set selection list; you need only specify allocation information (an ALLOCATE statement) or tape catalog information (an ACCOMPANY statement).

10.3.5 Cancellation of individual HSM tasks

This enhancement allows you to cancel active DFSMShsm data movement tasks, including ABARS tasks that process in their own address space. Previously, when you needed to cancel an active data movement task, you had to bring down the entire DFSMShsm address space. Now an active data movement task can be cancelled without impacting other ongoing DFSMShsm activity.

This enhancement will promote better DFSMShsm availability and higher end user satisfaction.

Support is planned to be made available via PTFs for z/OS V1.4 and higher in 2005.

10.3.6 Using wildcards with HMIGRATE

Current processing of an **HMIGRATE** command when using a wildcard data set filter (*) will generate error messages for any data set which is already migrated. With z/OS V1R7 DFSMShsm will bypass any data set that is already migrated when you specify the **HMIGRATE** command with a wildcard filter (*).

z/OS V1R7 processing of HMIGRATE using wildcards:

- HMIGRATE without the ML2 parameter - Data sets that reside on Level 0 DASD will be migrated and any data sets that are already migrated will be skipped.
- HMIGRATE with the ML2 parameter - Data sets that reside on Level 0 DASD or ML1 DASD will be migrated and any data sets residing on ML2 will be skipped.

This change will eliminate the error messages that were previously issued.

10.3.7 Saving LRECL of migrated data sets in the MCD

Currently the record length of a migrated data set is not available for reporting, which means that the data set must be recalled in order to have access to this information.

With z/OS V1R7 DFSMShsm the LRECL and a *data set empty indicator* are added to the MCD record for migrated data sets, which means that this information is now available for

query and reporting purposes. This information will be added to the LIST command output and to the DCOLLECT MIGRATEDDATA output.

To use the LIST DSNAME command to query the LRECL of a migrated data set, enter the command as follows:

```
LIST DSNAME(datasetname) MCDS
```

To display only those migrated data sets that were empty at the time of migration, use the new SELECT(EMPTY) option:

```
LIST DSNAME(datasetname) MCDS SELECT(EMPTY)
```

This change will make it easier for those installations that need to report on data set characteristics.

Migration and coexistence considerations

If a data set was migrated on a z/OS V1R7 HSM system, a FIXCDS DISPLAY command issued from a lower level system will be able to display this new information.

If a data set was migrated on a lower level system, the LRECL and data set empty information is not available in the MCDS and is not available for query.

10.3.8 New recycle processing option for connected sets

A *connected set* is a chain of volumes that become linked together when a data set starts on one volume and spans to another or multiple other volumes. The RECYCLE function uses the PERCENTVALID parameter on the RECYCLE command to determine if a volume or connected set is eligible for DFSMShsm recycle processing.

For pre z/OS V1R7 systems the first volume in the connected set must meet the PERCENTVALID threshold (as well as the connected set as a whole).

With z/OS V1R7, DFSMShsm allows you to specify with a new keyword, CHECKFIRST, that the entire connected set's average percentage of valid data is to be used to determine whether to recycle a connected set.

```
RECYCLE PERCENTVALID(percent) CHECKFIRST(Y | N).
```

The default value of CHECKFIRST(Y) is the status quo (that is, recycle processing works as before).

CHECKFIRST(N) will direct recycle processing to use the entire connected set's percentage of valid data to determine its eligibility for reconnection. Note that specifying CHECKFIRST(N) may increase the overall RECYCLE processing time since DFSMShsm will now be looking at the percentage valid criteria for the entire connected set rather than just the first volume in a connected set. If the first volume in a connected set does not meet the percentage valid criteria, but the entire connected set does, then recycling will now take place.

Installations will now be able to RECYCLE connected sets that previously were not being recycled. This will have the effect of potentially recycling a larger number of the volumes in a connected set sooner, thus releasing more volumes to be used for subsequent processing or to return to the tape scratch pool.

10.4 DFSMSrmm enhancements

DFSMSrmm provides management functions for removable media such as tape cartridges and reels.

z/OS V1R7 DFSMShsm has been enhanced to provide:

- Support for large format data sets
- Ability to issue DFSMSrmm TSO commands from the console
- Enterprise enablement
- DFSMSrmm CIM provider
- Improved security control over DFSMSrmm functions

10.4.1 Support for large format data sets

When all systems are running z/OS V1R7 or higher levels, you can implement a large format journal data set by deleting and reallocating the journal data set specifying DSNTYPE=LARGE in the JCL. The sample jobs EDGJNLAL and EDGPBKUP now allocate the journal and the journal backup with DSNTYPE=LARGE.

Note that you cannot write to a large format journal or journal backup from a lower level of DFSMSrmm. The system will fail the open with a 213 abend. For more information about large format data sets refer to “Large format data sets” on page 181.

DFSMSrmm will also allocate large format temporary work data sets created dynamically by:

- EDGUTIL during VERIFY and MEND processing
- DFRMM started task during inventory management when running the EDGHSKP utility

10.4.2 Issue DFSMSrmm TSO commands from the console

The MVS modify command can now be used to issue an RMM TSO command on the console, as follows:

```
F DFRMM,CMD=command
```

The command output is returned to the console and system log. For example:

```
F DFRMM,CMD=LISTCONTROL ALL
```

10.4.3 Enterprise enablement

In the past, you could use the high-level language application programming interface from C/C++ and Java (using the JNI) code running on the same z/OS system as the DFSMSrmm subsystem. A series of calls to the application programming interface were necessary to run the subcommand and to receive the output.

With z/OS V1R7 and the DFSMSrmm enterprise enablement enhancement, you can use the high-level language application programming interface as a Web service. This enables the API to be used from any system or platform that can run Java, C++, or any language that supports the Web services standards. Now, it is as if the high-level language application programming interface is available as a locally callable program. A single call to the application programming interface to run a subcommand and receive all the output is all that is needed.

A sample Java Web service application, rmmSampleWSClient.java, is located in /usr/lpp/dfsms/rmm/. The sample code shows how the application programming interface can be used via a Web service.

For further information, see:

- *DFSMSrmm Application Programming Interface,* SG26-7403
- *DFSMSrmm Implementation and Customization Guide,* SC26-7405

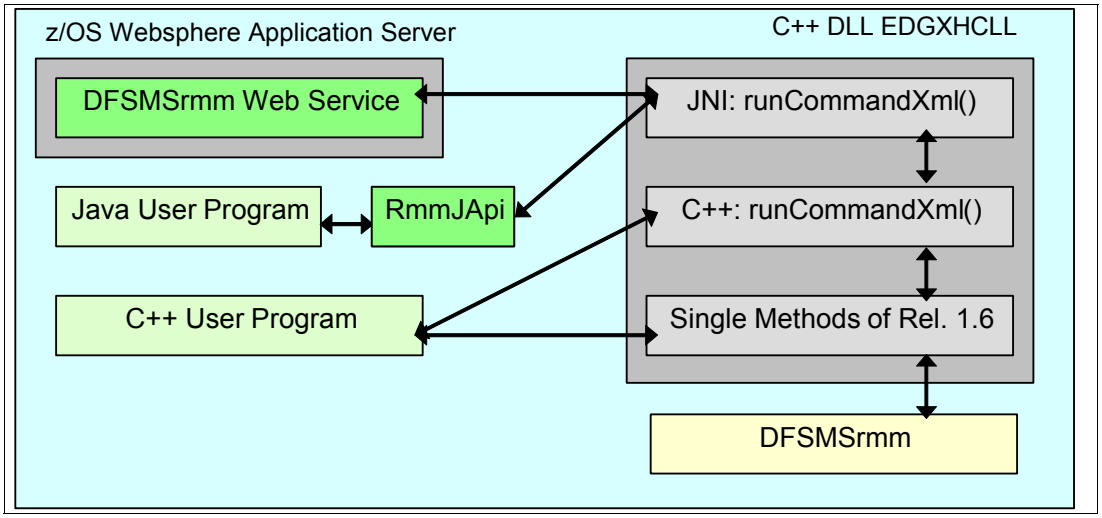


Figure 10-45 DFSMSrmm Web service overview

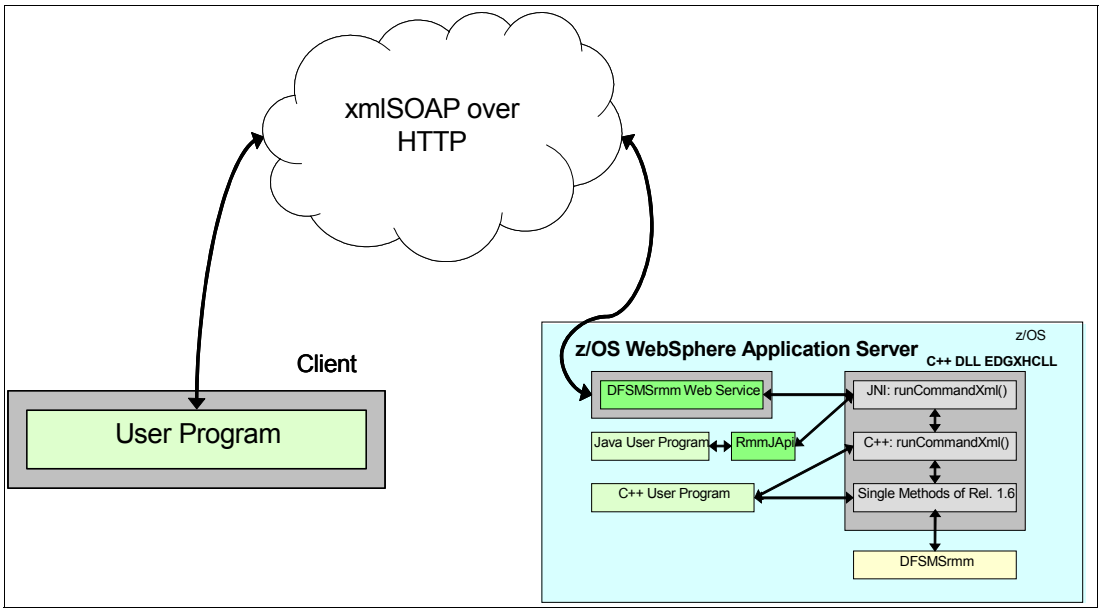


Figure 10-46 DFSMSrmm Web service overview (continued)

10.4.4 DFSMSrmm CIM provider

CIM (Common Information Model) is a set of standards that define a conceptual model representing IT resources, as shown in the overview in Figure 10-47 on page 216. The model presents this in a platform-independent, technology-neutral way. CIM provides the ability to manage a heterogeneous environment from a single management application.

The Distributed Management Task Force (DMTF) is the standards organization that is driving CIM. The mission of the DMTF is to lead the development of management standards for distributed desktop, network, enterprise, and internet environments. One of the goals of the DMTF is to “Promote interoperability among management solution providers.”

For further information about DMTF and CIM see:

```
http://www.dmtf.org/home
```

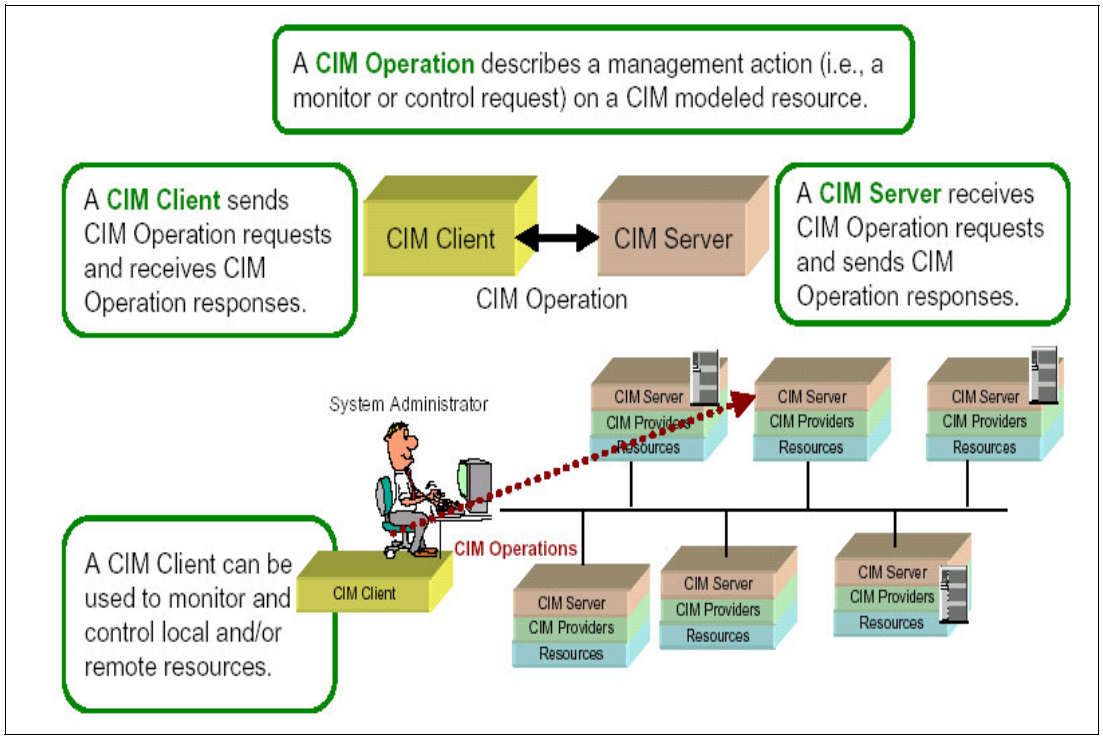


Figure 10-47 Overview of the CIM concept

The z/OS V1R7 DFSMSrmm includes a CIM provider that is a plug-in for the SNIA's Open Source CIM Object Manager (CIMOM). This enables RMM resources to be manageable via CIM.

The DFSMSrmm CIM provider application programming interface is a Java class that implements the CIM-specified methods required of providers.

10.4.5 Improved security control over DFSMSrmm functions

It is no longer necessary to have CONTROL access to STGADMIN.EDG.MASTER to run daily DFSMSrmm tasks. New resource profiles provide better control access to DFSMSrmm resources. You can use some or all of the new profiles to allow a subset of functions to be authorized.

11

Communications Server (CS) for z/OS V1R7

z/OS Communications Server is a network communication access method. It provides Systems Network Architecture (SNA) and Transmission Control Protocol/Internet Protocol (TCP/IP) networking protocols for z/OS.

The TCP/IP suite, also called the IP stack, includes associated applications, transport, network protocol layers, connectivity, and gateway functions that allow the transmission of data using IP networks.

The SNA protocols are provided by VTAM and include Subarea, Advanced Peer-to-Peer Networking® (APPN), High Performance Routing (HPR) and Enterprise Extended (EE) protocols. These protocols allow the sending of data between SNA network users.

The following enhancements in Communications Server were introduced for z/OS V1R7:

- Hardware exploitation OSA-Express2
- TCP/IP sysplex enhancements
- TCP/IP IPv6 enhancements
- TCP/IP FTP enhancements
- TCP/IP security enhancements
- TCP/IP CICS sockets
- TCP/IP OROUTED removed
- TCP/IP CTRACE optimization
- SNA enhancements

11.1 Communications Server z/OS V1R7 overview

Communications Server is a z/OS base element that supports secure TCP/IP, SNA, and z/OS UNIX networking on enterprise systems, connecting different types of communication subsystems and applications to each other and supporting usage of various communication devices. The major components of Communications Server are IP Services and SNA Services.

The progression of improvement from OS/390 V2R10 until z/OS V1R7 is shown in Figures 11-1 and 11-2.

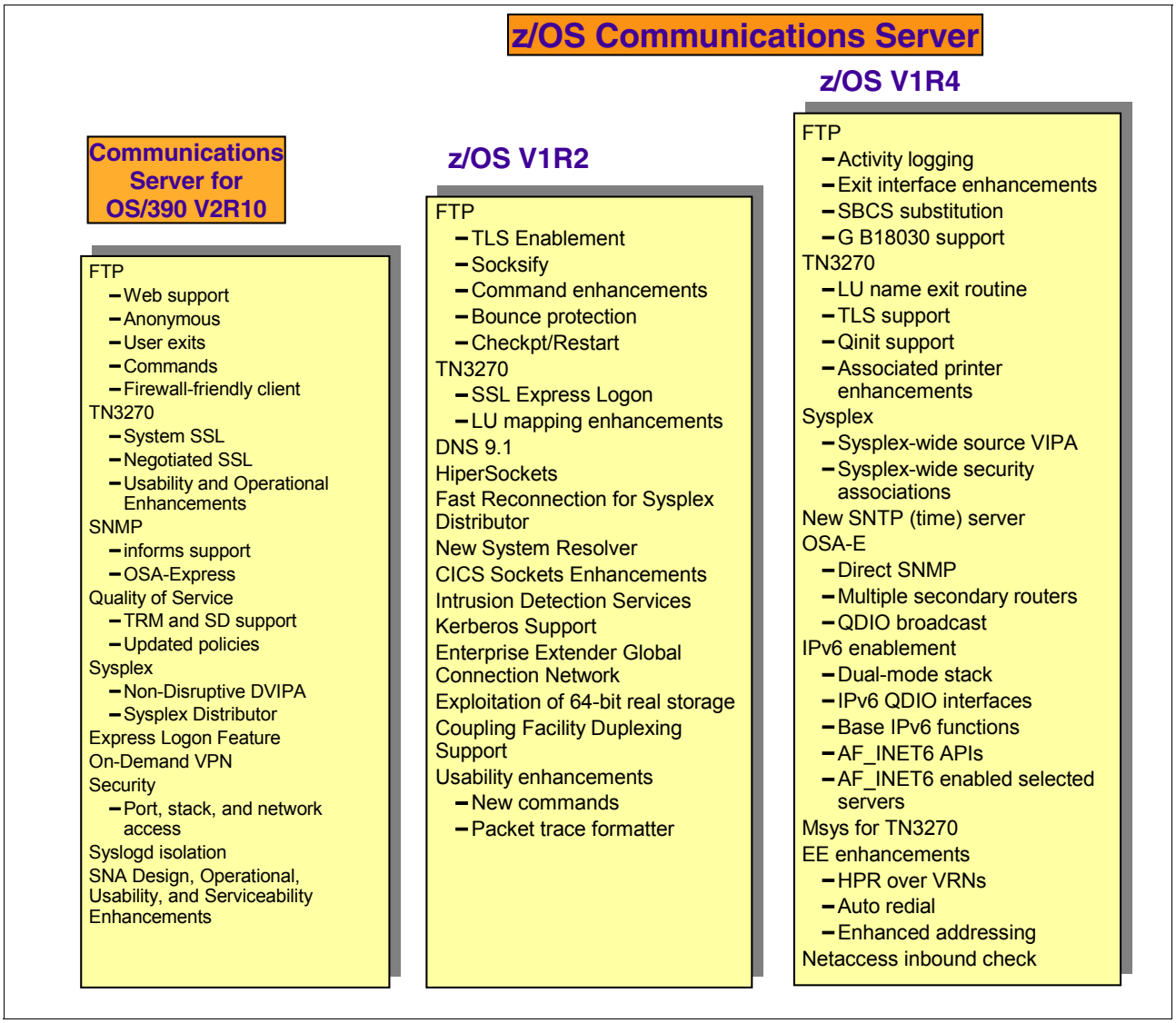


Figure 11-1 Communications Server since OS/390 V2R10 until Z/OS V1R4

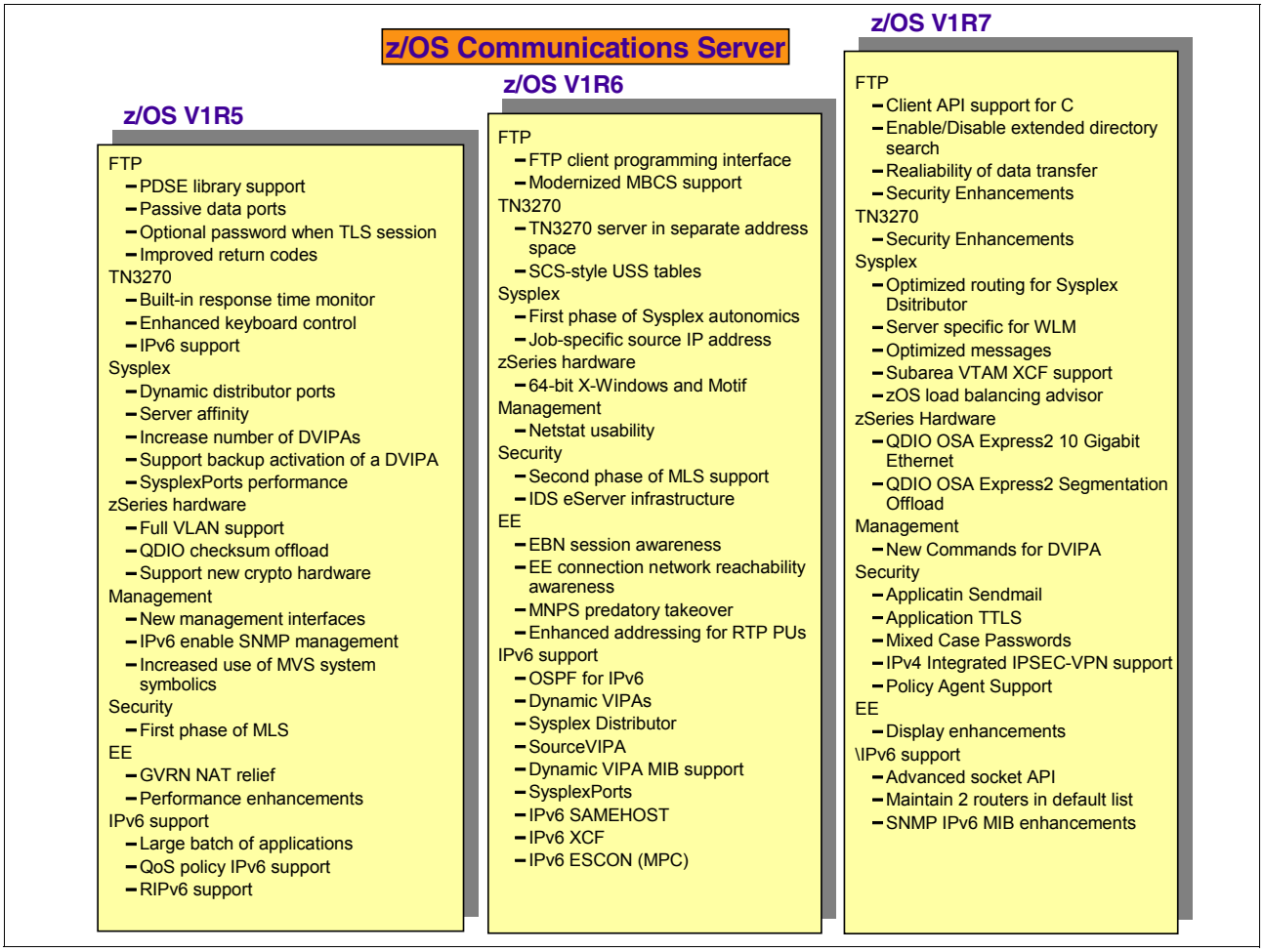


Figure 11-2 Communications Server since Z/OS V1R5 until V1R7

11.2 zSeries hardware exploitation - OSA-Express2

The z890 and z990 processors support the new OSA-Express2, the third generation of zSeries Ethernet technology. The following lists show the new OSA-Express2 interface features of both models, the 10 Gigabit ethernet (Gbe) and the Gbe model:

- Newest member 10 Gigabit Ethernet LR (long reach):
  - One port per feature
  - 9 micron single mode fiber, SC Duplex connector
- New Gigabit Ethernet features:
  - Gbe LX (Long wavelength)

    9 micron single mode fiber, LC Duplex connector
  - Gbe SX (Short wavelength)

    50 or 62.5 micron multimode fiber, LC Duplex connector
  - Each port is designed to achieve line speed
  - 1 Gbps in each direction

- Support offered by both 10 GbE and GbE is as follows:
  - Queued Direct Input/Output (QDIO) for TCP/IP traffic only
  - Use TN3270 or Enterprise Extender for SNA traffic
  - Layer 2 support for flexible and efficient data transfer
  - 640 TCP/IP stacks for improved virtualization
  - Large send for CPU efficiency
  - Concurrent License Internal Code (LIC) update to minimize network traffic disruption

Attention: The new support is QDIO mode only (CHPID type OSD).

The next two sections discuss the z/OS Communications Server V1R7 improvements for the new OSA Express2 models.

11.2.1 QDIO OSA-Express2 10 Gigabit Ethernet support

z/OS Communications Server V1R7 adds support for OSA-Express2 10 Gbe LR. This requires a z990 or z890 processor and is configured and managed exactly like Gbe. The support is transparent except that the following reflects the actual speed:

- The speed field on the **`Netstat DEVLINKS/-d`** command report output, as shown in Figure 11-3 and Figure 11-4.

```
DEVNAME: OSA2CE0           DEVTYPE: MPCIPA
  DEVSTATUS: READY         CFGROUTER: NON  ACTROUTER: NON
  LNKNAME: OSA2CEOLNK        LNKTYPE: IPAQENET    LNKSTATUS: READY
    NETNUM: N/A  QUESIZE: N/A  SPEED: 0000001000
    IPBROADCASTCAPABILITY: NO
```

Figure 11-3 Gigabit Ethernet Netstat DEVLINKS

```
DEVNAME: OSA2CE0           DEVTYPE: MPCIPA
  DEVSTATUS: READY         CFGROUTER: NON  ACTROUTER: NON
  LNKNAME: OSA2CEOLNK        LNKTYPE: IPAQENET    LNKSTATUS: READY
    NETNUM: N/A  QUESIZE: N/A  SPEED: 0000010000
    IPBROADCASTCAPABILITY: NO
```

Figure 11-4 10 Gigabit Ethernet Netstat DEVLINKS

- The SNMP MIB object ifHighSpeed (from the IF-MIB)
  - The ifName.23 is a 10 gigabit ethernet link, and the ifHighSpeed.23 is 10000. The unit for this display is in 1 million bits per second, so 10000 is 10,000 million bits per second or 10 gigabits per second, as shown in Figure 11-5 on page 221.

ifName.1=LOOPBACK	ifHighSpeed.1=0
ifName.2=LOOPBACK	ifHighSpeed.2=0
ifName.3=LOOPBACK6	ifHighSpeed.3=0
ifName.4=RS6K	ifHighSpeed.4=0
ifName.5=LRS6K	ifHighSpeed.5=100
ifName.6=MIT2CH22	ifHighSpeed.6=0
ifName.7=LMIT2CH22	ifHighSpeed.7=4
ifName.8=FIT2BO34	ifHighSpeed.8=0
ifName.9=LFIT2BO34	ifHighSpeed.9=4
ifName.10=MIT2BO23	ifHighSpeed.10=0
ifName.11=LMIT2BO23	ifHighSpeed.11=4
ifName.12=ITVIPA1	ifHighSpeed.12=0
ifName.13=LITVIPA1	ifHighSpeed.13=0
ifName.14=CISCO228	ifHighSpeed.14=0
ifName.15=LCISCO228	ifHighSpeed.15=100
ifName.16=CISCO226	ifHighSpeed.16=0
ifName.17=LCISCO226	ifHighSpeed.17=100
ifName.18=OFETHC	ifHighSpeed.18=0
ifName.19=LOFETHC	ifHighSpeed.19=100
ifName.20=OFETHB	ifHighSpeed.20=0
ifName.21=LOFETHB	ifHighSpeed.21=100
ifName.22=OTGETH2	ifHighSpeed.22=0
ifName.23=LOTGETH2	ifHighSpeed.23=10000

Figure 11-5 SNMP MIB ifHighSpeed object sample

PTF support for earlier releases

The following list shows the PTFs available for 10 Gbe support for current releases:

- VTAM (APAR OA09759)
  - V1R4 - UA15927
  - V1R5 - UA 15928
  - V1R6 - UA 15929
- TCP (APAR PQ96769)
  - V1R4 - UQ 95921
  - V1R5 - UQ 95922
  - V1R6 - UQ95923

11.2.2 QDIO OSA-Express2 segmentation offload

Communications Server V1R7 supports the new OSA-Express2 segmentation offload feature also referred to as *Large Send*. This new function decreases host CPU utilitzation and increases data transfer efficiency for IPv4 packets. This support is automatically enabled without requiring configuration changes when available in the adapter and it is displayable via the **NETSTAT DEVLINKS** command. This is similar to the existing checksum offload function and checksum is offloaded whenever segmentation is offloaded.

Restriction: This support has the following restrictions:

- IPv4 only
- TCP transport only
- Outbound packets only
- Packets written to the LAN only (not to another stack sharing the OSA)
- Packets larger than MSS only
- For multipath, only when all devices in the multipath group support segoffload
- No IPSEC packets

Figure 11-6 illustrates in which layer the IP stack makes the segmentation.

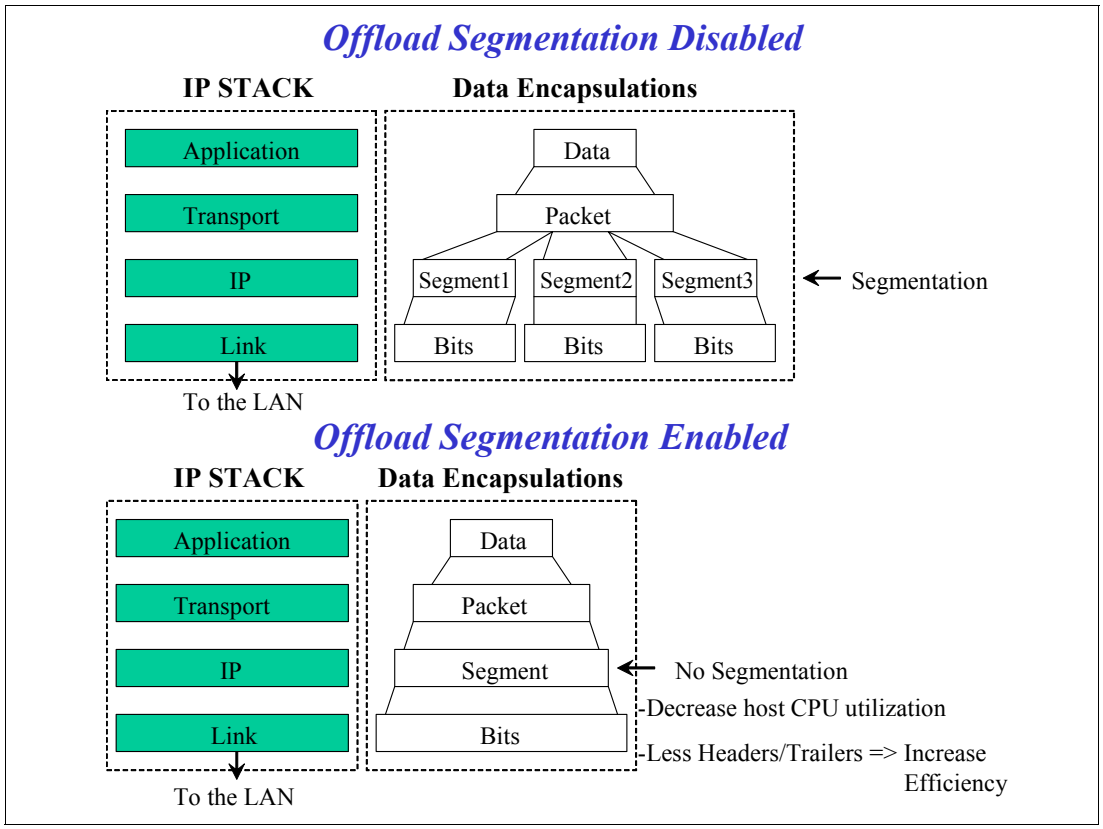


Figure 11-6 Comparative enabled/disabled offload segmentation

As shown in Figure 11-7, we only can have offload segmentation with packets to the LAN and the target interface does not need the offload segmentation feature.

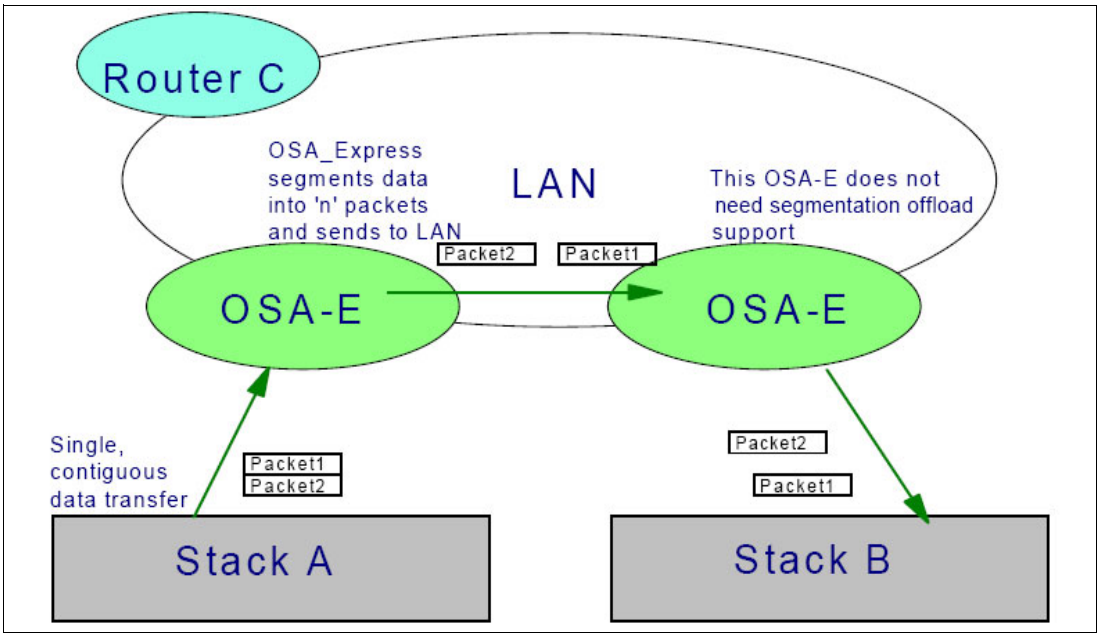


Figure 11-7 Offload segmentations only packets to the LAN

In the netstat links display shown in Figure 11-8, the bit `SegmentationOffload` indicates if the segmentation offload is enabled or disabled.

```
DevName: OGETHD            DevType: MPCIPA
  DevStatus: Ready
  LnkName: LOGETHD           LnkType: IPAQENET LnkStatus: Ready
    NetNum: n/a QueSize: n/a   Speed: 0000001000
   IpBroadcastCapability: No
   CfgRouter: Pri                   ActRouter: Pri
   ArpOffload: Yes                  ArpOffloadInfo: Yes
   ActMtu: 8992
   VLANid: 3                        VLANpriority: Enabled
   ReadStorage: GLOBAL (4096K)      InbPerf: Balanced
   ChecksumOffload: Yes             SegmentationOffload: Yes
   SecClass: 255
```

Figure 11-8 Offload segmentation netstat devlinks

SNMP MIB, new bit

The MIB ibmMvsIfFlag object now contains the tcpSegOffloadEnabled(6) bit, as shown in Figure 11-9.

```
# snmp -v walk ibmMvsIfFlag
ibmMvsIfFlag.2 = '20'h
ibmMvsIfFlag.3 = '20'h
ibmMvsIfFlag.5 = 'a4'h
ibmMvsIfFlag.7 = 'aa'h <-- x'02' - on - Segmentation Offload Enabled
```

Figure 11-9 Segmentation offload, ibmMvsIfFlag MIB object

The SNMP ibmMvsIfFlag MIB object is part of the ibmMvsIfTable, which is defined in the IBM MVS TCP/IP enterprise-specific MIB module. This MIB module is installed in the /usr/lpp/tcpip/samples HFS directory as file mvstcpip.mi2.

11.2.3 VTAM TNSTAT diagnosis

Figure 11-10 shows 2 VTAM TNSTAT responses. The TNSTATs were gathered from a single data transfer operation (180K transmission). The top half of the example shows TNSTATs from the sending host and the bottom section shows TNSTATs from the receiving host. The packet count is the same but the byte count is different. The reason the byte count is different is because the OSA-Express2 generated headers for each segment.

- VTAM TNSTATs counts OSA generated segments:

```
IST924I -------------------------------------
IST1233I DEV = 0E2A DIR = WR/
IST1755I SBALMAX = 2 SBALAVG = 1
IST1756I QDPTHMAX = 0 QDPTHAVG = 0
IST1723I SIGACNTO = 0 SIGACNT = 6
IST1721I SBALCNTO = 0 SBALCNT = 6
IST1722I PACKCNTO = 0 PACKCNT = 21
IST1236I BYTECNTO = 0 BYTECNT = 184984
IST1810I PKTIQDO = 0 PKTIQD = 0
IST1811I BYTIQDO = 0 BYTIQD = 0

IST924I -------------------------------------
IST1233I DEV = 0E2E DIR = READ
IST1719I PCIREALO = 0 PCIREAL = 6
IST1720I PCIVIRTO = 0 PCIVIRT = 0
IST1750I PCITHRSO = 0 PCITHRSH = 0
IST1751I PCIUNPRO = 0 PCIUNPRD = 0
IST1752I RPROCDEO = 0 RPROCDEF = 0
IST1753I RREPLDEO = 0 RREPLDEF = 0
IST1754I NOREADSO = 0 NOREADS = 0
IST1721I SBALCNTO = 0 SBALCNT = 6
IST1722I PACKCNTO = 0 PACKCNT = 21
IST1236I BYTECNTO = 0 BYTECNT = 186084
IST1810I PKTIQDO = 0 PKTIQD = 0
IST1811I BYTIQDO = 0 BYTIQD = 0
```

Figure 11-10 Offload segmentation, VTAM TNSTATs counts

New commands

As part of the segmentation offload solution, support for four new undocumented commands have been added. One command forces disablement of checksum offload and another, segmentation offload. One command enables checksum offload, and another, segmentation offload. These commands do not have an effect on active interfaces. Active interfaces must be restarted in order for the commands to take effect, as follows:

- To force disable offload, use:
  - **MODIFY tcpprocname,nochkoffload** (can be abbreviated **F tcpprocname,nochkoff**)
  - **MODIFY tcpprocname,nosegoffload** (can be abbreviated **F tcpprocname,nosegoff**)
- To re-enable offload, use:
  - **MODIFY tcpprocname,chkoffload** (can be abbreviated **F tcpprocname,chkoff**)
  - **MODIFY tcpprocname,segoffload** (can be abbreviated **F tcpprocname,segoff**)

11.2.4 Migration concerns

When migrating to CS V1R7, consider the following PTFs and buffer sizes:

- PTFs are supplied for QDIO OSA-Express2 segmentation offload.
  - TCP/IP APAR PK02490: TCP Segmentation Offload transfers the overhead of segmenting outbound data into individual TCP packets to the QDIO attached OSA-Express2 device. Offloading segmentation of streaming type workloads reduces CPU utilization and increases throughput. The PTF numbers are UK04060 and UK04061.
  - SNA APAR OA11148: This APAR provides the VTAM code needed to support Segmentation Offload, a function on OSA-Express2 designed to improve performance of outbound IPv4 TCP traffic. The PTF number is UA18116.
  - Segmentation offload cannot be enabled unless this PTF is applied.
- The big send buffer (up to 56K) maximizes offloading. Configure the TCP send buffer size using the following existing mechanisms:
  - TCPSENDBfrsize on TCPCONFIG statement sets default for all applications.
  - SETSOCKOPT (SO_SNDBUF) by the application overrides default.
- Send buffer size is also limited by the receive buffer size at the other end of the connection.
  - TCPRCVBufrsize on the TCPCONFIG statement sets the default for all applications.
  - SETSOCKOPT (SO_RCVBUF) by the application overrides the default.

11.3 TCP/IP sysplex enhancements

z/OS V1R7 introduces improvements in the following areas:

- **Optimized routing for Sysplex Distributor**

  Prior to z/OS V1R7 Communications Server, Sysplex Distributor used dynamic XCF interfaces (IPCONFIG DYNAMICXCF, IPCONFIG6 DYNAMICXCF, or both) to distribute all incoming packets to target stacks. Using dynamic XCF interfaces had several advantages, such as simplified configuration. However, the dynamic XCF interfaces might have already been heavily used by other system components and therefore might not have performed as well as alternate interfaces, such as an OSA connection to a Gigabit Ethernet network. Therefore, z/OS V1R7 Communications Server introduces support to enable Sysplex Distributor to distribute incoming packets to target stacks using the best available IP route.

- **DVIPA management**

  The DVIPA management enhancements include new TCP/IP configuration options and operator commands to improve operational tasks for TCP/IP stacks in a sysplex, reducing the need to manage OBEYFILE profiles. New operator commands are also introduced for workload balancing. Specifically, the enhancements are as follows:

  - In the case where a TCP/IP stack has been removed from the sysplex, there are two new ways for the stack to rejoin the sysplex group and also restore its sysplex configuration:
    - Manually, by using a new operator command (**`VARY TCPIP,,SYSPLEX,JOINGROUP`**)
    - Automatically, by using a new configuration option (AUTOREJOIN) for the sysplex autonomics function

– Prior to z/OS V1R7, you could rejoin the sysplex group by issuing a **VARY TCPIP,,OBEYFILE** command; this function is replaced in z/OS V1R7 | by the **VARY TCPIP,,SYSPLEX,JOINGROUP** command.

– Two new operator commands are introduced to enable a TCP/IP stack to easily relinquish or reclaim ownership of a DVIPA:

  • The **VARY TCPIP,,SYSPLEX,QUIESCE,PORT** command enables you to stop distribution for a specific server application.

  • The **VARY TCPIP,,SYSPLEX,QUIESCE,TARGET** command enables you to stop distribution for all server applications.

  The existing TCP connections are not disrupted by either command.

Note: These commands can be useful in scenarios where a target application or system needs to be stopped (such as for applying maintenance). By issuing the commands before stopping the application or system, you can minimize the impact to end users by preventing new work while still allowing existing requests to be completed. These commands can also be useful in scenarios where you need to temporarily divert new connection requests away from a particular target application or target TCP/IP stack.

- **WLM and load balancing**

  Server-specific workload manager (WLM) recommendations for load balancing is a new sysplex distribution feature in z/OS V1R7 that enables connections to be distributed based on a target server's workload capacity instead of a target system's workload capacity or using round-robin distribution. Prior to z/OS V1R7, the Sysplex Distributor queried WLM, which returned a WLM system weight recommendation of available capacity for each target stack in the sysplex. Using this information, the Sysplex Distributor distributed new connections in a weighted round-robin fashion, with most of the new connections going to the targets on stacks with the most available capacity. This is known as the BASEWLM option.

11.3.1 Optimized routing for Sysplex Distributor

Dynamic XCF interfaces are used when IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF is coded. DVIPA traffic is forwarded over dynamic XCF interfaces when Sysplex Distributor is used to distribute DVIPA traffic to multiple targets or when another stack takes over a DVIPA. The takeover stack continues to forward traffic for existing connections to the original stack. This has several advantages, as follows:

- It leverages existing sysplex XCF communication links.
- Configuration is simplified since the communication paths between all systems are defined automatically.

Under the covers, in z/OS V1R6 dynamic XCF uses one of three transport technologies, which are shown in Figure 11-11. Which technology is used depends on the availability and location of a partner stack:

- Inside the same LPAR - IUTSAMEH (memory-link inside a Z/OS system)
- Inside the same zSeries CEC - HiperSockets™ (if enabled for that purpose via the IQDCHPID VTAM start option)
- Outside the CEC - XCF signalling

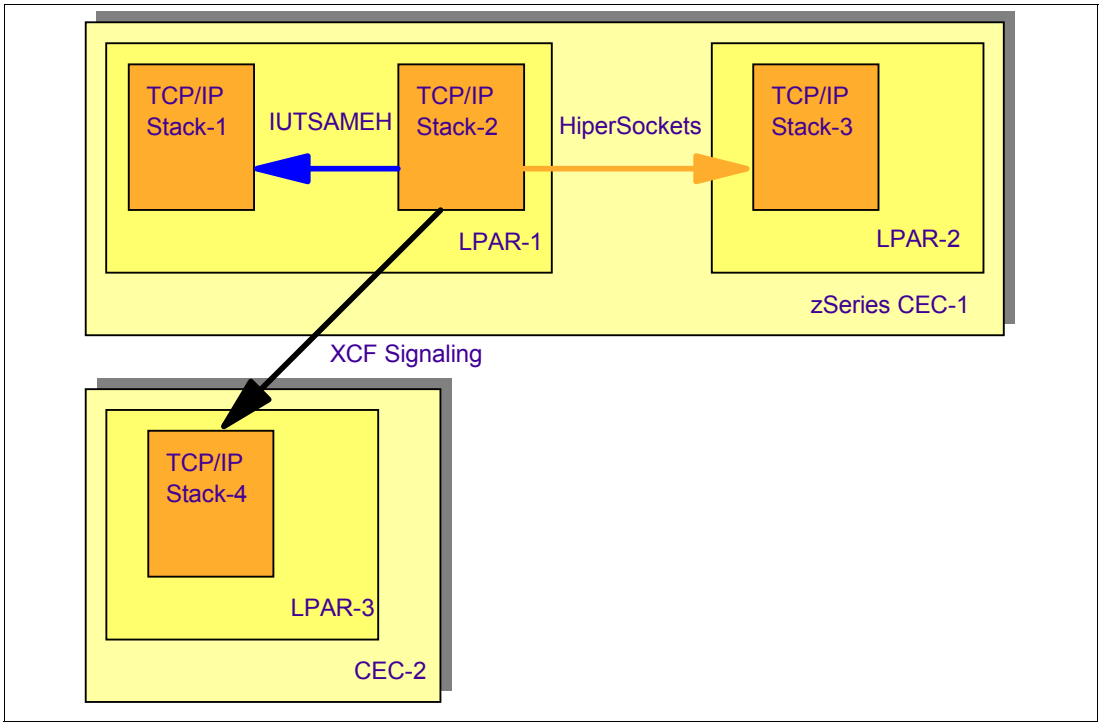


Figure 11-11 Dynamic XCF transport technologies

Performance considerations for forwarding DVIPA packets

IUTSAMEH or HiperSockets provide the best performance characteristic for forwarding DVIPA packets with the same MVS image or the same CEC. When the Sysplex Distributor and target stacks reside in a different CEC, other interfaces may be preferred over dynamic XCF interfaces. Sysplex XCF communication link are heavily used by other non-DVIPA communications. Low latency, wide bandwidth interfaces are available, for example Gbe segments using OSA Express. Using alternative interfaces can improve performance while reducing the utilitzation of sysplex XCF interfaces.

In the z/OS Communications Server V1R7, DVIPA IP forwarding functions have changed to use any IP network connectivity between TCP/IP stacks in a sysplex: shared Gbe, MPCPTP links, dynamic XCF, and any network.

Dynamic XCF with the target address for VIPADISTRIBUTE definitions must be defined in z/OS V1R7. Some workload will be routed via dynamic XCF, as follows:

- Sysplex-wide security association (IPSEC) packets
- Multi-level security (MLS) tagged packets
- Policy Agent QoS performance data collection

Note: To minimize XCF signalling, use HiperSockets for the same-CEC dynamic XCF.

TCP/IP sysplex improved distribution optimized routing

With z/OS V1R7, routing is optimized since you can use any available route (shown in Figure 11-12), and customized in one of the following ways:

- Configured with a new VIPAROUTE statement in VIPADYNAMIC block allowing one per target dynamic XCF address

- A configured address on NETSTAT VIPADCFG
- Using a current VIPAROUTE status on VIPADYN and VCRT displays

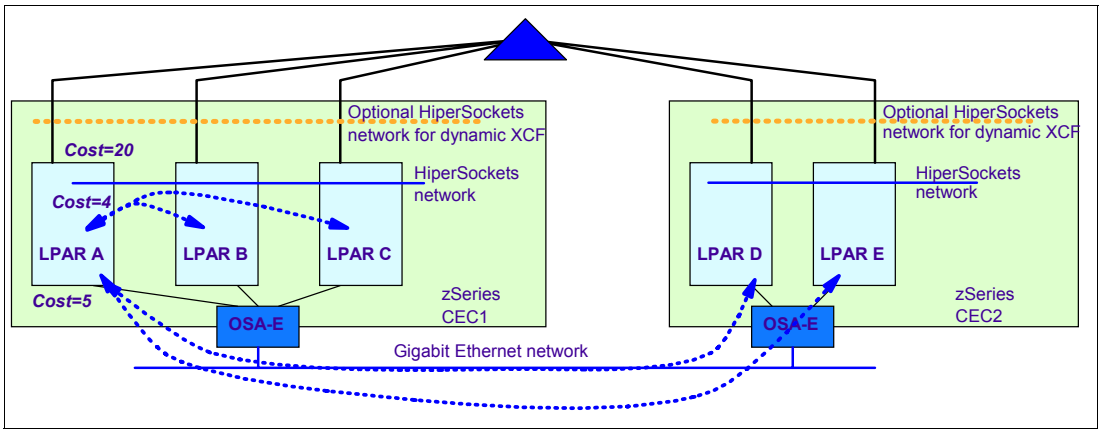


Figure 11-12 TCP/IP sysplex improved distribution optimized routing with V1R7

New VIPAROUTE statement

When a connection from the client needs to be processed by Sysplex Distributor, it will determine if a matching VIPAROUTE statement has been specified or not. If it has, the best available route will be determined using the normal IP routing tables. If no matching VIPAROUTE statement exists for that target, IP packets distributed by Sysplex Distributor to that target will use Dynamic XCF interfaces.

When a VIPAROUTE statement is in effect, packets are sent from the distributor to the target encapsulated in either a GRE wrapper (IPv4) or an IPv6 header. The outer IP header will contain the VIPAROUTE target IP address as it's destination IP address and the distributor's dynamic XCF address as the source IP address.

Note: VIPAROUTE statements will be used on distributing stacks, stacks which are backups for distributing stacks, and stacks which may be used for planned takeovers for non-distributed DVIPAs where the path for forwarding the packets would use XCF links. They are not needed on stacks that are purely target stacks. It may be desirable to have stacks within the same CEC share VIPAROUTE statement.

A VIPAROUTE statement is placed in a VIPADYNAMIC/ENDVIPADYNAMIC block. The VIPAROUTE statement shown in Figure 11-13 on page 229 has the following parameters:

DEFINE Specifies that the Sysplex Distributor should use the target_ipaddr to find the best available route to reach the target stack defined by the dynxcfip.

DELETE Specifies that a previously defined VIPAROUTE statement should be deleted. Sysplex Distributor processing for the target stack specified by the dynxcfip will revert to using dynamic XCF interfaces for existing and new connections after approximately 60 seconds.

dynxcfip Specifies the IPv4 or IPv6 Dynamic XCF address that uniquely identifies a target stack. The address is defined with IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF of that target stack.

target_ipaddr Specifies any fully qualified IPv4 address (in dotted-decimal format) or fully qualified IPv6 address (in colon-hexadecimal format) in the HOME list of the target stack except for a dynamic VIPA (DVIPA) or a loopback address.

It is a static VIPA, a dynamic XCF address, or a real IPv4/IPv6 address associated with a physical interface.

```
                .-DEFINE-.
>>-VIPAROUTE-----+--------+---+--dynxcfip--+--target_ipaddr--+--->
                .-DELEte-.
```

Figure 11-13 VIPAROUTE statement

An example of a VIPAROUTE statement is shown in Figure 11-14 for both IPv4 and IPV6.

```
VIPADYNAMIC
        VIPADEF.....
        VIPAROUTE 193.1.3.94 112.112.112.1       <--- IPv4
        VIPAROUTE 2OEC::193:1:3:94 2001::1:2     <--- IPv6
ENDVIPADYNAMIC
```

Figure 11-14 VIPAROUTE example with IPv4 and IPv6

Important: To change the current configured statement, you must specify the VIPAROUTE DELETE with the same dynxcfip and the same target_ipaddr first, and then specify the VIPAROUTE DEFINE with the same dynxcfip and the different target_ipaddr in a configuration data set on a **VARY TCPIP,,OBEYFILE** command.

If the VIPAROUTE is changed, it will affect active as well as new connections.

Generic routing encapsulation

Generic routing encapsulation (GRE) is a standard protocol described by RFC1701. GRE allows a wrapper to be placed around a packet during transmission of the data. A receiving stack that supports GRE will remove the GRE wrapper, allowing the original packet to be processed by the receiving stack. This is often used to deliver a packet to a stack using an alternate destination IP address. For more information regarding GRE, refer to RFC1701.

Recommendation: It is strongly recommended that all TCP/IP stacks participating in VIPAROUTE distribution must be at least z/OS V1R7.

If IP routing tables have changed or target connectivity success rate (TCSR) is low, Sysplex Distributor will perform a new route lookup to retrieve the current best route approximately every 60 seconds.

Note: z/OS V1R7 supports multipath routes used on a per connections basis if the stack has been configured to use any kind of multipath (per connection or per packet).

New netstat displays for optimized routing

Following are three new Netstat command examples for optimized routing:

- To display configured VIPAROUTE statements that are in effect, issue the **NETSTAT VIPADCFG/-F** command to display all configured VIPAROUTE information. To display the dynamic VIPA configuration for a local host, enter the following command, which produces the output shown in Figure 11-15:

  ```
  Netstat VIPADCFG/-F
  ```

Also supported are filters (IPAddr/l) to allow users to display only the information related to a specific DVIPA or dynamic XCF address.

```
MVS TCP/IP NETSTAT CS V1R7       TCPIP Name: TCPCS           15:51:43
Dynamic VIPA Information:
VIPA Define:
  IpAddr/PrefixLen: 103.1.1.94/24
.
VIPA Distribute:
  Dest: 103.1.1.94..701
    DestXCF: ALL
.
VIPA Route:
  DestXCF: 193.1.3.94
    TargetIp: 9.33.113.3
  DestXCF: 193.1.4.94
    TargetIp: 9.44.114.4
  DestXCF: 2ec0::943:f003
    TargetIp: 2ec0::943:f113
  DestXCF: 2ec0::943:f004
    TargetIp: 2000::4:4
```

Figure 11-15 Netstat VIPADCFG report

- To display the current dynamic VIPA and VIPAROUTE information for a local host, enter the following command, which produces the output shown in Figure 11-16:

  ```
  Netstat VIPADYN/-v
  ```

  Also, an additional optional modifier (DVIPA | VIPAROUTE) can be used to display the current dynamic VIPA information only, or the current VIPAROUTE information only. If no modifier is specified, both dynamic VIPA and VIPAROUTE information will be shown.

```
MVS TCP/IP NETSTAT CS V1R7       TCPIP Name: TCPCS       11:24:56
Dynamic VIPA:
  IpAddr/PrefixLen: 103.1.1.94/28
    Status: Active     Origin: VIPADefine     DistStat: Dist/Dest
..
VIPA Route:
  DestXCF: 193.38.2.2
    TargetIp: 213.38.1.2
    RtStatus: Active
  DestXCF: c1::38:2:2
    TargetIp: d5::38:1:2
    RtStatus: Active
```

Figure 11-16 Netstat VIPADYN report

The RtStatus (status of VIPAROUTE entry) shown in Figure 11-16 indicates the status of the route entry and can have the following values:

- **Active** - The target stack identified by XCF Address or DestXCF is active and TargetIP is defined at that target stack, and at least one route is available to TargetIP. The local stack will forward DVIPA packets to the target stack using the normal IP routing table to determine the best available route.
- **Defined** - The target stack identified by XCF Address or DestXCF is not active.

- **Inactive** - The target stack identified by XCF Address or DestXCF is active, and TargetIp is defined at that target stack; however, no route is available to TargetIp. As a result, the local stack cannot forward any DVIPA packets to the target stack.
- **Unavail** - The target stack identified by XCF Address or DestXCF is active, but TargetIp is not defined at that target stack. The local stack will forward DVIPA packets to the target stack using dynamic XCF interfaces.

- To display the dynamic VIPA Connection Routing Table (VCRT) used for Sysplex Distributor and moveable dynamic VIPA support, use the following command, which produces the output shown in Figure 11-17:

  Netstat VCRT DETAIL/-V

  Additional routing information is displayed in the output when VIPAROUTE profile statements have been configured to the stack.

```
MVS TCP/IP NETSTAT CS V1R7 TCPIP Name: TCPCS 11:17:34
Dynamic VIPA Connection Routing Table:
Dest:      203.38.1.1..801
  Source:  192.168.2.76..1037
  DestXCF: 193.35.1.1
    PolicyRule:    *NONE*
    PolicyAction:  *NONE*
    Intf:   EZAXCFI3
      VipaRoute: No         Gw: 0.0.0.0
Dest:      203.38.1.1..801
  Source:  192.168.2.76..1036
  DestXCF: 193.38.2.2
    PolicyRule:    *NONE*
    PolicyAction:  *NONE*
    Intf:  LTRLE1A
      VipaRoute: Yes        Gw: 213.116.38.1
```

Figure 11-17 Netstat VCRT DETAIL report

Impact to sysplex sockets

TCP sockets applications may benefit from knowing when the partner is in either the same MVS image or the same sysplex. When partners are in the same MVS image, for example, they can share information such as security contexts that are otherwise costly to generate; when both partners are in the same sysplex and communication is through a link that is not exposed outside the sysplex, applications can provide security without costly encryption or decryption of exchanged packets.

The socket option SO_CLUSTERCONNTYPE in getsockopt() allows sockets applications to interrogate the hosting stack about the partner application and to determine whether the partner is in the same sysplex, the same MVS image, or internal.

The internal indicator requested using the SO_CLUSTERCONNTYPE option will no longer be set if the destination IP address (that is, the partner's IP address) for a connection is a Dynamic VIPA or Distributed Dynamic VIPA residing in the sysplex. Traffic destined to these IP addresses can now be forwarded to the target TCP/IP stacks over links or interfaces that are external to the sysplex.

New MIB object and table

Add a new MIB object in the existing ibmMvsDVIPAConnRoute that does the following:

- Provides information about the routes used for distributed connections.

Add a new MIB table, ibmMvsDVIPARouteTable that does the following:

- Provides information about the VIPAROUTE information. Each entry in this table represents a VIPAROUTE profile statement.

Migration concerns

Applications exploiting the SO_CLUSTERCONNTYPE option on the GETSOCKOPT socket API should continue to function properly from a communications perspective, but they may no longer optimize their processing when the destination address being used is a dynamic VIPA or a distributed dynamic VIPA.

If you have applications that exploit this socket option with dynamic VIPAs or distributed dynamic VIPAs, you should consider modifying the configuration to use static VIPAs as the destination addresses.

11.3.2 DVIPA management

In z/OS V1R6 CS, installations must keep and maintain TCP/IP profiles and in z/OS V1R7 CS, DVIPA support allows manual movement of a VIPA by deactivating or reactivating it with the **VARY TCPIP,,SYSPLEX** command.

z/OS V1R7 CS introduces new commands that allow the following three new functions:

- Rejoin the sysplex group
- Deactivate or reactivate Dynamic VIPAs
- Quiesce or resume target applications

Rejoin the sysplex group

V1R6 introduced the following two ways to allow taking a stack out of the sysplex group. When the stack leaves the sysplex group, the current VIPADYNAMIC configuration is lost.

1. Automatically via the sysplex autonomics function

   Note: Starting with z/OS V1R6, Communications Server provided the sysplex autonomics function. If you enable this function, it is very important that you ensure that the WLM policy for the OMPROUTE address space receives sufficient resources in relationship to other work being managed on the system. Under high load conditions it is possible that OMPROUTE, if not properly classified, can trigger an autonomic response from the TCP/IP stack it has an affinity with, resulting in the TCP/IP address space removing itself from the sysplex group. It is recommended that the TCP/IP and OMPROUTE address spaces be placed in the SYSSTC service classification. Classification in another service class will leave the system vulnerable to a Sysplex Distributor outage.

2. Manually via the operator command **VARY TCPIP,,SYSPLEX,LEAVEGROUP**

   The way for the stack to rejoin the group previously was via the **VARY TCPIP,,OBEYFILE** operator command.

V1R7 introduces two new ways for a stack to rejoin the sysplex group and automatically reestablish a VIPADYNAMIC configuration:

1. Automatically via a sysplex autonomics function, as follows:
   a. When a recoverable problem (see Table 11-1 on page 233) that caused the stack to leave the group has been relieved, automatic rejoin takes effect.

b. Automatic rejoin is not possible if the problem was non-recoverable.

c. RECOVERY and AUTOREJOIN must be configured on the GLOBALCONFIG SYSPLEXMONITOR profile statement (see Figure 11-18 on page 234).

2. Manually via the **VARY TCPIP,,SYSPLEX,JOINGROUP** operator command (see Figure 11-19 on page 235). This replaces the existing method of rejoining the sysplex group using the **VARY TCPIP,,OBEYFILE** command and can be used after a non-recoverable problem.

   LEAVEgroup requests the TCP/IP stack to leave the sysplex group. This causes the stack to leave the sysplex group, delete all dynamic DVIPAs, and inactivate all its configured VIPADYNAMIC definitions. The VIPADYNAMIC configuration information is retained for possible future use by the **SYSPLEX,JOINGROUP** command. To rejoin the sysplex group it is necessary to issue a **VARY TCPIP,,SYSPLEX,JOINGROUP** operator command, which also reprocesses the stack's saved VIPADYNAMIC configuration.

Note: Now the **VARY TCPIP,, OBEYFILE** command fails for the new DYNAMICXCF and VIPADYNAMIC statements used with z/OS V1R7 CS.

Table 11-1 Recoverable and non-recoverable problems

Type	Message ID	Message	Additional information
Recoverable	EZZ9671E	tcpstackname DETERMINED THAT VTAM WAS INACTIVE FOR AT LEAST timevalue SECONDS	The problem is cleared when VTAM is started.
Recoverable	EZZ9672E EZZ9678E	tcpstackname DETERMINED THAT OMPROUTE WAS NOT RESPONSIVE FOR AT LEAST timevalue SECONDS	The problem is cleared when OMPROUTE is restarted.
Recoverable	EZZ9673E	tcpstackname DETERMINED THAT DYNAMIC XCF CONNECTIVITY TO ALL PARTNERS WAS NOT AVAILABLE FOR AT LEAST timevalue SECONDS	The problem is cleared when any XCF route is successfully activated.
Recoverable	EZZ9679E	tcpstackname DETERMINED THAT CSM WAS CRITICAL FOR AT LEAST timevalue SECONDS	The problem is cleared when CSM storage is no longer critical.
Recoverable	EZD1172E	tcpstackname DETERMINED THAT ALL PARTNERS WERE UNREACHABLE FOR AT LEAST timevalue SECONDS	The problem is cleared when any configured route (VIPAROUTE or XCF) to a partner is activated.
Recoverable	EZD1187E	tcpstackname WAS NOT ABLE TO GET TCP/IP storagetype STORAGE Due to storage limits set by GLOBALCONFIG, the requested storage (ECSA or PRIVATE) was not available	The problem is cleared when the requested storage is no longer critical.
Non Recoverable	EZD1170E	tcpstackname WAS NOT ABLE TO GET TCP/IP storagetype STORAGE	The requested storage was not available and GLOBALCONFIG storage limits were not set for the requested storage.
Non Recoverable	EZZ9670E	tcpstackname SYSPLEX PROCESSING ENCOUNTERED A NONRECOVERABLE ERROR - abendcode - abendreasoncode	
Non Recoverable	EZZ9674E	tcpstackname SYSPLEX PROCESSING WAS NOT RESPONSIVE FOR AT LEAST timevalue SECONDS	

The VIPADYNAMIC configuration information is saved when the stack leaves the group, and reprocessed when the stack rejoins the group, and includes the following VIPADYNAMIC definitions:

- VIPADEFINE
- VIPABACKUP
- VIPADISTRIBUTE
- VIPARANGE
- VIPASMPARMS
- VIPAROUTE

The following dynamic VIPA definitions are not saved when the stack leaves the sysplex group:

- Target DVIPAs are automatically re-created when the stack rejoins the group if this stack is still a target for that DVIPA from another (distributing) stack.
- BIND or IOCTL created DVIPAs must be re-created by the applications or by the MODDVIPA utility after the stack has rejoined the group.

When a stack has left the sysplex group, a saved VIPADYNAMIC configuration (if any) can be displayed by using the **NETSTAT VIPADCFG/-F** command.

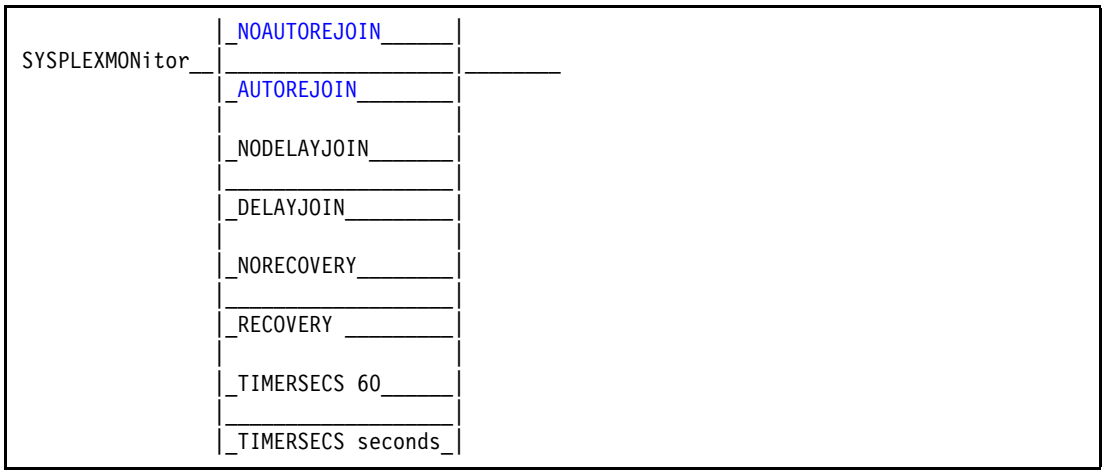


Figure 11-18 GLOBALCONFIG profile statement

The two new options shown in Figure 11-18 are as follows:

NOAUTOREJOIN The stack will not automatically rejoin the TCP/IP sysplex group when a detected problem is relieved. This is the default.

AUTOREJOIN When all detected problems (that caused the stack to leave the sysplex group) are relieved, the stack will automatically rejoin the sysplex group and reprocess the saved VIPADYNAMIC configuration.

Restriction: AUTOREJOIN requires RECOVERY to also be configured. AUTOREJOIN may not be configured when NORECOVERY is configured (or defaulted). If you use the **VARY TCPIP,,SYSPLEX,LEAVEGROUP** command to take the stack out of the group, automatic rejoin will not occur. You must use the **VARY TCPIP,,SYSPLEX,JOINGROUP** command to cause the stack to rejoin the sysplex group.

Guideline: AUTOREJOIN should be used when RECOVERY is configured to allow the stack to rejoin the sysplex group without operator intervention.

AUTOREJOIN/NOAUTOREJOIN considerations

AUTOREJOIN or NOAUTOREJOIN can be changed via a **VARY TCPIP,,OBEYFILE** command whether or not the stack is currently in the sysplex group. Changing from AUTOREJOIN to NOAUTOREJOIN will prevent the stack from automatically rejoining the sysplex group when a problem detected by the sysplex autonomics function is relieved. Changing from NOAUTOREJOIN to AUTOREJOIN will allow the stack to automatically rejoin the sysplex group when all problems detected by the sysplex autonomics function are relieved. If you change from NOAUTOREJOIN to AUTOREJOIN after the stack has left the sysplex and before the problem which caused it to leave has been relieved, the stack will automatically rejoin the sysplex group when the problem is relieved. However, if you change from NOAUTOREJOIN to AUTOREJOIN after the problem which caused the stack to leave the group has been relieved, a **VARY TCPIP,,SYSPLEX,JOINGROUP** command will be needed to cause the stack to rejoin the sysplex.

JOINgroup requests the TCP/IP stack to join the sysplex group. If this command is issued after the stack has left the sysplex group, it will also reprocess the stack's saved VIPADYNAMIC configuration.

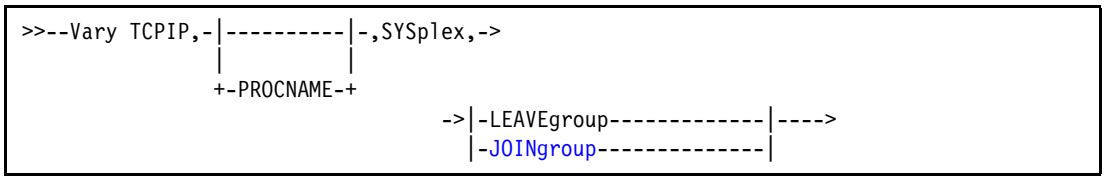


Figure 11-19 New keyword, JOINgroup, on the VARY TCPIP,,SYSplex command

Tip: Before issuing the **VARY TCPIP,,SYSPLEX,JOINGROUP** command, use the **VARYTCPIP,,SYSPLEX,DEACTIVATE** command to deactivate any DVIPA that you do not want restored when the stack rejoins the sysplex group.

JOINgroup considerations

When this command is issued, the following message is issued:

```
EZD1178I THE VARY TCPIP,,SYSPLEX,JOINGROUP COMMAND WAS ACCEPTED
```

Consider the following conditions:

- If VTAM is not running, or if the DELAYJOIN parameter is configured for GLOBALCONFIG SYSPLEXMONITOR and OMPROUTE is not initialized, the join will not take place until after VTAM (and OMPROUTE, if DELAYJOIN is configured) are initialized.
- When the join has completed, the following message is displayed:

  ```
  EZD1176I TCPCS HAS SUCCESSFULLY JOINED THE TCP/IP SYSPLEX GROUP
  ```

- If the stack had previously left the group and VIPADYNAMIC configuration had been saved, you will see either of the following messages:

  ```
  EZD1192I THE VIPADYNAMIC CONFIGURATION WAS SUCCESSFULLY RESTORED FOR TCPCS
  EZD1193I ALL OF THE VIPADYNAMIC CONFIGURATION DEFINITIONS FOR TCPCS COULD
  NOT BE RESTORED
  ```

Note: If the **VARY TCPIP,,SYSPLEX,LEAVEGROUP** command is used to take the stack out of the sysplex group, a **VARY TCPIP,,SYSPLEX,JOINGROUP** command is required to bring the stack back into the group. The sysplex autonomics function will not automatically bring the stack back into the group after a **VARY TCPIP,,SYSPLEX,LEAVEGROUP** command has been issued.

The **VARY TCPIP,,SYSPLEX,JOINGROUP** command will not be accepted if the Sysplex Problem Detection cleanup function was unsuccessful and message EZZ9675E was issued. If this has occurred, you have to restart the stack before it will be able to rejoin the sysplex group.

If any configuration conflict is detected while reprocessing the saved VIPADYNAMIC configuration, specific informational messages are issued and the saved VIPADYNAMIC definitions that fail are discarded.

DEACTivate or REACTivate dynamic VIPAs

New keywords, DEACTivate and REACTivate, are introduced for the **VARY TCPIP,, SYSplex** command in V1R7, as shown in Figure 11-20.

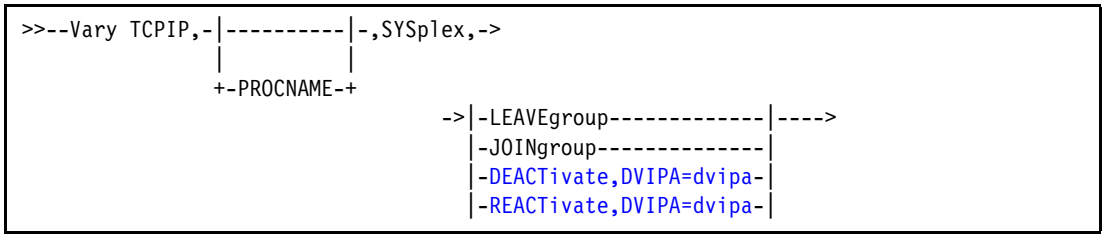


Figure 11-20 New keywords on the Vary TCPIP SYSplex command

DEACTivate Request the TCP/IP stack to deactivate a Dynamic VIPA. When you deactivate a Dynamic VIPA, it appears as though the DVIPA has been deleted, but the DVIPA's configuration is saved.

DVIPA=dvipa dvipa is the IPv4 address, IPv6 address, or IPv6 interface name of a DVIPA that is currently defined by VIPADEFINE or VIPABACKUP on this stack. The DVIPA can be in ACTIVE, BACKUP, or MOVING status.

REACTivate Request the TCP/IP stack to redefine a deactivated Dynamic VIPA and any distribution defined for that DVIPA using its saved configuration.

DVIPA=dvipa dvipa is the IPv4 address, IPv6 address, or IPv6 interface name of a DVIPA that is currently deactivated.

Example of DEACTivate

As shown in Figure 11-21 on page 237, deactivating a DVIPA removes the DVIPA resources from that stack, but saves the DVIPA's configuration information, and behaves as though the DVIPA has been deleted on that stack. The commands **Netstat VIPADYN/-v** and **DISPLAY TCPIP,,SYSPLEX,VIPAD** will not show this DVIPA. The **Netstat VIPADCFG/-f** command report will include this DVIPA's information under a new heading.

- **If DVIPA is active** - The stack stops supporting it. The DVIPA is removed from the HOME list except for the following:
  - When the DVIPA has existing connections, the commands **Netstat VIPADYN** and **DISPLAY TCPIP,,SYSPLEX,VIPAD** will show the DVIPA in QUIESCING status. The DVIPA will disappear when the last connection ends.

– When this stack is a target from the taking-over stack, Netstat VIPADYN and **DISPLAY TCPIP,,SYSPLEX,VIPAD** will show DVIPA as Active and Dest.

A backup stack, if any, will activate a DVIPA (takeover).

- **If DVIPA is backup** - A deactivate removes this stack as a backup, preventing this stack from taking over if the current owner fails.

Example of REACTivate

Reactivating a DVIPA reprocesses the deactivated configuration definition. The reactivated DVIPA will be active if any of the following conditions are defined:

- The DVIPA's origin was VIPADEFINE.
- The DVIPA's origin was VIPABACKUP.
- The MOVEABLE parameter was configured, and the DVIPA is not active elsewhere in the sysplex.

If DVIPA is active, the stack advertises the DVIPA, DVIPA is added to the HOME list (if not already there), and the current owner, if any, will give up the DVIPA (takeback).

Reactivation will fail if there is conflicting configuration on this stack or in the sysplex. If this occurs, the DVIPA remains deactivated, and the deactivated DVIPA can be deleted (VIPADELETE will delete the DVIPA definition and any VIPADISTRIBUTE definitions for that DVIPA), or remove the conflicting configuration and reissue the **VARY TCPIP,,SYSPLEX,REACT** command.

Deactivating or reactivating can be done when the stack is not in the sysplex group, then deactivated DVIPAs remain deactivated when the stack rejoins the group and the VIPADYNAMIC configuration is reprocessed. Deactivation and reactivation only applies to VIPADEFINE or VIPABACKUP DVIPAs. It is not possible to deactivate a target DVIPA (unless it is also configured as VIPADEFINE or VIPABACKUP), or a VIPARANGE DVIPA is created by BIND, SIOCSVIPA, SIOCSVIPA6 ioctl, or the MODDVIPA utility.

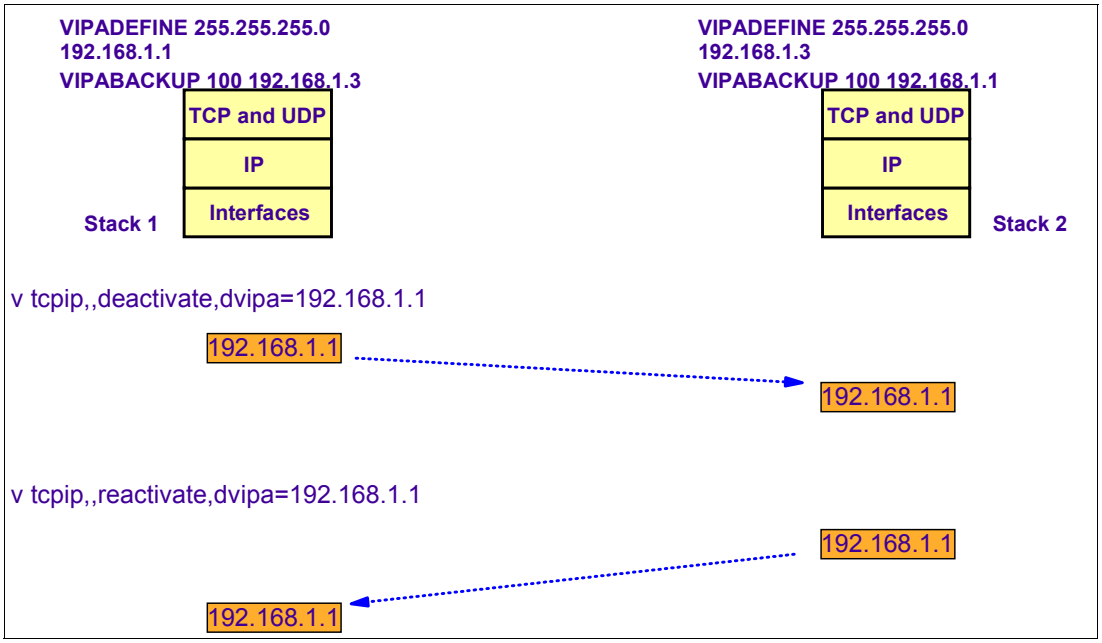


Figure 11-21 Deactivate/Reactivate DVIPA

Quiesce/Resume target applications

The **VARY TCPIP,,SYSPLEX** command has been enhanced to allow operators to use **quiesce** or **resume** to target an application or the entire target stack for sysplex distribution.

Quiesce

Two new operator commands enable you to stop DVIPA Sysplex Distributor workload balancing to a specific server application or all server applications on a TCP/IP stack, as follows:

- The **VARY TCPIP,,SYSPLEX,QUIESCE,PORT** command enables you to stop distribution for a specific server application.
- The **VARY TCPIP,,SYSPLEX,QUIESCE,TARGET** command enables you to stop distribution for all server applications.

The existing TCP connections are not disrupted by either command. These commands can be useful in scenarios where a target application or system needs to be stopped (such as for applying maintenance). By issuing the commands before stopping the application or system, you can minimize the impact to end users by preventing new work while still allowing existing requests to be completed. These commands can also be useful in scenarios where you need to temporarily divert new connection requests away from a particular target application or target TCP/IP stack.

Resume

Two new operator commands enable you to resume DVIPA Sysplex Distributor workload balancing to a specific server application or all server applications on a TCP/IP stack, as follows:

- The **VARY TCPIP,,SYSPLEX,RESUME,PORT** command enables you to resume distribution for a specific server application.
- The **VARY TCPIP,,SYSPLEX,RESUME,TARGET** command enables you to resume distribution for all server applications.

Quiesce/Resume sysplex distribution for individual applications (identified by port and, optionally, jobname and ASID) stops or resumes sysplex distribution of new connections to the application on the stack where the command was entered, with no impact to existing connections. This must be issued on the stack where the application runs.

Quiesce/Resume sysplex distribution for the entire target stack stops or resumes sysplex distribution to the entire stack where the command was entered, with no impact to existing connections. This must be issued on the stack where the application runs.

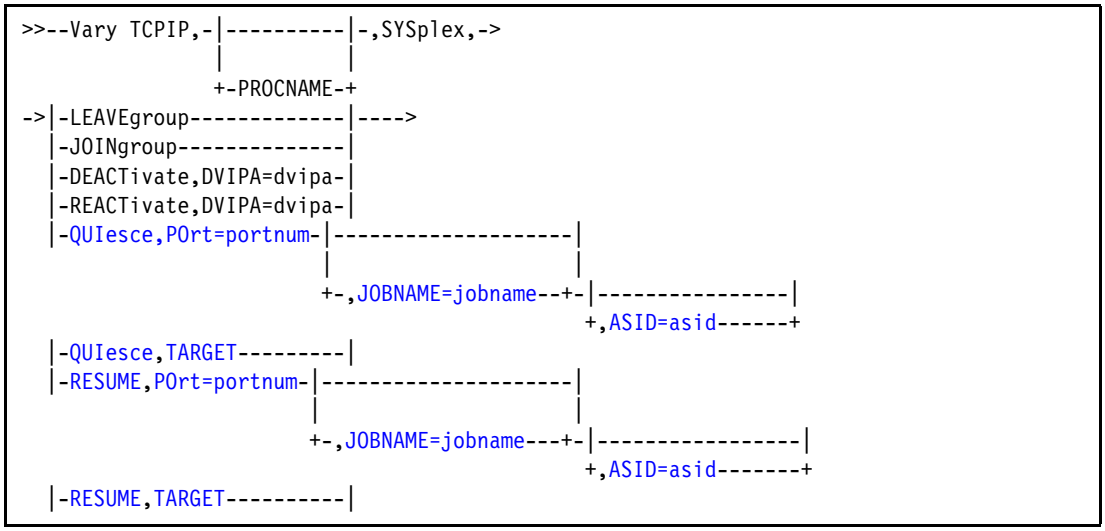


Figure 11-22 Quiesce and Resume to the Vary TCPIP sysplex command

The parameter options that are new with this release are:

QUIesce Request the specified application, or all applications on a particular TCP/IP stack, be quiesced from DVIPA Sysplex Distributor workload balancing. After the command is issued, Sysplex Distributor will no longer route new TCP connection requests to one or more specified application. Existing connections to these applications are not affected. This command must be issued on the local system where the applications are to be quiesced. This command can be useful in scenarios where you would like to temporarily divert new TCP connection requests away from a specific application or target system. One such scenario is when a particular application or system will be shut down (for example, in order to apply maintenance). Issuing this command prior to the shutdown can allow applications to gracefully complete any existing workload requests.

RESUME Request the specified application, or all applications associated with a TCP/IP stack, be resumed for DVIPA Sysplex Distributor workload balancing (that is, become eligible for new TCP connection requests).

POrt The port number is an integer between 1 and 65,535. PORT or TARGET must be specified following the QUIesce/RESUME keyword.

TARGET Request that all applications on this TCP/IP stack be quiesced or resumed for DVIPA Sysplex Distributor workload balancing. Port or Target must be specified following the QUIesce/RESUME keyword.

JOBNAME Jobname is an optional parameter. If the portnum specifies a port that has more than one instance of an application bound to it, then either JOBNAME or JOBNAME and ASID must be specified to identify a unique specific application instance to be resumed. The jobname specifies the MVS jobname of the application with which the resume command will be associated. Jobname can be up to 8 characters in length. The environment in which the application is run determines the jobname to be associated with a particular client or server application.

Applications submitted as batch jobs use the batch job name. The jobname associated with applications started from the MVS operator console using the **START** command will be determined as follows:

- If the START command is issued with the name of a member in a cataloged procedure library (for example, S APP1), the jobname will be the member name (for example, APP1).
- If the member name on the START command is qualified by a started task identifier (for example, S APP1.ABC), the jobname will be the started task identifier (for example, ABC).
- The JOBNAME parameter can also be used on the START command to identify the jobname (for example, S APP1,JOBNAME=XYZ).
- The JOBNAME can also be included on the JOB card.

Applications run from the z/OS shell normally have a job name that is the logged on user ID plus a one-character suffix, as follows:

- Authorized users can run applications from the z/OS shell and use the _BPX_JOBNAME environment variable to set the job name. In this case, the value specified for the environment variable is the job name.
- z/OS UNIX applications started by INETD typically use the jobname of the INETD server plus a one-character suffix.

ASID ASID is an optional parameter. If the portnum specifies a port that has more than one instance of an application bound to it and the jobname is not unique, then an asid must be specified to identify a unique specific application instance. The asid is the hexadecimal address space ID associated with the application to be quiesced or resumed.

QUIESCE/RESUME considerations and guidelines

The new commands must be issued on the system and TCP/IP stack where the application instance is running. The following conditions must also be considered:

- The commands apply to a single TCP/IP stack's application instance. If the server needs to be quiesced or resumed over multiple stacks in a CINET environment the commands will need to be issued on each stack.
- Any Sysplex Distributor timed affinities will be terminated. Existing connections are not affected.
- The QUIESCE state is associated with the application's active listening socket. If the application is recycled or if the application closes and opens a new listening socket on the specified port it will no longer be in a quiesced state.
- If the application is bound to the unspecified address, it may continue to receive connection requests that are not using a distributed DVIPA as the destination IP address.
- The QUIESCE state for a TARGET persists for all applications (existing and new) running on this TCP/IP stack, until the TCP/IP stack is recycled or a **V TCPIP,,RESUME,TARGET** command is issued.
- When an entire TCP/IP stack is quiesced via the TARGET option, you cannot resume individual applications for workload distribution. You can, however, resume distribution for the entire TCP/IP stack using the **V TCPIP,,RESUME,TARGET** command.
- RESUME with the TARGET option is the only valid command following a QUIESCE with the TARGET option command.

- When a TCP/IP stack is quiesced, the “ready count” (Rdy) field that appears on the **Netstat VDPT** display (issued on the Sysplex Distributor routing stack) will be zero for all entries associated with this target TCP/IP stack.

Migration concerns

When a stack leaves the sysplex group by operator command or Sysplex Autonomics:

- The current VIPADYNAMIC configuration information is saved.
- New DYNAMICXCF and VIPADYNAMIC statements are ignored.
- An informational message is issued.
- Rejoining the sysplex group is changed. The existing method (**VARY TCPIP,,OBEYFILE** command with DYNAMICXCF or VIPADYNAMIC statements) is replaced by 2 new methods:
  - Automatically, via Sysplex Autonomics
  - Manually, via **VARY TCPIP,,SYSPLEX,JOINGROUP** operator command
- The VIPADYNAMIC configuration is automatically reprocessed.

Netstat VIPADCFG/-F report changes are:

- The report may contain data when the stack has left the group.
- VIPADISTRIBUTE information will be displayed for all DVIPAs, regardless of status.
- Prior releases displayed VIPADISTRIBUTE information only for DVIPAs in active status.
- A new section is included. Deactivated Dynamic VIPA Information will contain VIPADEFINE, VIPABACKUP and VIPADISTRIBUTE information for all deactivated DVIPAs on the stack. Information in this section is formatted the same as in the Dynamic VIPA Information section.
- Any of the following Informational messages may precede the report:

```
EZZ2502I ...the stack is not a member of the TCP/IP sysplex group
EZZ2503I ...VIPADYNAMIC configuration is currently inactive
EZZ2505I ...VIPADYNAMIC configuration information is not available while
            the stack is delaying sysplex profile processing
```

11.3.3 Sysplex Distributor and WLM before z/OS V1R7

Before z/OS V1R7, using the Sysplex Distribution function, incoming connections for a DVIPA and port are distributed to multiple target stacks. Target selection is determined using the following:

- RoundRobin
- BaseWLM - Capacity Recommendations from WLM (weights) for each system. Weights are normalized, optionally modified with Policy information (QoS fractions) from each target.
- Shareport - Balance active connections (no backlog queue).

Determining weights

When determining a base WLM weight, WLM assigns a relative weight to each system in the sysplex with the highest weight going to the system with the most available CPU capacity. The weights range between 0 and 64. If all systems in the sysplex are running at or near 100% utilization, WLM will assign the highest weights to the systems with the largest amounts of lower importance work. In this way, new connection requests will be distributed to the systems with the highest displaceable capacity.

Normalizing and determining the QoS modified WLM weight proceeds as follows:

- WLM weights are normalized - the WLM weights range in value from 1 to 64. These returned system weights are divided by the smallest system weight. For example, if BaseWLM system weights of 50, 30, and 10 are returned, the normalized weights are 5, 3, and 1.
- A QoS Service level fraction is received from the target for each group of connections that map to a DVIPA/PORT for that service level. The fraction represents the performance of this group of connections.This is based on maximum connection limit for the service level, the target to client performance (ratio of retransmits and timeouts to number of packets sent, overall throughput and throughput/connection against desired values). The lower the fraction, the better the performance.
- The normalized WLM weight is reduced by the QoS Fraction percentage. For example, if the normalized WLM weight is 5 and the QoS Fraction is 20%, the modified weight is 4, calculated as follows: (5 - (5 * 20%)).

Figure 11-17 and the following discussion refer to an example of the distribution of connections to DVIPA1, Port 8000:

- Connections come in destined for DVIPA1, Port 8000.
- Based on the QoS modified WLM weights for this DVIPA/Port and service level, as seven connection requests are received, three connections are distributed to Target 1 and four connections to Target 2.
- Target 1 is configured with Shareport for Port 8000. Connections are evenly distributed among the servers that have no backlog queue such that each server has the same number of active connections.

The following describes the distribution problems with the base WLM:

- The WLM weight is based on a comparison of the available capacity for new work on each system, not how well each server is meeting the goals of its service class.
- If all systems are using close to 100% of capacity, then the WLM weight is based on a comparison of displaceable capacity - the amount of lower importance work on each system, but if the service class of the server is of low importance then it may not be able to displace this work.
- If a target stack or a server on the target stack is not responsive, new connection requests will continue to be routed to the target - the stack may even appear to be lightly loaded since applications are not processing any new work.
- QoS fractions monitor Target to Client performance, but do not monitor the path to the target.
- Shareport ensures an even distribution of active connections among servers that have not exceeded their backlog queue limit. However, it does not account for how well each server is meeting the goals of its service class.

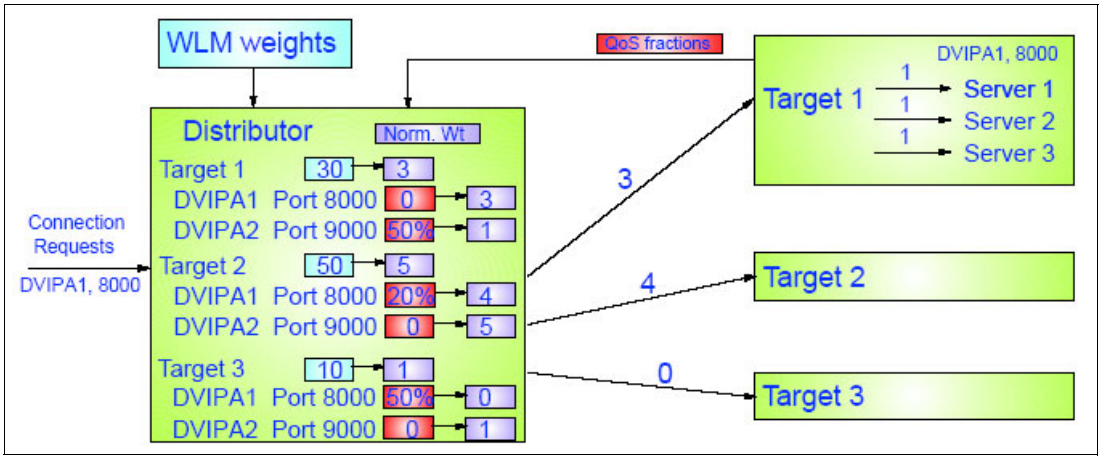


Figure 11-23 Base WLM distribution

11.3.4 Sysplex Distributor and WLM with V1R7

This new release solves the problems or limitations existing in the previous releases. The new release is based in the server-specific WLM and the sysplex autonomics health monitor for target stacks.

WLM is polled per server on each target stack shown in Figure 11-24 on page 244. WLM passes back, per server, the following information:

- How well each server is meeting the goals of its service class
- The displaceable capacity on the target system of new work, based on importance of the server's service class

In the server-specific WLM, WLM assigns a weight based on how well a server is meeting the goals of its service class and the displaceable capacity for new work based on the importance of its service class. Server-specific WLM weights are received for each server at the target. The shareport distribution algorithm will also be able to use the server-specific weight. Quality of Service (QoS) fractions are applied before normalizing.

A server-specific weight is sent from the target to the distributor for each DVIPA/Port. In the case of multiple shareport servers, an average weight is sent to the distributor. Determining the QoS modified WLM weight and normalizing to preserve more of the distinctions between different weights, the QoS fraction is applied to the raw WLM weight before normalizing, and the normalization algorithm is changed. Figure 11-24 shows the weight for Target 2's DVIPA1, Port 8000 Server is calculated as follows:

- The QoS Service level fraction is applied against the raw WLM weight before the WLM weight is normalized. The WLM weight is 40 and the QoS fraction is 20%, so the QoS modified WLM weight is 32 (40 - (40 * 20%).
  - The normalized weight is 8 - determined by dividing by 4.
  - The exception to this would be if all of the received WLM weights associated with a DVIPA/Port were less than or equal to 16. In that case normalization is not done. After the QoS fraction is applied against the raw weight, the weights are left unchanged.
  - To change a Server weight, WLM depends on the Server receiving work. So even if a Server weight is zero, a connection request will still be forwarded infrequently to that Server to generate new WLM values.

- Continuing with the example shown in Figure 11-24, consider the following aspects of the distribution of connections to DVIPA1, Port 8000:
  - Connections come in destined for DVIPA1, Port 8000.
  - Based on the QoS modified WLM weights for this DVIPA/Port and service level, as 18 connection requests are received, 10 connections are distributed to Target 1 and 8 connections to Target 2.
  - Target 1 is configured with Shareport for Port 8000. As 30 connections are received, 15 will be distributed to server 1, 10 to server 2, and 5 to server 3.

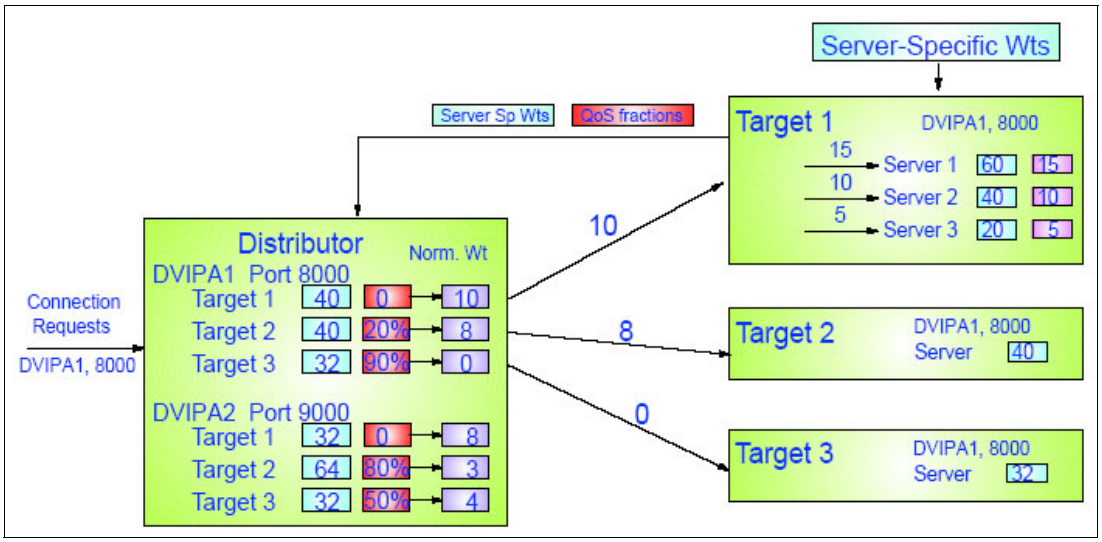


Figure 11-24 Server-specific WLM

11.3.5 Sysplex autonomics health monitor

The sysplex autonomics health monitor for target stacks determines if target or server WLM capacity is okay, but considers the following conditions that could occur:

- Connection requests are not reaching the target. Monitoring connectivity between the distributing stack and the target stack considers whether any new connection requests are reaching the target. The target connectivity success rate (TCSR) is the answer to this question, as follows:
  - Monitoring network connectivity between the server and client determines whether any new connections are being established. This determines the connection establishment rate (CER).
  - Monitoring target server responsiveness determines if the server is accepting new work. This determines the server accept efficiency fraction (SEF).

    The target sends SEF values and server statistics, shown in Figure 11-25 on page 245, to the distributor, which creates a target server responsiveness fraction (TSR) based on the TCSR and SEF (which includes CER). All values are expected to be 100 unless there is a problem. TCSR dropping to 25 or lower will drive the Optimized Routing function to do a new route lookup.

    Target stacks push key TCP/IP health statistics for target applications to the distributor. The following commands can be used to display them:

    - **`NETSTAT VDPT/-O`** for statistics per DVIPA/Port
    - **`NETSTAT ALL/-A`** for SHAREPORT statistics per socket

- Network path down between server and client.
- Server is not accepting new connection requests.

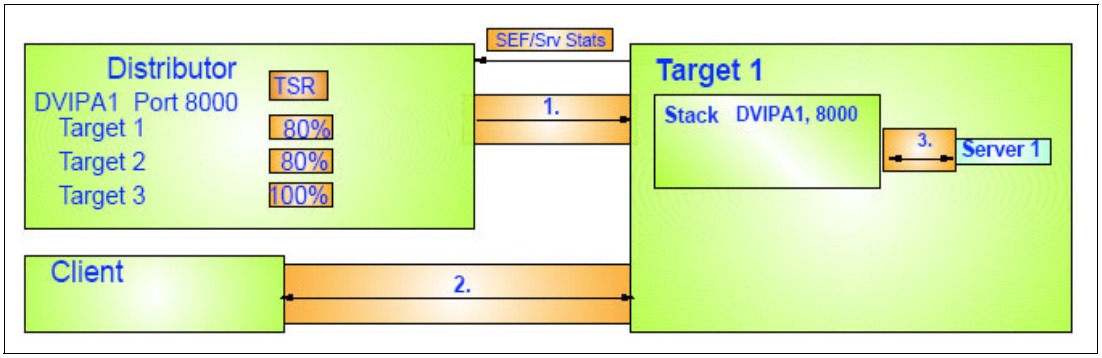


Figure 11-25 Sysplex autonomics health monitor for target stacks

Server-specific WLM and sysplex autonomics health monitor for target stacks allows the distributor to receive WLM server-specific weights, QoS fractions, SEF fractions, and server statistics, as shown in Figure 11-26, from the targets for each server. The distributor calculates a normalized weight from the raw server-specific weight, QoS fraction, and TSR fractions. If SHAREPORT WLM is being used, the target will use the SEF fractions and apply it to server-specific weights.

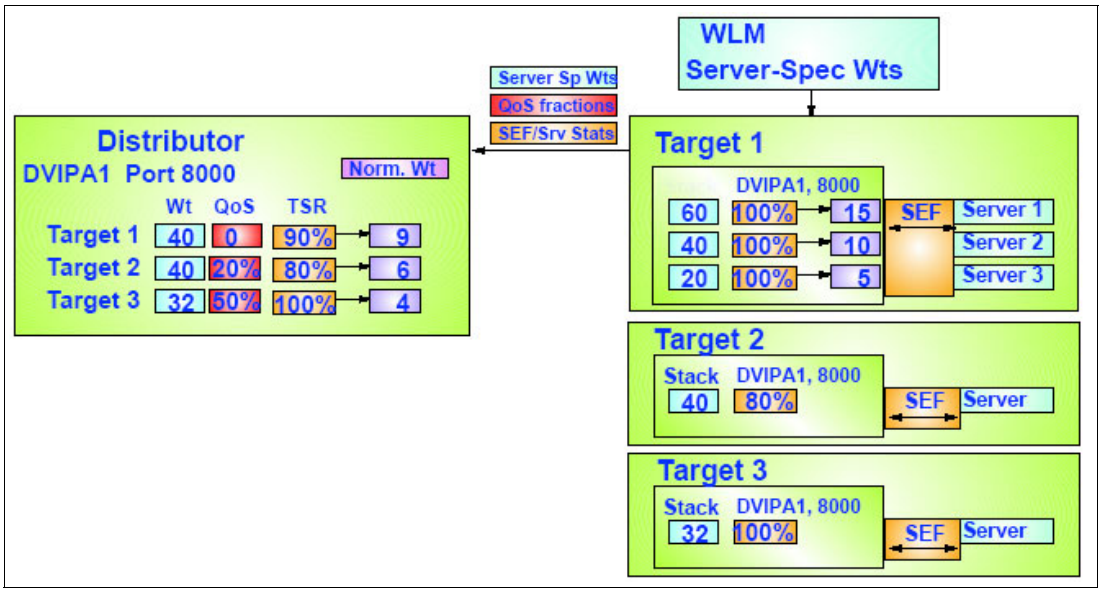


Figure 11-26 Server-specific WLM and sysplex autonomics health monitor for target stack

Sysplex autonomics example

As shown in Figure 11-26, the weight for Target 2's DVIPA1, Port 8000 Server is calculated as follows:

- The QoS Service level fraction is applied against the raw WLM Server weight. For example, if the WLM weight is 40 and the QoS fraction is 20%, the QoS modified WLM weight is 32 (40 - (40 * 20%)).
- A TSR fraction is calculated from the SEF value and server statistics that are received from the target and from information that the distributor keeps for each server. A higher

fraction means a healthier server. So if the QoS modified WLM weight is 32, and the Server fraction is 80%, the new modified weight is 25 (32 * 80%).

- Finally, the normalized weight of 6 is determined by dividing by 4.

SHAREPORT WLM distribution: The target calculates an SEF from the statistics for each shareport server. At the target, the fractions are applied against the raw server weights and normalized. The average of the SEF values and statistics is sent to the distributor. The average of the raw Server weights is sent to the distributor.

SHAREPORT distribution: If the existing SHAREPORT parameter is used, distribution is changed to use the SEF value alone. The SEF value is applied against an assumed raw weight of 64 (the highest weight) and a normalized weight is calculated as described previously. It is no longer based on balancing the number of active connections among the servers.

11.3.6 Configuring sysplex distribution using server-specific weights

Configure the existing parameter, SYSPLEXROUTING, on the IPCONFIG statement on the Distributor and each target stack. Configure a new parameter, SERVERWLM, on the VIPADISTRIBUTE statement, as shown in Figure 11-27.

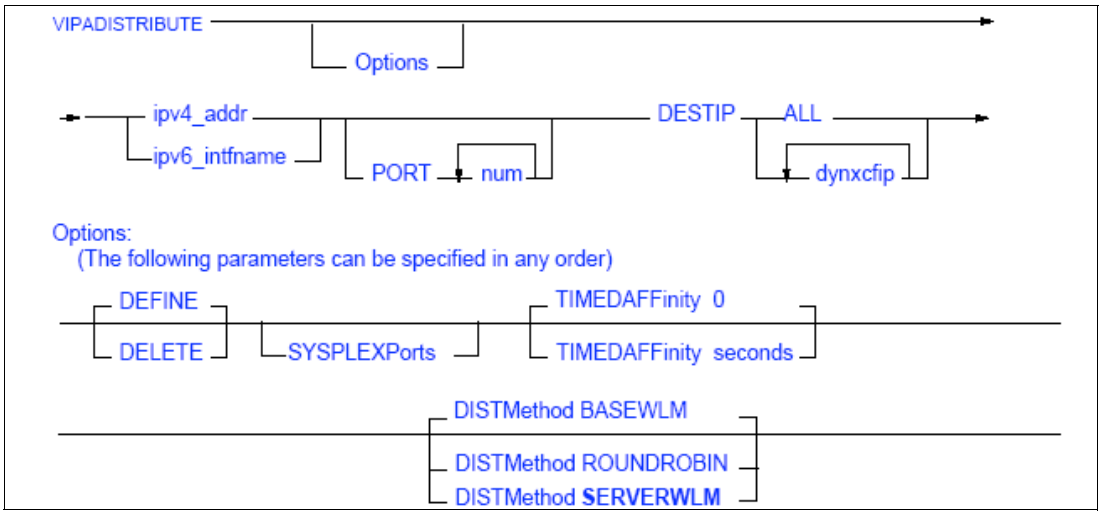


Figure 11-27 Configuring sysplex distribution assigns server-specific weights

Configuring shareport distribution to use server-specific weights

Configure a new parameter, SHAREPORTWLM, on the PORT statement. The SHAREPORTWLM parameter is independent of SERVERWLM.

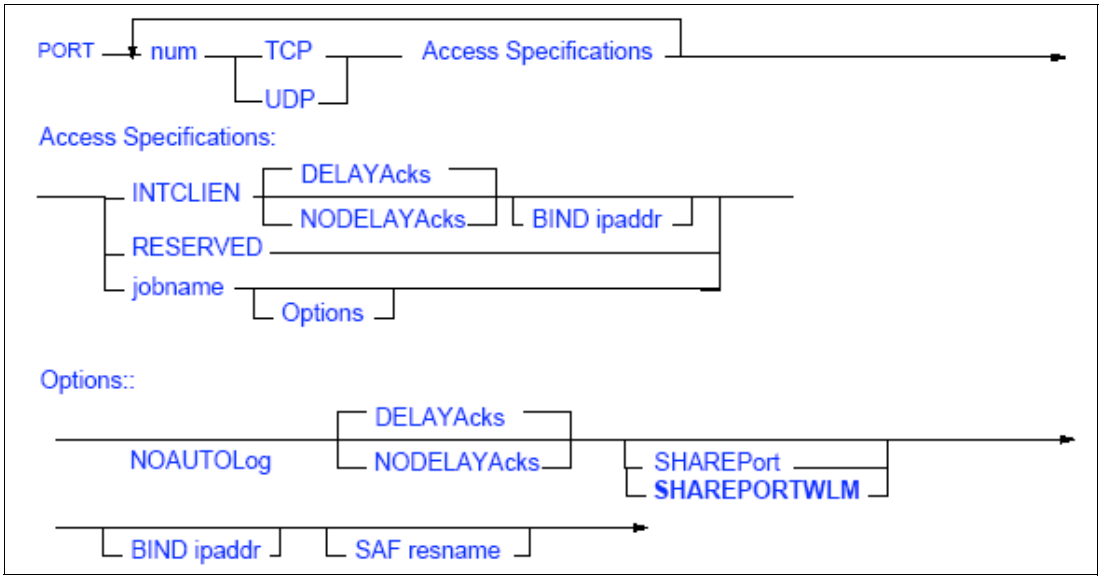


Figure 11-28 Configuring shareport distribution to use server-specific weights

Netstat display commands

The TCP/IP sysplex improved distribution server-specific WLM implementation can be displayed using the following operator commands:

- **`Netstat VIPADCFG/-F`** report - Display the configured distribution method.
- **`Netstat VDPT/-O`** report - Display distribution method, TSR values, WLM weights (after TSR adjustment).
- **`Netstat VDPT/-O DETAIL`** - Display TCSR, CER, and SEF values.
- **`Netstat PORTList/-o`** report - An existing flag is used to indicate that port sharing is being used (`S`). An additional new flag will be used to indicate that port sharing with Server-Specific weights is being used (`W`), as shown in Figure 11-29.

.

```
Long Format:
MVS TCP/IP NETSTAT CS V1R7 TCPIP Name: TCPCS 08:58:11
Port# Prot User Flags Range
----- ---- ---- ----- -----
08000 TCP CICS1 DASW
```

Figure 11-29 Netstat portlist display

Migration concerns

WLM server recommendations can only be used if the Sysplex Distributor and all target stacks for a distributed DVIPA port are V1R7 or later. If all targets for a DVIPA do not provide server-specific weights, then BASEWLM will be used.

Target Server responsiveness values can only be used if the Sysplex Distributor and all target stacks for a distributed DVIPA port are V1R7 or later. TSR values can be used with BASEWLM or SERVERWLM weights.

11.3.7 Subarea VTAM XCF support

z/OS V1R7 includes the capability to establish XCF connectivity between TCP/IP stacks in the sysplex from an APPN node without having to establish APPN connectivity between the nodes, and to establish TCP/IP XCF connectivity in the sysplex between TCP/IP stacks residing on pure Subarea VTAM nodes.

- For APPN nodes:
  - A new value for the XCFINIT start parameter, DEFINE, is now allowed.
  - If XCFINIT = DEFINE:
    - VTAM will join the ISTXCF Sysplex group.
    - VTAM will build the definitions necessary for XCF connectivity between this node and other nodes in the sysplex.
      The XCF APPN PU and XCF TRLE definitions will be built.
    - VTAM will not activate those connections.
    - TCP/IP connectivity is allowed, using either static or dynamic XCF definitions.
  - XCFINIT=YES will remain the default for APPN nodes.
- For pure subarea nodes:
  - XCFINIT start parameter is now allowed.
  - Allowed values are NO or DEFINE.
  - If XCFINIT = DEFINE:
    - VTAM will join the ISTXCF Sysplex group.
    - VTAM will build the definitions necessary for XCF connectivity between this node and other nodes in the sysplex.
      The XCF TRLE definition will be built.
  - No SNA XCF connectivity will be established between this node and other nodes in the Sysplex.
  - TCP/IP connectivity is allowed, using either static or dynamic XCF definitions.
- XCFINIT=DEFINE is the default for pure Subarea nodes.

Important: Pure subarea nodes will automatically try to join the ISTXCF group. Other down-level APPN nodes may try to establish SNA connectivity; if this is not desired, specify XCFINIT=NO start option.

11.3.8 z/OS Load Balancer Advisor solution

The z/OS Load Balancer Advisor (called *Advisor* in this discussion) is an application that communicates with outboard load balancers (LBs) and one or more z/OS Load Balancing Agents (called *Agents* in this discussion).

The purpose of the z/OS Load Balancing Advisor and z/OS Load Balancing Agents is to provide information to an outboard load balancer (such as a Cisco Content Switching Module (CSM)) about the availability of various resources (applications) and their relative ability to handle additional workload with respect to other resources that have the ability to handle the same workload. The outboard load balancer takes data that the advisor passes to it and makes a determination about where to route new workloads. This load balancing solution is dissimilar to existing load balancing solutions such as Sysplex Distributor and Cisco multi-node load balancing (MNLB) in that in this implementation, the actual decision of where to route work is made outside of the sysplex.

Note: z/OS Load Balancing Advisor is new to z/OS V1R7 Communications Server. It is also available in several previous releases through APARs.

Several existing IP load balancing solutions already exist, such as:

- External load balancing solutions with little sysplex awareness
- Sysplex-aware external load balancing solutions:
  - Network Dispatcher
- Internal load balancing solutions:
  - Sysplex Distributor
  - DNS/WLM

Attention: The z/OS Load Balancing Advisor and Agents implement a load balancing solution that should not be mixed with Sysplex Distributor for the same workload. Both load balancing solutions can exist within the same sysplex simultaneously; however, both should not attempt to balance the same workload.

Consider the following topics when choosing a load balancing solution:

- Administration and configuration
- Support for TCP and UDP applications
- Extra network flows
- Client/server affinities supported
- Network address translation required
- Use z/OS network QoS policy
- Use WLM recommendations
- Availability of load balancing when load balancing component is unavailable
- Caching
- Server health information available
- Sysplex-aware

11.3.9 Server/application state protocol (SASP)

The z/OS Load Balancing Advisor is a key component that allows any external load balancing solution to become sysplex aware. SASP is the protocol used between load balancers and workload managers. SASP provides a mechanism for workload manager to give distribution recommendations to load balancers, as shown in Figure 11-30 on page 251. Among the characteristics of SASP are the following:

- Little implementation complexity
- Little processing overhead
- Little additional user configuration
- Extendible
- SASP will not handle the transport or the actual distribution of work, only give recommendations

- Open protocol: currently being pursued as an individual Internet Draft RFC submitted to the IETF (current draft)

Load Balancer

The Load Balancer uses SASP to register members it is interested in load balancing. Load Balancer registers groups of clustered servers it is interested in load balancing. Each group designates an application cluster to be load balanced.

Each group consists of a list of members (target servers):

- **System-level cluster** - A list of target systems identified by IP address (in lieu of individual application servers). No specific target application information is returned in this case.
- **Application-level cluster** - A list of applications comprising the “load balancing” group. Applications are identified by protocol (TCP/UDP), an IP address of the target system they reside on, and the port the application is using.

  SASP allows for target servers in a load balancing group to use different ports (and even different protocols such as TCP/UDP), which is probably not applicable for real application workloads.

SASP supports both a “push” and a “pull” model for updating the Load Balancer with workload recommendations such as:

- Support of either by the Load Balancer is implementation dependent.
- Load Balancer tells global workload manager (GWM) which model it wants to use, as follows:
  - **Pull model** - GWM “suggests” a polling interval to the Load Balancer. The z/OS Load Balancing Advisor uses the configurable update_interval value for this purpose. The Load Balancer has the option to ignore this value or request updates for each polling interval.
  - **Push model** - GWM sends updated information to the Load Balancer on an interval basis. The z/OS Load Balancing Advisor uses the configurable update_interval value for this purpose. GWM may send data more frequently than the interval period.

Products that support SASP

The following list enumerates the products that support SASP:

- SASP global workload managers (GWMs)
  - EWLM (enterprise workload manager) - part of IBM Virtualization Engine 1.0 Supported Platforms:
    - IBM AIX® 5L™ Version 5.2
    - Microsoft® Windows® 2000 Advanced Server, 2000 Server, 2003 Enterprise Edition, 2003 Standard Edition
    - Sun™ Microsystems™ Solaris™ 8 (SPARC Platform Edition), 9 (SPARC Platform Edition)
- z/OS Load Balancing Advisor
  - Part of z/OS Communications Server (z/OS V1R4 and higher)
- Load balancers
  - CISCO CSM level 4.1 (2.5)

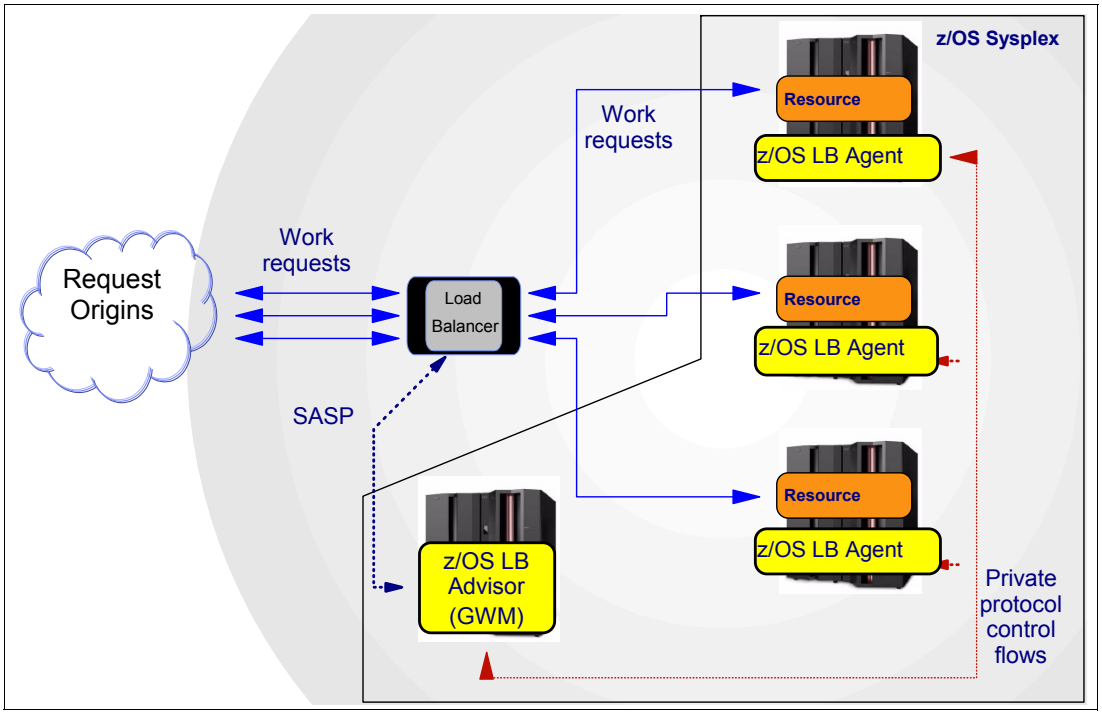


Figure 11-30 SASP with Load Balancing Advisor

11.3.10 z/OS load balancing implementation

z/OS Load Balancing (LB) Advisor acts as a global workload manager. It allows external load balancers to obtain workload balancing recommendations from z/OS (target- or server-specific) and TCP/IP health monitor statistics.

Load Balancer

Load Balancer determines which server instance in the sysplex should receive new workload using information from the Advisor. Workloads arrive from clients and workload balancing is performed to z/OS applications according to server availability, system capacity, and server ability to handle new workloads. CISCO support for SASP is available on their content switching module (CSM), and requires CSM level 4.1 (2.5). The Load Balancer has the following characteristics:

- Group definition rules and guidelines:
  - Multiple groups may be defined representing different server clusters.
  - All Members of a group must belong to the same sysplex; hence, members of the same group must all be managed by the z/OS Load Balancing Advisor or EWLM. A group may not be managed by both.
  - A group may not contain mixtures of application members and system members.
  - Clients connect to the cluster IP address of the group.
- Member definition rules and guidelines:
  - IP addresses should be VIPA addresses for availability reasons and should *not* contain the following types of addresses:
    - Distributed DVIPAs
    - “Deprecated” IPv6 addresses

- Addresses that are not unique within the sysplex
- Addresses that would not be reachable from the load balancer, including loopback addresses and unavailable IPv6 addresses

z/OS Load Balancer Advisor

This application collects information from agents using a private protocol, and communicates with external load balancers using the SASP protocol and provides them with information about z/OS resources. Only one instance per sysplex can be started. z/OS Load Balancer Advisor can be started as follows:

- New, stand-alone application:
  - Started via an MVS started task
  - Accepts MVS operator commands for display and modification purposes

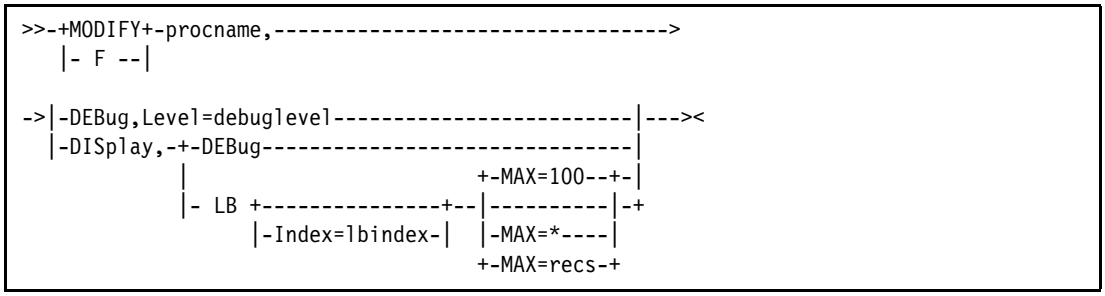


Figure 11-31 LB advisor console command syntax

z/OS Load Balancer Advisor executes on any system within the sysplex and has the following characteristics:

- Provides load balancing advice for any TCP/UDP server applications within the sysplex
- Acts as a TCP server application supporting SASP (port 3860 by default, but can be customized)
- Supports multiple LBs concurrently
- Only one active instance allowed per sysplex
- Does not require Sysplex Distributor to be configured
- Communicates with local Load Balancing Agents
- Uses TCP connections, acts as TCP server (on separate port from SASP)
- Obtains server topology information and workload balancing recommendations from each target system and for each target application

z/OS Load Balancer Advisor configuration

The configuration parameters shown in Figure 11-32 have the following definitions:

agent_connection_port Specifies the port the Advisor should listen on for connections from agents.

agent_id_list Specifies which agents are allowed to connect to the Advisor.

debug_level (optional) Specifies the level of debug information that will be logged. Default 7 (Error, Warning, Event).

lb_connection_v4/v6 Specifies the local IPv4 or IPv6 address and port the advisor should listen on for connections from load balancers. Use both to listen for either type of connection. Default port value is 3860.

lb_id_list Specifies remote addresses from which load balancers are allowed to connect to the Advisor.

port_list (optional) Specifies a list of ports and the type of WLM server recommendation that should be used for each. Overrides value from the wlm statement on a port basis. Only valid for V1R7.

update_interval (optional) Specifies how often agents update the Advisor with new information. May also determine how often the Advisor updates the load balancer if the load balancer supports the “push” flag. Default is 60 seconds.

wlm (optional) Specifies the default type of WLM recommendations that will be attempted; the choices are serverwlm and basewlm. The default is basewlm and is only valid for z/OS V1R7 CS.

```
debug_level             15    # Error, Warning, Event, Info
update_interval         120   # Agent updates every 2 minutes
lb_connection_v4        10.67.5.1..3860   # DVIPA
lb_id_list
{
  10.67.1.11                              # SDBAV4
}
agent_connection_port   8100
agent_id_list
{
  10.67.1.1..8000                         # SD1AV4
  10.67.1.2..8000                         # SD2AV4
  10.67.30.22..8000                       # SD2A2V4
  10.67.1.10..8000                        # SDAAV4
}
wlm serverwlm           # Request server-specific WLM weights
port_list
{
  21  wlm basewlm       # Use system WLM weights for FTP
}
```

Figure 11-32 LB advisor configuration definition

Important: If the load balancer has multiple source IP addresses it can use, make sure lb_id_list contains the address the Load Balancer will use as a source IP address when connecting as a SASP client. If your load balancer-to-advisor connection is failing, examine the Advisor log for a message with the text, `'Unauthorized connection attempt from <ip_address>'`. If this message is present, <ip_address> is the address the load balancer is using as a source IP address for connecting to the Advisor. Insert this address into the lb_id_list statement, restart the advisor and reconnect from the load balancer.

For CISCO CSM, this is the client VLAN interface IP address, not the server VLAN IP address.

If the advisor system fails, the best action is to restart the advisor in another system. This is the main reason for using a unique application-instance DVIPA.

z/OS Load Balancer Agent

The z/OS Load Balancer Agent collects information about z/OS systems and applications and sends it to the Advisor for aggregation using a private protocol. There is one instance per MVS system. The Agent is a new stand-alone application that is started via an MVS started

task. It accepts MVS operator commands for display and modification purposes and has the following characteristics:

- Executes on every target system in the sysplex or at least on every system in the sysplex that is a target of a load balanced request.
- Provides load balancing advice for specified TCP/UDP server applications on local system.
- Only one active instance allowed per MVS system.
- Supports multiple TCP/IP stacks and all known server types such as stack-affinity, generic, bind-specific, and shareport groups.
- Computes weights based on WLM, server availability, server health (dropped connections due to backlog queue full or dropped datagrams due to UDP queue limit exceeded).
- Uses server-specific WLM weights (V1R7) or system WLM weights.

z/OS Load Balancer Agent configuration

There are two aspects of Advisor configuration, the Advisor configuration file itself, and the underlying PROFILE.TCPIP changes that go along with the remainder of the z/OS Load Balancer Advisor system configuration.

The z/OS Load Balancer Agent, shown in Figure 11-33, is defined as follows:

advisor_id Specifies the remote IP address and port of the Advisor this Agent will communicate with.

host_connection Specifies the local IP address and port the Agent will bind to for communications with the Advisor.

debug_level (optional) Specifies the level of debug information that will be logged.

```
Agent #1:

debug_level          15    # Error, Warning, Event, Info
advisor_id           10.67.5.1..8100   # DVIPA
host_connection      10.67.1.2..8000   # SD2AV4

==================

Agent #2: same as above, except:

host_connection      10.67.30.22..8000
```

Figure 11-33 Agent configuration definition

The relationship between the z/OS Load Balancer Advisor and the Load Advisor Agent is shown in Figure 11-34.

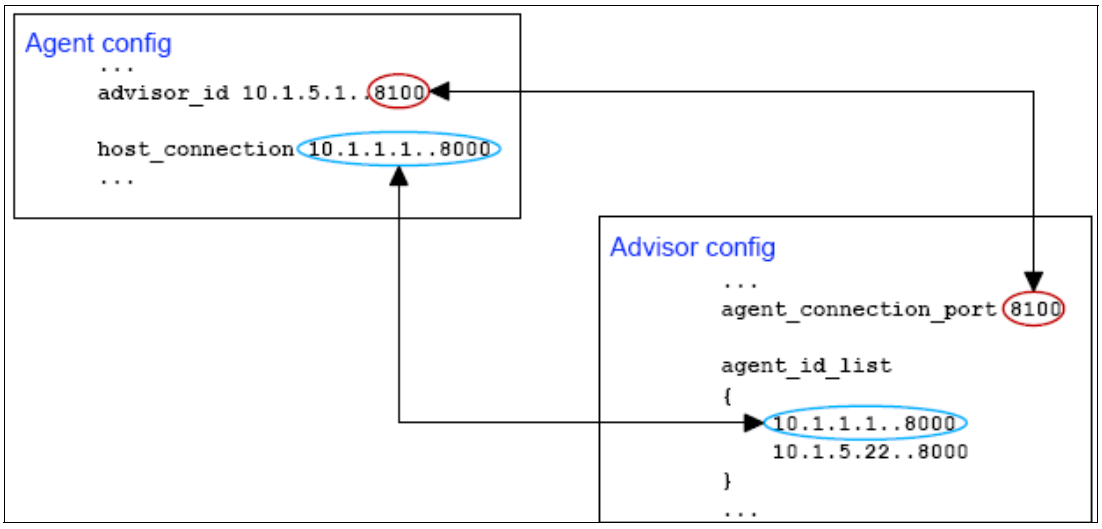


Figure 11-34 Configuration relationships between Agent and Advisor

Important: The IP address specified in the advisor_id statement can be any IP address belonging to the TCP/IP stack the advisor is running on. However, it is recommended that this be a unique application-instance DVIPA.

Figure 11-35 shows a typical configuration for the z/OS Load Balancers.

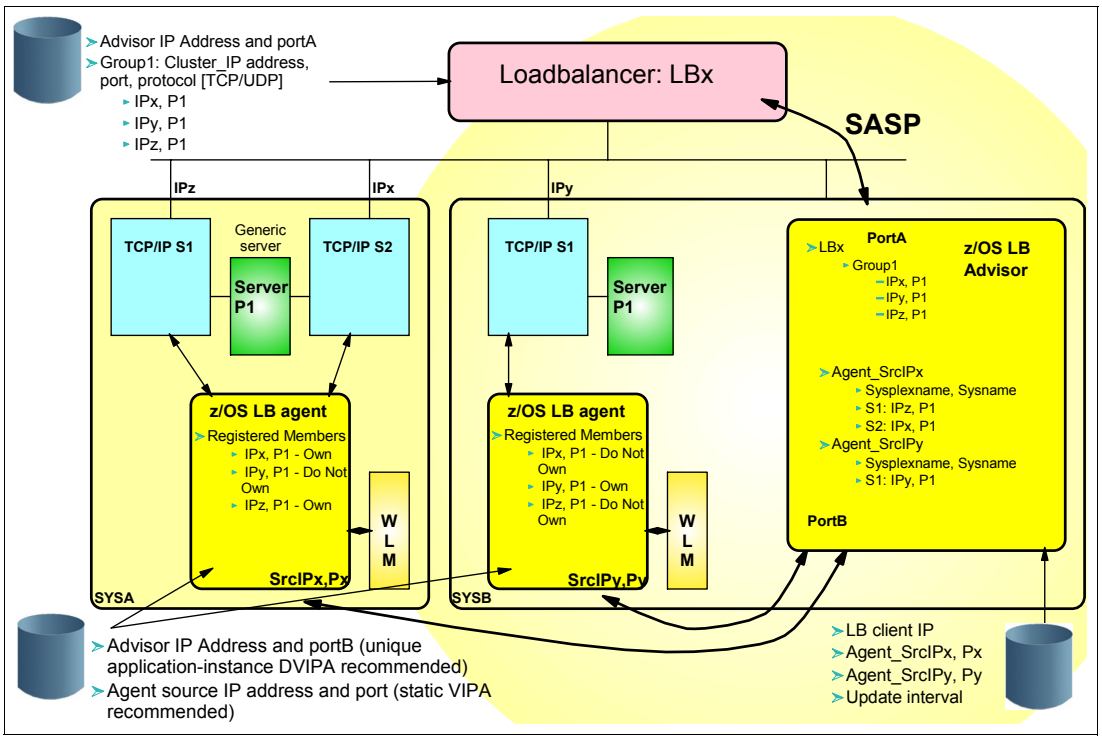


Figure 11-35 z/OS load balancing

Note: Figure 11-35 does not show an Agent on the system where the Advisor is located, but this can be done and is required if applications on that system want to be targets of workload distribution.

11.4 Recovery scenarios for Advisor and Agent

For recovery actions to take place, ensure that automation is in place to be able to restart the Advisor. An automatic restart can be on the same or another system in the sysplex in cases of failures of either the Advisor or the system. An automatic restart of the Agent takes place in the same system when the agent terminates abnormally.

Only one Advisor can be active per sysplex, and only one Agent per system. Internal checks will prevent the starting of multiple Advisors (within the sysplex) or multiple Agents within the same system. The Advisor can be started, or restarted, while Agents are already running, and vice versa.

Automatic Restart Management

Automation can be accomplished with an Automatic Restart Management (ARM) policy or other automation. The Advisor registers with ARM using the following values:

```
ELEMTYPE=SYSTCPIP
ELEMNAME=LBADV
TERMTYPE=ALLTERM
```

This indicates that the Advisor should be restarted only on the same system in cases where the Advisor itself fails, and also restarted on a different system if the system fails. Using an ARM policy, you can indicate which systems are eligible for running the Advisor in the case of system failures. You also need to ensure that the specified backup systems have all the necessary configuration in place to enable the Advisor to be restarted there.

Note: While AUTOLOG can be used to start the Agent, it can *not* be used to monitor the availability of the Agent after initial startup. You should place the Advisor in the AUTOLOG statement list to ensure that it is started when TCP/IP is started on that system. However, you should specify the NOAUTOLOG parameter on the PORT reservation statements for the Advisor ports in the TCP/IP profile. This prevents TCP/IP from monitoring and attempting to restart the Advisor, as that could interfere with your automation logic or the ARM policy that you have put in place.

11.4.1 Advisor failure

Consider the situation illustrated in Figure 11-36, in which the Advisor fails. If the Advisor or its underlying system were to fail, the load balancer might continue to distribute workload requests according to the last set of information received from the Advisor, it might resort to preconfigured weights, or it might even stop distributing new work requests to the cluster. Therefore, it is important that the Advisor be restarted as soon as possible when a failure occurs, so that it can begin communicating with the load balancer and workload request distribution can resume normally. This restart capability should cover scenarios where the Advisor itself fails, and where the system that the Advisor is running on fails.

The Advisor can run on any system in the sysplex and thus can be restarted on any system in the sysplex, as long as it is configured to use dynamic VIPAs and dynamic routing is in effect.

Recommendation: Make the Advisor and Agent automatically restart. Configure the Automatic Restart Manager or other automation software to restart the Advisor on any MVS system in the sysplex, and to restart the Agent on the same MVS system as it was previously running.

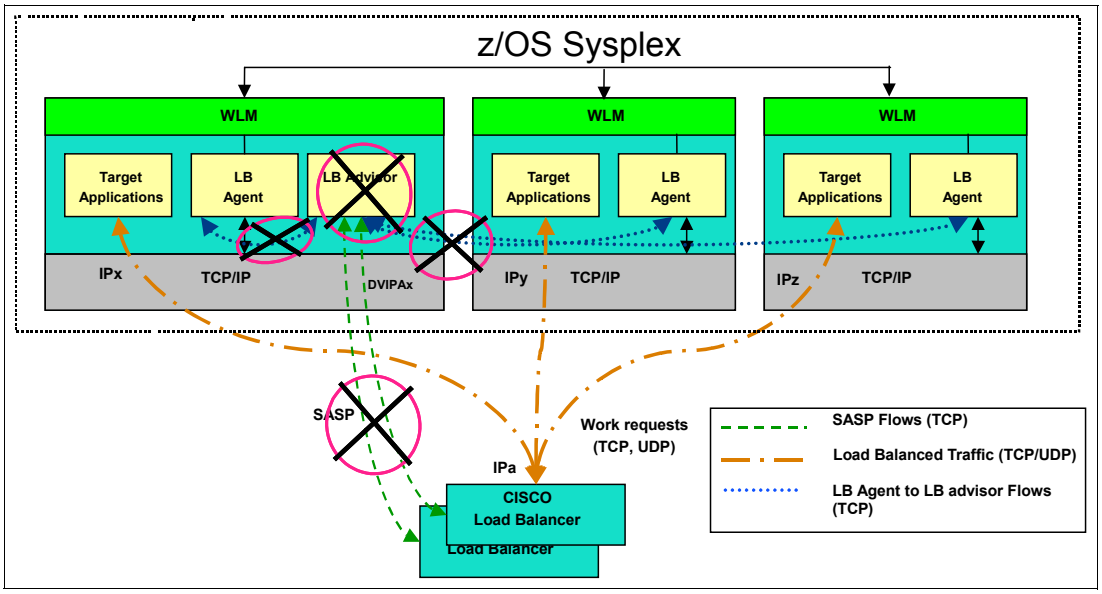


Figure 11-36 Recovery scenario if the Advisor fails

Advisor recovery

Figure 11-37 on page 258 shows a recovery scenario for the Advisor. The recovery steps are as follows:

- Restart the LB Advisor with one of the following methods:
  - Automatic Restart Management (ARM)
  - TCPIP AUTOLOG processing
  - Automation
- LB reconnect (new TCP connections)
  - Re-registers groups/servers
  - Begins polling again
- LB Agents reconnect

The LB Advisor rediscovers server topology and weights.

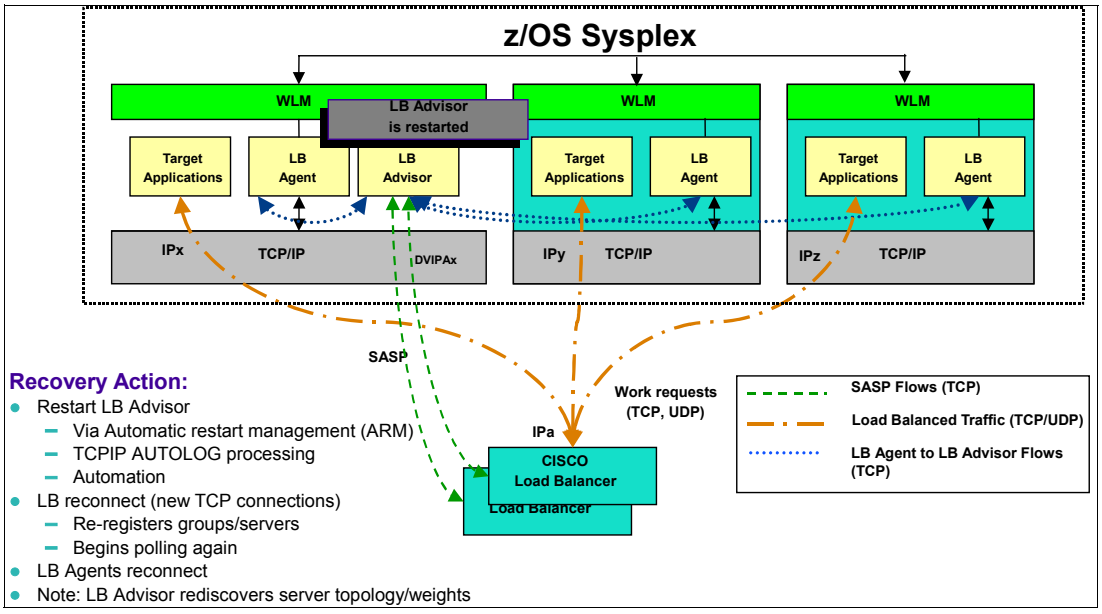


Figure 11-37 Advisor recovery scenario restart

11.4.2 TCP/IP stack on Advisor system fails

When the TCP/IP stack on the Advisor system fails, restarting the TCP/IP stack and LB Advisor is the best option. LB and the LB Agents reconnect. Figure 11-38 is an example of this situation.

Some special considerations exist for scenarios where ARM is used and the TCP/IP stack address space terminates, as the result of a failure or of a planned operation. When the TCP/IP stack becomes unavailable the Advisor also terminates, since it can no longer establish any TCP/IP communications. An ARM restart of the Advisor will likely fail because the TCP/IP protocol stack will not be available when the restarts occur.

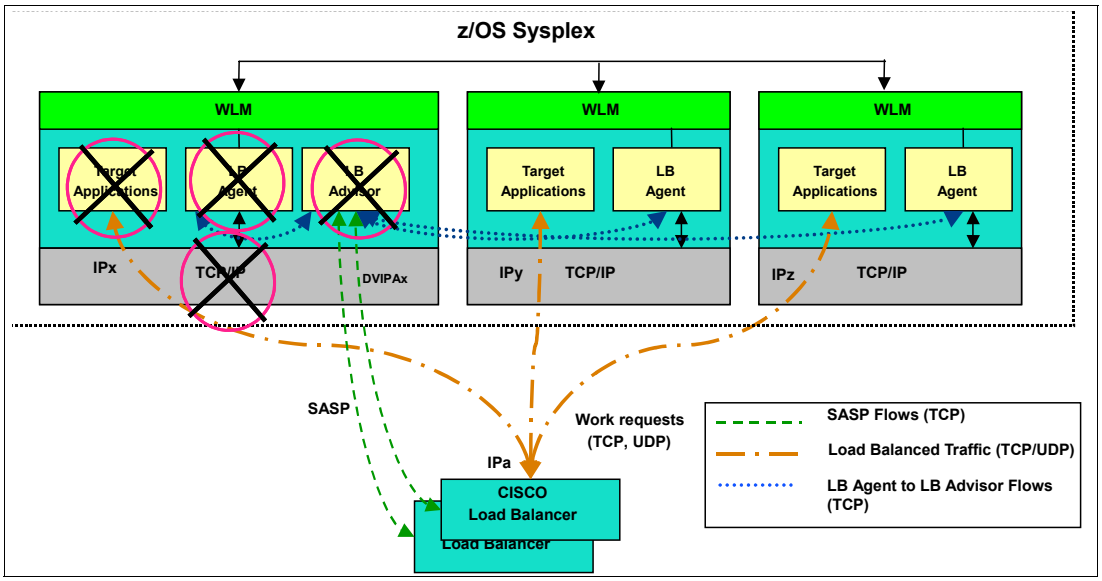


Figure 11-38 TCP/IP stack fails

TCP/IP stack recovery

You can handle the recovery in the following ways:

- For planned outages of the TCP/IP stack, manually start the Advisor on another system as soon as the Advisor terminates on the system where TCP/IP is stopped.
- For unplanned outages of the TCP/IP, ensure that an ARM policy (or other automation) is in place to quickly restart the TCP/IP stack on the same system, as shown in Figure 11-39. The Advisor also needs to be quickly restarted on the same system. This can be done by using an automation software package, or by using the TCP/IP profile AUTOLOG statement.
  - You should place the Advisor in the AUTOLOG statement list to ensure that it is started when TCP/IP is started on that system.
  - However, you should specify the NOAUTOLOG parameter on the PORT reservation statements for the Advisor ports in the TCP/IP profile. This prevents TCP/IP from monitoring and attempting to restart the Advisor since that could interfere with your automation logic or the ARM policy that you have put in place.

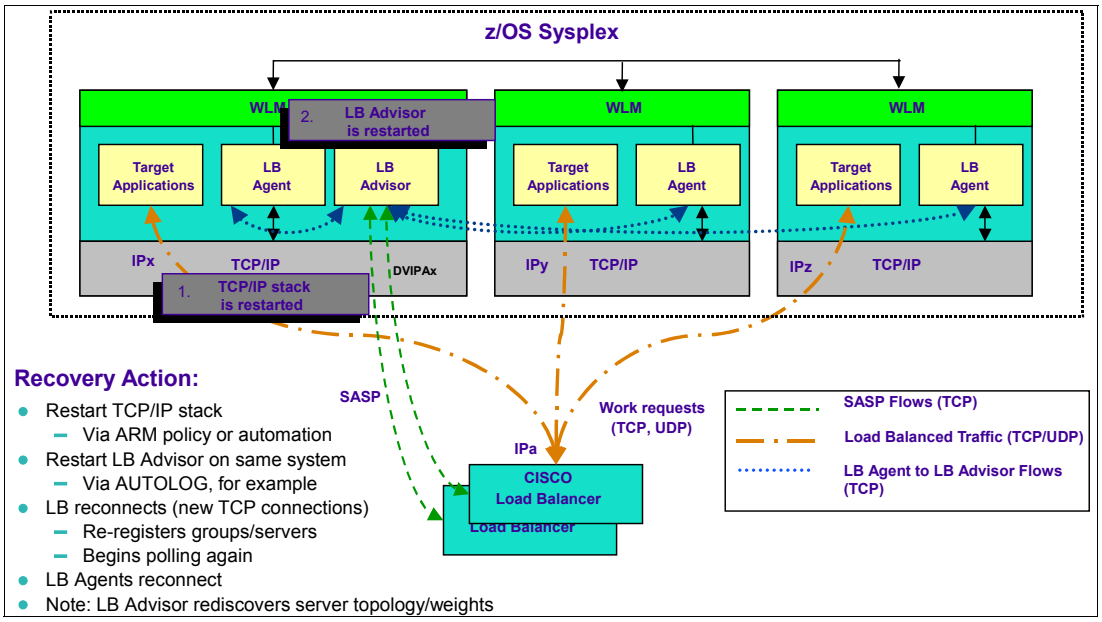


Figure 11-39 Recovery for TCP/IP stack failure

11.4.3 Advisor system fails

The Advisor should be restarted on a different system if the system fails, as shown in Figure 11-40 on page 260. Using an ARM policy, you can indicate which systems are eligible to run the Advisor in the case of system failures. You also need to ensure that the specified backup systems have all the necessary configuration in place to enable the Advisor to be restarted there.

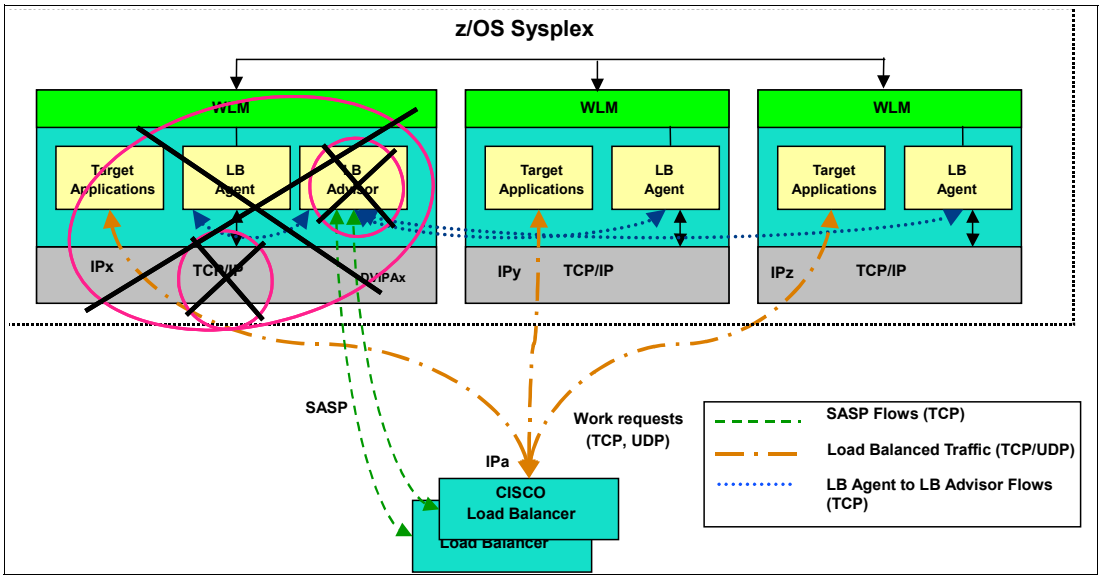


Figure 11-40 System fails where the Advisor is running

Advisor system failure recovery

Figure 11-41 shows the recovery actions for a system that fails that is running the Advisor.

DVIPA associated with the LB Advisor moves to the new system. This is the main reason for recommending a unique application-instance DVIPA for the LB Advisor.

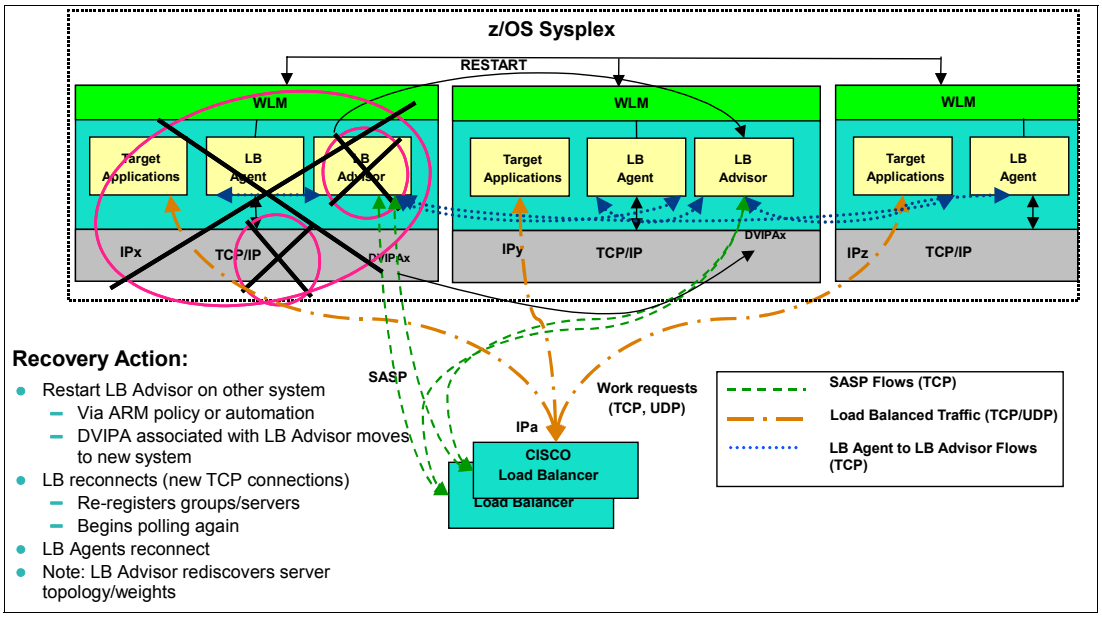


Figure 11-41 Recovery for the Advisor when the system fails

11.4.4 Advisor not responding

When the Advisor is suddenly not responding, as illustrated in Figure 11-42, the reason for this may be one of the following:

- Network connectivity loss
- TCP/IP stack is not healthy

- System itself is not healthy
- LB Advisor is active but not healthy

Advisor not responding recovery actions

Depending on the particular problem and the LB configuration, the potential recovery actions are affected. Make sure the LB probes the Advisor periodically using SASP. If the Advisor does not respond, the following happens:

- LB reverts back to its own load balancing algorithms, which defaults to round-robin.
- LB terminates the connection with LB Advisor and notifies the administrator.
- Periodically, an attempt is made to reconnect to LB Advisor.
- When a reconnect is successful, LB resumes using LB Advisor's weights.

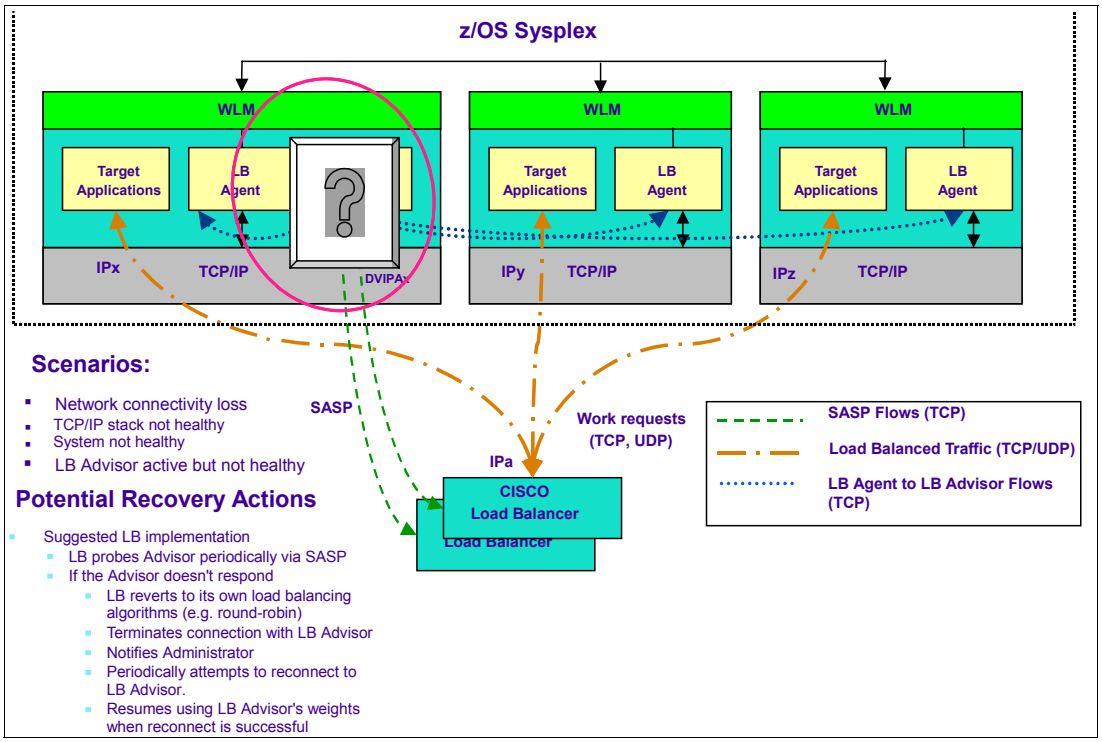


Figure 11-42 Advisor not responding recovery actions

11.4.5 Agent not responding

When the Agent is suddenly not responding, as illustrated in Figure 11-43 on page 262, the reason for this may be one of the following:

- Network connectivity loss
- TCP/IP stack is not healthy
- System itself is not healthy
- LB Agent is active but not healthy

Agent not responding recovery actions

The LB Advisor monitors the "health" of the LB Agent by periodically monitoring the "heartbeat." If the LB Agent is not responding, the operator is alerted.

The TCP connection is then severed to the LB Agent. This loss of connectivity is conveyed to the LB; none of the target resources in the target system can be contacted.

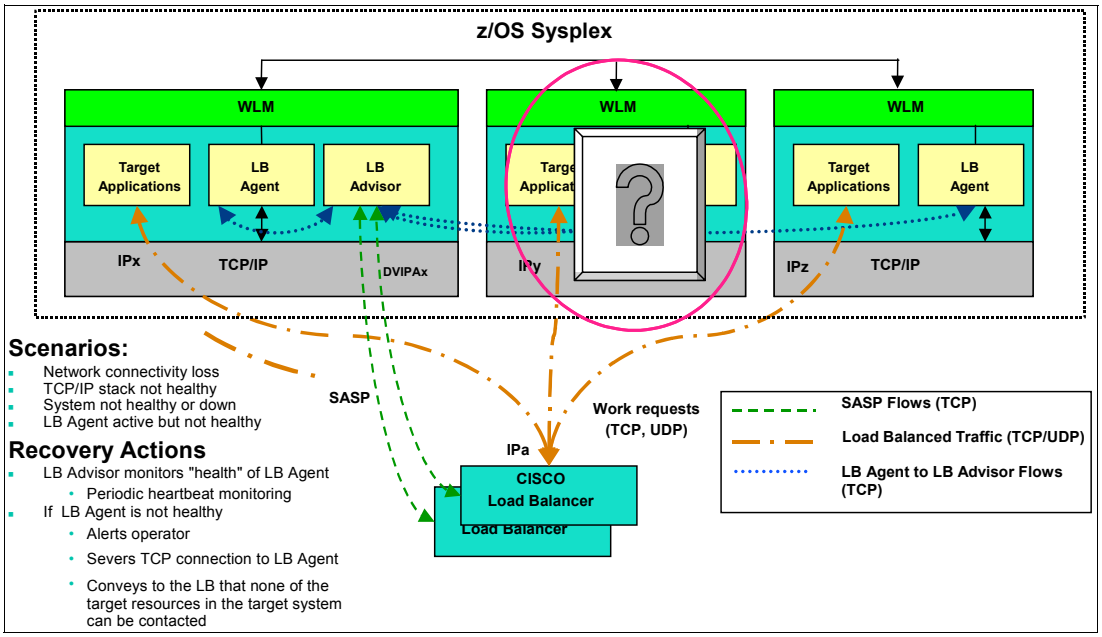


Figure 11-43 Agent not responding recovery actions

11.4.6 Target application fails

When a target application fails, the LB Agent notifies the LB Advisor and then no longer forwards requests to the target, as shown in Figure 11-44.

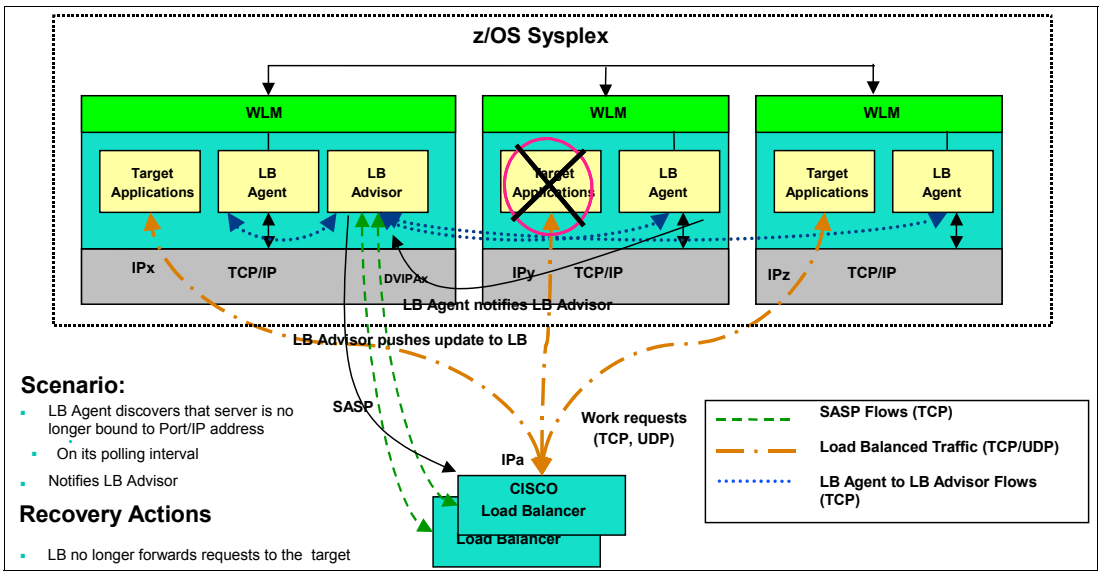


Figure 11-44 Target application fails

11.4.7 Network connectivity loss between LB and target application

There is the possibility that the following scenario can take place:

- A network connectivity loss can occur and there are no redundant paths to target systems.
- The LB Advisor may still believe the target system is reachable and healthy.

Recovery actions

The LB should be configured to perform IP layer health probes to the target systems.

If a target system is not reachable, the LB should suspend forwarding requests to that system.

IP layer health probes should continue to be issued so that distribution to the target system can once again resume if connectivity to that system is restored.

The possible scenarios and their potential recovery actions are shown in Figure 11-45.

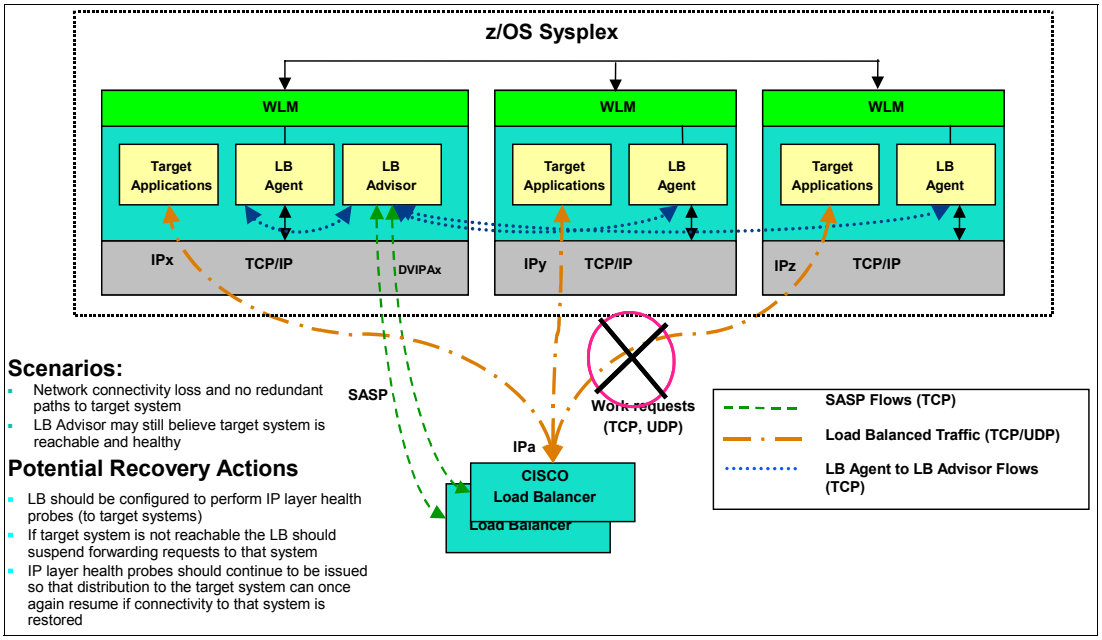


Figure 11-45 Loss of connectivity between the LB and target applications

11.4.8 Migration and coexistence considerations

There are z/OS Load Balancing Advisor PTFs for this support for previous releases, as follows:

- z/OS V1R4 APAR - PQ90032
- z/OS V1R5 and z/OS V1R6 APAR - PQ96293

The PTFs only support system WLM recommendations; the PTFs do *not* support server-specific WLM recommendations.

There is APAR documentation at the following Web address:

`http://www.ibm.com/support/docview.wss?rs=852&uid=swg27005585`

Server-specific WLM recommendations

The following support is provided:

- z/OS V1R7 supports server-specific WLM recommendations.
- Previous releases use system WLM recommendations exclusively.

- Server-specific versus system WLM recommendations are determined on a Group basis, as follows:
  - Server-specific WLM recommendations are used when all of the following are true:
    - Advisor is V1R7.
    - All members of a group are owned by V1R7 Agents.
    - Group only contains application members (rather than system members).
  - Otherwise, system WLM recommendations are used for the group.
  - Server-specific WLM will usually result in better workload distribution, except when servers serve as access points to applications which run in their own address space. Therefore, they should use a different WLM service class (TN3270, FTP, and INETD)

IPv6 considerations

If using IPv6 and DVIPA for the Advisor-Load Balancer connection or an Advisor-Agent connection, movement of the Advisor will be limited to those z/OS releases that support IPv6 DVIPAs (V1R6 and higher).

EWLM considerations

A group defined to a Load Balancer may not contain a mix of members managed by EWLM and z/OS Load Balancing Advisor. All members of a group must be managed by one or the other.

Sysplex Distributor considerations

Sysplex Distributor may coexist with z/OS Load Balancing Advisor since each would manage a workload to disjoint sets of target applications. However, nothing prevents you from using both for the same workload.

A possible scenario is to test one workload distribution methodology while the other is currently deployed. This requires that target applications must bind to INADDR_ANY instead of the DVIPAs configured for Sysplex Distributor. Only Sysplex Distributor should use the target application DVIPAs. For example, do not code the DVIPAs Sysplex Distributor uses in the members defined to the external load balancers. Coding the distributed DVIPA in a member in a load balancer would result in the workload being distributed to the distributed DVIPA being redistributed by Sysplex Distributor. This is not recommended.

Swappable versus non-swappable considerations

z/OS V1R7 runs non-swappable by default. Prior to z/OS V1R7, customization is required to run non-swappable.

11.5 TCP/IP IPv6 enhancements

z/OS V1R4 Communications Server introduced support for both IPv4 and IPv6 IP addresses for certain functions and applications. IPv6 is the next generation in the IP networks. Following is the evolution up to and including z/OS V1R7:

- z/OS V1R4 Basic IPv6 Support:
  - TCP/IP Stack protocol layers
  - Socket APIs - USS/LE and TCP/IP
  - Connectivity with OSA Express QDIO
  - Resolver and DNS

- FTP and UNIX applications
- Netstat and messages

- z/OS V1R5 and above:
  - Platform-specific with latest standards
  - CICS Sockets
  - Enable SNA applications over IPv6:
    - Enterprise Extender
    - Tn3270
  - Connectivity with support for point-to-point links (MPC, XCF, SAMEHOST)
  - OMPRoute support for RIPng
  - QOS Policy Agent (differentiated services) and SLAP MIB
  - MVS applications (remote execution applications)
  - SNMP standard TCP/IP MIBs (network management)
  - Additional SMF records
  - More applications (sendmail, tftpd, dcas)
- z/OS V1R6
  - Sysplex Functions (DVIPA, Distributor)
  - OMPRoute support of OSPFv3
  - SNMP Enterprise-specific MIB and standards update
- z/OS V1R7
  - SNMP UDP standard MIB
  - Updates to MVS TCP/IP Enterprise-specific MIB for UDP
  - IPv6 Advanced Socket Options
  - Maintain two IPv6 Routers in Default List

11.5.1 IPv6 advanced socket API

z/OS V1R4 Communications Server provided partial support for the advanced socket API options using the Language Environment, C/C+, and z/OS UNIX Assembler Callable Services APIs. z/OS V1R7 Communications Server adds support for the remaining IPv6 advanced socket API options (as described by RFC3542) to these APIs.

Note: RFC3542 provides an application with the ability to access information in IPv6 headers and extension headers on inbound packets. The application can influence IPv6 headers and extension headers on outbound packets with the network interface selection and routing path. The basic set was implemented in Communications Server in V1R4 as RFC2292, and now is superseded by RFC3542.

IPv6 socket API functions

These IPv6 socket API functions are geared towards advanced IPv6 applications that require access to low-level IPv6 protocol information and control the following:

- Access to information in IPv6 headers and extension headers on inbound packets.

- Ability to influence the contents of IPv6 headers and extension headers on outbound packets. This includes the ability to influence the network interface selection and network routing path for outbound IPv6 packets.
- Access to detailed path maximum transmission unit (MTU) information.

These advanced socket API functions are very flexible and quite powerful since they provide applications with direct control over the content of IPv6 packets leaving the z/OS host. As a result, access to these advanced socket API functions is limited to authorized applications (APF authorized or z/OS UNIX System Services Superuser authority) or applications that have access to security profiles defined in RACF or an equivalent external security manager.

11.5.2 Application use of APIs

The SERVAUTH profiles used by TCP/IP provide the ability to control whether an application is permitted to set IPv6 advanced socket API options. IPv6 advanced socket APIs let applications modify and receive information about packets, as follows:

- Control and modify outbound packet information, such as:
  - First hop address and routing headers
  - Packet fragmentation
  - MTU discovery
- Receive inbound packet information, such as:
  - Arriving interface
  - Destination IP address
  - Hop limit
  - Source routing
  - IPv6 options (routing headers, destination options, etcetera) set by the sender

New IPPROTO_IPv6 support

Table 11-2 on page 267 shows the set of the options newly provided with z/OS V1R7.

Note: For the IPV6_NEXTHOP, IPV6_TCLASS, IPV6_RTHDR, IPV6_HOPOPTS, IPV6_DSTOPTS, IPV6_RTHDRDSTOPTS, and IPV6_PKTINFO socket options, to set the socket option on setsockopt() or to use the corresponding ancillary data item on sendmsg(), an application must meet one of the following criteria:

- Be APF authorized.
- Have superuser authority.
- The corresponding SERVAUTH resource name is defined, and the application has at least READ access to the resource (see Table 11-3 on page 273).

Table 11-2 IPv6 Advanced Socket API options provided with z/OS V1R7

Option name	Data path	Transport supported
IPV6_HOPOPTS	Outbound	UDP, RAW
IPV6_RECVHOPOPTS	Inbound	UDP, RAW
IPV6_RTHDR	Outbound	UDP, RAW
IPV6_RECVRTHDR	Inbound	UDP, RAW
IPV6_RTHDRDSTOPTS	Outbound	UDP, RAW
IPV6_DSTOPTS	Outbound	UDP, RAW
IPV6_RECVDSTOPTS	Inbound	UDP, RAW
IPV6_TCLASS	Inbound	TCP, UDP, RAW
IPV6_TCLASS	Outbound	TCP, UDP, RAW
IPV6_NEXTHOP	Outbound	UDP, RAW
IPV6_RECVPATHMTU	Outbound	UDP, RAW
IPV6_PATHMTU	Outbound	UDP, RAW
IPV6_DONTFRAG	Outbound	UDP, RAW

IPv6_HOPOPTS

This option name provides options executed for the packet by every intermediate router, as follows:

- “Hop by hop” options.
- Predefined set defined in RFC.
- Applications can define their own options.
  - Intermediate routers not understanding options will do the following:
    - Discard packet if first two bits are 11 or 10.
    - Ignore all others.
  - LE and C/C++ APIs provide helper functions to build these options.

The IPv6_HOPOPTS options are set as follows:

- Use **setsockopt** to specify the hop by hop options built for all packets.
- Ancillary data on sendmsg to specify the hop by hop options for one packet.
- Zero length data on setsockopt or ancillary data clears the hop by hop options.

The IPv6_HOPOPTS options are queried as follows:

- By **getsockopt**, to find what the current hop by hop options are.

Figure 11-46 on page 268 shows an example application using IPV6_HOPOPTS to reserve buffer space for the next packet.

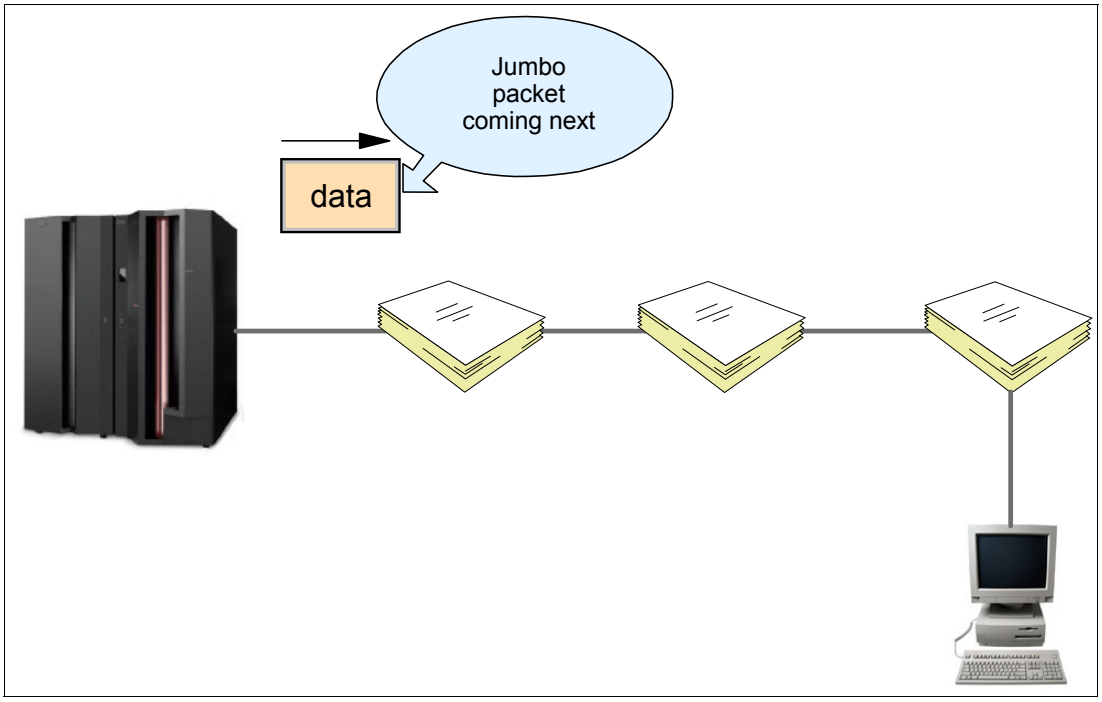


Figure 11-46 Example application using IPV6_HOPOPTS

IPV6_RECVHOPOPTS

This option allows applications to receive the hop options as ancillary data along with the received data. The options are used as follows:

- Set by **`setsockopt`** to specify receiving the hop by hop options for all packets.
- Queried by **`getsockopt`** to determine if you are set to receive the hop by hop options.

IPV6_RTHDR

This option provides source routing in IPv6 that contains the list of hops to go through to get to the destination, as follows:

- Hops need not be directly connected to each other.
- Each hop is considered the next hop as the destination of the packet, as follows:
  - Overrides the routing rules of the stack for that packet.
  - LE C/C++ APIs provide helper functions to build and read a routing header.
- Set by:
  - **`setsockopt`** to specify the route for all packets used by this socket.
  - Ancillary data on sendmsg to specify the route one packet should take.
  - Zero length data on setsockopt or ancillary data clears the routing headers and uses the standard stack routing.
- Queried by **`getsockopt`** to find what the current routing headers are.

Figure 11-47 is an example application using IPV6_RTHDR to redirect large packets without causing fragmentation.

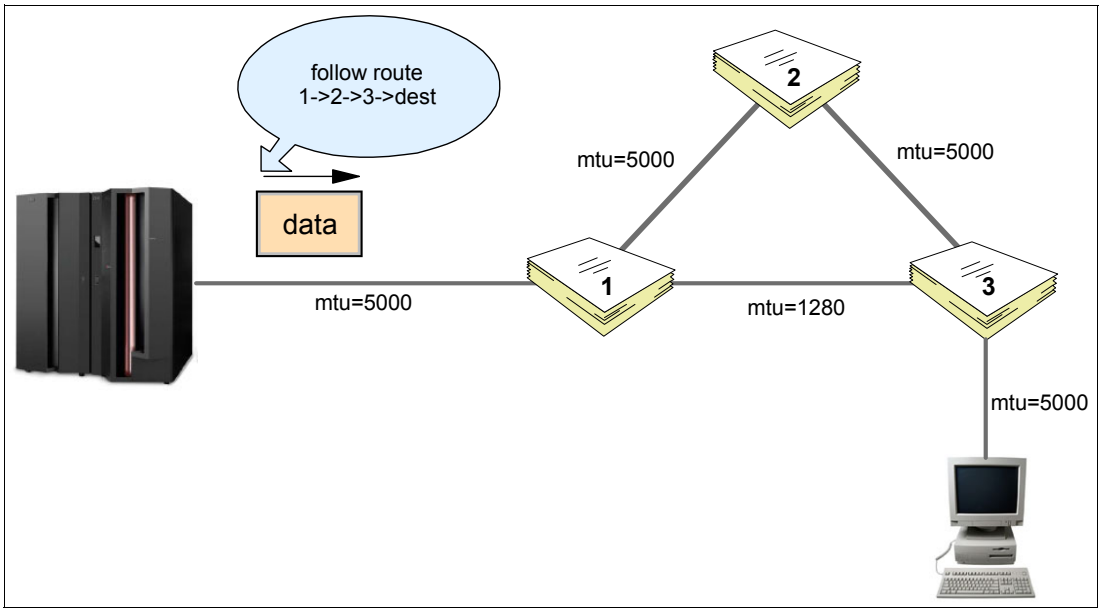


Figure 11-47 Example application using IPV6_RTHDR

IPV6_RECVRTHDR

This option allows applications to receive the routing headers to be used as follows:

- Received as ancillary data along with the received data.
- LE C/C++ APIs provide helper functions to read a routing header.
- Set by **`setsockopt`** to specify receiving the routing headers set for all packets.
- Queried by **`getsockopt`** to find if you are set to receive the routing headers

IPV6_RTHDRDSTOPTS

This option provides options for each intermediate routing header destination as follows:

- Should always be used with IPV6_RTHDR or the option will be silently ignored.
- Similar to setting hop by hop options.
- Set by:
  - **`setsockopt`** to specify the options used by the intermediate destinations used by this socket.
  - Ancillary data on sendmsg to specify the options used by the intermediate destinations used by a single packet.
  - Zero length data on setsockopt or ancillary data clears the intermediate destination options.
- Queried by **`getsockopt`** to find what the routing header destination options are.

Figure 11-48 on page 270 is an example application using IPV6_RTHDRDSTOPTS to RSVP resources along its path.

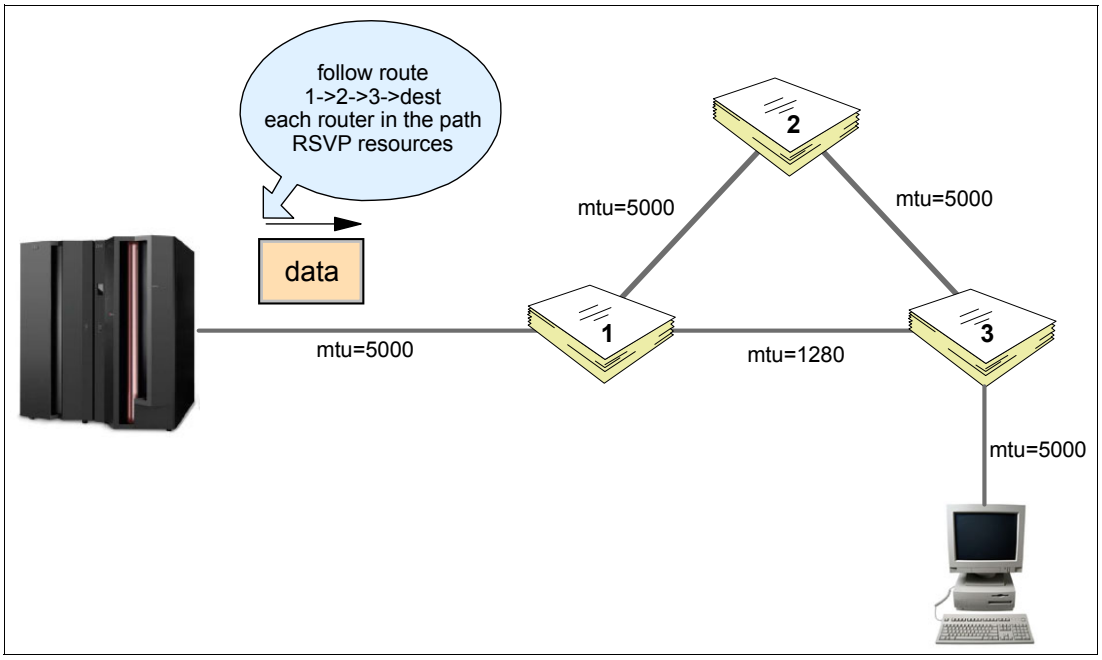


Figure 11-48 Example application using IPV6_RTHDRDSTOPTS

IPV6_DSTOPTS

This option sets options to be performed by the final destination, similar to setting hop by hop options, as follows:

- Set by:
  - **`setsockopt`** to specify the destination options used by this socket.
  - Ancillary data on sendmsg to specify the destination options for one packet.
  - Zero length data on setsockopt or ancillary data clears the destination options.
- Queried by **`getsockopt`** to find what the destination options are.

Figure 11-49 shows an example application using IPV6_DSTOPTS to request the destination to send an ICMP message.

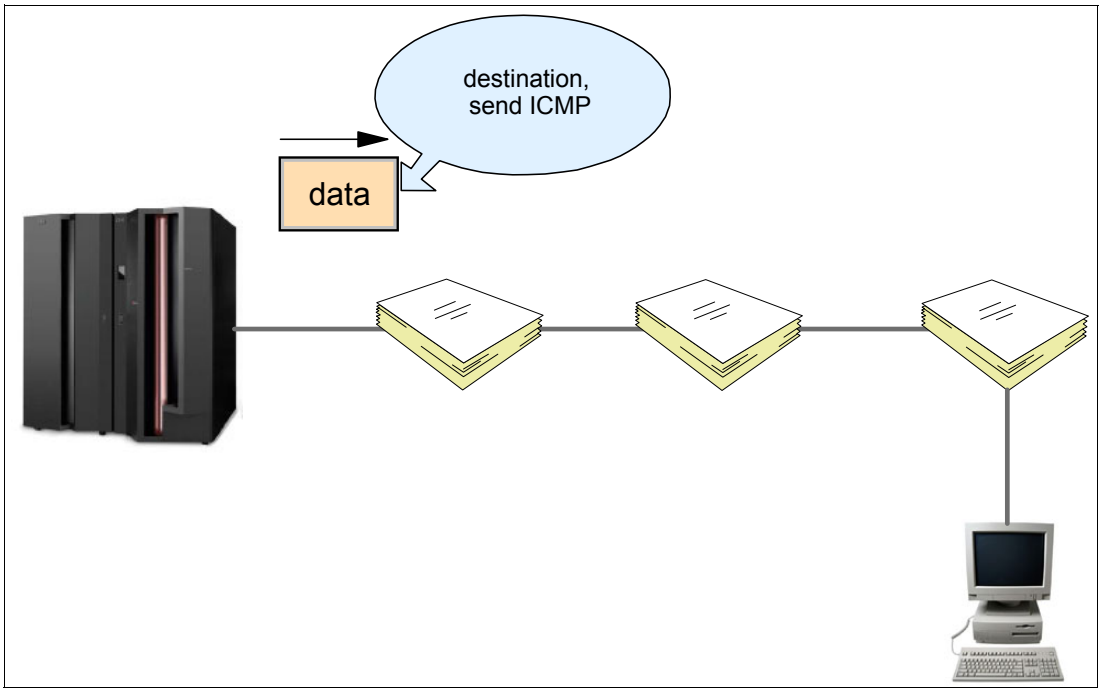


Figure 11-49 Example application using IPV6_DSTOPTS

IPV6_RECVDSTOPTS

This option allows applications to receive the destination options specified in the following ways:

- Receives both destination options and routing destination options; the routing header destination options are followed by a routing header.
- Received as ancillary data along with the received data.
- Set by **setsockopt** to specify receiving the routing headers set for all packets.
- Queried by **getsockopt** to find if you are set to receive the routing headers.

IPV6_TCLASS

This option sets the traffic class for a packet in the following ways:

- Influences quality of service.
- The traffic class set by Policy Agent overrides a traffic class set by an application.
- Is supported by TCP, UDP, and RAW.
- Is set by:
  - **setsockopt** to specify the traffic class used by this socket
  - Ancillary data on sendmsg to specify the traffic class for one packet
- Is queried by **getsockopt** to find what the traffic class is set to.

IPV6_RECVTCLASS

This option allows applications to received the traffic class and received as ancillary data along with the received data, as follows:

- Set by **setsockopt** to specify receiving the traffic class set for all packets.
- Queried by **getsockopt** to determine if you are set to receive traffic class updates.

IPV6_NEXTHOP

This option is useful for applications on multi-homed machines and determines which network to use to get to the destination, as follows:

- Set by:
  - **setsockopt** to specify the next hop used by this socket.
  - Ancillary data on sendmsg to specify the next hop for one packet.
  - Zero length data on setsockopt or ancillary data clears the next hop option and uses the default stack routing.
- Queried by **getsockopt** to find what the next hop is set to.

Figure 11-50 is an IPV6_NEXTHOP example.

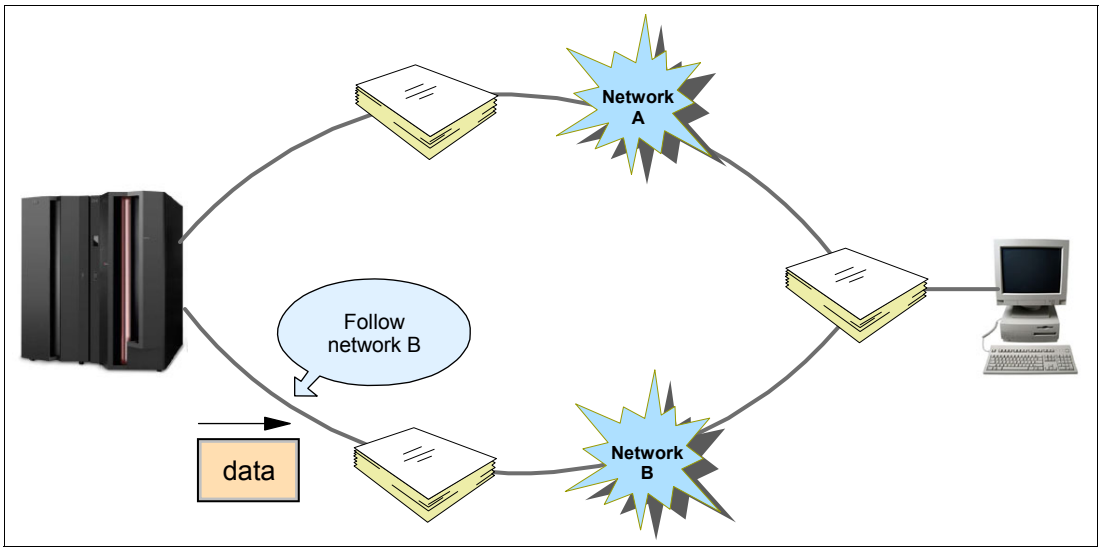


Figure 11-50 IPV6_NEXTHP example

IPV6_DONTFRAG

This option indicates to the stack to not fragment a datagram. It is used in the following ways:

- Overrides the stack default MTU selection policy.
- Used by applications to find the largest unfragmentable packet that can be sent to a destination, as follows:
  - Most often used with IPV6_RECVPATHMTU.
  - Applications start MTU discovery by the following options:
    - Setting IPV6_DONTFRAG and IPV6_RECVPATHMTU
    - Issue **getsockopt** for IPV6_DONTFRAG to get an initial guess of the MTU, which does the following:
      - Sends data grams as large as the best guess.
      - Waits for ICMPv6 packet too large messages.
      - Loops until no ICMPv6 packet too large message is received.
- Set by:
  - **setsockopt** to specify no fragmentation for datagrams used by this socket.
  - Ancillary data on sendmsg to specify no fragmentation for a single datagram.
  - Zero length data on setsockopt or ancillary data uses the stack default MTU selection policy.
- Queried by **getsockopt** to find the stack default MTU selection policy that will be used.

IPV6_RECVPATHMTU

This option allows applications to receive the MTU of a packet, is most often used with IPV6_DONTFRAG, and is received as ancillary data along with the received data. It is invoked as follows:

- Set by **setsockopt** to specify receiving the path MTU for all packets.
- Queried by **getsockopt** to find if you are set to receive the MTU for packets

Migrations concerns

Migration concerns are only pertinent for a Multilevel Security environment. IPV6_PKTINFO changed authorization in an MLS environment. To use the options in a MLS environment, the resource name must be defined and the application must have at least READ access to the resource.

Table 11-3 shows the resource names.

Table 11-3 RACF resource name

API option	RACF Resource Name
IPV6_NEXTHOP	EZB.SOCKOPT.sysname.tcpname.IPV6_NEXTHOP
IPV6_TCLASS	EZB.SOCKOPT.sysname.tcpname.IPV6_TCLASS
IPV6_RTHDR	EZB.SOCKOPT.sysname.tcpname.IPV6_RTHDR
IPV6_HOPOPTS	EZB.SOCKOPT.sysname.tcpname.IPV6_HOPOPTS
IPV6_DSTOPTS	EZB.SOCKOPT.sysname.tcpname.IPV6_DSTOPTS
IPV6_RTHDRDSTOPTS	EZB.SOCKOPT.sysname.tcpname.IPV6_RTHDRDSTOPTS
IPV6_HOPLIMIT	EZB.SOCKOPT.sysname.tcpname.IPV6_HOPLIMIT
IPV6_PKTINFO	EZB.SOCKOPT.sysname.tcpname.IPV6_PKTINFO

11.5.3 Maintain two IPv6 Routers in default List

IPv6 Standards require a minimum of two default routers. In some situations, z/OS CS does not meet this requirement. z/OS V1R7 allows the addition of the default routers back to the routing table when the last default route is deleted from the routing table.

11.5.4 SNMP IPv6 UDP MIB enhancements

SNMP (Simple Network Management Protocol) is a set of standards that enables similar management data from different platforms to be provided to management applications. The management data is defined in files called Management Information Base (MIB) modules. These files are often just called *MIBs*. The data definitions are written in SNMP syntax. Each piece of data is called a *MIB object* and is identified by a name and a dotted-decimal value called an *object identifier* (OID). There are three types of data:

Tables Contain rows of data, each with its own unique index value

Scalars Global pieces of data

Traps Provide information about asynchronous events

z/OS CS SNMP supports management data from the following types of MIBs:

- Standard MIBs, as defined in IETF internet drafts or RFCs.

- Enterprise-specific MIBs, which are proprietary MIBs not reviewed or approved by the IETF.

Figure 11-51 shows the SNMP diagram in z/OS.

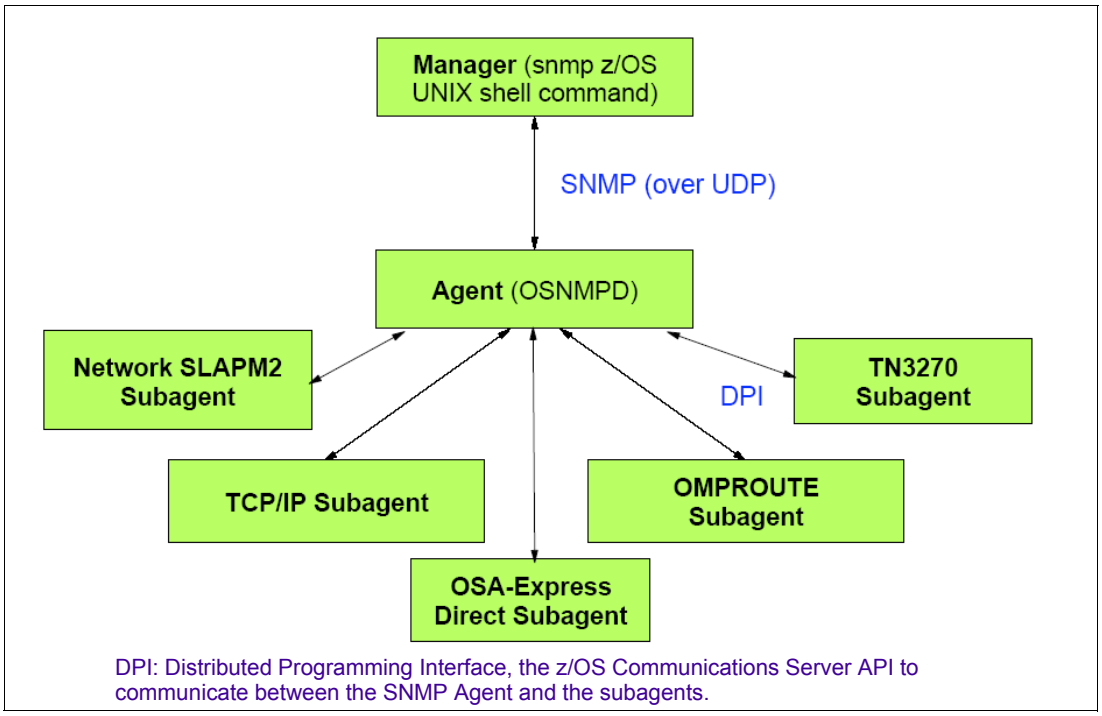


Figure 11-51 SNMP z/OS structure

SNMP history with CS

Following are the CS releases that support SNMP prior to z/OS V1R7:

- z/OS CS V1R5
  - First release to provide SNMP support for IPv6 management data.
  - The SNMP TCP/IP subagent was enhanced to support version-neutral MIB modules from IETF internet drafts. *Version-neutral* means that the MIB modules support both IPv4 and IPv6 data.
  - IP, routing, interfaces, and TCP SNMP data.
- z/OS CS V1R6
  - Support upgraded to match the current versions of the drafts.
  - Added additional data to our IBM MVS TCP/IP enterprise-specific MIB module (IP, routing, interfaces, and TCP SNMP data).

z/OS CS V1R7 SNMP support

This release has enhanced the TCP/IP subagent to support the version-neutral UDP management data in the IETF internet draft version of the UDP-MIB. Data is defined in UDP-MIB from draft-ietf-ipv6-rfc2013-update-03.txt, which was implemented in 4/2004, as follows:

- udpHCInDatagrams/udpHCOutDatagrams:
  - 64-bit UDP transport layer counters
  - Complements existing 32-bit counters

- udpEndpointTable:
  - Provides local/remote IP address and port information for all UDP endpoints

This release added the following version-neutral UDP management data to the TCP/IP Enterprise-specific MIB module:

- ibmTcpipMvsUdpEndpointTable - provides counters and augments the entries in the udpEndpointTable:
  - 32-bit and 64-bit datagram and byte counters
  - Connection ID and resource name
  - Last activity value
  - Socket options
  - Information regarding UDP sockets that are sending multicast data
  - ibmTcpipMvsUdpMcastTable - provides data regarding UDP sockets that are receiving multicast data.

This release also enhanced the following Netstat reports to display the remote IP address and port values for connected UDP sockets:

- ALL/-A
- ALLCONN/-a
- BYTEINFO/-b
- CONN/-c
- SOCKETS/-s

Note: IETF UDP-MIB internet draft shipped with the product because IETF internet drafts expire in six months. The version of the IETF UDP-MIB internet draft supported by z/OS V1R7 is shipped with the product and installed in the HFS in the /usr/lpp/tcpip/samples directory as file udpmib-mi2.

Migration concerns

Network management applications may not support deprecated SNMP UDP management data.

The new UDP-MIB from IETF internet draft deprecates the SNMP table, udpTable.

Deprecated UDP management data from the TCP/IP enterprise-specific MIB is as follows:

- ibmTcpipMvsUdpTable
- ibmTcpipMvsUdpEndpMcastTable

Netstat display of remote IP address and port for connected UDP sockets obtained using automated programs that process Netstat report output may have to be updated.

Note: *Deprecated* status with regard to SNMP MIB objects means that the MIB objects are still supported, but they will either become obsolete in the future, or they have been replaced by better objects. If the objects were deprecated because they have been replaced, then management applications should plan on migrating their support to the replacements.

11.6 TCP/IP FTP Enhancements

z/OS V1R7 introduces the following enhancements to FTP:

- FTP client API support for C/C++ programming languages
- Enable and disable FTP extended directory search
- FTP confidence-of-success level reporting
- FTP modification of end of line (EoL) sequence for ASCII transfers
- FTP security enhancements

11.6.1 FTP client API support for C/C++ programming languages

z/OS V1R7 provides a C interface for applications to invoke the FTP client programmatically. This API support extends the existing FTP client API. A sample C program is also provided. The supported API calls are:

- FAPI_INIT initializes the interface between the C program and the FTP client.
- FAPI_SCMD issues an FTP client subcommand.
- FAPI_POLL checks the status of an outstanding subcommand.
- FAPI_GETL_COPY retrieves output related to a subcommand and copies the output to a user buffer.
- FAPI_GETL_FIND searches the output related to a subcommand for a line of a specific type. If output of that type is available, it copies a single line of output to a user buffer.
- FAPI_TERM ends the interface between the C program and the FTP client.

In z/OS Communications Server V1R6 an assembler, COBOL, and PL/I API were made available for the FTP client. This allowed customers great flexibility in programming the FTP client to transfer files. Batch files and REXX scripts could not retrieve return codes for conditional execution, so programs could not be made to logically change actions or tasks, depending on the return values or data.

In z/OS V1R7, FTP client API support for C was added to make the availability of a programmable API to the FTP client more robust. With the addition of the C programming language to assembler, PL/I, and COBOL, the use of an API to the FTP client is more available. The C support allows C applications to start multiple FTP clients, control their interaction with the FTP server, retrieve detailed output, check return codes, logically process data while transferring data, use blocking and non-blocking mode, and so forth. Figure 11-52 shows how the client uses the APIs.

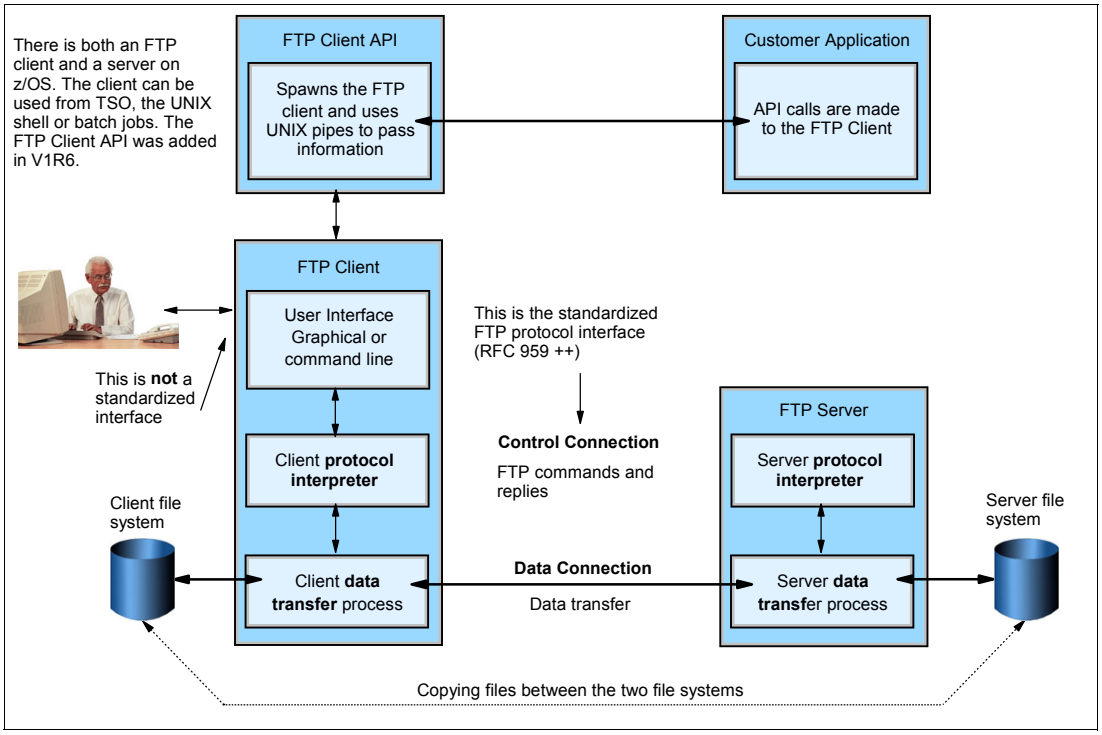


Figure 11-52 FTP APIs and customer application

11.6.2 Enable and disable extended directory search

The LISTSUBdir statement in the client FTP.DATA file enables an installation to select whether **`mput *`** should apply to the current working directory only or span to one subdirectory under the current working directory. The LISTSUBdir statement in the server FTP.DATA file enables an installation to select whether **`mdelete *`**, **`mget`** *, or **`ls *`** should apply to the current working directory only or span to one subdirectory under the current working directory. The following two methods for modifying the LISTSUBdir setting are provided as enhancements in z/OS V1R7:

- The FTP server supports new **`SITE`** command options that enable you to change the server LISTSUBdir setting for your login session.
- The FTP client supports new **`LOCSIte`** subcommand options that enable you to change the client's LISTSUBdir setting.

LISTSUBDIR parameter and duplicate file names

Figure 11-53 on page 278 shows a directory and file example where problems occurred. *Directory current* is the current directory, *Y* is a subdirectory, and *filex* is a file under directory Y. Notice that under directory current, there is also a file named filex.

If you had issued **`mdelete *`**, the operation would apply to both files listed: one in the current directory and one in the spanned subdirectory. Since you are deleting files, duplicate file names are not a problem.

For **`mget *`**, however, duplicate file names are a problem. While LISTSUBDIR is TRUE, and files of the same name exist in the working directory and one or more subdirectories, or in the working directory subdirectories, only the last file of that name copied to the client directory is the survivor. Any earlier files of that name copied during the mget are overwritten.

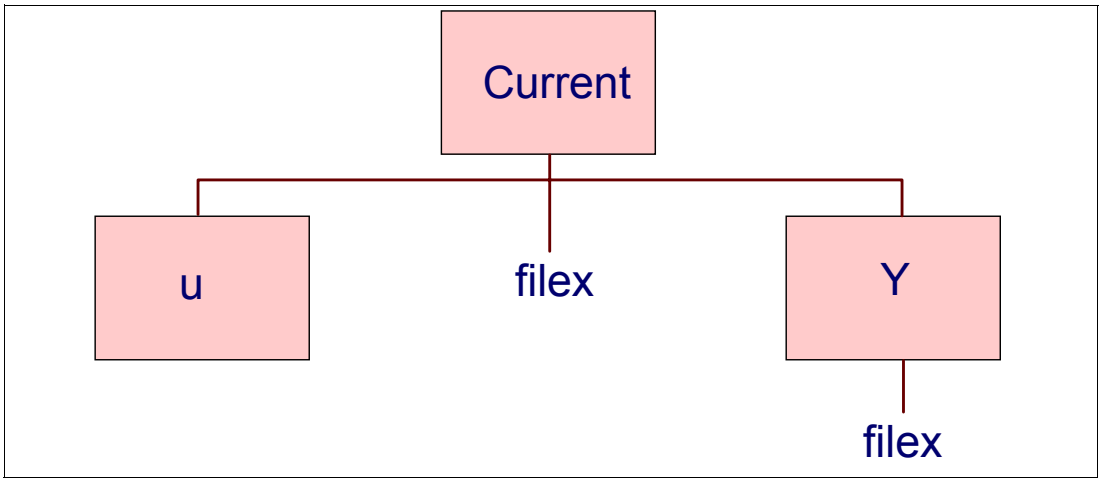


Figure 11-53 LISTSUBDIR file structure

Note: Only one subdirectory level is spanned by listsubdir. Therefore, both /current/filex and current/y/filex were copied. If a file /current/y/z/filex had existed, it would not have been retrieved by mget because that file would not have resided in the first subdirectory level of the working directory.

For more examples of server NLST processing, see the *z/OS Communications Server IP User's Guide and Commands*, SC31-8780 for the following subcommands: **`ls`**, **`mget`**, and **`mdelete`**.

If the LISTSUBdir option is not specified on the **`SITE`** subcommand and the LISTSUBDIR statement is not specified in the server FTP.DATA file, the default is as if the LISTSUBdir option was specified on the **`SITE`** subcommand. In addition, if the z/OS FTP server has the NOLISTSUBDIR option on the **`SITE`** subcommand or LISTSUBDIR FALSE in the server FTP.DATA file, then an **`mget *`** gets only the files in the current directory.

Restriction: The LISTSUBDIR statement applies to HFS file operations only. MVS data set operations are not affected. The FTP client must be communicating with a z/OS V1R7 or later FTP server or an unrecognized parameter response results.

LISTSUBDIR example 1

This first client example is using **`mput`** with LISTSUBDIR FALSE, and is shown in Figure 11-54. Client processing when LISTSUBDIR FALSE is coded in FTP.DATA; the diagram in Figure 11-54 shows the directory structure. The client is putting from the directory /current into the directory /u/user1. Only one file filex is copied (the shaded file). Contrast this with the next **`mput`** example, shown in Figure 11-55 on page 280, where LISTSUBDIR is TRUE.

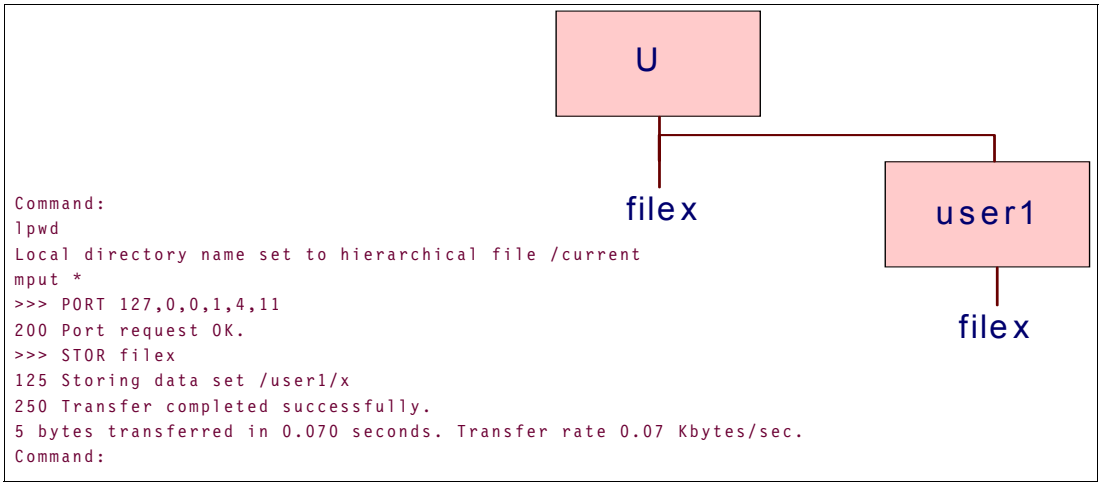


Figure 11-54 Client example mput with LISTSUBDIR FALSE

LISTSUBDIR example 2

The second client example is using **mput** with LISTSUBDIR TRUE, and is shown in Figure 11-55 on page 280. The client's local working directory was /user1. LISTSUBDIR TRUE is coded in the client's FTP.DATA and mget * was issued. The arrows point out that both files named filex were transferred from the client to the server and the server stored both files in /u/user1/X. As you can see in the server replies marked with arrows, since both files were written to the same location, the second filex file overwrote the first filex file.

The z/OS server is not 'aware' that an mput is in progress. To the server, two sequential STOR (put) operations for file filex occurred. Rather, the client sends two files called filex because two files called filex were in the scope of the wildcard search (the * argument causes the client to do a wildcard search for files in the user1 directory).

Since LISTSUBDIR is TRUE in the client's FTP.DATA, this behavior is what the user requested. One way to avoid this conflict is to use the **sunique** (stor unique) subcommand. When sunique is ON, the client uses the **STOU** command in place of the **STOR** command (store-unique instead of store).

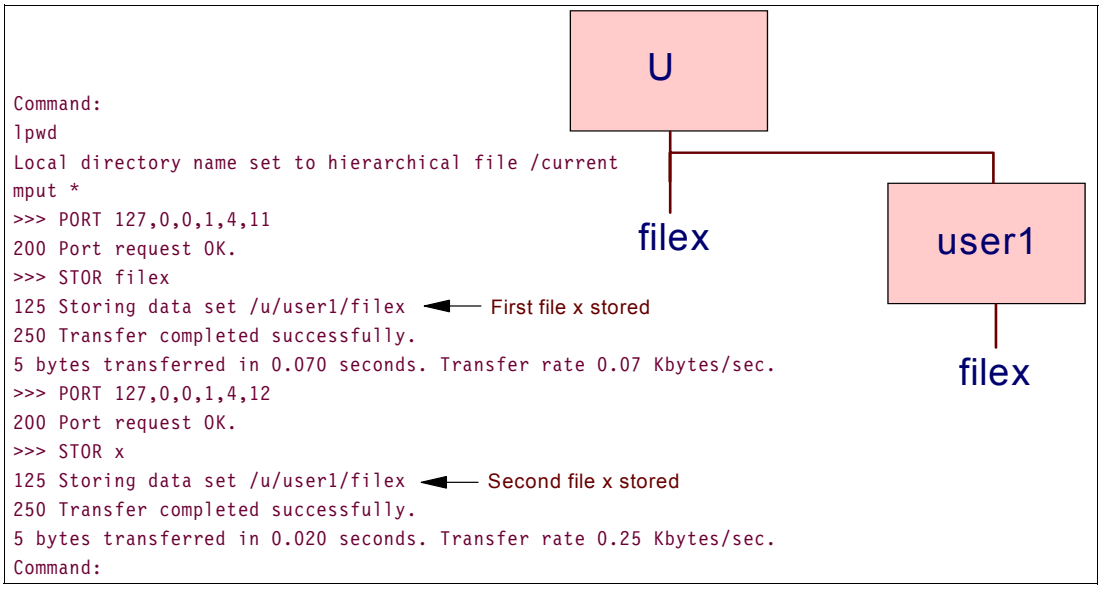


Figure 11-55 Client example mput with LISTSUBDIR TRUE

LISTSUBDIR with TRUE and sunique example 3

This client example mput with LISTSUBDIR TRUE and the **sunique** command, is shown in Figure 11-56. Both files filex are transferred, but the client uses the **STOU** command in place of the **STOR** command to send the file. Two files called filex are transferred, but the **STOU** command directs the server to never overwrite an existing file, but to create a unique name for each incoming file. The server created a new name /tmp/filex1 since the first incoming file used the name /tmp/x.

```
Command:
sunique
Store unique is ON
mput *
Mput sample/x (Yes|No|Quit|Stop prompting)? y
>>> PORT 9,42,103,37,4,49
200 Port request OK.
>>> STOU x
125 Storing data set /tmp/x (unique name)
250 Transfer completed successfully.
41 bytes transferred in 0.020 seconds. Transfer rate 2.05 Kbytes/sec.
Mput x (Yes|No|Quit|Stop prompting)? y
>>> PORT 9,42,103,37,4,50
200 Port request OK.
>>> STOU x
125 Storing data set /tmp/x1 (unique name)
250 Transfer completed successfully.
116 bytes transferred in 0.020 seconds. Transfer rate 5.80 Kbytes/sec.
```

Figure 11-56 Client example mput with LISTSUBDIR TRUE and the sunique command

Prior to V1R7, the LISTSUBDIR is FTP.DATA. This implies:

- Server: Restricted to FTP.DATA setting when logging in.
- Client: Must modify FTP.DATA, restart client to modify LISTSUBDIR.

With V1R7, the user is allowed to change the LISTSUBDIR setting after logging into the server using the **locsite** or **site** command.

- LOCSITE
  - Options: LISTSUBDIR / NOLISTSUBDIR.
  - Affects the **mput** subcommand.
- SITE
  - Options: LISTSUBDIR / NOTLISTSUBDIR.
  - Affects **ls ***, **mget ***, and **mdelete *** subcommands in an ftp client, and the **NLST** command in an ftp server.

11.6.3 Reliability of data transfer

z/OS V1R7 has two enhancements related to the reliability of data transfer: check confidence and end-of-line terminator selection. They are described in this section.

Check confidence

This feature allows the user to determine with some level of certainty that the transfer of a file structure, stream mode file completed successfully. There are three ways in which the confidence level can be conveyed:

- FTP server logging:
  - Requires the FTP.DATA statement FTPLOGGING to be set to TRUE.
  - Uses message EZYFS86I.
- FTPOSTPR user exit.
- FTP client message, uses message EZA2108I.

The confidence checking has five levels of granularity:

- High: No error was detected.
- NoEOF: An EOF marker was not found in STRUCT R or MODE B or C transfer.
- Low: The client did not respond following the transfer or another error was reported.
- Unknown: Only for outbound transfers and only set if checking is active.
- Inactive: Only reported by the FTPOSTPR user exit.

Outbound transfers are given an UNKNOWN confidence level as the highest confidence level that they can obtain. This is due to the fact that the receiver is not necessarily a z/OS FTP client, and we have no mechanism for determining with 100% accuracy that the file transfer completed successfully. If an error is detected, then the confidence level is lowered from UNKNOWN to LOW.

Figure 11-57 shows the statement configuration in the FTP.DATA file.

```
FTPLOGGING      TRUE       ; FTP server logging
CHKCONFIDENCE   TRUE       ; Confidence
```

Figure 11-57 FTP.DATA configuration

Figure 11-58 on page 282 shows the messages in the log and to the client.

```
- FTP server logging: Confidence level is reported in syslog.

May 24 18:35:23 WTSC70OE ftps[50593890]: EZYFS86I ID=FTPMVS100003 TRANS  Confidence=High

- FTP client message: Confidence level is reported to the client.

get myfile /tmp/myfile
EZA1701I >>> EPSV
229 Entering Extended Passive Mode (|||1117|)
EZA1701I >>> RETR myfile
125 Sending data set MUNTANE.MYFILE
250 Transfer completed successfully.
EZA2108I Confidence=High for GET of /tmp/myfile
EZA1617I 154 bytes transferred in 0.010 seconds. Transfer rate 15.40 Kbytes/sec
```

Figure 11-58 Log and client messages

The current setting of CHKCONFIDENCE can be determined by examining the reports generated by **STAT** and **LOCSTAT** command options.

```
- CHKCONFIDENCE -> TRUE
Command:
stat
>>> STAT
211-Server FTP talking to host fec0::9:67:115:4, port 1027
211-User: USER2 Working directory: USER2.
* * * * *
211-JESINTERFACELEVEL is 1
211-Confidence level of data transfers is checked and reported
211-ENcoding is set to SBCS

- CHKCONFIDENCE -> FALSE
Command:
stat
>>> STAT
* * * * *
211-JESINTERFACELEVEL is 1
211-Confidence level of data transfers is neither checked nor reported
211-ENcoding is set to SBCS
* * * * *
```

Figure 11-59 STAT command

CHKCONFIDENCE FALSE is the default setting, and there are no migration issues in most cases. Users of the FTPOSTPR user exit do have a migration issue, new parameter sent to exit. For more information, consult the V1R7 Configuration Reference documentation.

EOL terminator selection

The FTP protocol as defined in RFC 959 demands that the EOL sequence be Carriage Return Line Feed <CRLF>. However, z/OS V1R7 allows you to choose from among the following four EOL terminators:

- CRLF: Carriage Return Line Feed. Default used in the past.
- CR: Carriage Return only.
- LF: Line feed only.
- NONE: No EOL Terminator.

11.6.4 Security enhancements

z/OS V1R7 improves security in the File Transfer Protocol (FTP) in regard to the following areas:

- RACF
- Encryption
- SAPI interface

Advanced encryption standard (AES) exploitation

RFC2264 explains the TLS protocol version 1.0 for securing FTP using TLS.

The TLS protocol provides authentication, integrity, and encryption in a client-server communication. TLS use a collection of cryptographic algorithms, as follows:

- A key exchange algorithm
- A bulk encryption algorithm
- A message digest algorithm

This suite is known as a *cipher suite*. Prior to V1R7 the cipher suite had SSL_RC4_MD5, SSL_3DES_SHA and SSL_DES_SHA protocols. CS V1R7 supports the AES protocol, which is more secure than DES. AES supports 128 and 256 bit keys, RSA key exchange, and SHA-1 (secure hash algorithm).

Security Server enhancements

When an MVS operator enters **`S FTPD`** at the MVS console, a daemon process is started and waits for clients to connect to it, as shown in Figure 11-60 on page 284. When the daemon process receives a client connection, an FTP server is started with user ID=FTPUSER. Figure 11-60 shows a client connection process.

The user ID of the FTP server is FTPUSER. If the FTP uses TLS, the FTPUSER needs access to cryptographic services, which is unacceptable. The solution is to delegate resource profiles.

Delegate resource profiles

Delegate resource profiles is a new RACF V1R7 function. Some applications and daemons initiate requests that require access to resources to which the client who invoked the daemon may not otherwise need access. For example, the FTP daemon (FTPD) shipped with z/OS Communication Server requires access to sensitive ICSF resources that the FTP client does not. Generally, you must authorize the client user IDs to access resources that are needed by the daemon. However, if instructed by the documentation for the application or daemon, such as the FTP daemon, you can define a particular resource as a delegated resource and authorize it for use by the daemon's user ID rather than by the client user IDs.

Delegated resources are general resources that are eligible to be accessed by specially programmed applications that request RACF to check the application, or daemon's, authority for a resource when the client's authority is insufficient. Applications programmed in this way, such as the FTP daemon, are said to contain support for *nested* ACEEs because the identity of the application or daemon is said to be nested beneath the identity of the client for authorization purposes.

With this generic solution to the problem of permitting servers to sensitive resources profiles, an application code change is required before any daemon/server application can exploit resources profiles. Any V1R7 profile can be marked RACF-delegated; however, if the

application isn't coded to exploit delegated profiles, the delegation will have no impact. See V1R7 RACF publications for details.

The following examples are commands that define a resource as a delegated resource:

```
RDEFINE CSFSERV CSFENC APPLDATA ('RACF-DELEGATED')
RALTER  CSFSERV CSFENC APPLDATA ('RACF-DELEGATED')
RALTER  CSFSERV CSFENC APPLDATA ('THIS RESOURCE IS A RACF-DELEGATED RESOURCES')
```

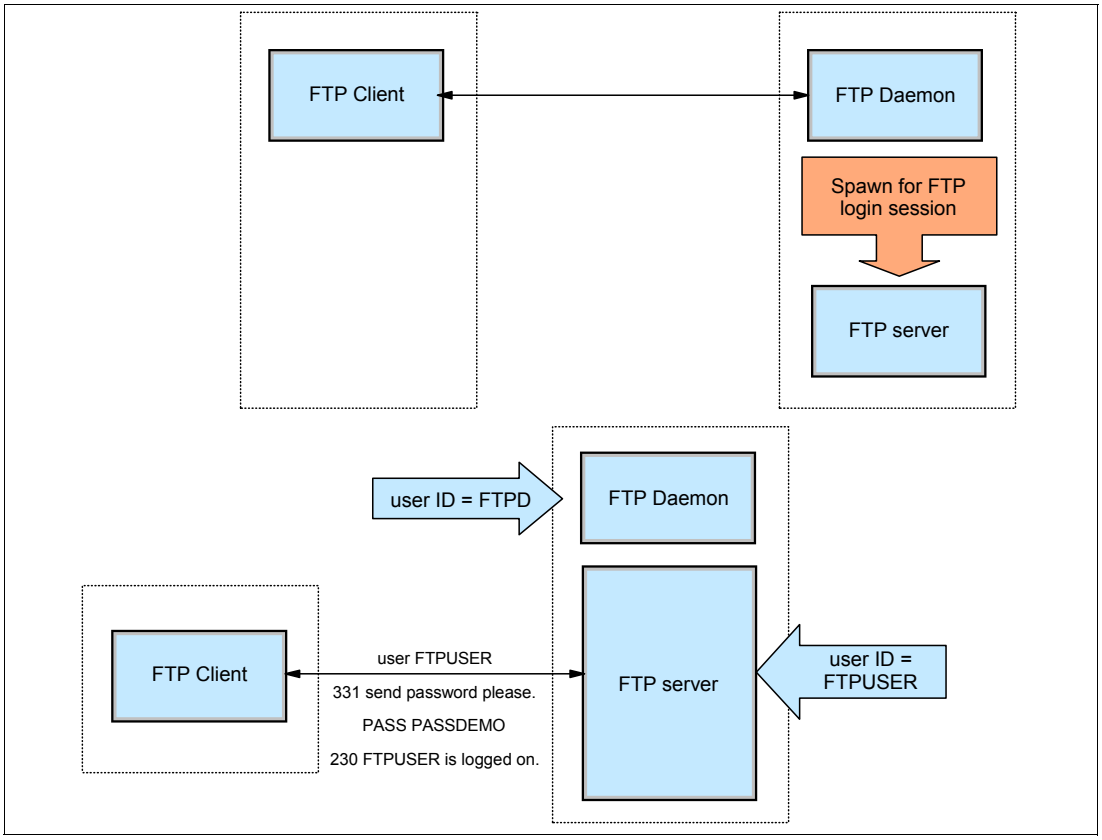


Figure 11-60 Client connection process

JES SAPI enhancements

Figure 11-61 on page 285 shows how FTP submits JES jobs and retrieves output. Using z/OS FTP, you can submit jobs to the server, display job output, retrieve job output (spool files) and delete a job. For details, see *z/OS Communications Server IP User's Guide and Commands*, SC31-8780 and *z/OS Communications Server IP Configuration Reference*, SC31-8776.

Prior to V1R7, SDSF and FTP required UPDATE access to display and retrieve job output. Now, only READ access is required to display or retrieve job output.

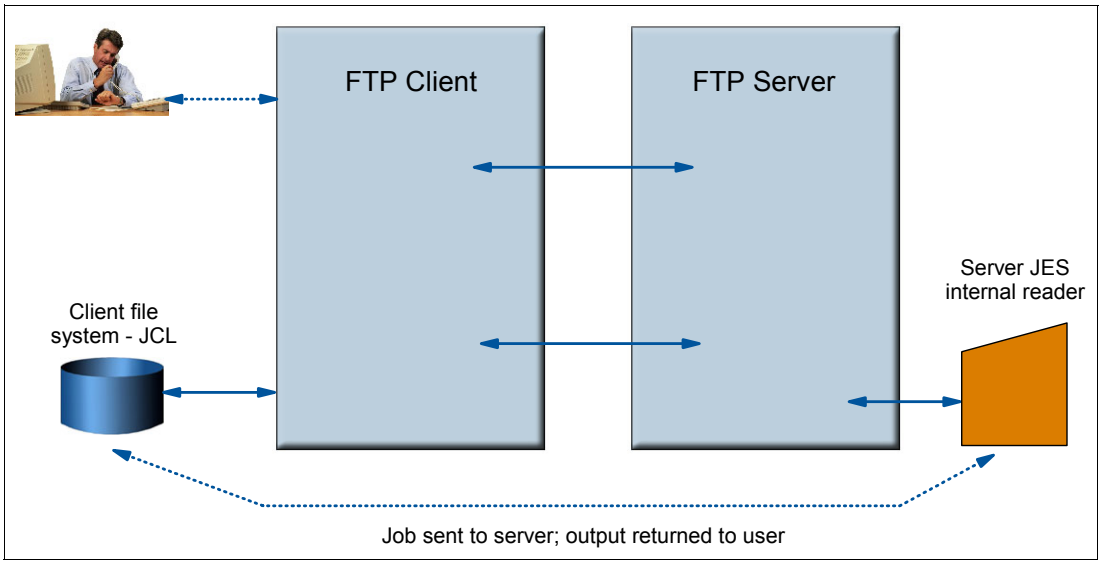


Figure 11-61 FTP and JES

11.7 TCP/IP security enhancements

Prior to z/OS V1R7 CS, the security in the z/OS communications server has two types of protection, as follows:

- Access to the system and its resources
- Services for transmitting and receiving data using secure connections

11.7.1 Access to the system and its resources

Access to the system and its resources has the following components:

- Firewall technologies on z/OS

  You can choose to use the z/OS firewall technologies to set up a traditional firewall structure where one or more firewalls reside in a z/OS LPAR. You can also choose to use the z/OS firewall technologies on your normal z/OS LPARs to add an extra layer of network access protection through IP filtering or add support for VPN end-points on z/OS.

- Intrusion detection services (IDS)

  Attacks can occur from the Internet or intranets. A firewall can provide some level of protection from the Internet but this perimeter security strategy alone may not be sufficient. z/OS IDS broadens intrusion detection coverage.

- SAF SERVAUTH

  All the “traditional” SAF protection of data sets, authorized functions, and so forth on a z/OS system, applies to TCP/IP workload just as it applies to all other types of workload. (Be careful with anonymous services such as anonymous FTP or TFTP services.) The SERVAUTH resource class is used to specifically define and protect a number of TCP/IP unique resources (stack access control, local port access control, network access control).

- Multilevel security (MLS)

  Multilevel security was provided mainly for government use. All users and data are associated with a Security Label via RACF. This means that information can never be written to a user/resource with a lower security label and the information can never be

read from a user/resource with a higher security label. This component prevents declassification of information.

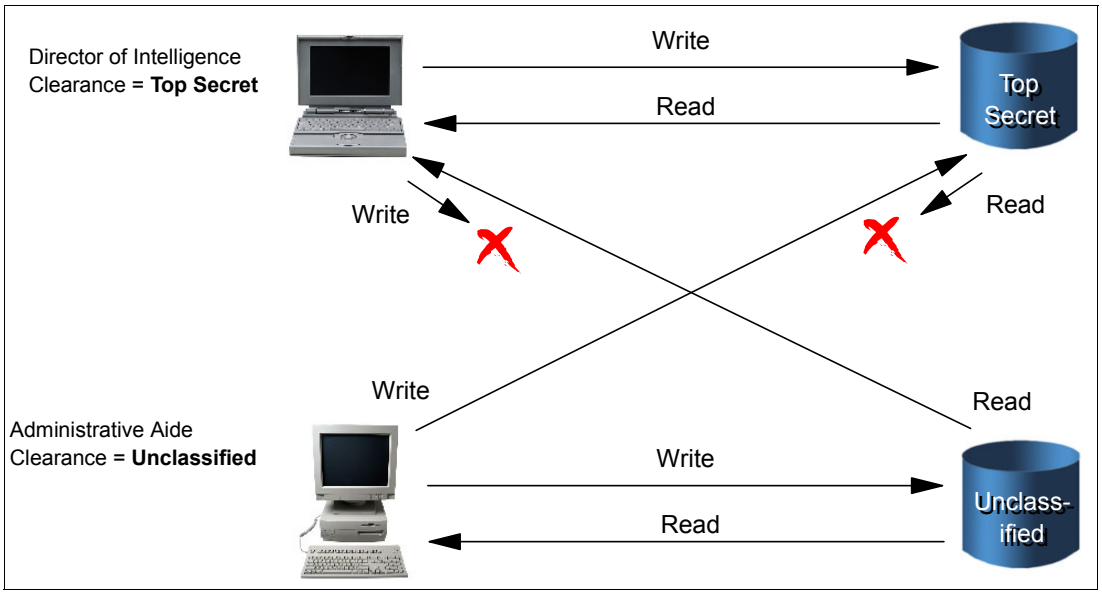


Figure 11-62 Multilevel security protection

11.7.2 Services for transmitting and receiving data using secure connections

These services are provided by the following components:

- Transparent application security IPSec and VPN provides authentication, integrity, and data privacy at the IP layer, as follows:
  - AH protocol provides authentication and integrity.
  - ESP protocol provides data privacy.
  - IKE protocol includes key exchange using public key cryptography and negotiation of security parameters.
  - Management of crypto keys and SAs can also be manual.
  - IP node authentication, not user authentication.
  - Use of IPSec is transparent to upper layers, including application blanket level protection for upper layer protocols.
- Built-in application security with SSL:
  - Provides authentication, integrity, and data privacy above the TCP layer with SSL handshake protocol that includes key exchange using public key cryptography and negotiation of security parameters.
  - User authentication if client certificates are used.
  - Applications must be changed to use SSL APIs.
  - UDP applications cannot be SSL-enabled.
  - z/OS provides a System SSL library for z/OS applications available to LE C/C++ sockets programs only.

On the following page, Figure 11-63 shows the relationship between IPSec or SSL and the TCP/IP stack; Figure 11-64 shows all components and their places in the TPC/IP stack, and Figure 11-65 shows where the firewall technologies functions are supported.

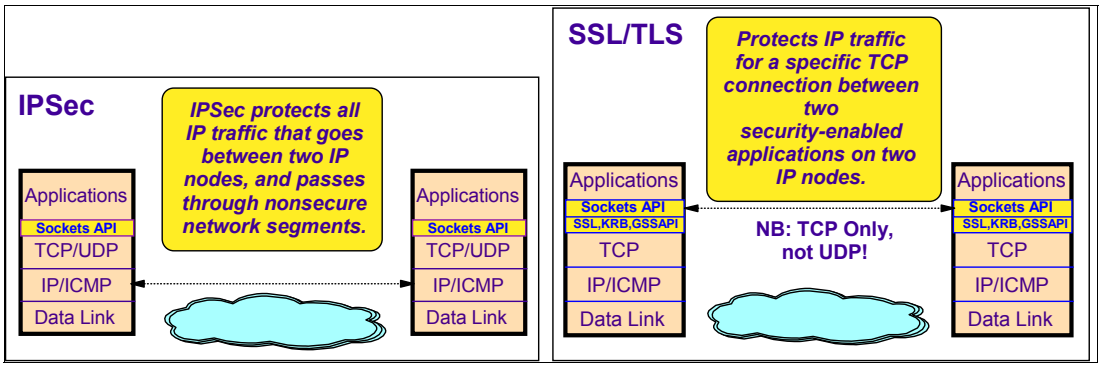


Figure 11-63 IPSec / SSL and TCP/IP stack relation.

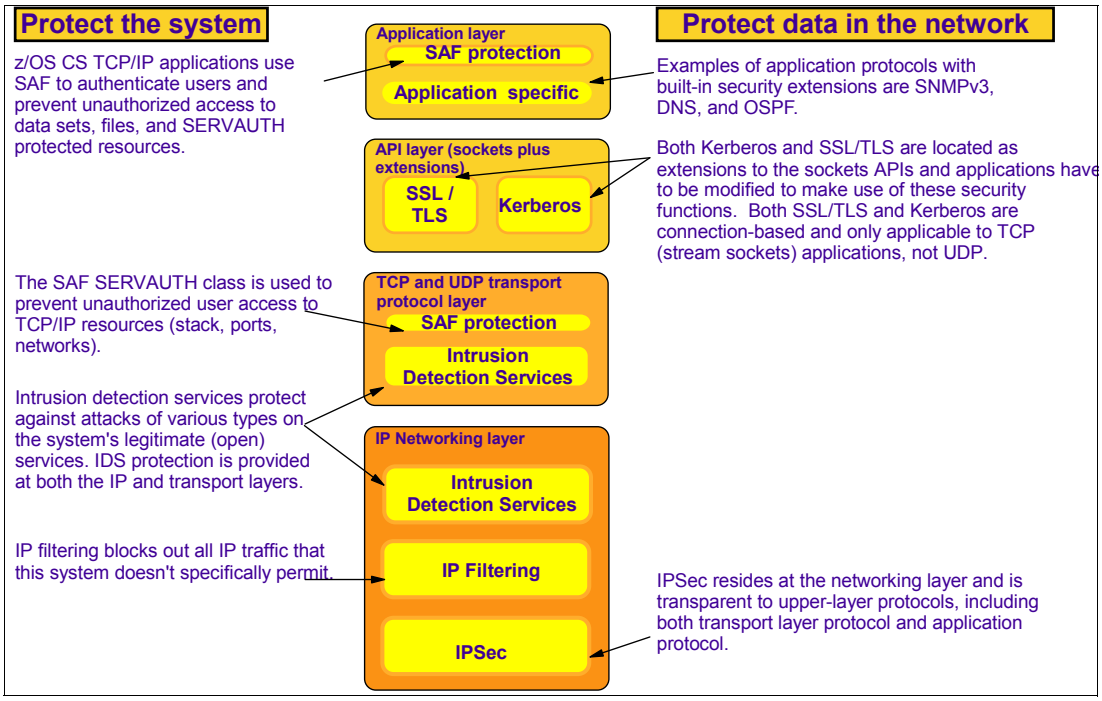


Figure 11-64 IP-based security technology overview and introduction.

The firewall technologies functions that are shipped with z/OS	Included in Communications Server	Included in Security Server Free	Included in Security Server Non-free	Useful in firewall configuration	Useful as self-protection layer in z/OS
IPv4 packet filters	✓			✓	✓
IPv4 IPSec (VPN)	✓			✓	✓
IPv4 Network Address Translation	✓			✓	
Internet Key Exchange (IKE)		✓		✓	✓
Command-line configuration		✓		✓	✓
GUI configuration		✓		✓	✓
FTP proxy server			✓	✓	
SOCKS V4 server			✓	✓	

Figure 11-65 Firewall technologies functions prior to z/OS V1R7

11.7.3 Solutions to previous problems (New in V1R7)

Previously, z/OS Communications Server did not provide all the elements required for IP Security. In particular, the following issues are encountered:

- Firewall technologies must be installed and configured.
- Documentation is split across multiple z/OS elements.
- Configuration does not exploit z/OS Communications Server configuration techniques, Policy Agent (PAGENT).
- The firewall command set is large.
- Overhead is incurred to maintain firewall servers (fwkern, fwstackd, isakmpd, and cfgsrv).
- Service ambiguity involving which service group is responsible for an IP Security problem.

The z/OS V1R7 Communications Server provides an alternative to using the Firewall Technologies IP Security support, including the following capabilities:

  - Provides a Comm Server equivalent to Firewall's ISAKMPD.
  - Eliminates the need to run Firewall's fwkern, fwstackd, and cfgsrv.
  - Provides a PAGENT-based configuration file to replace the existing Firewall configuration commands.
  - Provides one new UNIX System Service command to replace the multiple existing Firewall IP Security management commands.

Note: z/OS V1R7 continues to ship the firewall technologies IPSec support. In a future release, firewall technologies will no longer be shipped. This includes the IP security functions and the additional traditional firewall functions (NAT, SOCKS, and FTP proxy).

Restrictions:

- A stack can use Integrated IPSec or Firewall IPSec, but not both.
- IPv4 only solution, IPv6 is not supported.
- IPv4 Integrated IPSEC/VPN does not support the traditional firewall functions of Network Address Translation, FTP proxy server and SOCKS V4 proxy server.

Additional solutions

IPv4 integrated IPSec addresses three main concerns posed by firewall technologies; Table 11-4 shows the solutions to each concern.

Table 11-4 Additional solutions in V1R7

Concern	Prior V1R7 topic	Solution in V1R7
Configuration	Traditional firewall orientation.	Optimized configuration for a host role. Continue to provide IPSec support for routed traffic.
	Default rules allow all local traffic.	Default filter rules can be specified in the TCP/IP profile. If no default rules are specified then all traffic is denied.
	The GUI mimics the Firewall command set: - No way to share common policy definitions across multiple z/OS images. - All stacks on one image share the same filter rules.	The z/OS IP Security Configuration Assistant GUI provides a higher level of abstraction. Does not mimic PAGENT's IPSec configuration statements. (Hides the syntax and statement hierarchy of the IPSec configuration file.) Minimizes the amount of configuration information required. Advanced options to access additional configuration settings.
	Difficulties defining policy for multiple stacks.	PAGENT supports both a stack-specific and a common IPSec Policy file. The z/OS IP Security Configuration Assistant GUI allows configuration on a per stack basis and also allows configuration objects to be shared across stacks and images.
	Incompatibilities with NAT.	Allow IPSec to coexist with NAT in select configurations (NAT traversal support). Based on two new RFCs issued in January 2005 dealing with NAT/IPSec compatibility issues, RFC 3947 - Negotiation of NAT-Traversal in the IKE and RFC 3948 - UDP Encapsulation of IPsec ESP Packets. Support for pre-RFC drafts 2 and 3. NAT traversal support can be selectively disabled.
	Remote security endpoints configured individually.	Provides the ability to wildcard remote security endpoints.
Serviceability	Inadequate information in ISAKMPD syslog messages.	- IKED issues more detailed messages. - Certain redundant messages issued by IKED are suppressed by default, but can be enabled if needed. - Most IKED messages are included in a message instance block that contains detailed information about the impacted negotiation.
	Information useful to debugging dynamic VPN configuration errors buried in internal trace output.	Information now displayed in syslog output: - Unformatted IKE messages - Formatted IKE messages - Additional debug information related to a negotiation failure - Redundant messages - Information about IKE's certificate caches Display can be enabled/disabled.
	Obtaining diagnostic information for IBM service is difficult.	- IKED utilizes CTRACE. - IKED generates automatic SVC dumps for unexpected internal conditions (message EZD0922I). - IKED shuts down gracefully upon abend and generates an SVC dump.
Performance	ISAKMPD has no protection to prevent thrashing due to high bursts of activity.	- IKED contains logic to defer the start of new work during periods of high activity. - More selective handling of retransmitted messages.

11.7.4 Firewall IP security

Figure 11-66 compares the firewall IP security implemented prior to z/OS V1R7 and the integrated IP security in V1R7, as follows:

- PAGENT is not new to the z/OS Communications Server. Prior to z/OS V1R7 it was used to install QoS and IDS policy. In z/OS V1R7 it will be used to also load IPSec policy as well as Application Transparent Transport Layer Security (AT-TLS) policy.
- New PAGENT configuration statements are defined for IPSec-related configuration. These statements will reside in a separate IPSec policy file. The IPSec policy file eliminates the need for Firewall configuration commands.
- Trmd is not new to z/OS Communications Server. Prior to z/OS V1R7, it was used by the TCP/IP stack to write SYSLOG messages for TR- and IDS-related messages. In z/OS V1R7, it is used to write stack-related IPSec messages.
- The IKE daemon is new to the z/OS Communications Server in V1R7. It is a replacement for the firewall's FWKERN, FWTSTACKD, and ISAKMPD servers.
- The **ipsec** command is new to z/OS Communications Server in V1R7. It replaces the need for firewall management commands.
- The z/OS IP security configuration assistant GUI is new to the z/OS Communications Server in z/OS V1R7. It is a downloadable tool that creates IPSec configuration files. It is highly recommended that this GUI be used for an integrated IPSec configuration. It replaces the configuration aspects of the FW GUI. It does not address the management aspects of the FW GUI. The **ipsec** command needs to be used for IPSec management purposes.

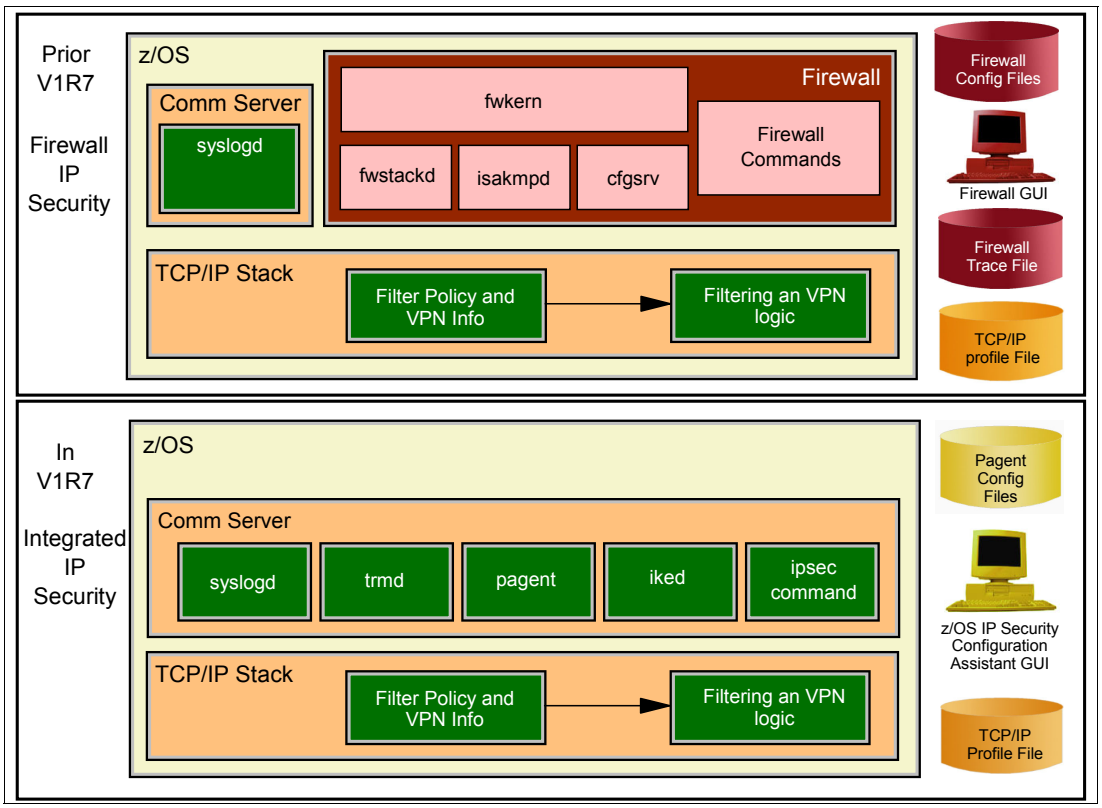


Figure 11-66 Firewall IP security and the new Integrated IP security

11.7.5 IPv4 integrated IPSec-VPN support

IPSec is a suite of protocols and standards defined by the IP Security Protocol working group. It is implemented in the network layer and provides blanket protection for all IP applications, end-to-end protection or just a segment of data path, and does not require modification to existing applications.

- IPSec RFCs implemented by integrated IP security include the following:
  - RFC2401: Security Architecture for the Internet Protocol
  - RFC2402: IP Authentication Header
  - RFC2403: The Use of HMAC-MD5-96 within ESP and AH
  - RFC2404: The Use of HMAC-SHA-1-96 within ESP and AH
  - RFC2406: IP Encapsulating Security Payload (ESP)
  - RFC2407: The Internet IP Security Domain of Interpretation for ISAKMP
  - RFC2408: Internet Security Association and Key Management Protocol (ISAKMP)
  - RFC2409: The Internet Key Exchange (IKE)
  - RFC2410: The NULL Encryption Algorithm and Its Use with IPSecR
  - FC 2451: The ESP CBC-Mode Cipher Algorithms
  - RFC3947: Negotiation of NAT-Traversal in the IKE
  - RFC3948: UDP Encapsulation of IPsec ESP Packets

Table 11-5 identifies the principal IPSec specifications.

Table 11-5 IPSec specifications

Specifications	**IPSec**
Traffic protected with data authentication and encryption	All protocols
End-to-end protection	Yes
Segment protection	Yes
Scope of protection	1) All traffic 2) Protocol 3) Single connection
Requires application modifications	No
Type of security	Device to device
Type of authentication	Peer-to-peer
Authentication credentials	1)Preshared keys 2) X.509 certificates
Authentication principals	Represents host
Session key generation/refresh	Possible

Virtual private network (VPN)

A VPN is a logical network of connected nodes that communicate over unsecure networks using a secure channel. Commonly called a *tunnel*, it uses authentication or encryption, or both. The VPN provides point-to-point security using ip security protocols defined by the IPSec working group.

A VPN connection starts when an application (this application needs a secure connection and this is configured in the security configuration) running in the endpoint A sends a request for connections to an application running in the endpoint B. The IP stack from endpoint A reviews

the security configuration and starts the parameters set negotiation with IP stack B. This parameters set is known as the *security association* (SA).

11.7.6 Security association (SA)

The security association parameters set agreed to between the endpoints is what allows them to get a secure connection and protect traffic. Each SA is unidirectional, so you need one for inbound and one for outbound. The cryptographic keys are different, but generally both SA parameters are symmetrical regardless of the algorithms used. Each SA is identified by the security parameter index (SPI) with 32-bit values added in the AH or ESP header. All the SAs known to the stack are collected in the security association database (SAD). Table 11-6 shows the parameters.

Table 11-6 Security association parameters

Parameter	Options
Security protocol	- Authentication header (AH) and/or - Encapsulating security payload (ESP)
Encryption algorithm (used by ESP)	- DES - Triple DES
Authentication algorithm (used by AH and ESP)	- HMAC_MD5 - HMAC_SHA
Cryptographic keys	Need to decide how to safely exchange keys: - Manually - Dynamically via Internet Key Exchange (IKE)
Encapsulation mode	- Tunnel - Transport
Lifetime/lifesize	For dynamic SA

Types of security associations

There are two types of security associations, as follows:

- **Manually defined SA** - This type of association is not commonly used because it does not provide a scalable solution and in the long run is difficult to manage. It requires the IPSECURITY option on the IPCONFIG statement and this option is mutually exclusive with the FIREWALL option. The SA is defined in a PAGENT IPSec configuration file; take care with the following statements:
  - Cannot be used when default filter policy is in effect.
  - Utilized by filter rules with an action of ipsec.
  - SA is defined by a manual VPN action and can be generated by the z/OS IP security configuration assistant GUI.
- **Dynamically defined SA** - This type is scalable and in the long run easier to manage, although it initially requires more configuration than a manual SA. Requires the IPSECURITY option on the IPCONFIG statement and this option is mutually exclusive with the FIREWALL option. Dynamic SAs are negotiated by the IKE daemon and cannot be used when default filter policy is in effect. Dynamic VPN policy defined in a PAGENT IPSec configuration file can be generated by the z/OS IP security configuration assistant GUI.

Important: The remote endpoint may need to reactivate a manual SA if you locally deactivate the SA and then locally activate the SA.

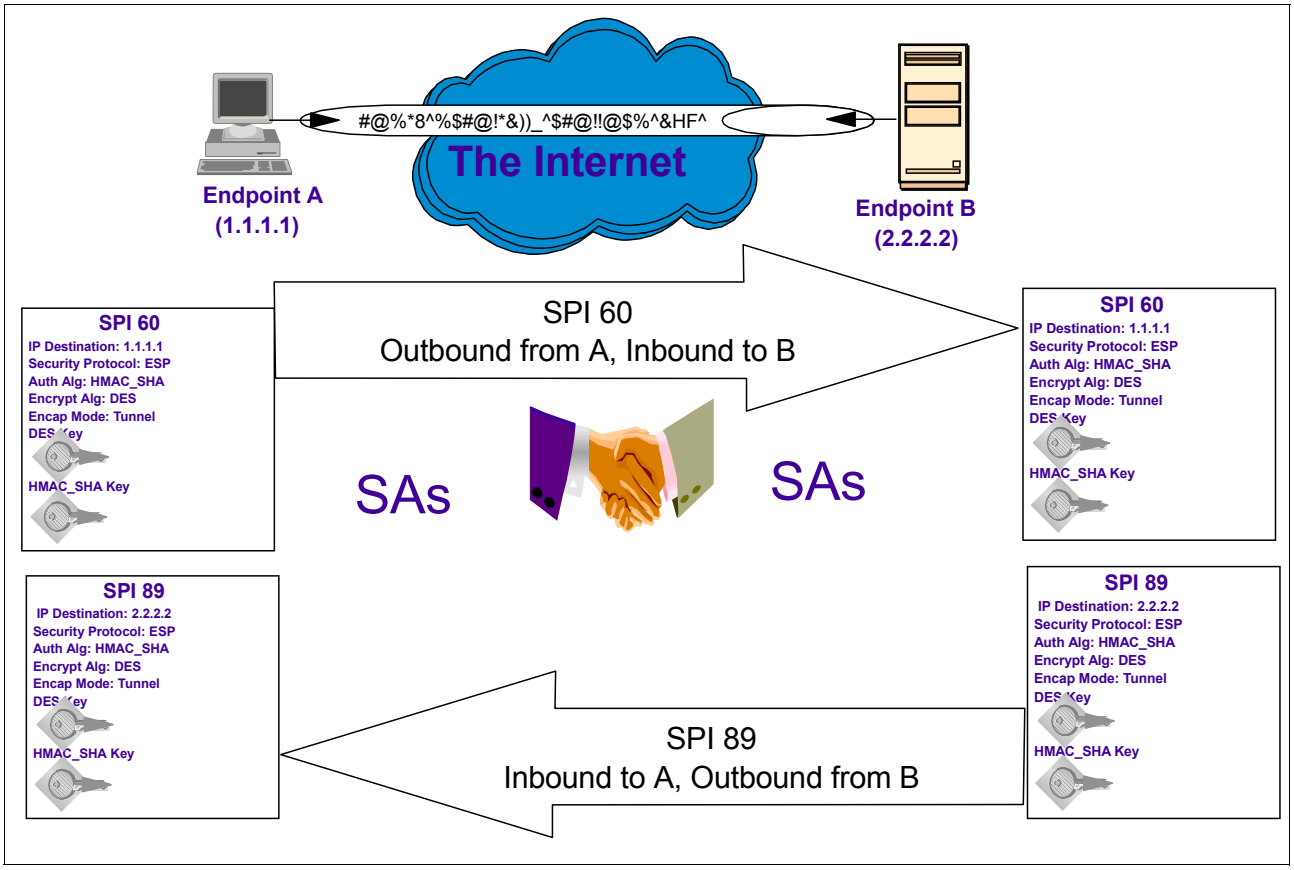


Figure 11-67 IPSec security association example

Security protocols

Integrated IPSec tunnels utilize IP security protocols defined by the IPSec working group, as follows:

- Authentication header (AH) protocol
  - RFC2402 provides:
    - Data Authentication
    - Data origin authentication
    - IP header authentication

  This protocol applies the authentication algorithm to the IP header and the payload to get the authentication data field in the AH header. The target goes through the same process and compares his authentication data result with the same field in the AH header. If they are the same, nobody has changed the IP header and the payload. With the AH protocol, anybody can look at the payload because it isn't encrypted. This problem is solved with encapsulating security payload. Figure 11-68 on page 294 shows the AH header fields.

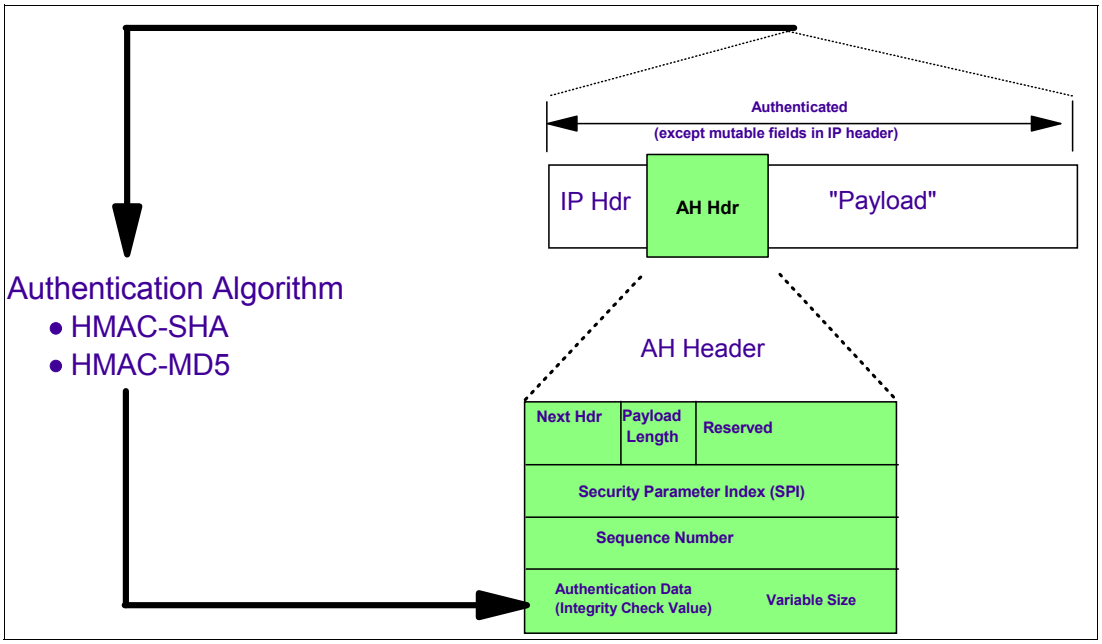


Figure 11-68 Authentication header fields

- Encapsulating Security Payload (ESP)
  - RFC2406 provides:
    - Data authentication
    - Data origin authentication
    - Data privacy

  This protocol provides data encryption but doesn't provide IP header authentication. This protocol is illustrated in Figure 11-69.

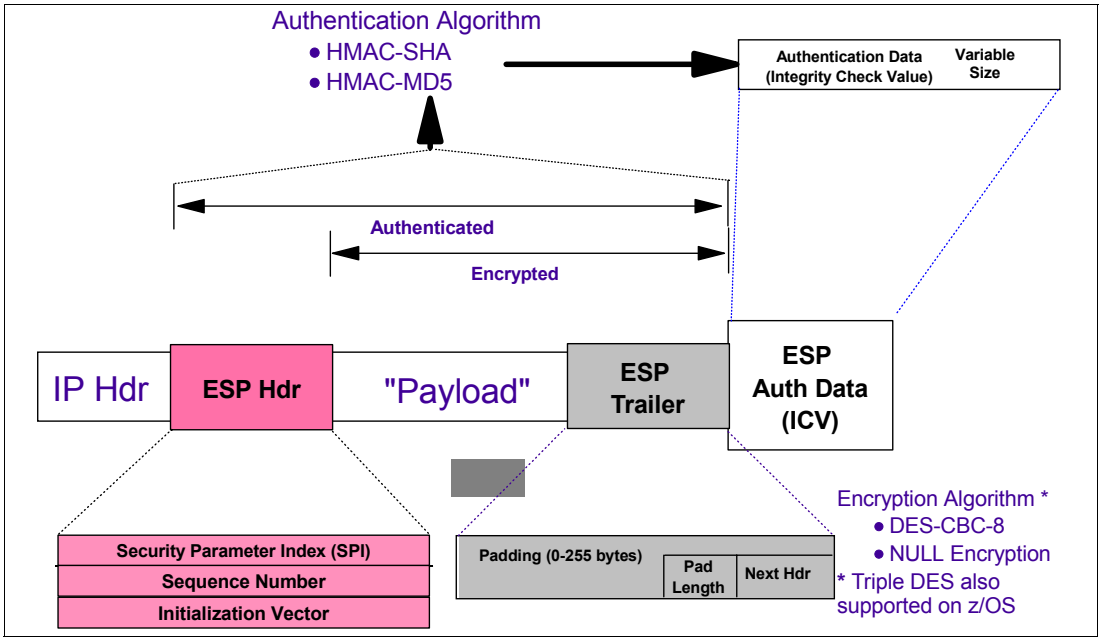


Figure 11-69 Encapsulating security payload headers.

Encapsulation mode

The encapsulation mode indicates how to construct an IPSec packet. Before we explain the different modes we need to review some concepts about the security endpoints:

- Security endpoints have two different parameters, as follows:
  - **Host** - Local data endpoint and tunnel endpoint are the same IP address.
  - **Gateway** (or security gateway) - Local data endpoint and tunnel endpoint are different IP addresses.
- The following modes explain the two encapsulation modes:
  - **Transport mode** - Inserts AH and/or ESP headers between the original IP header and protected data. If neither security endpoint is acting as a gateway, usually transport mode is used. We could use tunnel mode, but there is no need to incur extra cost of adding a new IP header in this case.
  - **Tunnel mode** - Creates a new IP header with an AH or an ESP header, or both. This new header is followed by the original IP header and protected data. If one or both security endpoints are acting as a gateway, tunnel mode must be selected.

In gateway-to-gateway, gateway-to-host, or host-to-gateway configuration, you must use tunnel mode. If you use transport mode, then when the target calculates the authentication data and compares it with the same field from the packet, it will be different because the IP destination from the local host is the gateway IP and this changes the destination IP.

Algorithms

There are two type of algorithms:

Encryption algorithm Encrypts the data in the ESP header. The IETF defines two algorithms: DES and triple DES.

Authentication algorithm Allows obtaining the authentication data field from the payload or the IP header in the ESP and AH security protocols. The IETF defined two algorithms: HMAC_MD5 (RFC 2403) and HMAC_SHA (RFC 2404).

Cryptographic keys

In IPSec, the most important point is how the hosts exchange the cryptographic keys. The way is different in manually defined SA than in dynamically defined SA, as follows:

Manually defined SA Need to decide how to safely exchange keys:

- Sneaker net
- Secure e-mail
- US mail
- Phone

The decision on how to refresh keys must be coordinated with the remote tunnel endpoint's administrator. Manual SA must be deactivated and activated when refreshing keys.

Dynamically defined SA The IKE protocol (RFC2409) is used.

IKE protocol (RFC2409)

RFC2409 implements the internet key exchange (IKE) used in dynamically defined SAs. IKE uses UDP ports 500 and 4500 to communicate with remote security endpoints, negotiating

SAs and sending informational messages. IKE obtains a policy from PAGENT; the policies used are the following:

- Policy information for negotiating IPSEC SAs is used in dynamic VPN actions.
- A policy for creating a secure channel is used to negotiate IPSec SAs using a key exchange policy.
- There is a policy for the **ipsec** command activation and autoactivation using the local dynamic VPN policy.

IKE phases

IKE negotiates the SA in the following phases:

- Phase 1 creates a secure channel with remote security endpoint negotiating for an IKE SA, and generates cryptographic keys that will be used to protect the phase 2 negotiation and the informational exchanges. This phase authenticates the identity of the parties involved, and requires processor-intensive cryptographic operations that are done infrequently.

  There are two different phase 1 exchange modes. Both exchange the same information, but one utilizes fewer messages, as follows:

  - **Main mode**: All IPSec implementations must support main mode. Main mode utilizes six messages. The last two messages contain identity information and are encrypted. This provides identity protection.
  - **Aggressive mode**: Some IPSec implementations do not support aggressive mode. Aggressive mode utilizes three messages. No messages are encrypted.

  Identity information is used to locate a policy. Phase 1 identity types supported by Integrated IPSec include the following:

  - An IPv4 address (this identity type should not be used when behind a NAT)
  - RFC822 name (e-mail address)
  - Fully qualified domain name (FQDN)
  - x500 distinguished name (DN)

  Authentication mode

  There are two methods of host authentication:

  - Pre-shared key - Security endpoint administrators agree to this value. The key is not directly used to encrypt data. Often used during the initial stages of dynamic SA deployment.
  - RSA signature - Requires X509 certificates. Certificates need to contain an endpoint's identity in the certificate's SubjectName (for DNs) or the SubjectAlternate name (for RFC822 names, FQDNs, or IPv4 addresses). Often used when dynamic SA are widely deployed.

  Diffie-Hellman is an algorithm that allows IKE to produce cryptographic keying material. Diffie-Hellman groups are defined in RFC2409 (IKE). Integrated IPSec supports groups 1 and 2. Group 2 provides better security characteristics, but it also requires more computational power.
- Phase 2 negotiates an IPSec SA with a remote security endpoint and generates cryptographic keys that are used to protect data. The keys are:
  - Authentication keys for use with AH
  - The authentication and/or encryption keys for use with ESP

IPSec configurations

For manually defined SAs, you need to modify the following profiles:

- The TCP/IP profile. Define the following options:
  - A new IPSECURITY option in the IPCONFIG statement to enable integrated IP security filtering and IPSec tunnel support. This option is mutually exclusive with the FIREWALL option and cannot be modified by the **`VARY TCPIP,,OBEYFILE`** command.
  - A new IPSEC statement allows you to define default IP filter rules. User-defined rules are always “permit” rules and cannot include routed traffic. The default rules are in effect when IPSec policy rules are not available or specifically enabled by the **`ipsec`** command.
  - A new optional SECCLASS option, when DYNAMICXCF is specified, is used to uniquely identify an interface or group interfaces with similar security requirements or as an IP filtering criteria, which:
    - Can only be specified on rules with an action of permit or deny
    - Allows broad rules to be written for all IP traffic that uses a group of interfaces without explicit knowledge of IP address

    Restriction: This can be specified for all link types except virtual.

- Policy Agent (PAGENT) configuration file options:
  - A new CommonIPSecConfig statement in the main configuration file. Identifies an IPSec configuration file containing policy definitions that are common to all stacks.
  - A new IPSecConfig statement in the image configuration file. Identifies an IPSec configuration file containing policy definitions that are specific to a a stack.
  - A new IPSec configuration file. This file can be generated by the z/OS IP Security Configuration Assistant GUI. This files contains three policy definitions: IpFilterPolicy, KeyExchangePolicy, and LocalDynVpnPolicy.

For dynamically defined SAs you need to modify the following profiles:

- TCP/IP profile: The same as the manual definition.
- Policy Agent (PAGENT) configuration file: The same as the manual definition.
- IKE daemon configuration, define the following:
  - Policy information for negotiating IPSec SA using dynamic VPN actions.
  - Policy for creating a secure channel used to negotiate IPSec SAs in the key exchange policy.
  - Policy for **`ipsec`** command activation and autoactivation in the local dynamic VPN policy.

IPSEC enablement

The IPSec activation can be done in the following ways:

Manual defined SA The **`ipsec`** command activates and deactivates manual SAs. It is automatically activated when policy is installed.

Dynamically defined SA This is a command line activation using the **`ipsec -y activate`** command, which requires the definition of a local dynamic VPN policy. It is autoactivated when a stack connects to a IKED or

when an IP security filter policy is reloaded and requires definition of a local dynamic VPN policy.

- On-demand activation when the stack receives an outbound packet requires the protection of a new dynamic tunnel and does not require definition of local dynamic VPN policy.
- Remote activation when a remote security endpoint initiates the negotiation of a new SA.

11.7.7 Application-transparent transport layer security

Prior to V1R7, there are two types of security technologies in the transport and application layers:

SSL Secure Sockets Layer created by Netscape and originally implemented inside Web clients and servers.

TLS Transport Layer Security as defined by the IETF and based on secure socket layer (SSL originally defined by Netscape to protect HTTP traffic). TLS defines SSL as a version of TLS for compatibility; the TLS clients and server should drop to SSL V3 based on a partner's capabilities. This traditionally provides security services as a socket layer service, and requires a reliable transport layer. z/OS applications can be TLS-enabled with system SSL that is part of the z/OS Integrated Security Services element.

Figure 11-70 shows the IP stack and the TLS/SSL service.

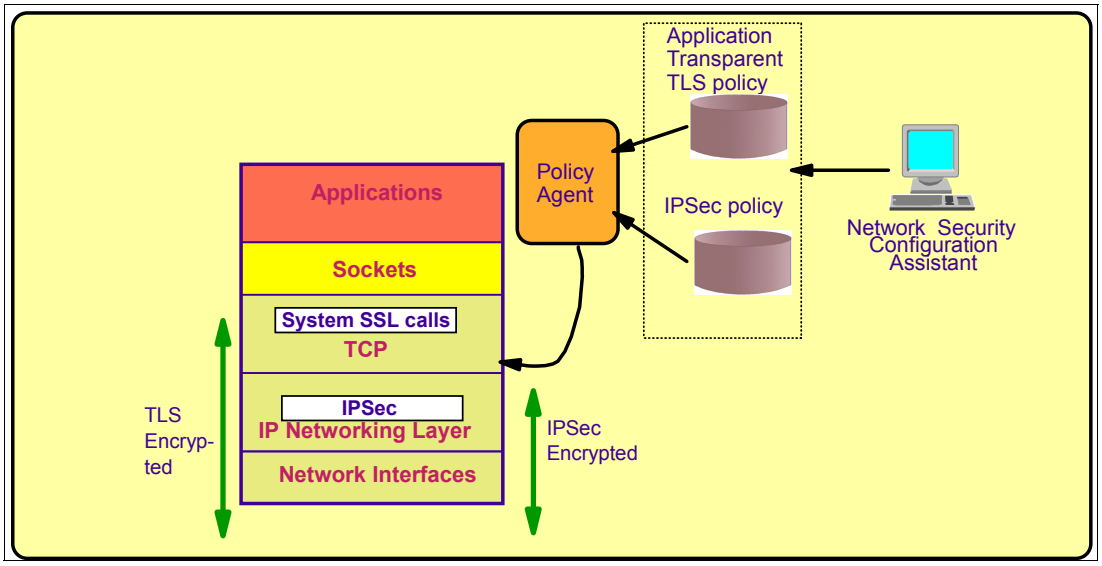


Figure 11-70 IP stack and TLS/SSL service.

Problems associated with these definitions

The following limitations are inherent in these security definitions.

Application layer implementation:

- Development expense is repeated for each application.
- Toolkits available:
  - C and Java
  - Forking or threaded POSIX model

  - Require application change
- Many existing z/OS applications do not fit this model.
- Many applications purchased or otherwise not available for change.

Application-specific deployment:

- Unique configuration for each application.
- Different levels of SSL/TLS architecture support.
- Not all toolkits support or exploit these z/OS features:
  - RACF keyrings
  - Certificates associated with userIDs
  - Hardware cryptography

Solutions provided in z/OS V1R7

The problems explained in the previous section are barriers to implementation and deployment. z/OS V1R7 implements some solutions, allowing the application to be transparent to the Transport Layer Security (TLS). The application can exploit and control advanced features like simple ioctl, permit clear text negotiation prior to starting secure connections, or extract status, certificate, or the associated user ID. The AT-TLS is implemented inside the TCP layer of the IP stack; uses a common configuration through a policy; and exploits RACF, System SSL, ICSF, and hardware cryptographics. The AT-TLS allows multiple security protocols, including TLS (SSL V3.1), SSL V3.0, and SSL V2.

Figure 11-71 shows the connection process.

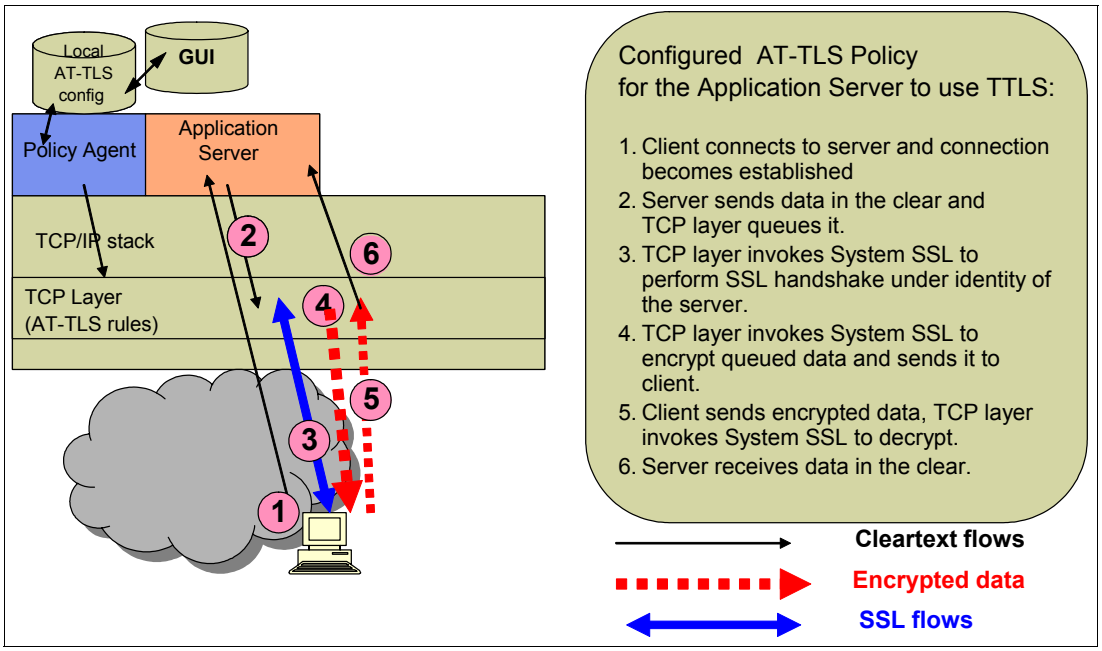


Figure 11-71 AT-TLS connection process

Dependencies for the new support

z/OS Cryptographic Services System Secure Sockets Layer (System SSL) requires the following:

- The PDS pdsename.SIEALNKE contains the System SSL DLLs.

- It must be in the program search order for TCPIP and Policy Agent.
- If it is not in the linklist or LPA, do the following:
  - Use the STEPLIB DD statement in your TCPIP JCL
  - Use the STEPLIB environment variable in the shell, as follows:

    ```
    export STEPLIB=$STEPLIB:pdsename.SIEALNKE
    ```

A z/OS UNIX APAR may be required for keyrings in the file system, as follows:

- The preferred keyring location is RACF.
- z/OS UNIX file systems are supported.

  Access to file system keyrings is controlled by file system ACLs, RWX (read, write, execute) permission bit settings, and z/OS UNIX process identity such as:

  - The owner (uid), group (gid), and the other category. These attributes can be modified during a login session, and are accessed as follows:
    - Stored in z/OS UNIX USP control block
    - By a user: **`su sgrp`**
    - By a program: **`setuid()`**, **`setgid()`**
    - By an ACL of the executed command: **`setuid`** or **`setgid`** attribute
  - If you rely on modified process attributes to control access to an AT-TLS keyring, you must apply a z/OS UNIX APAR to enable BPX1TLS with the new function code TLS_TASK_ACEE_USP#.

Migration considerations

z/OS CS ships the following applications with native TLS support. Some may use either the native support or AT-TLS. Do not configure both for the same application!

- Digital Certificate Access Server (DCAS)
  - Not currently an AT-TLS aware application; do not use with AT-TLS
- FTP client and FTPD server
  - Must specify Secondary Map in AT-TLS policy
  - If using implicit secure socket 990, see the following policy sample for guidance:

    ```
    /usr/lpp/tcpip/samples/pagent_TTLS.conf (/usr/lpp/tcpip/samples/IBM/EZAPAGFT)
    ```
- TN3270E server
  - Must specify Basic Port (no security information in TN displays)
  - No security parameters accepted

    (Keyring/LDAP/Encryption/ConnType/SAFCert/ExpressLogon)
- Sendmail

Restrictions

AT-TLS does not support the following applications.These connections will not map to AT-TLS policy. They will be permitted to proceed in clear text.

- Applications using the Pascal API to access TCP/IP:
  - Line print daemon and the following commands:
    - **`LPD`**, **`LPQ`**, and **`LPRM`**
  - Simple Mail Transfer Protocol (JES spool server)
  - TSO Telnet client

- Web servers using Fast Response Cache Accelerator
- Network administration applications permitted to EZB.INITSTACK profile:
  - Connections established and mapped prior to installation of AT-TLS policy will proceed in clear text.
  - Connections established and mapped after installation of AT-TLS policy are subject to policy installed.

11.7.8 Policy Agent support of IPSec and Application Transparent TL

The Policy Agent (PAGENT) manages the Quality of Service (QoS) and the security profiles in z/OS. The PAGENT policies are an administrative means to define control for traffic prioritization, bandwidth management, security, network behavior, and resource balancing. These definitions are contained in local configuration flat-files or an LDAP server, or both.

PAGENT provides the following configuration related to IPSec and AT-TLS:

- IPSec
  - IP filtering (including manual VPN tunnels)
  - Key exchange
  - Local dynamic VPN tunnel policies
- AT-TLS
  - AT-TLS policies

Figure 11-72 shows all the components related to the Policy Agent.

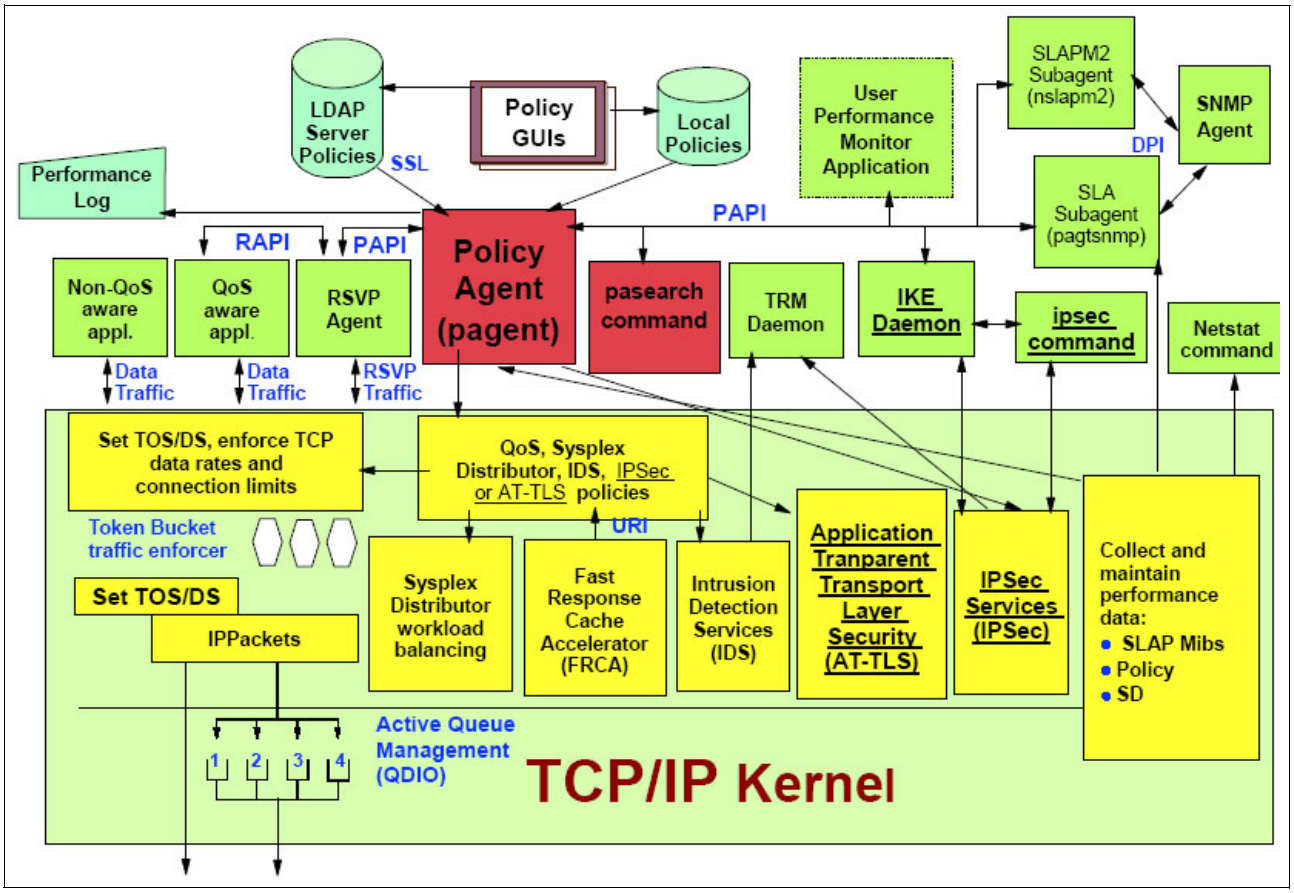


Figure 11-72 Policy Agent components

Where to define the policies

- Security GUIs for IPSec and AT-TLS policies:
  - z/OS IP Security Configuration Assistant.
  - This GUI will generate the configuration flat-file for IPSec or AT-TTLS policies, or both.
- Configuration flat-file: The IPSec and AT-TLS policies contain two files: the main configuration flat-file where you configure basic operational parameters, and the image configurator flat-file where you configure the functional parameters for a specific TCP/IP image.
  - Non-complex QoS policies (only 1 condition)
  - IPSec policies
  - AT-TLS policies
- LDAP Server central policy definitions for many hosts.
  - Non-complex or complex QoS policies
  - Non-complex or complex IDS policies

Note: For details on new IPSec and AT-TLS policies, see *z/OS Communications Server IP Configuration Reference*, SC31-8776.

11.7.9 Application TN3270

TN3270 supports many levels of SSL/TLS encryption: SSLv2 (considered a weak security protocol), SSLv3, TLSv1; and it supports many cipher suites: SSL_NULL_Null (NN), SSL_NULL_MD5 (NM), SSL_NULL_SHA (NS), SSL_RC4_MD5_EX (4E), SSL_RC4_MD5 (4M), SSL_RC4_SHA (4S), SSL_RC2_MD5_EX (2E), SSL_DES_SHA (DS), SSL_3DES_SHA (3S).

The SSLv2 protocol should not be a default choice because it is considered a weak security protocol. The solution is to do the following:

- SSLv2 protocol is not supported as a default choice. A new parameter (SSLV2/NOSSLV2) is added to turn on SSLv2 protocol if needed. It is specified as follows:

  `TelnetGlobals/TelnetParms/ParmsGroup`

- New AES cipher suites are added to the encryption statement, as follows:

  `SSL_AES_128_SHA (A1)` and `SSL_AES_256_SHA (A2)`

Important: A client that supported SSLv2 protocol will no longer work with default settings. The SSLv2 parameter must be specified in TelnetGlobal, TelnetParms, or ParmsGroup.

Figure 11-73 indicates whether SSLv2 is supported; Figure 11-74 displays which cipher suites are supported.

```
Display TCPIP,,Telnet,PROF
EZZ6060I TELNET PROFILE DISPLAY
PERSIS     FUNCTION     DIA  SECURITY  TIMERS  MISC
(LMTGQAK) (OATSKTQSWHT)(DRF)(PCKLECXN2)(IKPSTS)(SMLT)
 -------   -----------  ---  ---------  ------  ----
LMSM***    **TSBTQ*WHT  DJ*  BB******2  ***STS  SMD*
----- PORT: 23 ACTIVE PROF: CURR CONNS: 0
------------------------------------------------------------
FORMAT           LONG
TNSACONFIG       DISABLED
5 OF 5 RECORDS DISPLAYED
```

Figure 11-73 Display TCPIP,,Telnet,PROF

```
Display TCPIP,,Telnet,PROF,DETail
EZZ6080I TELNET PROFILE DISPLAY
  PERSIS   FUNCTION     DIA  SECURITY  TIMERS  MISC
 (LMTGQAK)(OATSKTQSWHT)(DRF)(PCKLECXN2)(IKPSTS)(SMLT)
  -------  -----------  ---  ---------  ------  ----
  *******  **TSBTQ***T  EC*  BB**D****  ***STS  *DD* *DEFAULT
  -------  -----------  DC-  ---------  ------  ---- *TGLOBAL
  -------  ---------H-  ---  SSS-DF---  ------  ---- *TPARMS
  *******  **TSBTQ**HT  DC*  SSS*DF***  ***STS  *DD* CURR
PERSISTENCE
  NOLUSESSIONPEND
...
SECURITY
  SECUREPORT 327
  CONNTYPE SECURE
  KEYRING SAF TNsafkeyring
  CRLLDAPSERVER NONE
  ENCRYPTION 4S,4M,A2,A1,3S,DS,4E,2E,NS,NM,NN (DEF)
  CLIENTAUTH SAFCERT
  NOEXPRESSLOGON
  NONACUSERID
  SSLV2
TIMERS
...
```

Figure 11-74 Display TCPIP,,Telnet,PROF,DETail

11.7.10 Application sendmail

Sendmail security can be improved by specifying a new AES cipher suite that is added to the cipherlevel parameter in the sendmail configuration file, as follows:

- For 128 bit keys, specify: cipherlevel 2F
- For 256 bit keys, specify: cipherlevel 35

11.7.11 Support for mixed case passwords

RACF Security Server now supports mixed case passwords by using the following commands:

```
SETROPTS PASSWORD(MIXED
SETROPTS PASSWORD(NOMIXED)
```

Password possibilities are increased (still eight characters) and the likelihood of random password login is reduced. The problem is that some CS clients and servers use upper case passwords. If CD clients and servers do not adapt, customers cannot use the RACF mixed case password support.

The solution is that more CS clients and servers now support mixed case passwords:

- The servers are the following:
  - FTP
  - Tn3270 for RestrictAppl function
  - Pop
  - z/OS UNIX System Services Telnet (otelnetd)

  - LPD
- The remote command servers:
  - Remote Execution Server (known as RSHD and RXSERVE)
  - z/OS UNIX System Services REXECD command (orexecd)
  - z/OS UNIX System Services RSHD command (orshd)
- VTAM commands:
  - APing

11.8 TCP/IP CICS sockets

z/OS V1R7 Communications Server introduces several areas of support in regard to CICS sockets, as follows:

- CICS sockets Application Transparent TLS exploitation

  Applications that use IP CICS sockets can take advantage of the Application Transparent Transport Layer Security (AT-TLS) support provided by the z/OS Communications Server TCP/IP stack. TCP/IP AT-TLS helps to reduce the total cost of ownership of providing TLS/SSL communication for applications because, in many cases, the application can take advantage of this without having to make code changes. If configured to use AT-TLS policy, z/OS IP CICS sockets-enabled applications can communicate with partner applications that support the Secure Socket Layer Protocol (SSL/TLS).

- CICS sockets performance enhancements

  Several enhancements to IP CICS sockets to improve performance for CICS transactions and the CICS dynamic storage area (DSA). As a result, you can now do the following:

  - Disable IP CICS sockets CICS tracing
  - Stop executing the EXEC CICS MONITOR command for event monitor points not contained in the CICS monitor control table
  - Provide storage relief below the 16 MB line by enabling the IP CICS sockets task-related user exit to be loaded above the 16 MB line

- Support for CICS Transaction Server (TS) Open Transaction Environment (OTE)

  IP CICS sockets provides the capability to run applications using the Open Transaction Environment (OTE), which reduces task control block (TCB) switching. This is especially beneficial to DB2 applications that also make use of IP CICS sockets functions. CICS OTE enables the resource manager's task-related user exits to execute on an Open API, L8, TCB rather than a privately managed TCB. Prior to z/OS V1R7 Communications Server, if a DB2 V6 program also included EZASOKET calls, then TCB switching occurred, which could be expensive in terms of CPU consumption.

- CICS Listener user ID support

  z/OS V1R7 Communications Server introduces a new enhancement for the IBM-supplied CICS Listener to provide the ability to set the user ID for a listener task. This enables unique and predictable user IDs to be associated with each CICS Listener instance in a CICS region regardless of the method used to start the listener (such as when using PLTPI or by way of the EZAO transaction). This feature can provide enhanced CICS resource control for each listener and can also be used to assign a unique user ID to all transactions that a specific CICS Listener starts.

See *z/OS Communications Server New Function Summary*, GC31-8771 for a description of this new support.

11.9 TCP/IP: OROUTED removed

z/OS V1R7 Communications Server discontinues support of the OROUTED routing daemon. OMPROUTE is the recommended routing daemon replacement. You must migrate from OROUTED to OMPROUTE if you have not already done so. The OMPROUTE routing daemon supports the following:

- RIPv1
- RIPv2
- RIPng
- OSPF
- OSPFv3

Tip: See *z/OS Migration*, GA22-7499 for detailed information on the OROUTED removal.

11.10 TCP-IP CTRACE optimization

Prior to z/OS V1R7, CTRACE needs to reduce processing for collecting SYSTCPIP tracing and reduce the set of trace data about sessions.

z/OS V1R7 improves CTRACE in the following ways:

- Reduce processing for SYSTCPIP tracing, add specialized trace calls that collect a minimum of information, and trace only one contiguous data area (less than 256 bytes).
- Add the PFSMIN and TCPMIN options to the SYSTCPIP CTRACE. The PFSMIN option writes a minimum trace entry for each PFS module. The TCPMIN option writes a minimum trace entry for specific events during TCP processing.
- There are four new CTRACE options for SYSTCPIP:
  - PFSMIN: The PFSMIN option writes a small amount of trace data for each PFS module.
  - TCPMIN: The TCPMIN option writes a small amount of trace data at strategic points in TCP protocol processing.
  - ALLMIN: Combines PFSMIN, TCPMIN, INIT, OPCMDS, and OPMSGS options.
  - EID: Allows tracing only for specific trace records. The EID keyword should only be used with the assistance of IBM service personnel.
- There are two new CTRACE options for IPCS:
  - PFSMIN
  - TCPMIN

Table 11-7 shows PFSMIN and TCPMIN data areas.

Table 11-7 PFSMIN and TCPMIN data areas

PFSE	Physical File System Entry trace record
PFSX	Physical File System Exit trace record
TCRE	TCP Read Entry trace record
TCRX	TCP Read Exit trace record
TCWE	TCP Write Entry trace record
TCWX	TCP Write Exit trace record
TCRV	TCP Recv Data trace record
TCSN	TCP Send Data trace record
TCSQ	TCP Send Queue Data trace record
TCRD	TCP Recv Drop Packet trace record

Migration concerns: Do not mix the PFS and PFMIN options and do not mix the TCP and TCPMIN options. You will be just collecting the data twice.

12

Networking with TCP/IP

This chapter provides information about NJE and the new TCPIP support provided with z/OS V1R7.

The following topics are covered:

- NJE basics
- SDSF support of TCP/IP NJE
- Exit impacts
- Enhanced functions

12.1 NJE basics

Before data can be sent from one *network job entry* (NJE) node to another via TCP, a virtual circuit must be established between the two nodes. A *virtual circuit* is a path between two applications over which TCP packets can be sent. An IP address is assigned to a system. Each TCP/IP service machine will have an IP address. Many applications on a system can use the TCP/IP service machine. To enable the TCP/IP service machine to separate incoming packets, the applications use port numbers to indicate which packets correspond to which applications. TCP allows an application to open a virtual circuit in either *passive* mode (waiting for incoming requests to open, also known as *server mode*) or *active* mode (sending requests to open, also known as *client mode*). In general, one TCP application (the server) issues a passive open for a port number (known as the *well known port*) and the other TCP application (the client) will issue an active open for the well known port on the system (IP address) where the first (server) application is located. Either side can attempt to connect to the other side's port. The TCP connection or virtual circuit path between the two applications will be completed and data can be exchanged over the path.

Geographically dispersed processors can communicate using the network job entry facility for the purpose of transmitting commands, messages, jobs, and job output between the host systems such as AS/400®, JES2, JES3, VM/RSCS, and VSE/POWER.

The facility can also provide communication among processors at the same location as an alternative to a JES2 multi-access spool configuration.

An NJE network can consist of up to 32,767 nodes. Each of the nodes can be either a single processor or a multi-access spool (MAS) configuration. A MAS configuration is one in which as many as 32 members share one set of JES2 spool and checkpoint volumes. NJE can also communicate between primary and secondary JES2 members on the same processor.

There are three ways to send and receive data with NJE:

- Binary synchronous communication (BSC)
- Systems Network Architecture (SNA)
- TCP/IP (this is new support for JES in z/OS V1R7)

12.1.1 Logical and physical configurations

The difference between logical and physical configurations is the most significant distinction between SNA, BSC, and TCP/IP protocols. In BSC NJE, these configurations are similar. In SNA NJE, the logical and physical configurations can be quite different. Various routing tables in the host and the communication controllers determine the logical configuration. In Figure 12-1 on page 311, Node A can have a direct session with Node C even though no direct line exists between them. Node A and Node C view themselves as being directly connected to each other. ACF/VTAM and Multi-Systems Networking Facility (MSNF) support this connection through the communications controllers (cc) and the network control programs (NCP), which are transparent to the nodes. Figure 12-1 represents a simple SNA NJE configuration.

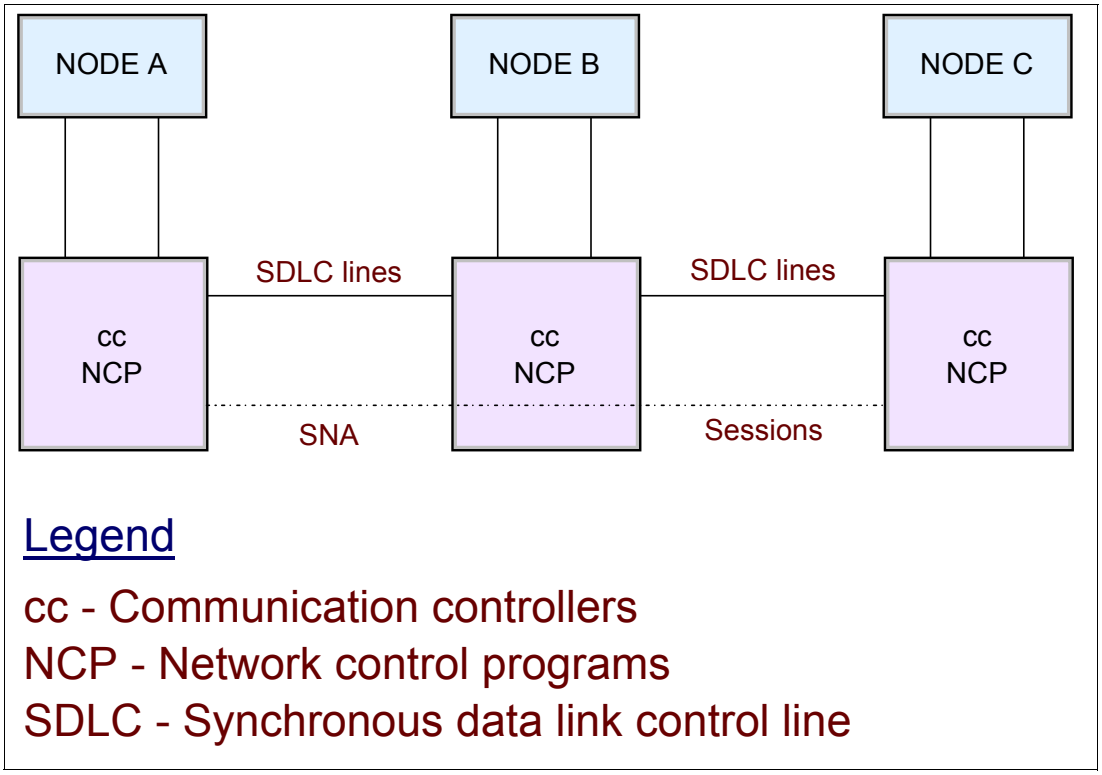


Figure 12-1 Sample SNA NJE configuration

12.1.2 Hardware considerations for NJE

The types of NJE communication links are the following:

- BSC lines

  BSC lines attached to IBM 3704, 3705, or 3725 Communication Controllers operate in emulation mode. Generate NJE 3704s and 3705s to emulate a 2701 communication controller.

- Channel-to-channel (CTC) adapters

  Channel-to-channel adapters supported as high-speed communication lines for NJE must be on block multiplexers because they can “lock up” selector channels. Use the LINE(nnnn) JES2 initialization statement to define these adapters. (See *z/OS V1R7.0 JES2 Initialization and Tuning Reference*, SA22-7533 for more details.)

  You can also use CTC adapters to support an SNA NJE environment; however, ACF/VTAM controls these CTC adapters. Consult *z/OS Communications Server: SNA Resource Definition Reference* for VTAM-CTC requirements.

  Channel-to-channel (CTC) connections are identical to BSC communications except CTC connections do not use EP or PEP. CTC connections are best suited for connections to nodes within the same computer facility. NJE protocols support an ESCON basic mode CTC (defined to the hardware configuration dialog as BCTC) and a 3088 CTC, but do not support an ESCON CTC (defined to the hardware configuration dialog as SCTC).

- SNA network lines participating in an NJE environment

  SNA NJE application-to-application sessions using SDLC lines require that the Advanced Communication Function (ACF) for the Network Control Program/VS (NCP/VS) or the partitioned emulator program (PEP) resides in the IBM 3705. Also, the processors

participating in these sessions must have ACF for VTAM and the Multi-System Networking Facility VTAM installed.

- TCP/IP NJE sessions

  This type of session takes advantage of TCP/IP's hardware independent layered stack to establish connections over a number of existing hardware protocols such as Ethernet and Token Ring. To use TCP/IP sessions, z/OS Communications Server TCP/IP requires z/OS UNIX System Services and ACF/VTAM to be configured and active. If there is support on the node being connected to, TCP/IP NJE sessions support IPv6 and TLS/SSL.

12.2 Using networking with TCP/IP

Every IBM networking partner now supplies a native TCP/IP implementation. As the importance of BSC and SNA networks decrease, the importance of an NJE over TCP/IP for MVS increases. It was a major requirement for customers to have this function available and it is now incorporated in z/OS V1R7.

NJE over TCP/IP was first implemented by VM (RSCS) and has since been implemented by AS/400 and VSE/POWER (the other major NJE partners). Since the VM implementation, a number of developments in TCP/IP required some updates to the existing protocol.

The TCP/IP NJE support in JES2 and JES3 now implements the protocol first introduced by RSCS and it extends that protocol to include the following new support for:

- IPv6
- Enhanced security (SSL/TLS)
- Changes to the basic sign-on protocols
- SYSIN data streams that have an LRECL of up to 32K

The goal of the redesign is to improve RAS and to get better performance (both for JES2 and for the NJE data transfers). JES3 is implementing TCP/IP NJE in parallel with the JES2 support. Some of the new code is part of a common component (common to JES2 and JES3).

A network can consist of any combination of SNA, BSC, and TCP/IP connections. Each of these has its advantages. Your choice depends on the available hardware and software at your installation.

A JES2 complex can use BSC, SNA, or TCP/IP protocol, or all. A user submitting an NJE job is not aware of whether JES2 is using BSC, SNA, or TCP/IP. To define the type of protocol that JES2 uses, code the TYPE parameter on the NJERMT initialization statement.

12.2.1 NJE over TCP/IP compatibility

The TCP/IP NJE protocol is an addition to the SNA and BSC protocols that JES already supported. There is no plan to alter the level of support for SNA or BSC (or BDT). Of course, in order to use the TCP/IP NJE support, both sides of the connection must support TCP/IP NJE. The implication is that z/OS V1R7 JES will be capable of using TCP/IP, SNA, or BSC to connect to other z/OS V1R7 JES2, z/OS V1R7 JES3, RSCS, AS/400, and VSE/POWER.

Compatibility: To connect to older levels of JES (z/OS 1.6 or earlier) will require the use of SNA or BSC connections.

Software and hardware dependencies

The z/OS V1R7 release of TCP/IP is needed for the NJE/TCP support.

JES2 coexistence APAR needed for new support

The JES2 migration policy of enforcing coexistence rules requires a compatibility APAR whenever a new release comes out. This allows the compatibility APAR to pre-req any service that may be required as well as deliver any code that is needed to support the new release. In this release, the coexistence APAR is needed to support a number of line items related to NJE and spool data set processing.

APAR OA08145 needs to be installed to handle the following problems:

- NJE truncates if support not on all nodes in the path.
- A spool reload fails without the APAR for jobs with long SYSIN.
- This APAR also adds support for long SYSIN records (greater than 254 bytes for NJE and spool OFFLOADs). Without the compatibility APAR, NJE truncates SYSIN records to 254 bytes. Spool reload of SYSIN records greater than 254 bytes causes errors and the loss of the job that is being reloaded. This APAR does not allow down-level nodes to create large SYSIN records. It only supports large SYSIN records received over NJE or spool reload.

12.2.2 Supported protocol

The protocol used is based on the same one used by VM (RSCS). Originally described as BITNET II, this protocol has been assigned:

- Port 175 for non-secure sessions
- Port 2252 for secure sessions

The protocol is essentially the BSC CTCA protocol with an IP wrapper and some additional sign-on protocols. The original protocol was designed for IPv4 and uses hard-coded IP addresses in some of the data records. This original protocol is tolerated; however, a newer protocol that uses IPv6 addresses and does not send IP addresses in the header is the preferred protocol. The formal documentation of this protocol, with upgrades to this release, is in publication: *z/OS Network Job Entry (NJE) Formats and Protocols*, SA22-7539.

It can be found on the Internet at the following site:

```
http://publibfp.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/iea1m503/6.5
```

Secure sign-on protocol

One of the enhancements being made is secure sign-on protocol using SAF/RACF. This new protocol applies to TCP/IP NJE as well as BSC and SNA sign-ons, and has the following characteristics:

- Uses SAF/RACF APPCLU class
- Is controlled by NODEnnn SIGNON=SECURE|COMPAT

TCP/IP NJE supports all the NJE constructs (ENDNODE, SUBNET, Store and forward, and so forth) that the owning JES supports.

12.2.3 TCP/IP NJE address space

One of the primary concerns customers have had with JES2 address space outages is that when the JES2 address space ABENDs, all NJE connections are lost. Even though the JES2 address space can be restarted, the NJE connections must also be re-initialized. Many

customers have automation to do this, but only when the system is IPLed. So, the JES2 address space outage turns into a system outage.

In order to address this, the TCP/IP NJE support is moved to a separate address space, where the connection can remain active when the JES2 address space is unavailable. This approach has multiple benefits:

- Availability

  Outages of the JES2 address space do not affect the availability of the NJE connection. Conversely, problems in the NJE address space do not result in a JES2 outage as they would have in the past.

- Performance

  Most of the work associated with the NJE connection (I/O, building headers and trailers, and so forth) is being done outside the JES2 address space, rather than under the JES2 main task. This frees up cycles in the already overtaxed JES2 main task to do other things.

The actual communication with TCP/IP is being done via a new common JES2 and JES3 component, IAZNJTCP.

The **$S NETSRVx** command is used to create the NETSERV address space. The name of the address space is based on the owning JES2 subsystem name and the associated NETSERV number. The PGM= for the address space is IAZNJTCP. The ASCRE macro is used with the IEESYSAS PROC. Figure 12-2 shows the new address space name.

```
jesxSnnn
Where: jesx – subsystem name
       nnn  – NETSRV number
```

Figure 12-2 JES2 NETSRV address space name

Figure 12-3 illustrates the new TCP/IP NJE address space.

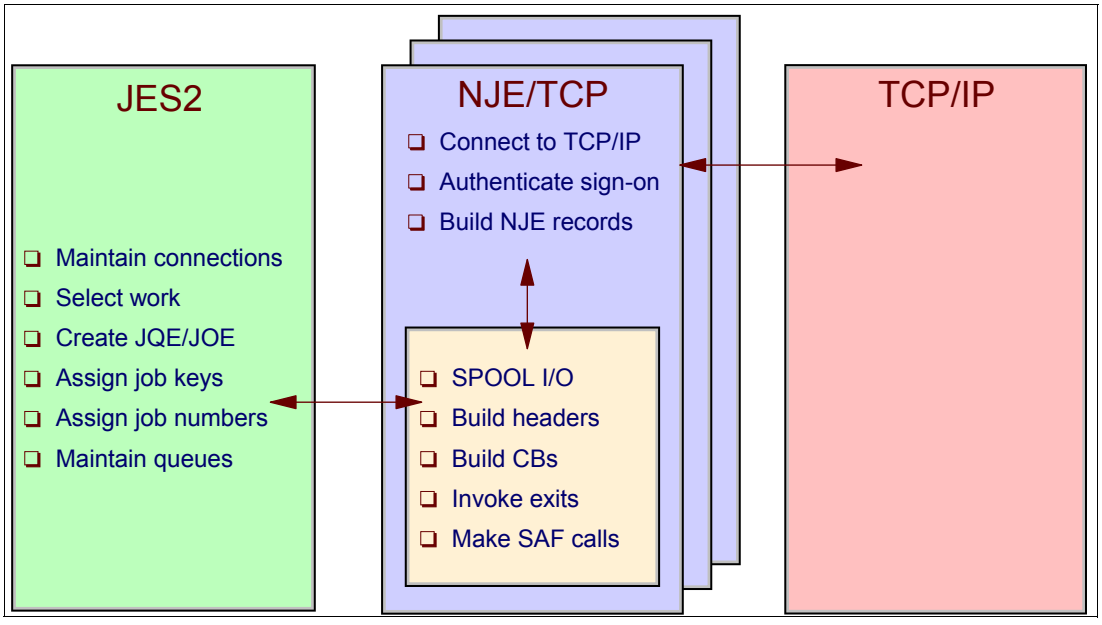


Figure 12-3 Networking address spaces with TCP/IP

JES definitions

The owning JES is responsible for:

- Initialization statements
- The command interface
- Network topology
- Selecting jobs or SYSOUT to transmit
- The creation of new job or output structures for inbound data

TCP/IP NJE address space functions

The TCP/IP NJE address space main functions are:

- Communicates with TCP/IP
- Create or process NJE data streams
- Read and write data from the JES spool

There can be multiple TCP/IP NJE address spaces associated with each JES.

The TCP/IP NJE address space can be associated with one of the following:

- A specific IP address and TCP/IP stack
- All IP addresses supported by a stack
- All IP addresses known to this system

However, each TCP/IP NJE address space must have a unique port number assigned to it for incoming connections.

TCP/IP NJE supports all the NJE constructs (ENDNODE, SUBNET, store and forward, and so forth) that the owning JES supports.

NETSRV devices

Each TCP/IP NJE address space is associated with a NETSRVn device in JES2. The name of the new TCP/IP NJE address space has been chosen to aid operations in associating the address space with the corresponding JES2 NETSRV device. Since there can be multiple address spaces per JES2 and multiple JES2s per MVS image, the name of this new address space must include the owning JES2 name as well as the internal JES2 structure identifier. The format that was chosen is *jesx*S*nnn* where *jesx* is the owning JES name (for example, JES2) and *nnn* is the associated JES2 network server number. For example, JES2S001 is the address space name associated with the JES2 subsystem, network server NETSRV1.

Code that runs in the new address space is common code between JES2 and JES3 (IAZ component) that calls JES-specific processing routines. It is the responsibility of the JES to start the address space and, in the address space initialization routine, initialize an interface control block.

Losing NJE connections

In addition to the new support for TCP/NJE, a number of external changes are being driven by the nature of how TCP/NJE operates. SNA/BSC NJE connections are lost when the JES2 address space terminates. As a result there is no need to preserve any state information over a JES2 hot start. However, TCP/NJE connections, by design, persist over a JES2 hot start. This implies that state information must be preserved over a hot start and that warm start (hot start) processing cannot just discard jobs that are on JOB and SYSOUT receivers.

The major state information that must persist over a hot start is connection information. This is the information used by the network path manager (NPM) to determine the network topology (how to get from node A to node B). Previously, it was assumed that when a node went down, any connection records ($NATs) could be marked inactive. Now, for TCP/IP NJE connections, they must be maintained unless the MVS also goes down. Furthermore, the node definitions cannot change over a hot start, since connections are based on node statements and changing node definitions can have unpredictable results.

12.2.4 Node definitions

To preserve node definitions (and to address a number of user requirements), portions of the NODE initialization statements are saved in a new checkpoint CTENT (NAME=, SUBNET=, PATHMGR=, PRIVATE=, ENDNODE=). The result is that the values now have a JESPLEX (MAS) scope and will persist over a hot start. These operands on a **$TNODE** commands are propagated to all members of the MAS. Unlike other MAS scope settings, the values in the initialization deck are used on all but a *hot* start. This is because most customers do not use operator commands to update NJE settings, but rather a restart of JES2.

Because of the new CTENT and other changes for TCP/IP NJE, the following new support is included in this support:

- You can increase the number of defined nodes via the following operator command:

  ```
  $TNJEDEF,NODENUM=num
  ```

  Support to decrease the value was not implemented.

- Change the local node name via the following operator command:

  ```
  $TNODE(n),NAME=name
  ```

  This can affect the XCF Group name if it defaults to the local node.

- Change the name of another node via the following operator command even if there are existing connections to the node:

  ```
  $TNODE(n),NAME=name
  ```

  There are two cases for this. **$TNODE(num),NAME=name** implies a new node and any existing connection records for the node are deleted. **$TNODE(oldname),NAME=newname** implies the node is being re-named and any connection records for the node are retained (and updated with the new name).

Warm start processing

Warm start processing is updated to properly handle new scenarios of a JES2 hot start with TCP/IP NJE address spaces active. JES2 must be able to reconnect to the existing TCP/IP NJE address space and properly handle jobs that were active on TCP/IP NJE. This includes jobs that were being transmitted as well as jobs that were being received. The code must also handle the case that while JES2 was down, the TCP/IP NJE address space terminated. Code must also deal with the case where the LINE or NETSRV associated with the TCP/IP NJE address was not specified in the initialization deck.

JES2 control blocks

The basic data structure at the LINE level in the JES2 address space has not changed. There is a DCT at the NETSRV level and the LINE level. Then each subdevice also has a DCT associated with it. The PCEs that represent the subdevices still exist.

12.2.5 Comparing TCP/IP NJE and SNA NJE

From an external view, TCP/IP NJE will operate very similarly to the current SNA. In SNA, there is an external called a LOGON that represents the JES2 connection to VTAM. In TCP/IP NJE the analogous structure is the NETSRV (Network server). This represents the connection from JES2 to TCP/IP. It also represents a TCP/NJE address space. A NETSRV initialization statement is created, along with the corresponding $ADD, $S, $T, $E, $P, and $D commands. The NETSRV initialization statement will also support a START=YES option. This automatically starts the TCP/IP NJE address space when JES2 is started. Up to 999 NETSRVs can be created. See Table 12-1 for a comparison of TCP/IP NJE and SNA.

In the JES2 SNA paradigm, there is an external called an *APPL*. This represents the correspondence between an NJE node (8 character name) and a VTAM APPLID. There exists an APPL for each LOGON and for each VTAM APPLID that JES2 can connect to over SNA. In TCP/IP, the analogous structure will be called a *SOCKET*. The socket will associate a node name with a IP address and port. It will also specify attributes for the connection (NJE resistance, line association, NETSRV association). A SOCKET initialization statement will be created, along with the corresponding **$ADD**, **$T**, and **$D** commands.

LINE externals will be extended to include TCP/IP constructs. Currently, UNIT= on a line can be a device address or “SNA”. UNIT= is now extended to support UNIT=TCP. TCP lines can only be used to support TCP/IP NJE connections. NJE sub-devices (SYSOUT receivers, SYSOUT transmitters, Job receivers and Job transmitters) will continue to be associated with TCP lines. The externals for sub-devices are unaffected by this support. However, with this release, the total number of lines (BSC, SNA, and TCP) that can be defined is increased to 64K (per JES2 image). Commands and initialization statements are unaffected except to the extent that operands that are specific to SNA or BSC lines will not be supported on TCP lines.

Commands to start and stop networking will be similar to what exists today.

NETSRVs are able to bind to all IP stacks available on the current system or a specific address. As part of this support, the code should support multiple stacks, VIPA, and Sysplex Distributor.

TLS/SSL support is available using the TCP/IP transparent TLS support. This configuration is done using TCP/IP and SAF/RACF.

Table 12-1 Comparison between SNA and TCP/IP NJE statements

SNA description	TCP/IP NJE description	Structure description
LOGON*(nnn)* • Open APPLID to VTAM • Represents the JES2 connection to VTAM	NETSRV(*nnn*) • Open SOCKET to TCP/IP and TCP/IP NJE server AS • Represents a connection from JES2 to TCP/IP	Connecting to network product
APPL(*name*) Associate an NJE node with a VTAM APPLID	SOCKET*(name)* Associate a node name with a IP address and port	Maps NJE node name to network construct
LINE(*nnnnn*) UNIT= SNA or a device address Connection to another node	LINE(*nnnnnn*) UNIT=TCP Connection to another node	Logical JES2 network line connection
Lnnnnn.SRn, Lnnnnn.STn, Lnnnnn.JRn, Lnnnnn.STn	Lnnnnn.SRn, Lnnnnn.STn, Lnnnnn.JRn, Lnnnnn.STn	Networking sub-devices

12.2.6 TCP/IP NJE initialization statements

For TCP/IP networking, there are new definitions in the JES2 initialization statements.

NETSRV statement

The NETSRV statement (which may also be specified as NETSERV or NSV) defines the characteristics of the network server. It must point to a socket that describes the local node:

```
NETSRV(nnn) SOCKET=,STACK=, TRACEIO=(JES=NO,COMMON=NO,VERBOSE=NO)
```

The parameters have the following meanings:

- SOCKET=

  Points to a SOCKET(xxxxxxxx) statement defining the local node's IP address and port.

- STACK=

  Indicates a specific TCP/IP stack to use. Default is all stacks.

- TRACE =
  - JES=YES/NO

    Traces communication within JES2 (between JES2 and NETSRV address space), and from JES2 to the common code.

  - COMMON=YES/NO

    Traces events and communication within the common component (such as TCP/IP API calls).

  - VERBOSE=YES/NO

    Issues additional messages to console/SYSLOG for diagnostics on live system.

LINE statement

The LINE statement represents a logical line used for networking. Lines used for NJE over TCP work like lines used for NJE over SNA. The major difference is the UNIT= parameter is set to TCP.

The maximum line number supported is increased from 32767 to 65535.

- UNIT=TCP
  - It defines a line as TCP/IP line.
- TRACE =
  - JES=YES/NO

    Traces communication within JES2 (between JES2 and NETSRV address space), and from JES2 to the common code.

  - COMMON=YES/NO

    Traces events and communication within the common component (such as TCP/IP API calls).

  - VERBOSE=YES/NO

    Issues additional messages to console/SYSLOG for diagnostics on live system.

12.2.7 JES2 commands to define TCP/IP networking

The commands that can be used to control TCP/IP NJE devices and connections are defined in this section. Processing is similar to the commands used for SNA.

- **$ADD NETSRVnnn** - Creates a new NETSRV device (1-999).
- **$T NETSRVnnn** - Change attributes of NETSRV.
- **$S NETSRVnnn** - Starts NETSRV address space.

  The **$S NETSERVx** command is used to create the NETSERV address space. The name of the address space is based on the owning JES2 subsystem name and the associated NETSERV number. The PGM= for the address space is IAZNJTCP. ASCRE is used with the IEESYSAS PROC to create the new address space.

  The address space name is *jesx*S*nnn*, where *jesx* is the subsystem name and *nnn* is theNETSRV number.
- **NETSRVnnn SOCKET=** - Defines the IP addr/port of this node.
- **$P NETSRVnnn** - Drains the NETSRV address space.
- **$E NETSRVnnn** - Resets connections in the NETSRV address space.
- **$ADD SOCKET** (socket) - Creates a new socket definition.
- **$T SOCKET** (socket) - Modify attributes of a socket.
- **$SN,S=socket** - Start networking with the specified socket.

  $SN,N=nodename - If NODE(nodename) LINE= points to TCP/IP line.

  The **$SN** command now supports starting networking connections over TCP/IP NJE. In addition to supporting the current **$SN,N=node**, a new form **$SN,SOCKET=name** (or S=name) is supported to connect the specified SOCKET and associated NODE.
- **$P NETSRVx** - Queues request to NETSRV address space to shut down cleanly.
- **$E NETSRVx** - Queues request to NETSRV address space to terminate all connections.
- NETSRV address spaces may be cancelled or forced if unresponsive to commands, as follows:
  - **P JES2S001**
  - **CANCEL JES2S001**
  - **FORCE JES2S001**

Examples

Figure 12-4 on page 320 shows initialization statements defining a simple 2-node TCP/IP NJE network. Numeric IP addresses were used in this example, but canonical names also work. In this case the socket names were set the same as the node names.

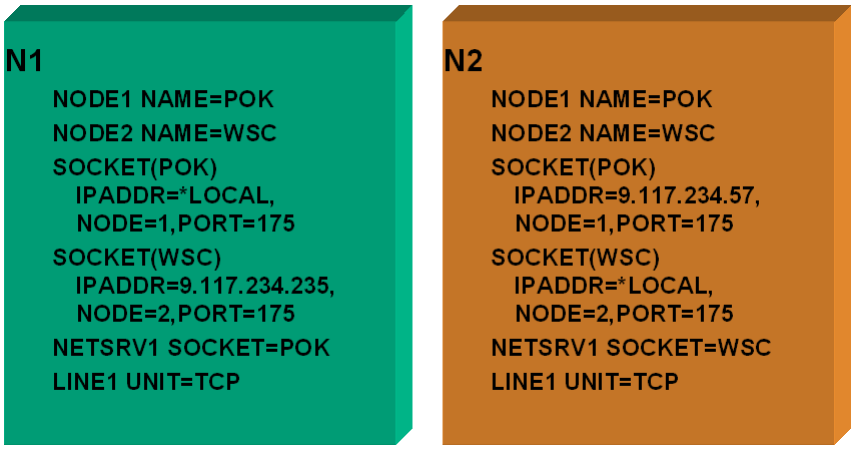


Figure 12-4 JES2 initialization statements for two communicating nodes

Commands to start TCP/IP networking

Figure 12-5 shows a series of commands to start the connection. The **$SNETSERV1** command starts the NETSERV address space. The **$SLINE1** commands prepare the lines for use. The **$SN,S=WSC** actually begins the exchange of records that results in the sign-on completing.

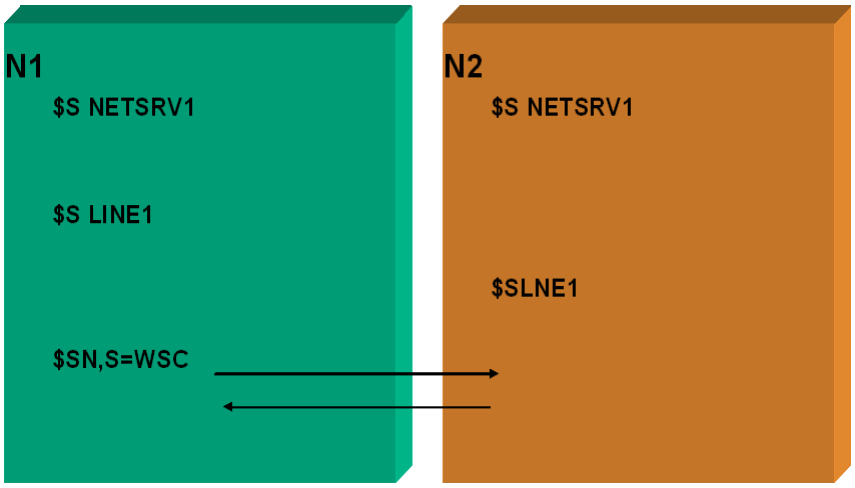


Figure 12-5 Two nodes establishing communication

12.3 MAS configuration considerations for TCP/IP NJE

A multi-access spool configuration (MAS) can participate in the JES2 network. JES2 considers the spool and its attached members to be one NJE network node.

Optionally, you can direct commands and messages to a particular JES2 member attached to the spool. Use *$Nnnn*M*mm* to route a command to a specific member, where *mm* is the member number as defined on the MEMBER(mm) statement.

The MASMSG= parameter on the CONDEF statement specifies the maximum number of commands or messages that can queue between any two members of the multi-access spool configuration before the network discards messages. This count also applies to the number of input messages or commands a member can queue to itself. When a member of a multi-access spool node queues a message to another member of that node, there is no

interrupt mechanism to inform the receiving member that a message exists. The receiving member periodically reads the shared job queue record and examines queue information for new messages.

MASDEF hold parameter

The following JES2 initialization parameter, on MASDEF initialization statement, control the length of time it takes for the receiving member to recognize the queuing:

```
HOLD=value
```

This parameter specifies the length of time the sending system holds the hardware reserve lock on the job queue. Small values increase system overhead while increasing responsiveness to queuing. If the posting of work occurs early in the time slot represented by *value*, then the receiving system might not retrieve the job queue in its attempt to read because the sending system still holds the lock.

MASDEF dormancy parameter

The MASDEF dormancy parameter is specified as follows:

```
DORMANCY=(mmmm,nnnn)
```

mmmm specifies the minimum length of time the receiving system will not consider requests to the job queue. This will also delay the sending system because it must own the job queue to queue the message.

nnnn specifies the maximum length of time between requests for the job queue by the receiving system.

Small values for both increase system overhead while increasing responsiveness to queuing.

The MASDEF statement defines various default values for a multi-access spool configuration. When a member of a multi-access spool configuration participates in a network, that member must have control of the shared queues before it can transmit or receive network data. Careful consideration when specifying these parameters will help the performance of your communications within the network.

The $T operator command can modify each of these parameters to make it easier to tune your multi-access spool node. You should experiment to determine the parameter values that yield satisfactory responsiveness without causing adverse increases in system overhead for the individual nodes. The values mentioned here are for illustrative purposes and are not recommendations.

```
DELAY=value
```

Figure 12-6 NJEDEF statement

Defining a multi-access SPOOL node (TCP/IP considerations)

For TCP/IP connections, an installation has a number of options when multiple MAS members connect to the network. An installation can define a unique IP address and port for each member of the MAS or for each NETSRV(nnn) in the MAS. In this way, connections into this node can be directed to a specific NETSRV(nnn) instance within the MAS. One advantage is that if you configure TCP/IP correctly, you can ensure there are multiple redundant connections between 2 nodes and lessen the impact of a failure (of a system or a piece of networking hardware) on your network. But you may not be able to make a connection when there is a problem with the target NETSERV(nnn). Another method uses

Sysplex Distributor to provide a single VIPA address for a sysplex. In this case, multiple NETSRV(nnn) address spaces (on multiple MVS images) will listen to the same IP address (VIPA). The connecting system only knows one address for the target system and Communication Server, with the assistance of WLM, will determine which NETSRV will handle the connection. An outage of a system or some network hardware will not impact the ability to establish an NJE connection. However, if there is a system outage, the active NJE connection that was assigned to that system will be terminated. New connections can then be established. Using this method, you cannot ensure that two parallel connections between two nodes will use different NETSRVs on different members, thus losing some redundancy, but it makes a simpler network setup and ensures that if at least one NETSRV is active, an NJE session can be established.

Internal reader enhancements

One of the goals of the TCP/IP NJE support is to be able to perform as much processing in the TCP/IP NJE address space as possible. This includes processing currently performed in HASPRDR. Since job input processing can now be performed in an address space other than the JES2 address space, internal reader processing will be updated to use this support. This implies that for all intents and purposes, internal readers as we know them will no longer exist.

12.4 Support for long SYSIN record lengths

SYSIN data set records were limited to a size of 254 bytes for JES2 and 4040 bytes for JES3 because the code in input processing for both JESes did not support creating spanned records. However, TCP/IP NJE processing for SYSIN streams and SYSOUT streams is the same. Thus long records are automatically supported. In addition, the JES2 internal reader enhancement will use the same spooling process that TCP/IP NJE uses, so long SYSIN streams will be practically free for internal readers and TCP/IP NJE streams in JES2. This support will allow a record size of up to 32760. A new feature flag will be added to the NJE sign-on NCC record to indicate support of long SYSIN records. JES2 will set this for TCP/IP NJE sessions only. If data is being sent to a node that does not support long records, the record will be sent truncated. Truncation is the standard procedure for cases where record size limits are not the same.

Support will also be added to SPOOL OFFLOAD to OFFLOAD and RELOAD job streams with long SYSIN records. This is the same logic used to send records over NJE to other nodes using BSC and SNA. As a result, SNA and BSC will also support sending and receiving long SYSIN records. As a further compatibility consideration with mixed levels in a MAS, the large SYSIN support for HASPRDR and HASPNJT is included in the compatibility APAR for z/OS V1R2, V1R4, and V1R5.

Implications of input processing changes

As stated previously, input processing for internal readers and TCP/IP NJE will be moving out of the traditional processing in the JES2 address space (HASPRDR) to processing in the NETSERV or user address space.

Error messages for syntax problems will now be more specific, indicating not just that there was a problem on a particular statement, but also stating what the problem on that statement was (as best we can detect it).

Internal reader support no longer buffers data in a data space before passing it to HASPRDR. As a result, a job will be made available to JES2 as soon as the next JOB card is seen (and not when a buffer is filled). This is slightly different than previous releases. Processing when a /*EOF card is written or when an ENDREQ is issued is not altered in this release.

Since processing of jobs on an internal reader is now done in the user's address space, the CPU associated with submitting the job will be assigned to the submitter and not JES2. This can affect accounting information (SMF records) for applications that submit large numbers of jobs. Furthermore, the priority of the submitting job will affect the speed at which a job is submitted. Lower priority jobs will not be able to submit jobs as fast as higher priority jobs.

Messages issued by input processing for internal reader processing will be placed in the JOBLOG of the submitting job as well as in SYSLOG and the JOBLOG of the submitted job (as they appear until z/OS V1R6). This allows a user to examine his own SYSLOG for information on jobs the user submits.

In previous releases, if the JES2 address space fails while jobs are being written to the internal reader, the jobs active on the reader are purged (including jobs that have been buffered in the data space). Now, jobs will not be lost over a JES2 hot start (except in rare timing situations where the job was submitted but not hardened to checkpoint before the JES2 failure).

12.5 NJE security considerations

In networking, you must:

- Secure the node, which involves ensuring the proper nodes and users have access to the network
- Secure the data, which involves ensuring that if an unauthorized node or user ID intercepts the data, that data is not usable by the interceptor

You should protect your resources at different levels. Security mechanisms include:

- RACF (or an equivalent security product) - Each node in the network should use RACF to protect its local resources. See *z/OS JES2 Initialization and Tuning Guide,* SA22-7532, for information on implementing security for your network.
- SSL (secure socket layer) - NJE/TCP network can have SSL or Transport Layer Security (TLS) defined for nodes.
- Encryption.
- JES2 passwords.
- JES2 exits.
- Security authorization facility (SAF) exits.
- SMF exits.
- MVS exits and modifications.

In larger networks, implementing security is most commonly done at the points where subnets interconnect, besides the individual node security. Therefore, it is important that the gateway nodes in the network enforce the proper level of security. NJE installation contains a complete discussion of security in large networks.

12.5.1 Password processing

JES2 defines passwords associated with NJE lines and with nodes in the NJE network during initialization. The way you code passwords varies depending on the type of protocol. Specify line passwords using the PASSWORD= parameter of the LINE(nnnn) initialization statement for BSC lines.

If you are using SNA lines, *do not* specify a line password for NJE. Use the NODE(*nnnn*) initialization statement to specify a node password for each node with which another node can communicate in your network.

As an alternative to traditional NODE passwords, you can use NODE(nnnn) SIGNON=SECURE along with an APPCLU profile as a secure form of NJE sign-on. It does not exchange nodal passwords in clear text.

With TCP/NJE you can provide passwords as a process of creating digital certificates to do SSL secure networking. You can use the SECURE= parameter on socket definitions to exploit the SSL security facility of TCP/IP.

Installations can encrypt passwords for jobs sent through the network. Use RACF on the submitting node to encrypt a password that is then verified by RACF on the job's execution node. Before sending jobs with encrypted passwords through the network, ensure that the execution node supports encrypted passwords. Use the PENCRYPT= parameter on the NODE(nnnn) statement to indicate the nodes that support encryption. See *z/OS JES2 Initialization and Tuning Guide,* SA22-7532 for a description of the PENCRYPT= parameter.

Encryption

You can encrypt a particular line (in which case everything sent on that line is encrypted) or a particular transmission. End-to-end encryption is the process of encrypting a teleprocessing line. You encrypt a teleprocessing line by using cryptographic modems. When using ACF/VTAM, there are software products you can use to encrypt individual sessions. You can also use software products to encrypt specific transmissions before sending them through the network. In this case, the receiver must have the same product and the encryption key to decode the data. When using TCP/IP for doing NJE, you can define a policy agent for the network and exchange digital certificates between nodes in network.

Using secure sign-on protocol for NJE sign-on

You can define passwords to JES2, associated with NJE lines and with nodes in the NJE network during initialization. However, using nodal passwords has the following notable drawbacks:

- The passwords are exchanged across the network in clear text, which could compromise the security of the password.
- The passwords are defined and maintained in the JES2 initialization stream by the JES2 system programmer, rather than in the system's security policy (that is, RACF) by a security administrator.

The secure sign-on protocol allows greater password security. In order to take advantage of this protocol, you need to specify the following:

- On the local node, specify **NODE(*node2*) SIGNON=SECURE** to indicate that the secure protocol is to be used when signing on to node *node2*
- Specify **RDEFINE APPCLU NJE.*node1*.*node2* SESSION(SESSKEY(*key*)) UACC(NONE)**, where:
  - *node1* is the name of the local node.
  - *node2* is the name of the adjacent node.
  - *key* is an agreed upon session key for the connection.
- Issue **SETROPTS CLASSACT(APPCLU)** to activate the APPCLU security class.

The node at the other end of the connection must define this setup as well.

SAF/RACF

One of the enhancements being made is secure sign-on protocol using SAF/RACF. This new secure sign-on protocol was added to provide additional authentication of NJE partners.

The new protocol applies to TCP/IP as well as BSC and SNA sign-ons. This function uses SAF/RACF APPCLU to encrypt a string passed to the other member for authentication, and the validation is the same as that used by other exploiters of the APPCLU class. It exchanges DES-encrypted passwords in I/J sign-on records. The entity used is:

```
NJE.node1.node2
Uses SESSKEY associated with profile for encryption
```

Figure 12-7 shows the definitions needed to establish a secure sign-on. The N1 and N2 in the profile names are the NJE node names of the these nodes. The key that is specified must match on both nodes. The sign-on can be completed only if both keys match.

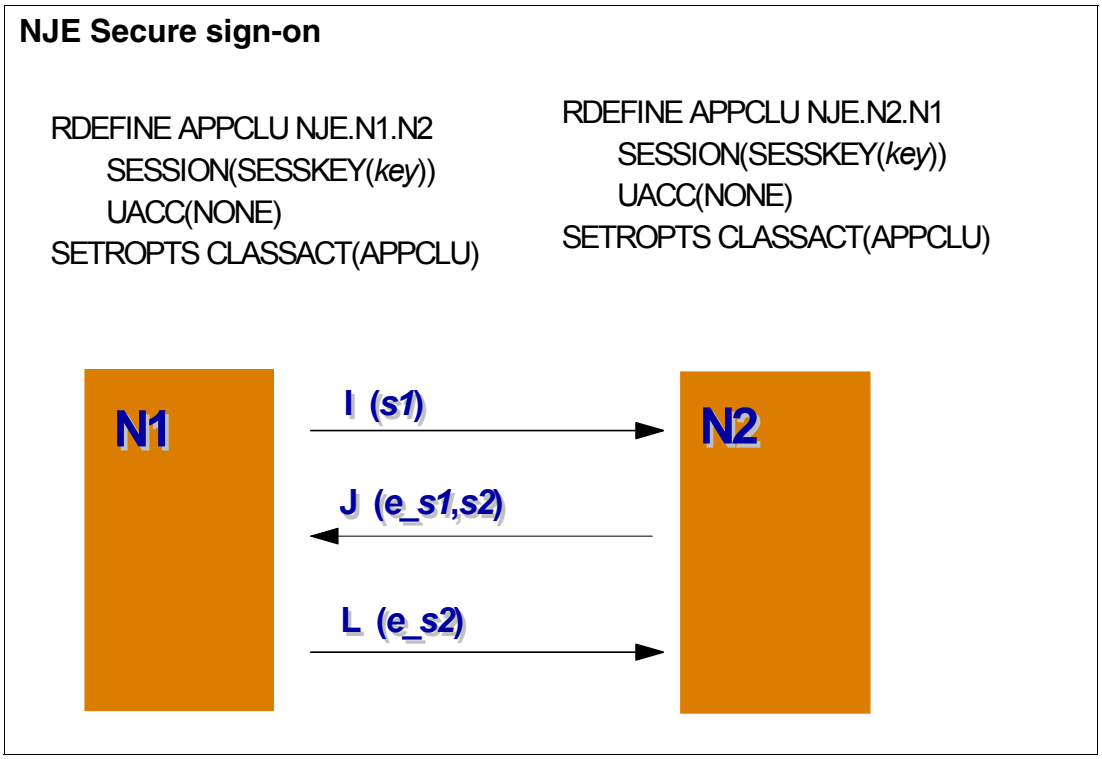


Figure 12-7 NJE Secure sign-on

TLS/SSL support

For NJE/TCP connections, JES2 will exploit the application transparent Transport Layer Security support introduced in this release by TCP/IP. The setup for this is controlled totally by TCP/IP and SAF/RACF. The only JES2 specification is the SECURE=YES/NO on the socket statements. To use this support, the TCPCONFIG statement must be updated, a policy agent must be defined, and the appropriate certificate and key rings set up.

12.6 NJE EXITS

The changes made on NJE to implement TCP/IP make it impossible to call the traditional HASPRDR exits in the JES2 main task. Similarly, changes to internal reader processing also make it impossible to call the traditional HASPRDR exits in that environment. To address this,

a new set of input processing exits has been defined. These exits run in the user environment in the NETSERV address space. In addition, a new exit (exit 51) was defined in the main task when jobs change phase. Data can be passed to exit 51 from other exits. Exit can be used as the ultimate end of input exit in the main task for all input sources.

For control block I/O, since the I/O is being done outside the main task, exit 8 instead of exit 7 is called.

For other exits, the change was more subtle. For exits 36 and 37, the exits are called, but in a different address space (was JES2 and is now the NETSERV or USER address space).

Figure 12-8 shows a list of the affected exits.

Exit	Function	Action
2	Input processing - JOB card	Additional exit defined
3	Input processing - Accounting field	Additional exit defined
4	Input processing - JCL/JECL	Additional exit defined
7	JES2 main task control block I/O	Document changes (exit 8 called for cases when exit 7 was called)
8	Non-JES2 main task control block I/O	Document changes (exit 8 called for cases when exit 7 was called)
13	NJE Mail notification	Obsolete (delete the exit point)
20	Input processing - End of input	Additional exit defined
36	Pre-SAF exit	Called in different address space
37	Post-SAF exit	Called in different address space
39	NJE SAF rejection	Additional exit defined
46	NJE header/trailer transmit	Additional exit defined
47	NJE header/trailer receive	Additional exit defined
49	$QGET veto exit	Now called for $S J (Start Job) processing to assist in exit migration

Figure 12-8 Exits that have been affected

New exit numbers were defined for exits that need to be called outside the main task. Figure 12-9 shows a list of the new exit numbers, the similar old exit, and the environment of the new exit.

New Exit	Similar exit	Environ	Function
50	20	USER	End of input
51	*	JES2	$QMOD - job phase change
52	2	USER	Input processing - JOB card
53	3	USER	Input processing - Accounting field
54	4	USER	Input processing- JCL/JECL
55	39	USER	NJE SAF rejection
56	46	USER	NJE header/trailer transmit
57	47	USER	NJE header/trailer receive

Figure 12-9 List of new exits

All exits pass XPLs. Old exits, described on Figure 12-8, have XPLs available as well as the current input registers. The XPLs for the new and old exits have the same data (but separate mappings). Some data areas that were in PCEs have been moved to new data areas that are common to both environments. The XPL formalizes some of the interfaces and simplifies some of the tasks commonly performed in each exit.

- Old style exits (2, 3, 4, 20, 39, 46, 47) are still used for:
  - Local card readers
  - RJE readers
  - SNA and BSC NJE transmitters and receivers
  - Spool Offload transmitters and receivers
- New style exits (52, 53, 54, 50, 55, 56, 57) are used for:
  - Batch Internal readers
  - STC and TSU internal readers
  - TCP/IP NJE transmitters and receivers
- New exit 51 receives control for all phase changes:
  - Job moves from $INPUT to $XEQ
  - Job requeued for execution

Figure 12-10 on page 328 is an example of input processing of a job from a card reader. In this case, the old exits are used.

Figure 12-11 shows the input processing of a started task. In this case the new exits are used.

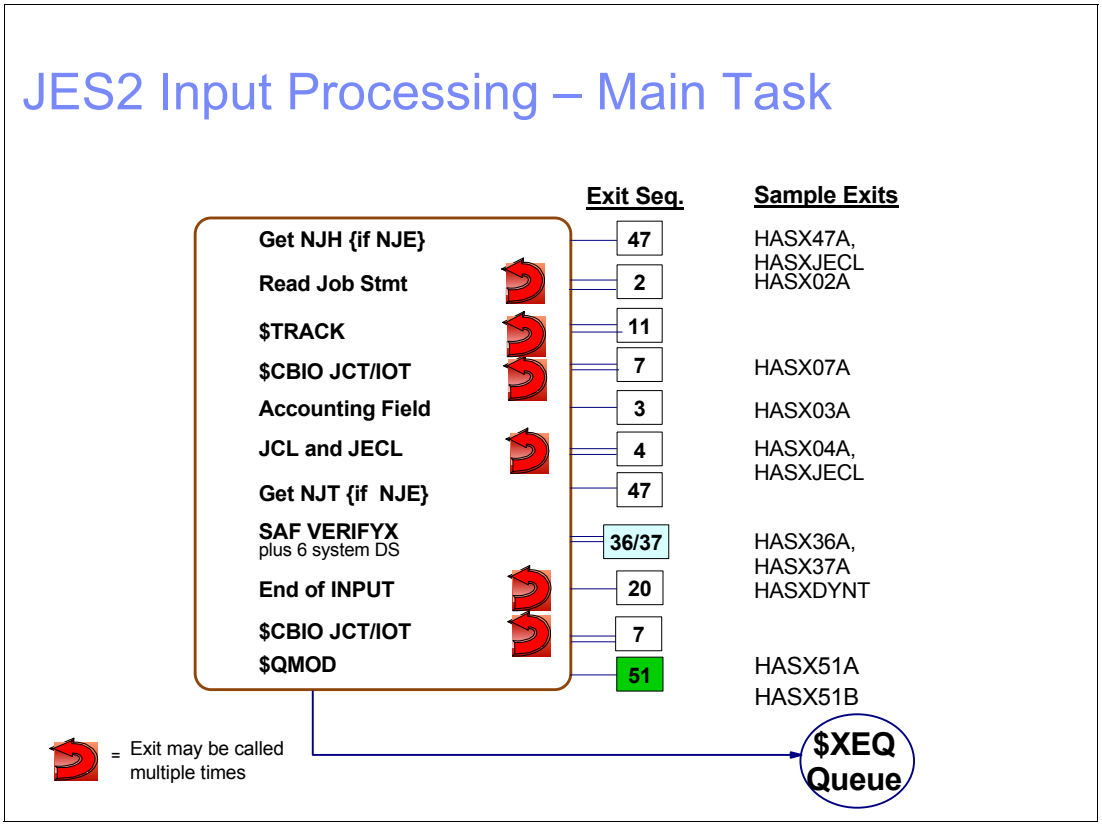


Figure 12-10 JES2 input processing of a job from a card reader (uses the main environment)

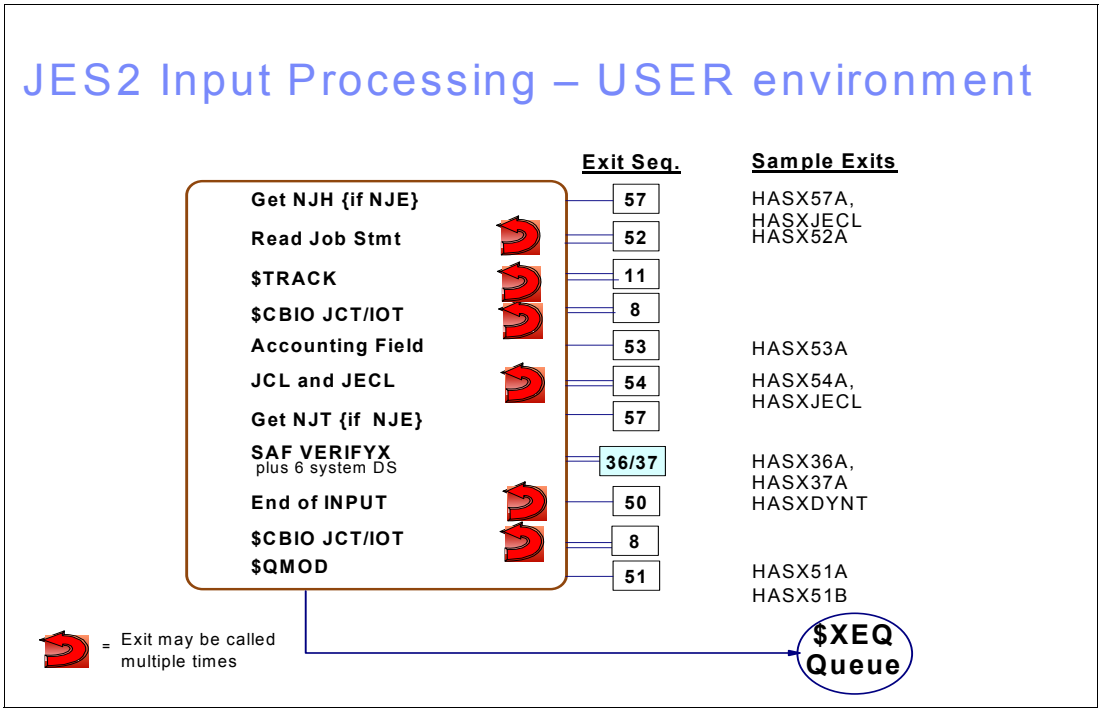


Figure 12-11 JES2 input processing of a started task (uses a USER environment)

New assembly environment

As part of implementing TCP/IP NJE, many of the existing service routines that were used for BSC and SNA needed to be made available for use in the new NETSERV address space. This included the services for input processing. To simplify the processing, a new JES2 assembly environment was created called USER ANY.

The $SAVE and $RETURN macros use either the main task or user save services based on the environment that is currently active. This allows access to the PCE in the main task as well as access to services such as $WAIT. General purpose code can be written to run in either environment by using a few special purpose, environment sensitive routines (such as FREEJCT) that perform environment specific tasks, minimizing the coding and maintenance effort.

This is a very effective way to implement a single exit routine that can run in multiple environments. A number of sample exits use this technique.

See *z/OS JES2 Installation Exits,* SA22-7534 for more details about the JES2 exits changes.

13

Tools and service aids

This chapter describes the changes to the tools and service aids that MVS provides for diagnosis. Tools include dumps and traces, while service aids include the other facilities provided for diagnosis. The following tools and service aids are been enhanced in z/OS V1R7:

- SPZAP
- SADMP
- SDUMP
- System trace
- External traces (GTF and CTRACE)
- SLIP
- IPCS

13.1 Overview of enhancements

For z/OS V1R7, the following enhancements have been made to the tools and service aids:

SPZAP SPZAP has been enhanced to support DSNTYPE=LARGE data sets. DSNTYPE=LARGE data sets are like conventional sequential data sets except for the fact that they may span more than 64K tracks per volume.

SADMP SADMP is the most fundamental diagnostic tool and the focus in z/OS V1R7 is to get SADMPs captured quickly and effectively when they are needed. Installations that are enlarging the sizes of their LPARs should consider the effect on SADMP production and analysis in their planning.

SDUMP SDUMP is the preferred dumping tool in MVS via its many faces: **DUMP** command, SYSMDUMP, and transaction dump. SDUMP is improved in a number of areas and have also focused on better analysis aids, partly to help the traditional audience of systems programmers and vendor support personnel and partly to help traditional users of formatted dumping tools who are migrating to unformatted dumping at an increasing rate in the last several years.

External trace The changes for external trace writing support last received attention around 1990. With the increased system speed, complexity, and size, external trace support again requires focus. The first phase of improvements is being delivered in z/OS V1R7.

SLIP For SLIP, improvements have been included in z/OS V1R7 to make it easier to trap circumstances where dumping, tracing, or related actions need to be taken.

IPCS For IPCS, enhancements in z/OS V1R7 include support for large block sizes, compression, and striping. You can limit the scope of analysis with the PROFLE command, and report handling is enhanced to enable you to focus only on pertinent information.

13.1.1 Migration considerations

Make sure to consider the following migration issues associated with the changes made in z/OS V1R7 to the tools and service aids.

AMDSADDD REXX exec

The AMDSADDD REXX exec has been moved from SAMPLIB to ABLSCLI0. Update any procedures that reference it.

AMDSADDD allocates and maintains SADMP dump data sets. It can be used as a migration aid as described in *z/OS MVS Interactive Problem Control System (IPCS) Customization*, SA22-7595.

IPCS subcommands

On most IPCS subcommands that support ASID selection criteria, the defaults for address space selection have been changed from both CURRENT and ERROR criteria to solely the CURRENT.

This allows many IPCS problem screening and component analysis functions to default to running much faster in z/OS V1R7.

The ERROR criteria is eliminated as a default but it is still available to choose. Its use would rarely add value to dump analysis and did cause significant delays when working with some

dumps. If your IPCS users have specific procedures where use of ERROR criterion to select ASIDs remains valuable, update those procedures to explicitly request the option.

IEBGENER or COPYDUMP subcommand

To copy a dump to a data set, the recommended method is IPCS COPYDUMP. IPCS COPYDUMP can run without a dump directory being employed. Use the DEFER option when initiating the IPCS session to tell IPCS to defer accessing a dump directory until one is required.

In z/OS V1R7, IPCS COPYDUMP has the ability to merge the records from a multi-volume SADMP and recapture the prioritized order used by SADMP to get the most important data into the dump data sets first. If SADMP is allowed to complete normally, IEBGENER and similar transcription programs can produce a logically complete dump data set that IPCS can process. However, IPCS performance, particularly IPCS dump initialization, will degrade as more volumes are added to the SADMP data set.

Note: SADMP to DASD can exhaust the pre-allocated space associated with the initial data set. You can designate second and subsequent data sets to cause a complete SADMP to be written. COPYDUMP accepts a list of data set names and can create a single dump data set for analysis from the several dump data sets to which SADMP wrote.

SADMP to DASD can exhaust the pre-allocated space associated with the initial data set. You can designate second and subsequent data sets to cause a complete SADMP to be written. COPYDUMP accepts a list of data set names and can create a single dump data set for analysis from the several dump data sets to which SADMP wrote.

13.2 SPZAP enhancements

The only change to SPZAP is to support DSNTYPE=LARGE. This is to handle viewing and altering DSNTYPE=LARGE data sets identified by the SYSLIB ddname.

No JCL or control statements need to be changed and no existing functions have been modified or changed.

SYSIN and SYSPRINT data sets may also be DSNTYPE=LARGE. The DSNTYPE=LARGE support is transparent to any use of SPZAP.

13.3 SADMP enhancements

SADMP can dump to extended format data sets and to DSNTYPE=LARGE data sets. However, it does not support striping or compression for the extended format data sets to which it writes.

- DSNTYPE=LARGE data sets are supported and parmlib default options may limit their use to a small group of z/OS applications.
- The extended format data sets can span more than 64K tracks per volume. We recommend using extended format data sets until all your systems are running z/OS V1R7.

When you enter IPCS, from the primary panel select Option 3. The panel shown in Figure 13-1 on page 334 is displayed. The IPCS dialog SAMDP dump data set utility (option 6) is new with z/OS V1R7.

There are no changes required to create a SADMP program.

```
--------------------------------- IPCS UTILITY MENU  -----------------------
 OPTION  ===>
                                                                 ******************
    1  COPYDDIR    - Copy dump directory data                    * USERID  - ROGERS
    2  COPYDUMP    - Copy a dump data set                        * DATE    - 05/08/23
    3  COPYTRC     - Copy trace data                             * JULIAN  - 05.235
    4  DSLIST      - Process list of data set names              * TIME    - 12:21
    5  DAE         - Process DAE data                            * PREFIX  - ROGERS
    6  SADMP       - SADMP dump data set utility                 * TERMINAL- 3278T
                                                                 * PF KEYS - 24
 Enter END command to terminate                                  ******************
```

Figure 13-1 IPCS Utility Menu with new option SADMP

When you choose Option 6, the new SADMP DASD Dump Data Set Utility panel shown in Figure 13-2 is displayed. Here you can clear, define, or reallocate a SADMP dump data set.

```
------------------- SADMP DASD Dump Data Set Utility ------------------
 Command ==>

 Enter/verify parameters.
 Use ENTER to perform function, END to terminate.

  Function  ==> R ( C - Clear, D - Define, R - Reallocate)
  DSNAME    ==>

  Volume serial numbers:  (1-32)
     1- 8  VOL001
     9-16
    17-24
    25-32

  Unit ==> 9345  (3380, 3390, or 9345)
  Cylinders ==> 500      (cylinders per volume)
  DSNTYPE(LARGE) ==> N (Y or N)

  Optional SMS classes: (May be required by installation ACS routines)
  StorClas ==>            DataClas ==>           MgmtClas ==>
```

Figure 13-2 Panel to clear, define, or allocate a SADMP dump data set

13.3.1 Using SADMP with z/OS V1R7

Many IPCS problem screening and component analysis functions default to running faster in z/OS V1R7. This is because the defaults for address space selection have been changed from both CURRENT and ERROR criteria to solely the CURRENT criterion. IPCS user experience has revealed that the use of the ERROR selection criterion rarely adds value to dump analysis, and it requires a broad survey of all ASIDs in the system to assess. Eliminating it as a default while keeping it available yields a more usable IPCS. If your IPCS users have specific procedures where use of ERROR criterion to select ASIDs remains valuable, update those procedures to explicitly request the option.

Operational changes

SADMP runs very much the same way as prior releases. From the perspective of the operator who runs SADMP, DSNTYPE=LARGE data sets are treated just the same as the ones used previously. The operational changes are as follows:

- SADMP tries harder to ensure that data needed to process every SADMP is written to it early. Several page data set pages may be brought in concurrently to achieve this acceleration if independent paths are available.
- An alteration of some messages tells the operator about progress through the 3 phases, and, if the operator is very sensitive to such things, a modest acceleration of capturing data from page data sets may be sensed. Some messages are changed to reflect the following logic and inform the operator about the phases, as follows:
  - Primary phase dumps vital MVS data (PSAs, CVT, and so forth)
  - Second phase dumps ASIDs 1-4
  - Third phase dumps the rest

If installation priorities mandate cutting the dumping process short, this makes it more likely that the truncated dump will be useful. We do not recommend truncation, but we recognize that your specific business priorities may require it.

SADMP analysis considerations

When doing SADMP analysis, consider the following when processing dumps:

- IPCS analysis of dumps in place is not recommended for multi-volume dumps to DASD.
- Use IPCS COPYDUMP since it produces a dump that IPCS can process more efficiently than one copied by IEBGENER or similar programs. See “IPCS COPYDUMP” on page 336.
- Use a compressed extended sequential data set as a target. This could save about 40% of DASD for large data sets.
- Ensure large CISIZE for DATA portion. BLSCDDIR CLIST is updated to help. A DSNTYPE=LARGE data set can only be used if the dump is both written and processed on an V1R7 system or a later release. A VSAM linear data set with either an extended format or conventional format with a control interval size (CISIZE) of 32K can be substituted. Neither extended sequential nor VSAM data sets, other than linear data sets with the required CISIZE, should be used.
- In addition, consider the following options:
  - Ensure large (but not excessive) BUFSPACE for the directory.
  - Consider striping.
  - Avoid compression because of intensive updating during IPCS analysis.

SADMP performance considerations

Figure 13-3 on page 336 shows an 87 GB dump, with SADMP, unloaded using IEBGENER. This performance test was to see whether dump directory performance could be improved by simply striping it. Appropriate SMS classes, with a dump directory striped 5 ways, was used to try to improve performance. The result was a dump initialization that completed in 36 minutes.

The version of IPCS with which all preceding runs had taken place was z/OS V1R6 IPCS. A dump directory striped 5 ways and using z/OS V1R7 IPCS resulted in a one third reduction in initialization time and brought it down to 24 minutes.

Dump initialization elapsed time (minutes)	IPCSDDIR characteristics
3600	4K CISIZE, V1R6 IPCS
54	24K CISIZE, V1R6 IPCS
36	24K CISIZE, 5 stripes, V1R6 IPCS
24	24K CISIZE, 5 stripes, V1R7 IPCS

Figure 13-3 Improvement in dump directory size for performance

Using IPCS COPYDUMP, dump allocation, and dump directory allocation are good ideas if you plan to run substantially larger LPARs.

13.3.2 IPCS COPYDUMP

IPCS **COPYDUMP** is enhanced as follows:

- Input may be a list of ddnames or dsnames to accommodate SADMP overflow.

  SADMP can fill one dump data set, ask the operator for another, and write overflow records to the second. It can also go from a 2nd to a 3rd and so on. IPCS **COPYDUMP** has been updated to accept a list of input data sets to bring such dumps back together for analysis.
- Original multi-volume SADMP detected:
  - All volumes accessed in parallel.
  - Records merged to recover SADMP placement of important data first.
  - DSNTYPE=LARGE supported for input and output.

Use **COPYDUMP** to copy the SADMP dump data sets from the data sets which they were initially written into to a second type of extended format dump data set. This makes the special repositories that the installation tends to set aside for SADMP use maximally available for reuse, and produces a dump data set that IPCS can process more efficiently. SADMP sees a multi-volume dump data set as though it were volume-count separate sequential repositories. DFSMS sees all records on volume 1 followed by all records on volume 2, and so on. Transcription multi-volume SADMPs using **COPYDUMP** reconciles the two views and produces a data set where the most important records appear early in the dump data set, not scattered across N volumes.

13.4 SDUMP enhancements

Prior to V1R7, when SDUMP was entered with the caller holding locks and requesting the SUSPEND SUMMARY option, capturing the status of the system trace table was deferred for a long period of time, which caused a loss of context at the time that the dump was requested. z/OS V1R7 SDUMP now schedules an SRB immediately in such a situation, reducing the possibility that this valuable diagnostic information will not be captured in time.

Prior to z/OS V1R7, SDUMP did not record the distinction between pages of storage that were not valid for use by an application and those that had been authorized for use but never

touched by the application. RSM uses the term “first reference status” for such pages. z/OS V1R7 SDUMP records the validity of such pages in dumped storage ranges using a technique employed by SADMP for a number of years – “zerodef” records that can compactly describe many all-zero pages of the same address space in a single record.

SDUMP was to set aside a limited amount of storage to hold information requested in the SUMMARY portion of a dump, but it has not made the amount of such storage known to applications prior to V1R7. V1R7 sets aside field RTCTSDSU for those sophisticated applications that may have a great deal of storage that they would like to capture in the SUMMARY portion – applications that have the further sophistication to know how to trim back large requests to ensure that room is available to capture the most important parts of that data.

Self-monitoring of the dumping process is introduced. SDUMP monitors the time spent in its data collection phases and adds the information to the dump data set. The following IPCS subcommand will format the statistics:

```
VERBEXIT IEAVTSFS /* Format statistics */
```

13.5 System trace

Several years ago zSeries processors began supporting not only a TRACE instruction but a TRACG instruction to record entries in the system trace table. There became a recognized need to increase the amount of information each GPR could place in a trace entry and a need to increase the precision of the time stamps recorded in new entries. User trace entries are the first that can exploit the TRACG instruction by having an application request TRACEMODE=TRACG on a PTRACE macro.

13.5.1 Trace instructions

z/Architecture Principles of Operation, SA22-7832 describes the branch instruction trace entries and the mode trace entries that MVS combines with them (and that are generated by the hardware). MVS enables or disables the production of these unformatted entries by manipulating control register bits by the instruction. The trace table entries that are not “branch (or mode)” entries are generated by MVS software through the TRACE or TRACG instructions.

The enhancements in the system trace with z/OS V1R7 are as follows:

- TRACG instruction used to capture user trace entry is enhanced to have:
  - More data per GPR
  - More precise time stamp
- The system trace table entries are mapped by IHATTE, which is now updated to describe the TTEE produced.
- Existing PTRACE macros continue to be supported and use TRACE, and are used as shown in Figure 13-4.

[label]	PTRACE	TYPE=USRn,…,SAVEAREA=F4SA,TRACEMODE=TRACG

Figure 13-4 System trace macro

Note: Because tracing branch instructions can significantly increase the number of trace entries being generated, you can increase the size of the trace tables from the default 256 kilobytes when you turn tracing on: `TRACE ST,999K,BR=ON`.

13.6 External traces

External trace writing support was enhanced significantly in 1990 and has been largely unchanged since that time. With z/OS V1R7, improved trace writing is implemented for much faster processors and more complex environments. Previous support was theoretically ample, allowing you to specify many ddnames to be used for external recording when trace entries were generated very fast. However, doing so meant that no single trace data set would show a whole picture of what happened. IPCS **COPYTRC** or **MERGE** operations would need to be used, and, if you were running a sysplex with several systems involved, more merging of information might be needed.

DFSMS has supported striping for several years, and modern processors are willing to rapidly compress data and reverse the compression process later. VSAM linear data sets are now used as the repository of choice for future tracing with z/OS V1R7. VSAM extended addressing is supported if you need very large trace data sets.

Some existing applications that directly read traditional trace data sets may not be able to handle the new VSAM linear data sets. IPCS **COPYTRC** can be used to transcribe trace entries from a VSAM linear data set to one suitable for use by those programs.

The following are the external traces enhancements:

- Common support for external traces in GTF, CTRACE, and IPCS are enhanced to support VSAM linear data sets with CISIZE of 32K.
- Performance addressed:
  - Large unit of data transfer.
  - Striping is supported.
  - Writer code compresses data.
  - IPCS decompresses for trace processing subcommands.
- VSAM extended addressing is supported if you need very large trace data sets.
- IPCS **COPYTRC** subcommand can be used to create traditional trace for programs written to directly process that format.

13.7 SLIP enhancements

With z/OS V1R7, the following changes have been made to SLIP processing:

- Changes to the PVTMOD and MSGID parameters on the SLIP command
- A new symbolic, BEAR, to hold the branch-from address

PVTMOD and MSGID

PVTMOD specifies the name of the load module as the entry point name.

With z/OS V1R7, the entry point name designated using PVTMOD may now be as long as 80 characters to allow more precise designation of HFS paths.

```
PVTMOD=(name,[start[,end]])
```

MSGID causes control to be passed to the SLIP action processor under the unit of work issuing the WTO when the MSGID of the WTO matches the message ID specified on the MSGID parameter. The message ID consists of the characters up to the first blank in the WTO, but is never more than 10 characters. The slip action processor gets control after SSI and MPF processing.

WIth z/OS V1R7, message identifiers are allowed to be placed in apostrophes where special characters employed by some products may be designated. Quoted message identifiers do not need to be terminated by a blank.

```
MSGID='a-msgid'
```

BEAR symbolic

SLIP supports a BEAR symbolic that is valid in the PER, RTM1, and RTM2 environments and yields zero elsewhere. BEAR holds the most recent branch-from address for a successful branch prior to the event that triggered entry into slip. The following example employs BEAR:

```
SLIP SET,ERRTYP=PROG,TRDATA=(BEAR?+4,+B),A=TRACE,E
```

The trap tells SLIP that when an erroneous program check occurs (for example, branch to low storage) this trap will match. Assuming GTF is active, the contents of +4 to +B from the last successful branch will be recorded in the trace.

13.8 IPCS enhancements

Files directly supported by IPCS may have the DSNTYPE=LARGE attribute in z/OS V1R7. If you are planning to run larger LPARs, it makes sense to set aside some time to plan for larger dumps and traces.

DSNTYPE=LARGE

The DSNTYPE=LARGE is supported in:

- Dumps
- Traces
- Other data sets viewed via RBA or BLOCK(n)
- Print file
- Table of contents file

Large dumps and traces

Large dumps and traces make performance more of a concern. So consider the following:

- Large BLKSIZEs, compression, and striping are all supported. Each can make a significant difference.
- Good allocation for dump directories can make a significant difference in IPCS efficiency. Compression is not recommended because directories are updated very rapidly during IPCS analysis, but focusing on primary space, secondary space, CISIZE, BUFSPACE, and striping can really help. If you anticipate the need to work with really large media, the VSAM extended addressing option should be used.

13.8.1 Limit Analysis

The PROFILE subcommand tells contention analysis and WHERE not to do processing whose value does not warrant the time needed to perform analysis. For example:

```
PROFILE EXCLUDE(ANALYZEABC:ANALYZEDEF +
WHERECSVCOMMON)
```

Sometimes IPCS can take too much time adding value to your reports. Subcommands that analyze contention may consider resources managed by resource managers that don't usually concern you with the dumps that come to you for analysis. The ability of IPCS WHERE to associate a common storage address with a module containing that address may not be sufficiently valuable for you to wait for all common area modules to be mapped. The z/OS V1R7 IPCS **PROFILE** subcommand gives you the ability to exclude this sort of processing, making it more responsive by doing less analysis.

The **EVALPROF** subcommand allows command procedures that you write to understand what PROFILE subcommand options are in effect.

13.8.2 Report handling

The following enhancements are added:

- IPCS report viewing now supports **EXCLUDE** primary command.
- The **FIND/RFIND** and **EXCLUDE** commands support all options against reports.

  Use this option with caution against very large reports.
- The **REPORT** primary command is added. It initiates processing of reports for the logical screen and does the following:
  - REPORT BROWSE/EDIT/VIEW - Shows report via ISPF services.
  - REPORT procedure-name - Command procedures supported.
  - REPORT EVALRPT - Makes data available to command procedure.

14

Console restructure

In z/OS V1R7, the console restructure stage 1B is available. It is the continuation of the work started in console restructure stage 1 and delivered as an update to z/OS V1R4.

The following reliability, availability, and serviceability (RAS) items and enhancements are introduced in stage 1B and are described in this chapter:

- EMCS console removal support
- Monitor message independence
- Console query interface
- 1-byte console ID elimination
- TRACK command elimination

14.1 Console restructure phase 1B

Console restructure Stage 1 (delivered in z/OS V1R4.2) was a Reliability, Availability and Serviceability (RAS) item. The consoles components were among the first exploiters of sysplex. The exploitation occurred before much of the sysplex software and hardware, such as ordered delivery and coupling facility, had been developed. The infrastructure of message processing (WTO and DOM) has been updated and enhanced to provide greater reliability and availability of the system and sysplex and to remove or reduce system outages caused by message floods.

14.2 EMCS console removal support

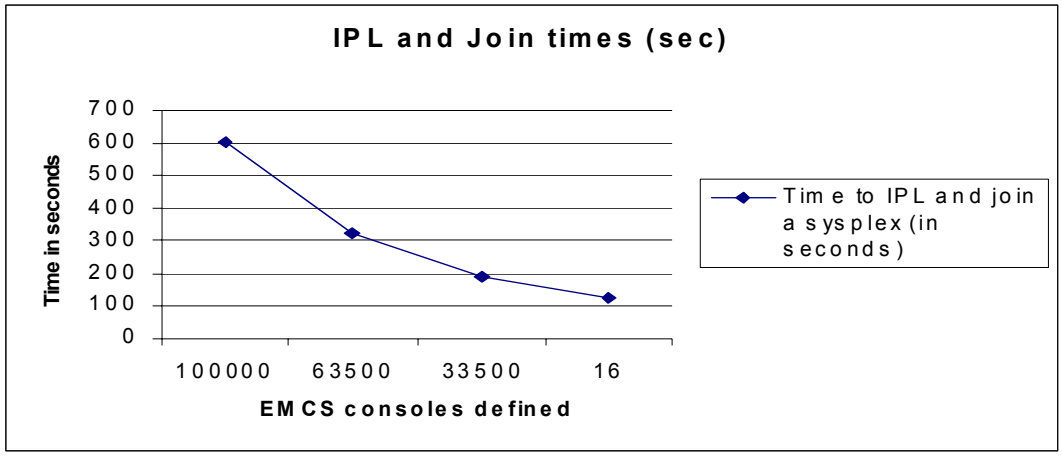
Information on all EMCS (extended multiple console support) consoles is sent across systems, including inactive EMCS consoles that are no longer needed. This resulted in long times for IPL and for console data to be refreshed in each system in a sysplex.

With this enhancement you are able to remove inactive EMCS consoles that are no longer needed. This reduces the system refresh and IPL times in a sysplex.

The impact of an IPL and of systems joining a sysplex when removing EMCS consoles on z/OS V1R7 was tested in four different situations, as follows:

1. Started with 100K consoles defined and had a system join the sysplex.
2. Removed 36.5K consoles and had a system join the sysplex with 63.5K consoles defined.
3. Removed another 30K and had a system join the sysplex with 33.5K consoles defined.
4. Removed 33484 consoles and had a system join the sysplex with only 16 consoles defined.

The results of these tests are shown in Figure 14-1.



Number of EMCS consoles defined	100000	63500	33500	16
Time to IPL and join a sysplex (sec)	605	324	187	122

Figure 14-1 IPL and join times

14.2.1 Migration and coexistence considerations

All systems in a sysplex must either be z/OS V1R7 or have APAR OA06857 installed on any level between z/OS V1R4 and z/OS V1R6.

Table 14-1 Coexistence PTFS

Release	PTF Number
z/OS V1R4	UA16035
z/OS V1R4.2	UA16037
z/OS V1R5	UA16038
z/OS V1R6	UA16036

Migration scenario 1

The following situation occurs if the systems in the sysplex do not follow the recommendations concerning the coexistence maintenance:

A system will partition from a sysplex if:

- A pre-z/OS V1R4 system joins a sysplex containing a z/OS V1R7 system.
- A pre-z/OS V1R7 system (without APAR OA06857) joins a sysplex containing a z/OS V1R7 system.

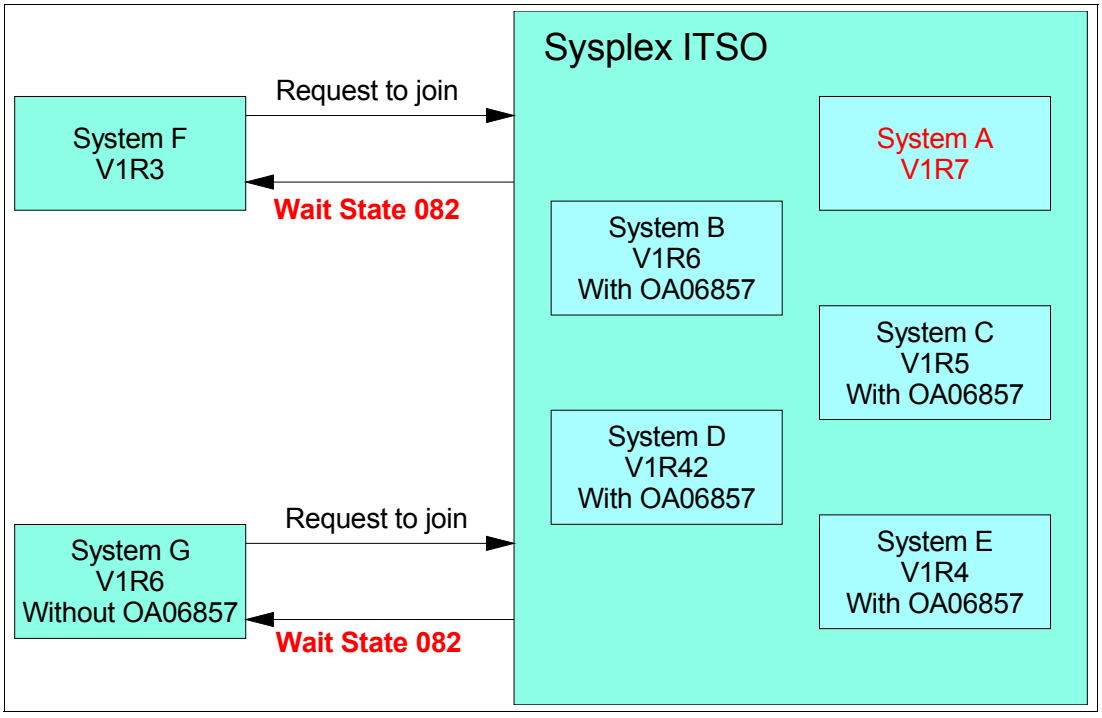


Figure 14-2 Systems requesting to join a sysplex

Migration scenario 2

A z/OS V1R7 system will partition itself out of a sysplex if it joins a sysplex containing a pre-z/OS V1R4 system.

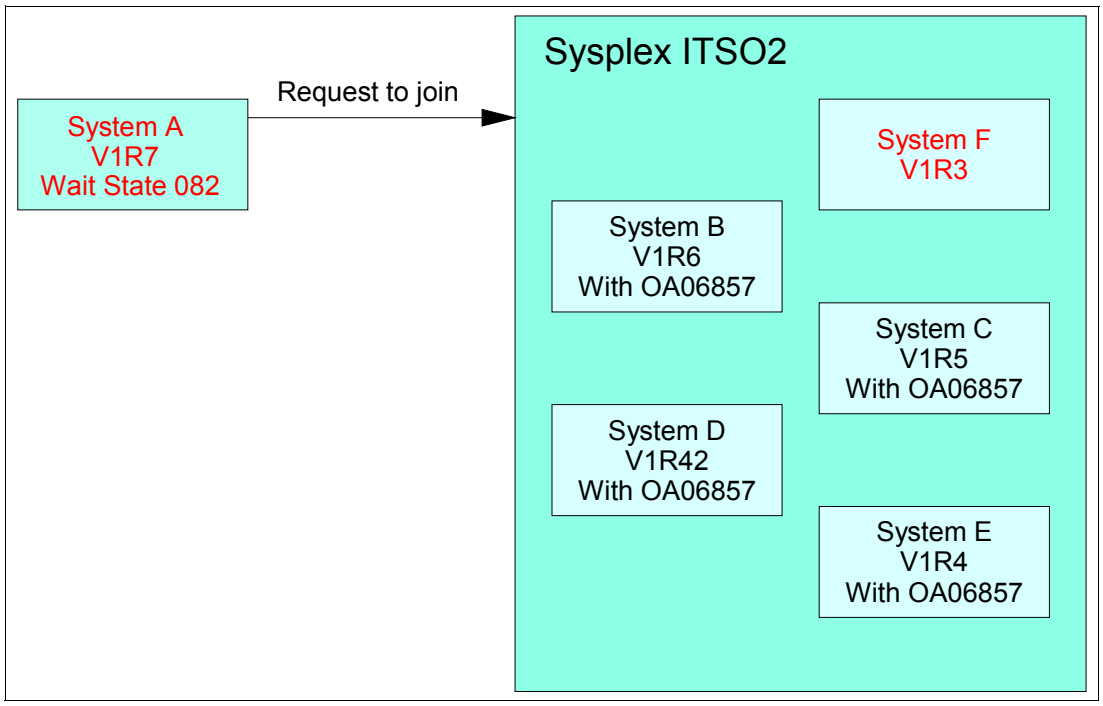


Figure 14-3 A V1R7 system attempting to join a sysplex with a pre-V1R4 system

Migration scenario 3

A z/OS V1R7 system will partition itself out of a sysplex if it attempts to join a sysplex containing a pre-z/OS V1R7 system (without APAR OA06857).

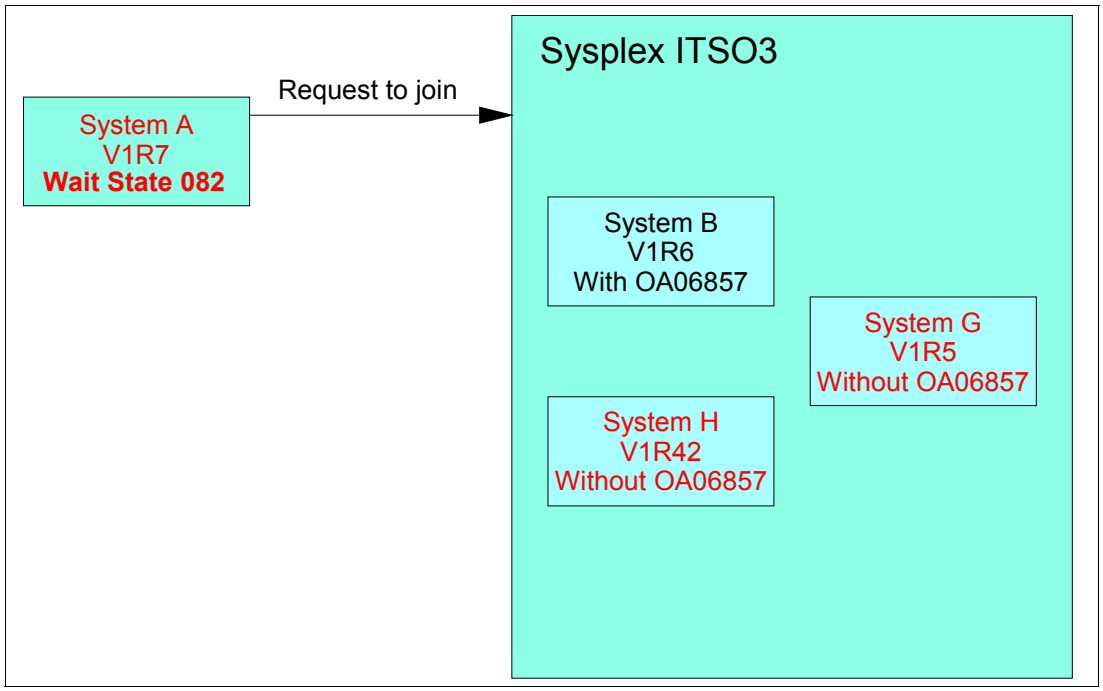


Figure 14-4 Attempts to join a sysplex with systems not having APAR OA06857

Migration scenario 4

A system will partition from a sysplex if:

- A pre-z/OS V1R4 system joins a sysplex containing pre-z/OS V1R7 systems (with APAR OA06857) after a console had been removed, and at one time, a z/OS V1R7 system existed.
- A pre-z/OS V1R7 system (without APAR OA06857) joins a sysplex containing pre-z/OS V1R7 systems (with APAR OA06857) after a console had been removed, and at one time, a z/OS V1R7 system existed.

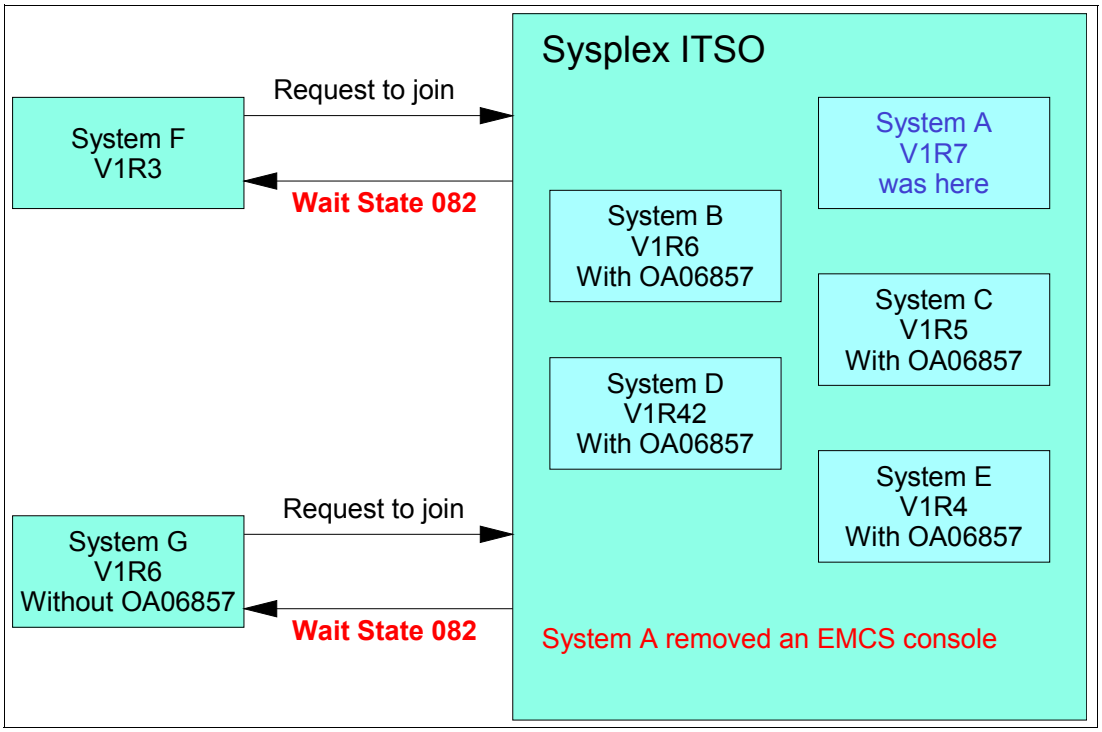


Figure 14-5 Two systems requesting to join where a previous V1R7 system has been removed

14.2.2 EMCS console removal implementation

A sample program IEARELEC in SYS1.SAMPLIB is provided to show how to use the service that deletes the EMCS consoles. It is similar to IEARELCN, which is used for MCS consoles.

The service is as follows:

- Supports the wild cards in console names
- Must be APF-authorized
- Can be modified to pass a CART and/or console name (to direct any system messages issued by the service)

To delete an EMCS console, you can do the following:

- Use the **`DISPLAY EMCS,ST=L`** command to obtain all the defined EMCS consoles in the sysplex.
- Determine which consoles can or should be removed.
- Use the IEARELEC sample program to remove the console definitions.

Defined EMCS consoles

The command and messages shown in Figure 14-6 display the defined EMCS consoles.

,

```
IEE129I 18.26.08 DISPLAY EMCS 866
DISPLAY EMCS,ST=L
NUMBER OF CONSOLES MATCHING CRITERIA: 97
SC63     *ROUTO02 SC64     *ROUTO03 *DICNS65 *DICNS70 BZZOSC63
SYSJ3RO1 BOZO     BZZOSC70 BART     HERING   HARJANS  KYNEF
*ROUTO06 KMT1     KMT3     PEGGYR   *ROUTO07 BWILSON  PAOLOR1
TSAUSER  AAUTO164 AUTSYS64 AIVO0164 *ROUTO10 *ROUTO13 AOPER164
AWRKO264 AWRKO764 AWRKO964 AWRKO864 AAUTO264 *SYSLG63 *OPLOGO1
*SYSLG64 *OPLOGO2 SC65     *ROUTE65 SC70     *ROUTE70 SYSJ3NO1
MORON    TROWELL  MERONI   ROGERS   RCONWAY  RC64     DONNAS
BOZOBOB  PAOLOR2  KMT6     KMT8     MBEAL    VBUDI    AUTGSS64
AUTMON64 AUTREC64 TIVOO2   HAIMO65  AOFAI64  AWRKO464 AWRKO364
AWRK1964 AWRK1364 AWRKO664 AOPER264 *DICNS63 HAIMO    *DICNS64
*SYSLG65 *OPLOGO3 *SYSLG70 *OPLOGO4 SYSJ3DO1 BZZOSC65 BZZOSC64
EMCSCON  WHITE    BIRD     KEYES    KMT2     DSNTWR   PAOLOR3
KMT7     SC64R64  AUTXCF64 AHWOO264 TWSRES3  CNMCRNM  CNMCR64
AWRK1464 AWRKO164 AWRK1O64 SC64T64  CBDHSSOO TWSRES1
```

Figure 14-6 Display of defined EMCS consoles

z/OS health checker for z/OS

There is a health checker that can detect when you have exceeded a certain threshold. The definitions are as follows:

```
CNZ_EMCS_Inactive_Consoles
Default threshold = 10000 consoles
Example of JCL to invoke IEARELEC
```

Sample JCL to invoke IEARELEC

All EMCS consoles that are inactive and are not IBM internal consoles are removed. Some JCL examples are shown in Figure 14-7.

```
//JOB1     JOB  ...
//AMR      EXEC PGM=IEARELEC,PARM='CONSNAME(consol01)'
 JOB1 will attempt to remove the console named ‘CONSOL01’.

//JOB2     JOB  ...
//AMR      EXEC PGM=IEARELEC,PARM='CONSNAME(consol*)'
 JOB2 will attempt to remove any console with a name that begins with ‘CONSOL’ (for
example, CONSOL01, CONSOL02, etc.)

//JOB3     JOB  ...
//AMR      EXEC PGM=IEARELEC,PARM='CONSNAME(sy?con*)'
JOB3 will attempt to remove any console whose name has as its first two characters,
‘SY’, and its fourth thru sixth characters, ‘CON’ (for example, SY1CON1, SY1CON2,
SY2CON1, SY2CON2, etc.)
```

Figure 14-7 JCL examples to remove consoles using IEARELEC

Sample output from invoking IEARELEC

The following JCL was used to remove consoles:

```
//JOB3     JOB  ...
//AMR      EXEC PGM=IEARELEC,PARM='CONSNAME(sy?con*)'
```

The JCL using the IEARELEC service generates the following hardcopy-only message:

```
CNZ4002I EMCS CONSOLE REMOVAL FOR WILDCARD PATTERN SY?CON*
         FOUND: 4 REMOVED: 4 NOT REMOVED: 0
         THE FOLLOWING EMCS CONSOLES WERE REMOVED:
         SY1CON1  SY1CON2  SY2CON1  SY2CON2
```

IEARELEC generates the following job log message:

```
MRC104I ALL EMCS CONSOLES MATCHING THE WILDCARD PATTERN OF SY?CON* HAVE BEEN REMOVED
```

14.3 Monitor message independence

This enhancement allows monitor messages to be produced without requiring that the messages be sent to a console. The messages that are produced have console routing attributes that are not used. Therefore, these messages are not sent to a console.

14.3.1 New MONITOR keyword for SETCON command

A new keyword, MONITOR, is used to enable or disable monitor message production. In addition, the keyword allows control of whether specific message types are logged. The **SETCON MONITOR** command has the following options:

```
JOBNAMES = ON | ON,LOG | ON,NOLOG | OFF
SESS = ON | ON,LOG | ON,NOLOG | OFF
STATUS = ON | ON,LOG | ON,NOLOG | OFF
T = ON | OFF
```

JOBNAMES

Message production is to be enabled for the display of the name of each job when the job starts and terminates, including unit record allocation when the step starts. If a job terminates abnormally, the job name will appear in a diagnostic message. These messages are also sent to the SYSLOG and OPERLOG. Following are examples of this option:

```
JOBNAMES=ON or JOBNAMES=(ON) or JOBNAMES=(ON,LOG) or JOBNAMES=OFF or JOBNAMES=(ON,NOLOG)
```

JOBNAMES=(ON,NOLOG) is exactly like JOBNAME=ON except these messages are not sent to the SYSLOG or OPERLOG.

Specifying JOBNAMES=OFF or JOBNAMES=(OFF) indicates that message production is to be disabled for the display of the name of each job when the job starts and terminates, including unit record allocation when the step starts.

Note: When a request to disable this message type is made, production of these messages is disabled only if there are no consoles in the sysplex currently receiving this message type.

SESS

Message production is to be enabled for the display of the user identifier for each TSO terminal when the session is initiated and when it is terminated. If the session terminates abnormally, the user identifier appears in the diagnostic message. These messages will also be sent to the SYSLOG or OPERLOG. Following are examples of this option:

```
SESS=ON or SESS=(ON) or SESS=(ON,LOG) or SESS=(ON,NOLOG)
```

SESS=(ON,NOLOG) is exactly like SESS=ON except these messages are not sent to the SYSLOG or OPERLOG.

SESS=OFF or SESS=(OFF) indicates that message production is to be disabled for the display of the user identifier for each TSO terminal when the session is initiated and when it is terminated.

Note: When a request to disable this message type is made, production of these messages is disabled only if there are no consoles in the sysplex currently receiving this message type.

STATUS

Message production is to be enabled for the display of data set names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG whenever they are freed. These messages are also be sent to the SYSLOG or OPERLOG. Following are examples of this option:

```
STATUS=ON or STATUS=(ON) or STATUS=(ON,LOG) or STATUS=(ON,NOLOG)
```

STATUS=(ON,NOLOG) is exactly like STATUS=ON except these messages are not sent to the SYSLOG or OPERLOG.

STATUS=OFF or STATUS=(OFF) indicates that message production is to be disabled for the display of data set names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG whenever they are freed.

Note: When a request to disable this message type is made, production of these messages is disabled only if there are no consoles in the sysplex currently receiving this message type.

T

The T option is for monitor messages that can optionally contain a timestamp, and if so, the timestamp is to be included in the message. Following are examples of this option:

```
T=ON or T=(ON) or T=OFF or T=(OFF)
```

For monitor messages that can optionally contain a timestamp, the timestamp is *not* to be included in the message.

14.3.2 Implementing monitor message independence

The new MONITOR keyword is added on the **SETCON** and **DISPLAY OPDATA** commands. For more information about these commands see *z/OS MVS System Commands*, SA22-7627.

On the **SETCON** command, use the MONITOR keyword as follows:

- Enable or disable monitor message production.
- Control whether specific message types are logged.

SETCON MONITOR command

The syntax for the command is as follows:

```
JOBNAMES= ON | (ON,LOG) | (ON,NOLOG) | OFF
SESS= ON | (ON,LOG) | (ON,NOLOG) | OFF
STATUS = ON | (ON,LOG) | (ON,NOLOG) | OFF
T= ON | OFF
```

DISPLAY OPDATA command

On the **DISPLAY OPDATA** command, the MONITOR keyword displays overall monitor message production status.

```
DISPLAY OPDATA,MONITOR [,FULL]
```

The system is to display the enablement status of the monitoring facility for all message types supported, including whether each of these message types are sent to the system SYSLOG or OPERLOG. The system also displays the number of consoles and, if applicable, TSO users that have requested to receive specific message types. An example follows:

```
DISPLAY OPDATA,MONITOR

CNZ1100I 10.03.57 MONITOR DISPLAY 849
SPACE=OFF  DSNAME=OFF  TIMESTAMP=OFF
MSGTYPE   SETCON MN  NUMBER OF RECEIVERS
JOBNAMES  ON,LOG          3 CONSOLES
SESS      ON,NOLOG        5 CONSOLES
STATUS    OFF             3 CONSOLES
```

The parameters have the following meanings:

SPACE=ON Message production is enabled for the display, in demount messages, of the available space on the direct access volume.

SPACE=OFF Message production is disabled for the display, in demount messages, of the available space on the direct access volume.

DSNAME=ON Message production is enabled for the display, in mount messages, of the name of the first non-temporary data set allocated on the volume to which the messages refer. No data set name appears in messages for data sets with a disposition of delete.

DSNAME=OFF Message production is disabled for the display, in mount messages, of the name of the first non-temporary data set allocated on the volume to which the messages refer. No data set name appears in messages for data sets with a disposition of delete.

TIMESTAMP=ON For monitor messages that can optionally contain a timestamp, the timestamp is included in the message.

TIMESTAMP=OFF For monitor messages that can optionally contain a timestamp, the timestamp is not included in the message.

An example of using the FULL keyword follows:

```
DISPLAY OPDATA,MONITOR,FULL

CNZ1101I 10.06.48 MONITOR DISPLAY 853
SPACE=OFF  DSNAME=OFF  TIMESTAMP=OFF
MSGTYPE   SETCON MN  RECEIVING CONSOLE NAMES
JOBNAMES  ON,LOG     MCSY13E0 EMCSY1   EMCSY2
SESS      ON,NOLOG   MCSY13E0 EMCSY4   EMCSY5   EMCSY3   EMCSY6
STATUS    OFF        MCSY13E0 EMCSY7   EMCSY8
```

Instead of displaying the number of consoles and TSO users that have requested to receive specific message types, the system lists the names of those consoles. If there is any TSO user information to display, an additional section listing their names is included.

14.4 Consoles query interface

An IBM-supported interface, called the consoles query interface, is provided to obtain “retained messages” data. This new interface is called CNZQUERY.

14.4.1 Implementation of CNZQUERY interface

The CNZQUERY interface has the following keywords:

```
WTOR = NO | YES
AMRF= NO | YES
ANSAREAALET=xansareaalet
```

WTOR=YES Return information about WTORs. A queue of OREs is returned.

AMRF=YES Return information about the AMRF. Three queues of WQEs are returned.

ANSAREAALET This is the ALET of the data space which is to contain the output information. The data space must be on the dispatchable unit access list or be a common area data space. It must include the address range x'1000' through x'7FFFEFFF' (that is, it is a 2G data space). It may contain the 0 and x'7FFFF000' pages.

Using the CNZQUERY interface

When you specify the interface options, this results in the following conditions:

- When WTOR=YES is specified, a serialized snapshot of the ORE control block chain is copied to the data space referenced by ANSAREAALET.
- When AMRF=YES is specified, a serialized snapshot of the AMRF WQE control block chains is copied to the data space referenced by ANSAREAALET.
- CNZQUERY additional output is mapped by the CNZMYQUA macro, as follows:
  - When WTOR=YES is specified, the address of the first ORE
  - When AMRF=YES is specified, the following addresses are obtained:
    - Address of the first immediate action WQE
    - Address of the first eventual action WQE
    - Address of the first critical eventual action WQE

14.5 1-byte Console ID elimination

The 1-byte console IDs are no longer supported and this removes the 99 MCS, SMCS, and subsystem consoles per sysplex constraint. This will then allow the 99 console constraint to be delivered in a future release.

Subsystem console IDs are now 4 bytes in length. The macros and commands that support 1-byte console IDs and migration IDs are removed.

Note: This is the last release to support 1-byte console IDs. For complete details on the 1-byte Console ID Tracking Facility, see APAR II13752.

The **DISPLAY OPDATA,TR** command is used to display recorded instances of 1-byte console ID usage and issues message CNZ1001I.

The **SET CNIDTR** command is used to activate an exclusion list that informs the facility of recorded instances that have already been reported and should no longer be tracked.

14.5.1 Migration and coexistence considerations

z/OS V1R7 has implemented changes to some macros and commands. The following macros are changed:

- The WTO and WTOR macros no longer support the MCSFLAG (REG0 or QREG0). A severity 12 MNOTE is issued if the field is used.
- The MCSOPER macro has the MIGID keyword removed. A severity 1 MNOTE indicates that the MIGID is unsupported.

Note: These changes will only affect programs being reassembled.

The commands changes are as follows:

- All commands supporting L=cc and L=name now only support L=name.
- The commands shown in Figure 14-8 used to support console ID or name, now only support console names.

Command	Response if Console ID specified
D C,CN=nn	IEE274I DISPLAY CONSOLE nn NOT VALID
D PFK,CN=nn	IEE274I DISPLAY CONSOLE nn NOT VALID
D R,CN=nn	IEE274I DISPLAY CONSOLE nn NOT VALID
RESET CN(nn)	IEE274I RESET CONSOLE nn NOT VALID
SWITCH CN=nn	IEE274I SWITCH CONSOLE nn NOT VALID
VARY CN(nn)	IEE274I VARY CN CONSOLE nn NOT VALID

Figure 14-8 Commands that now only support console names

14.5.2 Implementation

- MCSOPER
  - Add NAME keyword support on DEACTIVATE call
  - Mutually exclusive with CONSID keyword
- MCSOPMSG
  - Add NAME keyword support
  - Mutually exclusive with CONSID keyword

14.6 TRACK command elimination

The **TRACK** command is incompatible with the new console restructure infrastructure, so the **TRACK** and **STOPTR** commands are eliminated.

As an alternative, the JES2 **$TA** and **$VS** commands can be used to achieve the same function. **$VS** allows you to issue an MVS command and the **$TA** defines an automatic

command (starts every x seconds), so having a **$TA** issue a **$VS** which issues a **D A,L** command should do what you need. For example, to issue the **D A,L** command every 2 minutes, enter:

```
$TA,I=120,'$VS,''D A,L'''
```

14.6.1 Migration and coexistence considerations

When converting to z/OS V1R7, the following changes have been made during the console restructure:

- CONSOLxx parmlib member changes

  UTME(nnn) keyword is no longer supported. In the CONSOLxx parmlib member, an MCS or SMCS console would specify the UTME(nnn) keyword to set the interval in seconds for updating dynamic displays. It has been removed.

  If you specify this parameter, the following message is issued:

  ```
  IEA196I CONSOLJG 03E0: UNRECOGNIZED KEYWORD UTME(30) IGNORED.
  ```

- Command changes

  The commands shown in Figure 14-9 are no longer supported.

Command	Response if command issued
TRACK	IEE305I TRACK COMMAND INVALID
STOPTR	IEE305I STOPTR COMMAND INVALID
CONTROL T	IEE156I CONTROL INVALID OPERAND -T
CONTROL D,H	IEE156I CONTROL INVALID OPERAND -D
CONTROL D,U	IEE156I CONTROL INVALID OPERAND -D
MSGRT TR=A	IEE156I MSGRT INVALID OPERAND -TR=A

Figure 14-9 Commands that are no longer supported

The commands shown in Figure 14-9 provided the following functions:

TRACK Request a dynamic display of job information for a particular console.

STOPTR Stop a dynamic display of job information for a particular console.

CONTROL T Change the time interval for updating a dynamic display for a particular console.

CONTROL D,H Prevent the dynamic display from being updated.

CONTROL D,U Resume updating the dynamic display.

MSGRT TR=A The system is to route the TRACK A command display and the action of the STOPTR command to the specified console.

15

zFS enhancements

zFS is the z/OS Distributed File Service zSeries File System. It is a z/OS UNIX System Services file system that can be used in addition to the hierarchical file system (HFS). zFS file systems contain files and directories that can be accessed with z/OS UNIX application programming interfaces (APIs). zFS file systems can be mounted into the z/OS UNIX hierarchy along with other local (or remote) file system types (for example HFS, TFS, AUTOMNT, and NFS).

This chapter describes zFS enhancements introduced in z/OS V1R7, specifically the following:

- Common forwarding support
- Addition of valid aggregate name characters
- Unquiesce modify command
- Performance monitoring APIs
- zFS end of memory support
- RMF Monitor III support for zFS

15.1 zFS overview

zFS can be used for all levels of the z/OS UNIX System Services hierarchy (including the root file system) when all members are at the z/OS V1R7 level. Because zFS has higher performance characteristics than HFS, zFS is the strategic file system; HFS may no longer be supported in a future release and you will have to begin to migrate the remaining HFS file systems to zFS.

The data set that contains zFS file systems is called a *zFS aggregate*. A zFS aggregate can contain one or more zFS file systems. A zFS aggregate is a Virtual Storage Access Method Linear Data Set (VSAM LDS). Once the zFS aggregate is defined and formatted, one or more zFS file systems can be created in the aggregate.

A zFS aggregate that contains only a single zFS file system is called a *compatibility mode aggregate*. Compatibility mode aggregates are more like HFS. It is recommended that as you begin to use zFS, you use compatibility mode aggregates.

A zFS aggregate that contains multiple file systems is called a *multi-file system aggregate*. This kind of aggregate is not supported in a sysplex shared file system environment.

For full sysplex support, zFS must be running on all systems in the sysplex and all zFS file systems must be compatibility mode file systems, they cannot be file systems in multi-file system aggregates.

Note: Multi-file system aggregate support is not planned to be enhanced and might be removed sometime in the future. Therefore, you should only use compatibility mode aggregates.

15.2 zFS configuration options for sysplex

When the IOEFSPRM is specified in the IOEZPRM DD statement of the ZFS PROC, there can only be one IOEFSPRM file for each member of a sysplex. Otherwise, using PARMLIB, zFS configuration options can be specified in a list of configuration parm files. This allows an installation to specify the following:

- Configuration options that should be common among all members of the sysplex (for example, adm_threads).
- In a shared IOEPRMxx member, the configuration options that should be system-specific (for example, define_aggr) in a separate, system-specific IOEPRMxx member.

If a configuration option is specified more than once, the first one found is taken.

15.2.1 zfsadm configquery command

This command queries the current value of zFS configuration options that you specify either in the zFS procedure or in parmlib. The **zfsadm configquery** command displays the current value of zFS configuration options. The value is retrieved from ZFS address space memory rather than from the IOEFSPRM file. You can specify that the configuration option query request should be sent to another system by using the **-system** option. The configuration options as shown via the **zfsadm configquery** command are the following:

```
zfsadm configquery [-system system name] [-adm_threads] [-aggrfull] [-aggrgrow] [-all]
[-allow_dup_fs] [-auto_attach] [-cmd_trace] [-debug_dsn] [-fsfull] [-fsgrow] [-group]
[-log_cache_size] [-meta_cache_size] [-metaback_cache_size] [-msg_input_dsn]
[-msg_output_dsn] [-nbs] [-sync_interval] [-sysplex_state] [-trace_dsn]
```

```
[-trace_table_size] [-tran_cache_size] [-user_cache_readahead] [-user_cache_size]
[-vnode_cache_limit] [-vnode_cache_size] [-level] [-help]
```

New IOEFSPRM options

IOEFSPRM is the zFS sample parameter data set that you define in a DDNAME=IOEZPRM statement in the proclib JCL for ZFS started task. You can use IOEPRMxx to customize in your installation. The new options added are:

dir_cache_size=2M Dir_cache_size specifies the size of the directory buffer cache.

group=IOEZFS Specifies the XCF group name used by zFS.

xcf_trace_table_size=4M Specifies the size of the XCF trace table.

New configuration options with z/OS V1R7

The following options are new with z/OS V1R7 with the **zfsadm configquery** command:

-system *system name* Specifies the name of the system the report request will be sent to, to retrieve the data requested.

-group Displays the XCF group used by ZFS for communication between sysplex members.

Default Value: IOEZFS

Expected Value: 1 to 8 characters

Example group: IOEZFS1

-sysplex_state Displays the sysplex state of ZFS. Zero (0) indicates that ZFS is not in a shared file system environment. One (1) indicates that ZFS is in a shared file system environment.

Note: dir_cache_size and xcf_trace_table_size are not currently set via the **zfsadm config** command.

zfsadm config command

The **zfsadm config** command changes the value of zFS configuration (IOEFSPRM) options in memory. Following are the options with z/OS V1R7:

```
zfsadm config [-admin_threads number] [-user_cache_size number[,fixed]]
[-meta_cache_size number[,fixed]] [-log_cache_size number[,fixed]] [-sync_interval
number] [-vnode_cache_size number] [-nbs {on|off}] [-fsfull threshold,increment]
[-aggrfull threshold,increment] [-trace_dsn PDSE_dataset_name] [-tran_cache_size number]
[-msg_output_dsn Seq_dataset_name] [-user_cache_readahead {on|off}]
[-metaback_cache_size number[,fixed]] [-fsgrow increment,times] [-aggrgrow {on|off}]
[-allow_dup_fs {on|off}] [-vnode_cache_limit number] [-system system name] [-level]
[-help]
```

15.3 Common forwarding support

Before this release, zfsadm commands must be issued from the owning system. Now, all zfsadm commands that apply to zFS aggregates or file systems work against all aggregates and file systems across the sysplex.

Previously, zfsadm commands (and the pfsctl APIs) could only display or change information that is located or owned on the current member of the sysplex. A particular problem is quiescing of a zFS aggregate.

15.3.1 DFSMS backup (ADRDSSU)

When DFSMS ADRDSSU is used to back up a zFS aggregate, the aggregate is quiesced by DFSMS before the backup. Currently, in order for this quiesce to be successful, it must be issued on the system that owns the aggregate. If it is issued on any other system in the sysplex, the quiesce fails (and the backup fails). It is a problem to issue the backup on the owning system because ownership can change at any time as a result of operator command or system failure. The z/OS V1R7 **zfsadm** command forwarding allows the quiesce (or any other **zfsadm** command) to be issued from any member of the sysplex.

15.3.2 zFS zfsadm command forwarding

From any member of the sysplex, you can request display or change against zFS to another system if both systems are in z/OS V1R7.

The following **zfsadm** commands are now global in a sysplex:

aggrinfo, **clonesys**, **lsaggr**, **lsfs**

In *z/OS Distributed File Service zSeries File System Administration*, SC24-5989 there is a complete description of all **zfsadm** commands.

Now, **zfsadm** commands go to the correct system when you issue the next commands by any system in the sysplex:

aggrinfo, **clone**, **create**, **delete**, **grow**, **lsfs**, **lsquota**, **quiesce**, **rename**, **setquota**, **unquiesce**

The following **zfsadm** commands can optionally direct their operation to a particular member of the sysplex:

aggrinfo, **attach**, **clonesys**, **config**, **configquery**, **define**, **detach**, **format**, **lsaggr**, **lsfs**, **query**

zfsadm command

ZFSADM is a command suite with which you issue commands against zFS file systems. A new option for **ZFSADM** commands enables you to specify the name of the system that the request will be sent to, as follows:

-system *system name* specifies the name of the system that the request will be sent to.

The **ZFSADM** commands have the same general structure:

```
command {-option1 argument...|-option2 {argument1 | argument2}...}
        [-optional_information]
```

The following example illustrates the elements of a **ZFSADM** command:

```
zfsadm aggrinfo [-aggregate name | -system system name] [-fast | -long] [-level]
                    [-help]
```

Figure 15-1 is an example of use of the system option in **ZFSADM** commands.

```
USER1 @ SC70:/u/user1> zfsadm aggrinfo -system SC70
IOEZ00368I A total of 8 aggregates are attached to system SC70.
OMVS.TEST.MULTIFS.ZFS (R/O MULT): 20597 K free out of total 20880
ZFSFR.ZFSA.ZFS (R/O COMP): 6374 K free out of total 7200
ZFSFR.ZFSB.ZFS (R/O COMP): 57760 K free out of total 576000
OMVS.TEST.MIGR.HFS (R/W COMP): 380 K free out of total 29520
OMVS.USER2.TEST.ZFS (R/W COMP): 454 K free out of total 11520
ZFSFR.ZFSH.ZFS (R/W COMP): 292908 K free out of total 1440000
OMVS.USER2.ZFS (R/W COMP): 565 K free out of total 720
OMVS.USER2.TEST.DIRCACHE (R/W COMP): 3438 K free out of total 3600
```

Figure 15-1 Use of system option in zfsadm command

15.3.3 Sysplex considerations

Whether **ZFSADM** commands act globally across the sysplex depends on the BPXPRMxx parmlib member with the SYSPLEX option. If SYSPLEX(YES) is specified, then you are in a shared file system environment and **ZFSADM** commands will act globally. If IOEFSPRM SYSPLEX(OFF) is specified, or the BPXPRMxx SYSPLEX option specifies SYSPLEX(NO), then **ZFSADM** commands will not act globally. The parmlib members options are fully explained in the z/OS V1R7 initialization and tuning reference manual.

When all members in the sysplex are at z/OS V1R7, backups of the zFS file system should work from any member, regardless of which member of the sysplex owns the file system.

15.4 Addition of valid aggregate name characters

Previously, you could not create and format a zFS aggregate with special characters in the aggregate name. If any of these characters was in the name of an HFS file system that you wanted to move during a conversion to zFS, you had to use a different name for zFS.

In this release, any name you can use for an HFS file system can be used for a zFS aggregate. zFS supports the following additional characters in zFS file system names and zFS aggregate names:

- @ (at sign)
- # (number sign)
- $ (dollar)

Examples:

```
# zfsadm define -aggr PLEX.JMS.AGGR#06.LDS0006 -volumes CFC000 -cyl 10
IOEZ00248E VSAM linear data set PLEX.JMS.AGGR#06.LDS0006 successfully created.
# zfsadm format -aggr PLEX.JMS.AGGR#06.LDS0006 -compat
IOEZ00077I HFS-compatibility aggregate PLEX.JMS.AGGR#06.LDS0006 has been
                    successfully created
```

15.4.1 Migration consideration

It is highly recommended that APAR OA08611 be installed on previous releases before IPLing the first z/OS V1R7 system. This will allow zFS aggregates with special characters in the name to be processed successfully on previous releases.

15.5 Unquiesce modify command

Before this release, if a zFS aggregate was quiesced and the job failed to unquiesce it, you would need to use the OMVS **ZFSADM UNQUIESCE** command; you could not unquiesce from the operator console.

In z/OS V1R7, the operator can using the **F ZFS,UNQUIESCE** command from the console, as shown in the following example:

```
F ZFS,UNQUIESCE,PLEX.JMS.AGGR#01.LDS0001
```

This command must be issued from the owning system. It does not forward a request to other members of the sysplex.

15.6 Performance monitoring APIs

Previously, certain zFS performance counters could only be retrieved via operator console command. The data could not easily be retrieved by an application, and only an API is provided to obtain statistics on zFS activity.

In z/OS V1R7 a new programming interface is provided in pfsctl (BPX1PCT) to provide statistics that were previously available only from the **modify** command. RMF Monitor II will use the zFS monitoring APIs to provide performance information about the zFS environment. You can use this information to tune the zFS environment by monitoring things such as cache sizes, I/O balancing, and the sizes of zFS aggregates. This can help simplify zFS performance management.

You can retrieve performance counter information by:

- Using the pfsctl API (BPX1PCT) in an application. See the *z/OS Distributed File Service zSeries File System Administration*, SC24-5989 manual for details.
- **ZFSADM QUERY** command to display or reset performance counters.

The second set of performance monitoring pfsctl APIs are added in z/OS V1R7; the first set was added in release z/OS V1R6. There are also corresponding **ZFSADM QUERY** commands.

The format of the **ZFSADM QUERY** command as shown in Figure 15-2 on page 359 has been updated in z/OS V1R7.

```
zfsadm query [-locking]
          [-reset]
          [-storage]
          [-usercache]
          [-trancache] R7
          [-iocounts]
          [-iobyaggregate]
          [-iobydasd]
          [-knpfs] R7
          [-metacache] R7
          [-dircache] R7
          [-vnodecache] R7
          [-logcache] R7
          [-system system_name] R7
          [-level]
          [-help]
```

Figure 15-2 zfsadm query command parameters

The following pfsctl application programming interface command calls have been added to retrieve performance counters and other information from the zFS physical file system:

- Statistics Directory Cache
- Statistics Kernel
- Statistics Log Cache
- Statistics Metadata Cache
- Statistics Transaction Cache
- Statistics Vnode Cache

See informations about pfsctl APIs and **ZFSADM QUERY** commands in *z/OS Distributed File Service zSeries File System Administration*, SC24-5989 manual.

15.7 zFS end of memory support

Previously, an end of memory condition while executing in zFS could cause zFS to go down and all zFS file systems to be unmounted.

In z/OS V1R7, a new zFS end of memory (EOM) support is invoked when a calling application goes to end of memory while executing in zFS.

End of memory conditions are recovered by zFS, so it does not go down, and no zFS file systems are unmounted, so they do not need to be remounted.

15.8 zFS command and options enhancements

z/OS V1R7 provides new informational commands related to sysplex support.

zfsadm lssys This command shows the names of the members in a sysplex, as shown in Figure 15-3 on page 360.

```
USER1 @ SC70:/u/user1>zfsadm lssys
IOEZ00361I A total of 2 systems are in the XCF group for zFS
SC70
SC65
```

Figure 15-3 zfsadm lsys option

New informational zfsadm options

-system Specifies the name of the system that the request will be sent to. This option is described in 15.3, “Common forwarding support” on page 355.

-long This is a new option for the **AGGFRINFO** and **LSFS** subcommands. It causes the output of the command to be extended to display the following additional information about space usage in an aggregate: number of log file blocks, number of blocks in the file system table, number of blocks in the bitmap, and numbers of free blocks and free fragments, as shown in Figure 15-4. The -fast option is the default.

```
USER1 @ SC70:/u/user1>zfsadm aggrinfo -aggregate ZFSFR.ZFSH.ZFS -long
ZFSFR.ZFSH.ZFS (R/W COMP): 292908 K free out of total 1440000
         34238 free 8k blocks;      19004 free 1K fragments
         14400 K log file;             56 K filesystem table
           208 K bitmap file
```

Figure 15-4 Use of the **configquery** *command*

-group Displays the XCF group used by zFS for communication between sysplex members.

It displays the XCF group used by zFS for communication between sysplex members, as shown in Figure 15-5.

```
USER1 @ SC70:/u/user1>zfsadm configquery -group
IOEZ00317I The value for configuration option -group is IOEZFS.
```

Figure 15-5 Display XCF group for zFS

-sysplex_state This option for the **configquery** command displays the sysplex state of zFS, as shown in Figure 15-6. Zero (0) indicates that zFS is not in a shared file system environment. One (1) indicates that zFS is in a shared file system environment.

```
USER1 @ SC70:/u/user1>zfsadm configquery -sysplex_state
IOEZ00317I The value for configuration option -sysplex_state is 1.
```

Figure 15-6 Display sysplex state of zFS

-dircache This option for the **ZFSADM QUERY** command specifies that the directory cache report should be displayed.

```
USER1 @ SC70:/>zfsadm query -dircache
                 Directory Backing Caching Statistics

Buffers    (K bytes)  Requests     Hits    Ratio   Discards
---------- --------- ---------- ---------- ------ ----------
       256      2048          0          0   0.0%          0
```

Figure 15-7 Directory cache report

-knpfs This option specifies that the kernel counters report should be displayed, and is shown in Figure 15-8.

```
USER1 @ SC70:/>zfsadm query -knpfs
          zFS Kernel PFS Calls
          --------------------

Operation               Count          Avg Time
---------           ----------        ----------
zfs_opens                   0             0.000
zfs_removes                 0             0.000
zfs_reads                   0             0.000
zfs_writes                  0             0.000
zfs_ioctls                  0             0.000
zfs_getattrs               90             0.010
...
zfs_pfsctls            1813828             1.205
zfs_statfss                 74             0.017
zfs_mounts                   7        227424.030
zfs_unmounts                 0             0.000
zfs_vinacts                  0             0.000
------------       ----------        ----------
*TOTALS*               1814002             2.082
```

Figure 15-8 Display using the -knpfs option

-logcache Specifies that the log cache counters report should be displayed, as shown in Figure 15-9.

```
USER1 @ SC70:/>zfsadm query -logcache
                       Log File Caching Statistics

Buffers    (K bytes)  Requests     Hits    Ratio   Written
---------- --------- ---------- ---------- ------ ----------
     24584    196672          8          0   0.0%          0

New buffer: log full waits          0  NBS IO waits          0
```

Figure 15-9 Display log cache counters

-metacache Specifies that the metadata cache counters report should be displayed, as shown in Figure 15-10 on page 362.

```
USER1 @ SC70:/>zfsadm query -metacache
                       Metadata Caching Statistics

Buffers    (K bytes)  Requests     Hits    Ratio   Updates
---------- --------- ---------- ---------- ------ ----------
     12288     98304       6427       3358  52.2%          0

                    Metadata Backing Caching Statistics

Buffers    (K bytes)  Requests     Hits    Ratio   Discards
---------- --------- ---------- ---------- ------ ----------
         0         0          0          0   0.0%          0
```

Figure 15-10 Display metadata cache counters

-trancache Specifies that the transaction cache counters report should be displayed, as shown in Figure 15-11.

```
USER1 @ SC70:/>zfsadm query -trancache
                      Transaction Cache Statistics
                      ----------------------------

Trans started:        0 Lookups on tran:        0 EC Merges:        0
Allocated Trans:     2000 (Act=        0, Pend=       0,
                           Comp=       0, Free=    2000)
```

Figure 15-11 Display transaction cache counters

-vnodecache Specifies that the vnode cache counters report should be displayed.

```
USER1 @ SC70:/>zfsadm query -vnodecache
                               zFS Vnode Op Counts

Vnode Op                       Count   Vnode Op                       Count
-----------------------  ----------   -----------------------  ----------
efs_hold                           0   efs_readdir                        0
efs_rele                           0   efs_create                         0
efs_inactive                       0   efs_remove                         0
efsvn_getattr                     97   efs_rename                         0
...                                                                       0
efs_getanode                       0   efs_vmblkinfo                      0
efs_readdir_raw                    0                                      0
Total zFS Vnode Ops               97                                      0
                                                                          0
                          zFS Vnode Cache Statistics                      0
                                                                          0
 Vnodes      Requests     Hits     Ratio  Allocates   Deletes             0
 ---------- ---------- ---------- ------- ---------- ----------          0
     65536         10          3  29.999%          0          0          0
                                                                          0
zFS Vnode structure size: 184 bytes
zFS extended vnodes: 65536, extension size 692 bytes (minimum)
Held zFS vnodes:          7 (high          7)
Open zFS vnodes:              0 (high          0) Reusable: 65529
```

Figure 15-12 Display vnode cache counters

15.9 RMF support for zFS

When looking at zFS performance, you have to consider the zFS components that are involved in I/O processing to or from a zFS file system. The performance of zFS can be influenced by controlling the size of the caches used to hold file system and log data.

There are new zFS summary and activity reports which provide data on:

- zFS response times and wait times
- zFS cache activity
- zFS activity and capacity by aggregate
- zFS activity and capacity by filesystem

This data helps to control the zFS environment according to:

- Cache sizes
- I/O balancing
- Capacity control for zFS aggregates

The zFS data reported by RMF Monitor III can be categorized into:

- zFS performance data
- zFS capacity data

15.9.1 zFS cache monitoring

The following caches can be monitored by RMF Monitor III:

User file cache This cache is used for all user files and performs I/O for all user files greater than 7 KB. It is allocated in data spaces.

zFS has a structure for each file currently cached, each cached file is broken into 64K, and each segment is broken into 4K pages.

Vnode Cache Vnode is a virtual inode, an object in a file system that represents a file. The vnode cache resides in the zFS primary address space.

Metadata cache User files that are smaller than 7 KB have the I/O from this cache. The Metadata cache resides in the zFS primary address space. For performance reasons the allocated storage can be fixed.

Log file cache This cache is used to write file record transactions that describe changes to the file system. The log file cache is allocated in a data space.

Transaction cache Data structures representing transactions that change metadata. The transaction cache is stored in the zFS primary address space. zFS dynamically increases the cache based on the number of concurrent pending transactions.

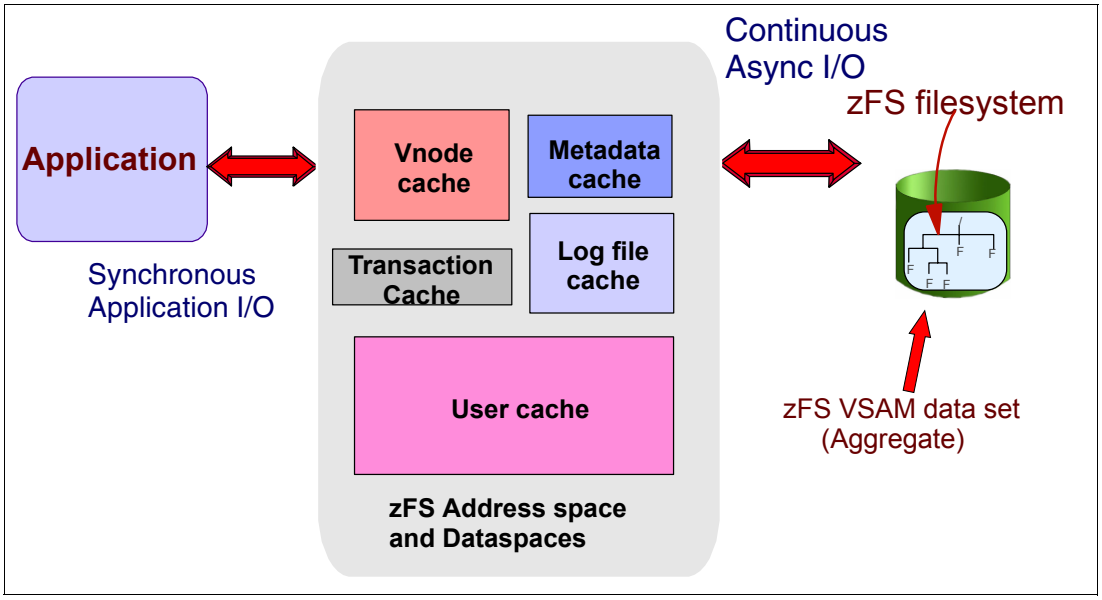


Figure 15-13 zFS address space and data spaces overview

Invoke zFS reports

The zFS reports are single system reports. They can be invoked from the RMF Overview Report Selection Menu, shown in Figure 15-14, or by using the following commands:

ZFSSUM or **ZFSS**: zFS summary report

ZFSACT or **ZFSA**: zFS activity report

Gathering of zFS activity data

Gathering of zFS activity data is controlled by a new Monitor III gatherer option:

NOZFS | **ZFS** (default: ZFS)

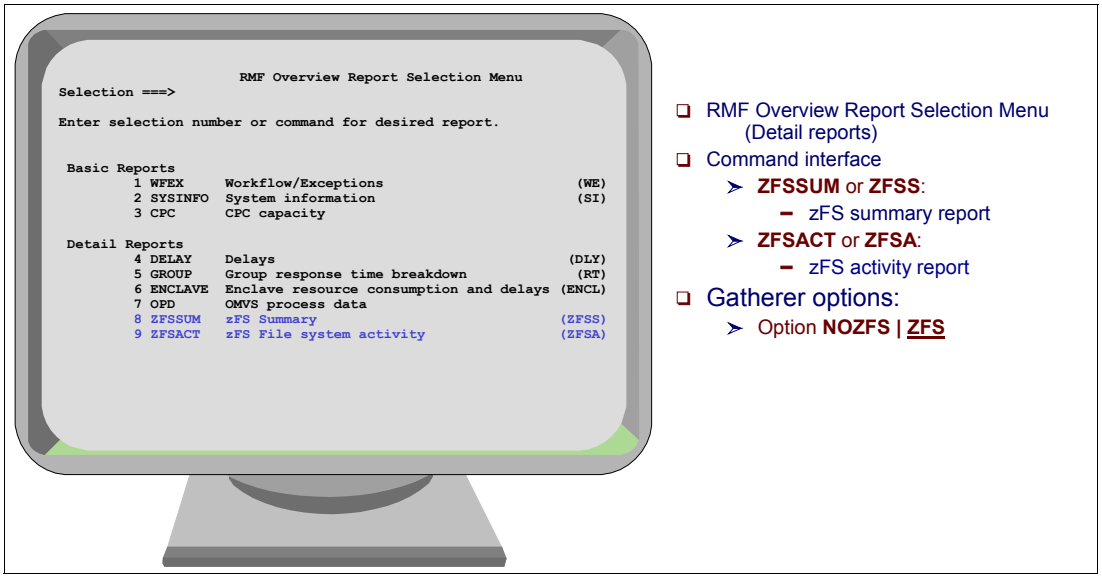


Figure 15-14 RMF Overview Report Selection Menu

15.10 zFS summary report

The zFS Summary Report in Figure 15-15 on page 366 shows the zFS activity and capacity data for the following three areas:

1. Summary for overall zFS response time as request response time and wait times.
2. Cache activity for the following four cache types:
   - User cache
   - Vnode cache
   - Metadata cache
   - Transaction cache
3. Aggregate activity and capacity data. The aggregate name is a cursor-sensitive field that shows the file systems statistics for the selected aggregate in the zFS activity report.

15.10.1 Report field descriptions

The important areas of the Response Summary Report are:

- In the response time section, the I/O, lock, and sleep wait percentages are workload dependent. A possible reason for high values is if the caches are too small. Small log files (small aggregates) that are heavily updated may result in I/Os, as follows:

  Total Average time in milliseconds to complete a zFS request.

  I/O wait% Percentage of time a zFS request has to wait for an I/O completion.

  Lock wait% Percentage of time a zFS request has to wait for locks.

  Sleep wait% Percentage of time a zFS request has to wait for events to occur.

- The Cache Activity section is as follows:

  User cache rate Number of requests per second made to the user file cache.

  User cache hit% Percentage of requests to the user file cache that completed without accessing the DASD.

  User cache read% Percentage of read requests to the user file cache, based on the sum of read and write requests.

  User cache dly% Percentage of requests to the user file cache that was delayed.

  Vnode rate Number of requests per second made to the vnode cache.

  Vnode hit% Percentage of requests to the vnode cache that completed without accessing the DASD.

  Metadata rate Number of requests per second made to the metadata cache.

  Metadata hit% Percentage of requests to the metadata cache that completed without accessing the DASD.

  Trx rate Number of transactions per second that started in the transaction cache.

- The Aggregate Name section, as follows:

  Aggregate name Name of the zFS aggregate. The name of the aggregate is the VSAM Linear data set name.

  Size Size of the aggregate.

  Use% Percentage of space used in the aggregate.

Mode Aggregate mode:

R/O CP R/W CP (CP: compatibility mode = aggregate contains one file system).

R/O MS R/W MS (MS: multi-file mode = aggregate can contain multiple file systems).

FS Number of file systems in the aggregate (one for CP mode aggregates).

Read (B/sec) Read data transfer rate (in bytes per second) for the aggregate.

Write (B/sec) Write data transfer rate (in bytes per second) for the aggregate.

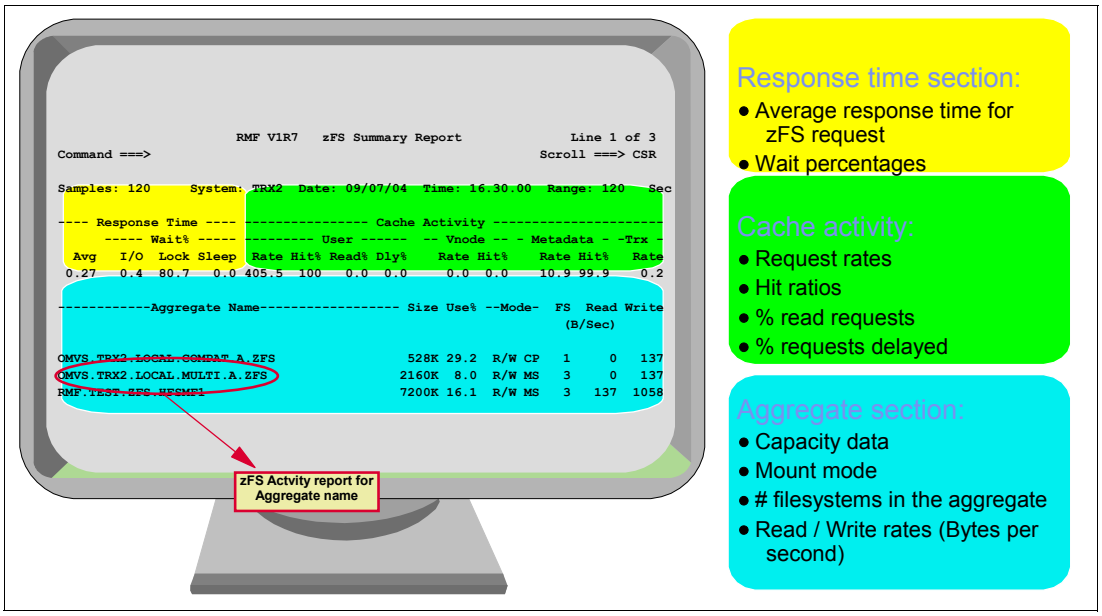


Figure 15-15 zFS Summary Report

15.11 Detail reports

From the zFS summary report, you can get detail reports by placing the cursor on the cursor-sensitive report fields.

15.11.1 I/O details report

Place the cursor on the response time section (on the I/O value). This will show the I/O details panel as in Figure 15-16 on page 367.

The report displays a breakdown of I/O requests for the following I/O request types:

- I/O for file system metadata
- I/O for log data
- I/O for user data

Field descriptions

Count Total number of requests.

Waits Number of requests waiting on an I/O completion of this I/O type.

Cancels Number of requests canceled (for example, a user deletes a file with pending I/O).

Merges Number of times two I/O requests are merged to one because of better performance.

Type Type of I/O request.

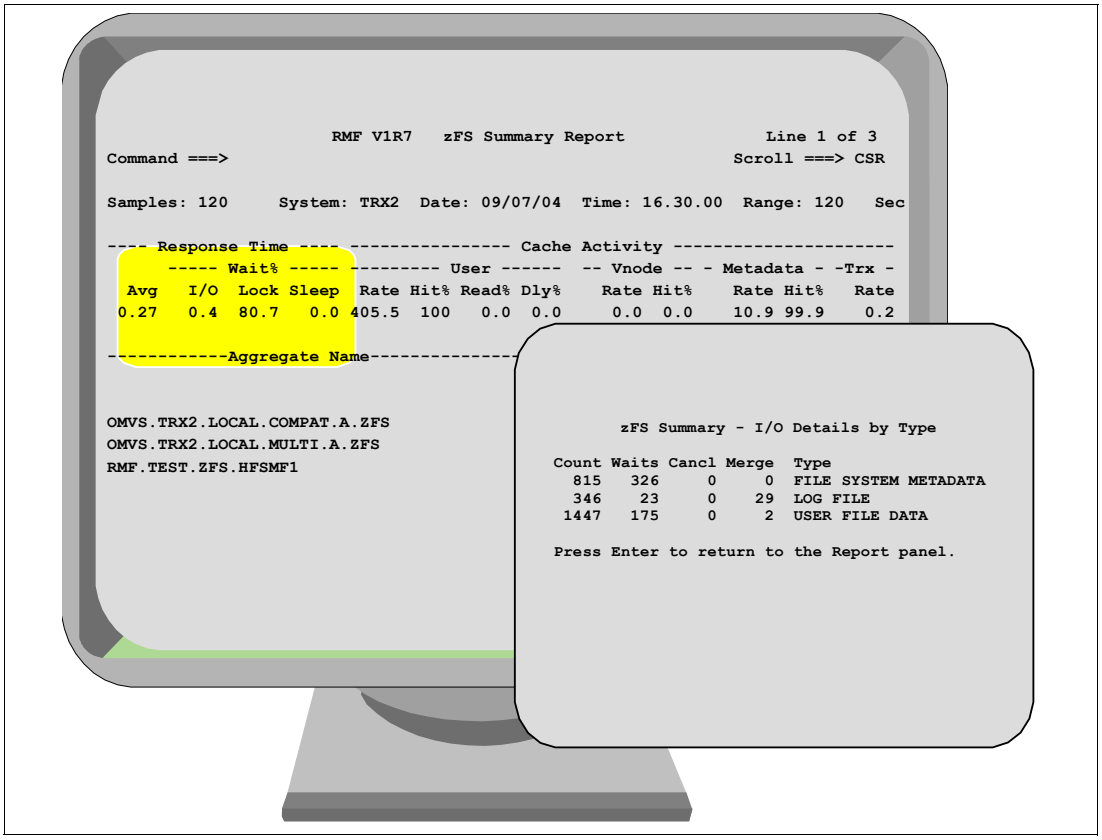


Figure 15-16 zFS summary: I/O details

15.11.2 User and vnode cache detail reports

Figure 15-17 on page 368 shows the user and vnode cache details. Place the cursor on the User section under Cache Activity to show the user cache details panel.

Field descriptions for user cache

The fields in the User Cache Details report have the following meanings:

Read Delay% Percentage of read requests to the user file cache that was delayed.

The following reason is counted as a read request delay:

Read wait: Read has to wait for pending I/O. (For example, a read of a file found the file data pending read because of asynchronous read-ahead from DASD to the user file cache.)

Write Delay% Percentage of write requests to the user file cache that was delayed.

The following reasons are counted as write request delays:

Write wait: Write has to wait because of pending I/O.

Write faulted: Write to a file needs to perform a read from DASD. If a write only updates a portion of a page of a file and that page is not in the user file cache, then the page needs to be read from DASD before the new data is written to the user file cache.

Async read rate Total number of read-aheads per second. Each read-ahead action for a file reads in one segment (up to 64K) from DASD to user file cache.

Scheduled write rate Total number of scheduled writes per second. Each scheduled write action for a file writes one segment (up to 64K) from user file cache to DASD.

Page reclaim writes Total number of page reclaim writes performed in the range time. A page reclaim write action writes one segment of a file from user file cache to DASD. Page reclaim writes are performed to reclaim space in the user file cache.

Fsyncs Shows how often applications requested that zFS sync a file's data to disk.

Vnode cache details

Cursor sensitivity in the Vnode section of the cache activity data displays the Vnode Cache Details panel.

The open count is important because an open file requires a USS and zFS vnode. If there were more open files than the zFS vnode cache size, zFS is forced to allocate more vnodes to meet the requirements of the system.

zFS will never have more than vnode_cache_size extended vnodes, but will have more than vnode_cache_size vnodes if it is forced to allocate above the vnode_cache_size due to USS requirements.

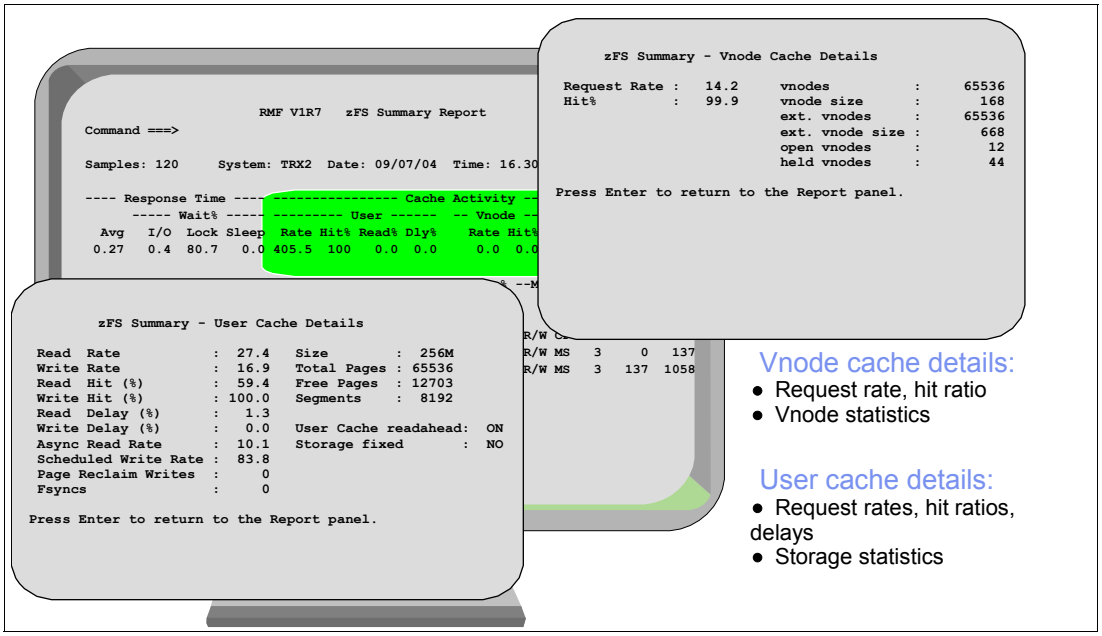


Figure 15-17 User and Vnode cache detail

15.11.3 Metadata and transaction cache detail reports

Figure 15-18 on page 369 shows the Metadata and Transaction cache details.

Metadata cache details

Cursor sensitivity in the Metadata section of the cache activity data displays the Metadata Cache Details panel.

The Metadata backing cache is:

- Optional
- Can be used as an extension to the Metadata cache
- Resides in a data space

Transaction cache details

Cursor sensitivity in the Trx section of the cache activity data displays the Transaction Cache Details pop-up panel. In this display, EC merge rate is the number of transaction class merges per second.

zFS decides when a transaction is related to or dependent on another transaction. When this determination is made the transactions are grouped into an *equivalence class*. Any transactions in the same equivalence class are committed together or backed out together in the event of a system crash. By using equivalence classes, threads running transactions simply run in parallel without added serialization between the two (other than locks if they hit common structures) and simply add their associated transactions to the same class. This increases throughput. The merge of equivalence classes occurs when two transactions that need to be made equivalent are both already in equivalence classes. In this case both classes are merged.

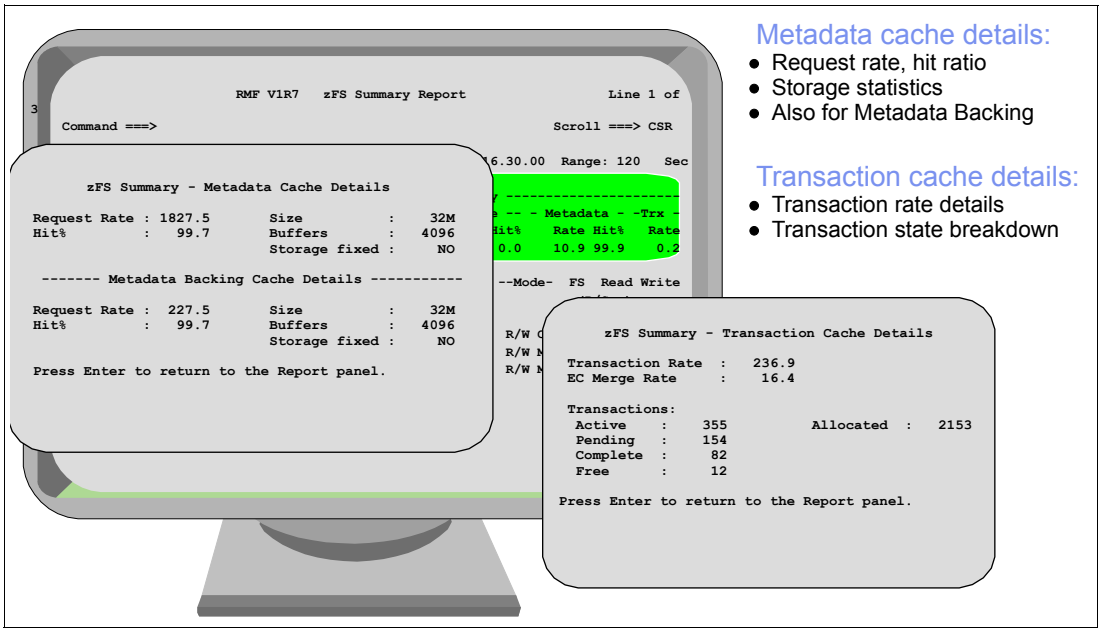


Figure 15-18 Metadata and Transaction Cache details

15.12 The zFS activity report

The zFS activity report on Figure 15-19 on page 370 shows information about zFS file systems. When invoked from the RMF Overview Report selection menu or by the **ZFSACT/ZFSA** command, the file systems for all attached aggregates are reported. If the report

is invoked by cursor-sensitive control from the zFS summary report, the file systems for one aggregate name are reported.

15.12.1 Field descriptions

Aggregate name	Set by cursor-sensitive control on an aggregate name in the zFS summary report. `ALL` indicates that file systems for all attached aggregates are reported.
File System Name	Name of the file system or USS file system name.
Mount Point	Mount point of the file system.
Mode	File system mount mode: R/W: read-write R/O: read-only N/M: not mounted QSC: file system not available because the aggregate is quiesced
Quota Limit	Maximum size of the file system (known as the *quota*). The quota is a logical number. When the quota is reached, the file system indicates that it is full.
Quota Usg%	Percentage of the quota currently used by the file system.
Operation Rate	Total number of vnode operations per second for this file system.

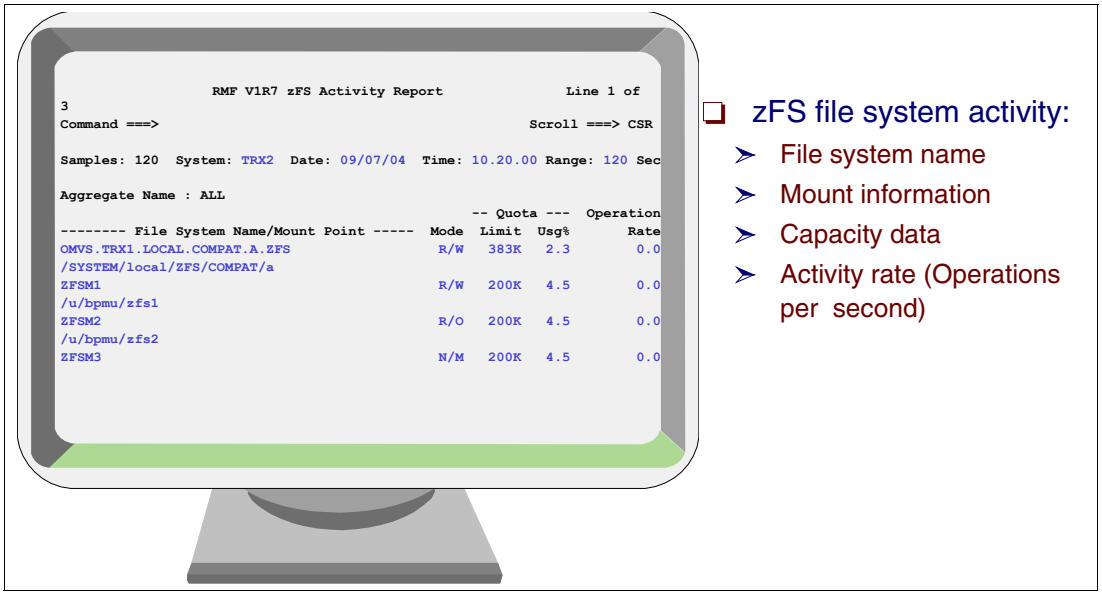


Figure 15-19 zFS Activity Report

15.13 New messages

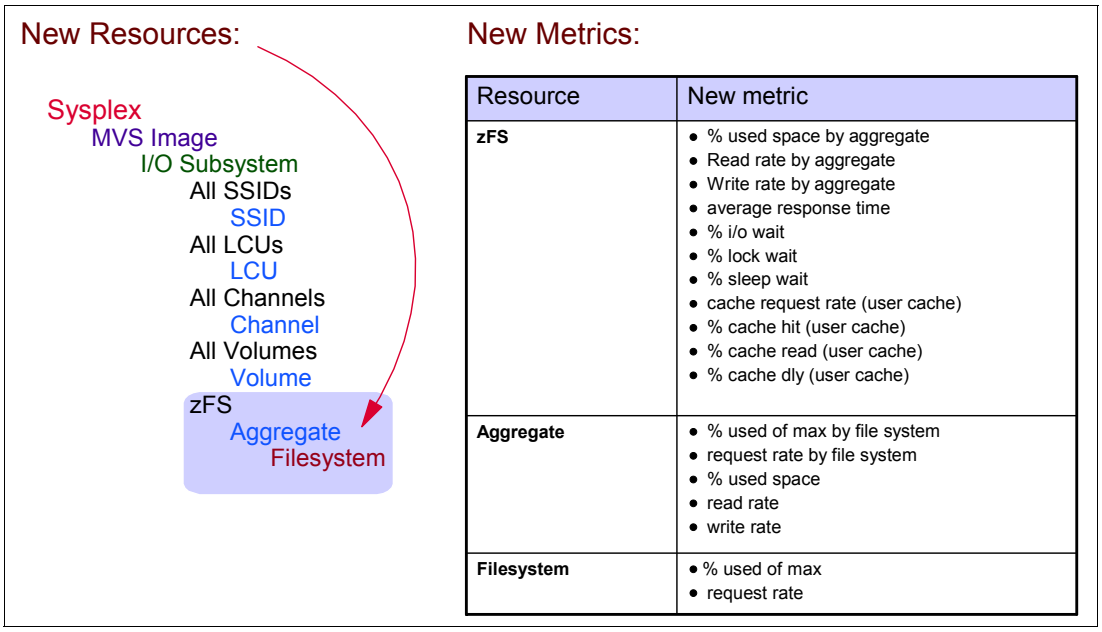
zFS support introduces new RMF Monitor III reporter messages, as follows:

```
ERB944I - Report is not available, reason code x.
1 = OMVS is not active (or not available)
2 = zFS is not active or shutting down
3 = backlevel data or no data from zFS interface
```

```
ERB945I – No aggregate found
Within the current report interval, RMF did not detect any zFS data for the aggregate.
```

15.14 New RMF PM resources and metrics

zFS support introduces new resources and metrics, which can be used by the RMF PM client.



Resource	New metric
zFS	• % used space by aggregate • Read rate by aggregate • Write rate by aggregate • average response time • % i/o wait • % lock wait • % sleep wait • cache request rate (user cache) • % cache hit (user cache) • % cache read (user cache) • % cache dly (user cache)
Aggregate	• % used of max by file system • request rate by file system • % used space • read rate • write rate
Filesystem	• % used of max • request rate

Figure 15-20 New RMF PM Resources and Metrics

15.15 Migration/coexistence considerations

With z/OS V1R7, RMF by default gathers the zFS activity data. You can suppress the gathering of zFS data by specifying the new Monitor III gatherer option `NOZFS`.

16

System Logger and XRC

The System Logger maintains log data in both the CF and DASD. Depending on the size of the CF log structure and the amount of activity, data could remain in the CF without having been off-loaded to the DASD offload data sets for many seconds, to minutes, to hours, which means that the CF resident log data would not be replicated by XRC to the remote site until it was offloaded to the DASD offload data sets.

This chapter describes the following enhancements:

- System Logger and XRC
- Coexistence support
- Setup for XRC
- Remote site recovery

16.1 System Logger and XRC

With the new DRXRC-type staging data sets, System Logger duplexed the log stream data written to a Coupling Facility structure in a staging data set in an asynchronous manner, as shown in Figure 16-1. Thus the existing eXtended Remote Copy (XRC) capabilities combined with the asynchronous writing to log stream data sets provides a more complete method of mirroring log data written into a coupling facility structure.

Note:

1. You can continue to duplex log data to log stream staging data sets for recovery purposes at your primary sysplex site in conjunction with the new DRXRC-type staging data sets, *but only one type of duplexing specification is allowed for a given log stream*. The log streams using conventional staging data sets will still be used for log data recovery at the primary site or at a secondary recovery site.
2. Peer-to-Peer Remote Copy (PPRC) or any non-XRC (LOGPLUS) configurations will not provide the correct environment for the proper use of DRXRC-type staging data sets.

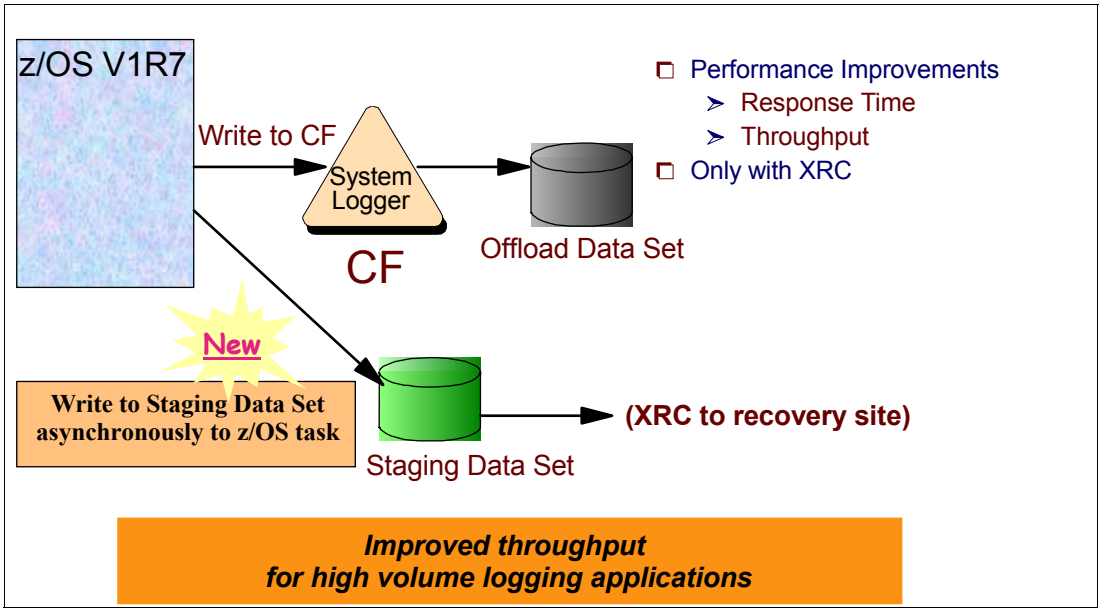


Figure 16-1 System Logger with XRC

16.2 Coexistence support

Following are the PTFs and APARs needed for z/OS V1R7 to coexist with down level z/OS systems in the same sysplex:

- There is an XRC/SDM and a System Logger APAR that are required prior to an installation activating the XRC+ support.
  - OA11515 for XRC/SDM
  - OA11568 for System Logger

- PTFs for compatibility APAR OA08661 are required on pre z/OS V1R7 systems in your sysplex:
  - UA17177 for z/OS V1R4
  - UA17178 for z/OS V1R5
  - UA17179 for z/OS V1R6

16.3 Setup for XRC

To use this enhancement, perform the following steps:

1. Make a System Logger definition as follows:
   - Define a **LOGSTREAM** with **STG_DUPLEX(YES)** and **DUPLEXMODE(DRXRC)** as follows:

     ```
     DEFINE LOGSTREAM
        STG_DUPLEX(YES)
        DUPLEXMODE(DRXRC)
     ```

2. Configure the volumes so only one DRXRC staging data set is allowed per volume because these volumes are defined to XRC. Since a single volume is used for each DRXRC-type staging data set, VSAM data striping cannot be used for these data sets.

   If the Enterprise Storage Server® (ESS) is used for DRXRC staging data sets, then we recommend sizing the DRXRC-type staging data set to be as large as the size of a maximum coupling facility structure space allocated for the log stream. This can be done since configuration of a proper size volume is possible when using ESS.

   System Logger only supports a staging data set up to 2 gigabytes in size. For fixed size storage devices (for example, a 3390-03), we recommend sizing of 2 GB for DRXRC staging data sets, which is specified as STG_SIZE(524288) on the log stream definition, to avoid having more than one staging data set on a single volume.

3. XRC definitions:

   Consider which XRC ERRORLEVEL specification on the **XADDPAIR** or **XSTART** commands to use for the LOGPLUS volumes. The ERRORLEVEL keyword indicates the level of detection and action to use if the XRC (logical) session consistency group time is compromised. Only the ERRORLEVEL(SESSION) preserves consistency across all volumes in the session. Use of VOLUME or group_name ERRORLEVEL allows session consistency time to advance for error conditions even for L+ volumes.

   Add the staging data set volumes using the new **LOGPLUS** parameter on the **XADDPAIR** TSO/E command or by using the ANTRQST XADD API. The **XADDPAIR** command is shown in Figure 16-2 on page 376.

```
XADDPAIR LOGXPLUS VOLUME(DRXRC1 DRXRC2 XRCP1C XRCUTL) ERRORLEVEL(SESSION) LOGPLUS

ANTA8101I XADDPAIR COMPLETE FOR VOLUME PAIR(DRXRC1,DRXRC2) FOR SESSION(TEST) WITH
ERRORLEVEL(SESSSION), SCSESSION(L+)

ANTA8004I XADDPAIR COMPLETE, VOLUME PAIR(XRCP01,XRCUTL) ADDED TO SESSION(TEST),
SCSESSION(L+)

ANTI8023I QUICK INITIALIZATION STARTED FOR XRC VOLUME PAIR(DRXRC1, DRXRC2)

ANTI8024I XRC VOLUME PAIR(XRCP0A,XRCS0A) IN DUPLEX STATUS AT TIME OF 2005.063
15:18:05.579805
```

Figure 16-2 XADDPAIR command and confirmation messages issued

The XQUERY report will show sc sessions, as shown in Figure 16-3.

```
ANTQ8200I XQUERY STARTED FOR SESSION(TEST)
ASNAME(ANTAS001) 862  --- OutPut
ANTQ8202I XQUERY STORAGECONTROL REPORT - 004
ANTQ8241I SC   SC S    RES   UTIL                  CURRENT
ANTQ8242I SSID SN ID T CNT  VOL    TIMEOUT  --STORAGE CONTROL TIME--
ANTQ8203I ------------------------------------------------------------
ANTQ8243I 9103 L+ F2   000A XRCP1C 00.04.05 2005.063 15:21:47.203313
ANTQ8243I 9103 L+ F3   000A XRCP1B 00.04.05 2005.063 15:21:47.205401
```

Figure 16-3 Output from the XQUERY report

Note: There is an L+ when System Logger is actively using the volume and an L- when System Logger is not actively using the volume.

System Logger commands

Examples of System Logger commands follow.

```
D LOGGER,CONN,LSN= TCHOPRA.TEST.LOGSTRM1, DETAIL

IXG601I   12.39.43  LOGGER DISPLAY 769
CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM SY1
LOGSTREAM                  STRUCTURE        #CONN  STATUS
---------                  ---------        ------ ------
TCHOPRA.TEST.LOGSTRM1       LIST01           000001 INUSE
DUPLEXING: LOCAL BUFFERS, STAGING DRXRC
    STGDSN: IXGLOGR.TCHOPRA.TEST.LOGSTRM1.SY1
            VOLUME=DRXRC1  SIZE=000300 (IN 4K)   % IN-USE=002
NUMBER OF LOGSTREAMS: 1
```

Figure 16-4 Display System Logger command for logstream connection

Staging data set information appears for both regular staging data sets and DRXRC staging data sets if the duplexing line on the connection reveals that staging data set duplexing is in place.

```
D LOGGER,L

XG601I   12.39.43  LOGGER DISPLAY 769
CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM SY1
LOGSTREAM                  STRUCTURE        #CONN  STATUS
---------                  ---------        ------ ------
TCHOPRA.TEST.LOGSTRM1       LIST01           000001 INUSE
DUPLEXING: LOCAL BUFFERS, STAGING DRXRC

NUMBER OF LOGSTREAMS: 1
```

Figure 16-5 Display System Logger command

16.4 Remote site recovery

To recover the volumes and staging data sets on a secondary system, make the following changes:

- New XRC ANTXINxx parmlib parameters for CONTIME category in:

  ```
  DefaultSessionid
  DefaultHlq
  ```

  These new parameters will need to be supplied for the logical session ID and the high-level qualifier of the state data set for System Logger to acquire the consistency time that is used for the DRXRC-type staging data sets. These XRC parmlib values may be dynamically changed by using the XSET command.

- XRC procedures before system IPL are:
  - Issue XEND or XSUSPEND.
  - Issue XRECOVER.

  XRECOVER will bring the data on the secondary volumes to a consistent, recoverable state where they are equivalent to the primary volumes, as of the given timestamp.

- IPL the first system in the sysplex with:
  - Specify **DRMODE=YES** on the IPL parms.
  - Reply **Y** to message IXG068D.

  This will confirm that the DRXRC-type staging data sets are to be included in log data recovery.

Note: The IBM Device Support facilities ICKDSF volume utility commands INIT and REFORMAT (XRCLOGGER parameter) aids in managing LOGPLUS designated volumes. For more information refer to *Device Support Facilities User's Guide and Reference, Release 17*, GC35-0033. The ICKDSF utility may be needed if a volume state is left as L+ even after Logger is no longer actively using the volume.

16.4.1 DRMODE support (DIL support for XRC+)

A new IEASYSxx parmlib member, shown in Figure 16-6 on page 378, specifies that a recovery system is being IPLed as part of a disaster recovery scenario and special handling of certain resources is required. This option (when YES is specified) causes System Logger to include, as part of its log data recovery processing, log data from log stream DRXRC-type staging data sets that were intended only for disaster recovery purposes.

```
IEASYSxx parmlib member:
    DRMODE={NO}|{YES}
    Default Value = NO
```

Figure 16-6 IEASYSxx parmlib member for specifying DRMODE

- **DRMODE = YES**
  When DRMODE=YES is specified, it indicates that a recovery system is being IPLed as part of a disaster recovery scenario and special handling of certain resources is required. The YES specification is intended to be used when the installation had previously configured the sysplex for specific disaster recovery capabilities. The YES option causes System Logger to include DRXRC-type staging data sets in its log data recovery for coupling facility structure-based log streams that had been connected prior to this IPL. That is, Logger attempts to recover log data for log streams with the STG_DUPLEX(YES), DUPLEXMODE(DRXRC) specification that had been connected prior to the IPL with the DRMODE=YES option. Also, a Logger confirmation message IXG068D is issued when the DRMODE=YES option is specified. When this option is specified, it is assumed that the necessary actions had been taken to establish the DASD consistency groups related to the Logger data sets.
- **DRMODE = NO**
  When DRMODE=NO is specified, or defaulted, the z/OS system is being IPLed without requesting special resource handling for disaster recovery purposes. The NO specification or default would normally be used for most system IPLs. The NO option causes System Logger to *not* include DRXRC-type staging data sets in its log data recovery for coupling facility structure-based log streams that had been connected prior to this IPL.

System Logger confirmation message

Figure 16-7 is a confirmation message issued by System Logger.

```
IXG068D CONFIRM LOGGER TO CONVERT DRXRC-TYPE RESOURCES ON THIS IPL. REPLY Y TO CONVERT
THEM OR N TO NOT CONVERT THEM
```

Figure 16-7 System Logger confirmation message

The possible responses are:

Y Allow Logger to include DRXRC staging data sets in its log data recovery for coupling facility structure based log streams that had been connected prior to this IPL.

N Request Logger to not convert DRXRC-type staging data sets for log stream recovery.

16.4.2 DRXRC considerations for DRMODE=YES IPL option

When the first recovery system at the secondary recovery site is initially being IPLed as part of a disaster recovery scenario and special handling of certain resources is required, the DRMODE=YES IPL option should be specified. For these specific IPLs, System Logger will issue decision message IXG068D and wait for a reply.

System Logger status

A **DISPLAY LOGGER, STATUS** command will result in an IXG601I message indicating the System Logger status. When a system is IPLed with the DRMODE=YES specification and

the IXGLOGR address space is started, this display command will result in an IXG601I message with a System Logger status of one of the following:

```
INITIALIZING - DRXRC CONVERSION NOT DONE
ACTIVE -  DRXRC CONVERSION NOT DONE
```

The `INITIALIZING` text in the status may appear before Logger issues message IXG068D shown in Figure 16-7. Once the status has the `ACTIVE` text, then before responding to message IXG068D, enter a **`DISPLAY LOGGER,L,LSN=*`** command and view the resulting IXG601I message to identify the log streams that have STAGING DRXRC included as a duplexing method. The log streams with duplexing method STAGING DRXRC will be affected by the response to the IXG068D message.

A reply of Y to message IXG068D will cause Logger to issue message IXG072I indicating the DASD consistency time that will be used for log data.

If Logger encounters errors while attempting to obtain the DASD consistency time from XRC, it will issue IXG070I and IXG071D. These messages will identify the error and request a response on how to continue. One option is to for Logger to retry obtaining the consistency time from XRC after the installation took steps to remedy the reason for the failure identified in message IXG070I.

Logstream log data recovery

The other option is to request that Logger use the current time as the consistency time and all the log data in the converted DRXRC-type staging data set resources will be used during log stream log data recovery. This means that the log data will be more recent (up to date) so there will not be any missing log data; however, some log data in the converted DRXRC-type staging data sets might not be time consistent with the rest of the Logger configuration.

DRXRC-type staging data sets

After identifying the log streams that had DRXRC-type staging data sets in use, these log streams will have their attributes updated by System Logger to the following specification:

```
STG_DUPLEX (NO), DUPLEXMODE ().
```

Message IXG224I is issued (to hardcopy) for each log stream that is updated.

This attribute change is made since the intent of DRXRC-type staging data sets is to allow the log stream recovery to occur on the secondary (remote) site. Once the log data has been recovered, the installation's need for using DRXRC-type staging data sets has been satisfied. It is expected that continued use of these recovered log streams on the secondary sysplex (recovery) site should not automatically cause new staging data set duplexing. The installation will have to re-establish any specific duplexing if the automatic log stream duplexing settings are not desired.

If Logger is unable to complete the conversion of all log streams that had DRXRC-type staging data sets in use before the IPL of this recovery system, then message IXG073D is issued requesting a response on how to continue. One option is to retry the process of converting (identifying and preparing) any remaining log stream DRXRC-type resources for log data recovery use.

The other option is to request that Logger cancel the DRXRC-type conversion process and complete Logger initialization. With this option, log streams that have DRXRC-type staging data sets that are not yet converted will be treated as if the system were IPLed with DRMODE=NO. If the only copy of these log streams' primary (interim) log data are in DRXRC-type staging data sets, then recovery for these log streams will most likely result in them being marked as damaged (that is, "possible loss of data"). A DRXRC-type staging data

set will only be used for recovery of log data if Logger was able to successfully complete the conversion of the corresponding log streams that had DRXRC-type staging data sets in use.

Log data recovery

After the appropriate log streams have their DRXRC-type staging data sets marked as eligible to be included for log data recovery, message IXG069I is issued and Logger performs its system-level recovery for log streams that have failed connections for the system being IPLed.

If message IXG069I has not been issued and this recovery system needs to be re-IPLed, then the DRMODE=YES option needs to be specified again to allow Logger to complete the preparation for converting the DRXRC-type staging data sets for use in log stream log data recovery.

Once message IXG069I is issued, then all the remaining recovery systems can be IPLed with DRMODE=NO, and Logger will use the previously defined and converted DRXRC-type staging data sets as part of the system level log stream recovery on those systems as well.

For this environment, if the appropriate XRC actions are not taken to ensure the Logger configuration on DASD is *consistent*, then unpredictable results can occur for Logger and its exploiters.

When a recovery system is IPLed with DRMODE=NO specified (or defaulted) or if the reply to IXG068D is N, then System Logger will not include DRXRC-type staging data sets in its log data recovery for coupling facility structure-based log streams that had been connected prior to this IPL. If the only copies of some log streams' primary (interim) log data are in DRXRC-type staging data sets, then the recovery for these log streams might not succeed (message IXG212E) and the log streams will be marked as damaged ("possible loss of data").

Regardless of the DRMODE= specification or the reply to an IXG068D message, System Logger's conventional log data recovery processing for log streams continues to be performed as part of system level/connection log stream recovery.

System Logger NOSTART support

If you do not want to bring the System Logger up on the secondary site, then specify IXGLOGR=NOSTART in one of the following parmlib members:

- IEFSSNxx parmlib member:

```
SUBSYS SUBNAME(LOGR) INITRTN(IXGSSINT) INITPARM(IXGLOGR=NOSTART)
```

- COMMNDxx parmlib member:

```
COM='SETSSI ADD,SUBNAME=LOGR,INITRTN=IXGSSINT,INITPARM=''IXGLOGR=NOSTART'''
```

17

ISPF enhancements

This chapter describes the following enhancements to ISPF in z/OS Version 1 Release 7:

- File tailoring trace
- Panel trace
- LIBDEF enhancements
- Support for large format sequential data sets
- Table utility
- HTML and XML highlighting
- UNICODE viewing
- DSINFO - APF and LINKLST status
- Enhanced sort
- Display system and user ID

17.1 File tailoring trace

File tailoring services read skeleton files and write tailored output that can be used to drive other functions. Frequently, file tailoring is used to generate job files for batch execution. In z/OS V1R7, support has been added to ISPF file tailoring to trace the file tailoring service calls (**FTOPEN**, **FTINCL**, **FTCLOSE**, and **FTERASE**) and the processing that occurs within the file tailoring code and processing of each skeleton statement.

The trace is started and stopped with the new **ISPFTTRC** command, which will either:

- Start the trace if it is not active.
- Stop and optionally view or edit the trace output where the trace is active.

The **ISPFTTRC** command can be used to trace file tailoring services that are invoked from any screen within the current ISPF session. Figure 17-1 shows the syntax for the **ISPFTTRC** command.

```
ISPFTTRC   [END]
           [VIEW]
           [QUIET]
           [READ( None | Summary | Detail ) ]
           [REC|RECORDS(* | All | None |
                      [Src|Source] [Data] [Cntl] |
                      [NOSrc|NOSource] [NOData] [NOCntl] ) ]
           [SCR|SCREEN( 0 | * | screenid )]
           [SVC|SERVICE( None | Detail ) ]
           [SKL|SKEL|SKELETON( * | skel_name | skel_mask ) ]
           [TBV|TBVARS( None | Detail ) ]
```

Figure 17-1 Syntax for the ISPFTTRC command

These are the options for the **ISPFTTRC** command:

END — Terminates the trace if active. No attempt is made to edit or view the trace data set.

VIEW — Terminates the trace if active and displays the trace data set.

QUIET — Prevents the issuing of the trace initialization and termination messages.

READ — Controls the generation of trace records when a skeleton member is being read into memory.

- None — No trace records are produced during the read processing.
- Summary — Generates summary information, including where the skeleton was loaded from (either ISPSLIB or LIBDEF data set), and the number of records read.
- Detail — Generates the same information as for the summary trace, but includes the skeleton source. This is the default setting.

RECORDS — Controls the generation of trace records during record processing of the skeleton member.

- * | All — Generates trace records for all skeleton record processing.
- None — No trace records are produced for any skeleton processing.
- Source — Generates trace records for the source skeleton record.
- Data — Generates trace records for the data records.

	`Cntl`	Generates trace records for the control statements.
	`NOSource`	Stops generation of trace records for the source skeleton records.
	`NOData`	Stops generation of trace records for the data records.
	`NOCntl`	Stops generation of trace records for the control statements.
`SCREEN`	Controls the generation of trace records based on the screen ID.	
	`0`	Generate trace records for the all logical screens. This is the default.
	`*`	Generate trace records for the current screen id.
	`Screenid`	Generates trace records only for the logical screen ID as specified. The *`screenid`* is a single character in the range 1-9, A-W.
`SERVICE`	Controls the generation of trace records for the file tailoring service calls, namely **`FTOPEN`**, **`FTINCL`**, **`FTCLOSE`**, and **`FTERASE`**.	
	`None`	No trace records are produced during the service call processing.
	`Detail`	Generates trace records for the **`FTOPEN`**, **`FTINCL`**, **`FTCLOSE`**, and **`FTERASE`** service calls, showing all the parameters. A trace record is produced both before and after the call processing, with the post record showing the return code from the service. This is the default setting.
`SKELETON`	Controls the generation of trace records based on the skeleton name.	
	`*`	Generate trace records for all skeletons. This is the default.
	`skel_name`	Generates trace records only for the skeleton name as specified.
	`skel_mask`	Generates trace records for skeletons matching the skel_mask. The mask may contain a percent sign (`%`) to represent a single character or an asterisk (*) to represent any number of characters.
	`TBVARS`	Controls the generation of trace records for table variables used in)DOT control statements.
	`None`	No trace records are produced during the)DOT control statement processing.
	`Detail`	Generates trace records showing the table key and name variables for each iteration of the table. Extension variables are not shown. This is the default setting.

The output from the trace is written to a dynamically allocated variable blocked data set that has a record length of 255. If the ddname ISPFTTRC is pre-allocated, this data set is used, provided it refers to a sequential, variable blocked data set with a record length that is at least 255.

Trace output

Figure 17-2 shows an example of the header information in the trace output:

- Date and time (GMT) when the trace was initialized
- ISPF level information as found in dialog variable ZISPFOS
- z/OS level information as found in dialog variable ZOS390RL
- The **`ISPFTTRC`** command with the invocation parameters
- The options that are in effect for the current execution of the file tailoring trace

- Module-level information for each of the ISPF modules associated with file tailoring and skeleton processing

```
******************************** Top of Data ********************************
========= ISPF File Tailoring Trace =========== 2005.123 15:42:09 GMT ==========
     ZISPFOS: ISPF FOR z/OS 01.07.00          ZOS390RL: z/OS   01.07.00
     ISPFTTRC Command: ISPFTTRC
    Options in Effect: SKELETON(*) SCREEN(0) RECORDS(SOURCE CNTL DATA)
                       READ(DETAIL) SERVICE(DETAIL) TBVARS(DETAIL)

     ISPFICWD Version: ISPFICWD 05045-BASE z/17
     ISPFICWL Version: ISPFICWL 04349-BASE z/17
     ISPFIEND Version: ISPFIEND 04349-BASE z/17
     ISPFIINT Version: ISPFIINT 04274-BASE z/17
     ISPFILBS Version: ISPFILBS 04349-BASE z/17
     ISPFITLR Version: ISPFITLR 04349-BASE z/17
     ISPFITRO Version: ISPFITRO 05017-BASE z/17
     ISPFITRV Version: ISPFITRV 04349-BASE z/17
================================================================================
TLD# Type Skeleton  Rec#   IM IF DO TB  Cd RC Data
---- ---- -------- ------  -- -- -- --  -- -- --------------------------------
****************************** Bottom of Data ******************************
```

Figure 17-2 Example of the header information in the trace output

The remainder of the trace is broken into a number of columns to show each trace record. These fields have the following meanings:

`TLD#` The task or screen identifier where the File Tailoring is being invoked.

`Type` The trace entry type. The valid types are:

- `Svc` These records are generated for calls to the ISPF File Tailoring Services and shows all the call parameters. This is limited to the FTOPEN, FTINCL, FTCLOSE, and FTERASE services. The generation of this type of trace record is controlled by the ISPFTTRC SERVICE parameter.
- `SvcR` These records are generated at return from the ISPF File Tailoring services. The trace includes the return code from the service. The FTCLOSE return trace entry will include an additional record showing the number of records written to the File Tailoring output data set.
- `Read` These records are generated reading a skeleton into storage. The generation of this type of trace records is controlled by the ISPFTTRC READ parameter. A summary trace does not show the skeleton source records.
- `Src` These records are generated as a skeleton record is selected for processing. The generation of this type of trace records is controlled by the ISPFTTRC RECORDS parameter.
- `CtlR` These records are generated at the conclusion of record processing when the record was determined to be a control statement. The generation of this type of trace records is controlled by the ISPFTTRC RECORDS parameter.
- `DatR` These records are generated at the conclusion of record processing when the record was determined to be a data record. The generation of this type trace records is controlled by the ISPFTTRC RECORDS parameter.
- `NoFT` These records are generated when the NOFT parameter is specified on the FTINCL service call, or the NT option is specified on the)IM control

statement. The generation of this type of trace records is controlled by the ISPFTTRC RECORDS parameter.

`Err` These records are generated when an ISPF File Tailoring processing error occurs and ISPF issues an error message. The records generated include both the short and long error messages.

`Skeleton` The ISPF skeleton name associated with the trace record.

`Record` Displays the record number associated with the trace entry type. For Read, Src, and CtlR the input record number from the skeleton member is displayed. For DatR and NoFT, the output record number is displayed. This field is blank for all other record types.

`IM` The current imbed level. The skeleton name specified on the FTINCL service is always level 1.

`IF` The current IF or SEL level. This field is blank if no)IF or)SEL statement is being processed.

`DO` The current DO level. This field is blank if no)DO structure is being processed.

`TB` The current Table level. This field is blank if no)DOT structure is being processed.

`Cd` The Condition value returned for the following skeleton control statements: **)IF**, **)SEL**, **)UNTIL**, or **)WHILE**

- **T** Indicates a True condition
- **F** Indicates a False condition

)ENDDO, or **)ENDDOT**

- **X** Indicates the corresponding)DO or)DOT control statement is terminating (that is, the exit condition has been met).

)IM with **OPT** parameter

- **X** Imbed member was not found, file tailoring processing will continue.

`RC` The Return Code, shown only for SvcR, DatR, and CtlR trace entries.

`Data` Trace data for the particular trace entry. The trace data will extend the full width of the output file and will wrap if required.

Figure 17-3 shows how the sample skeleton is traced by the **ISPFTTRC** command. A new appendix in the ISPF User's Guide, Volume I describes the new **ISPFTTRC** command and the contents of the file tailoring trace.

```
***************************** Top of Data ******************************
>>3>>START>> IF >>
)SET RC = 0
)IF &RC = 0 THEN )DO
>>3>>DO<<<<
)SET RC = 4
)ENDDO
>>3>>END<<<< RC=&RC
***************************** Bottom of Data ***************************

================================================================================
TLD# Type Skeleton  Rec#   IM IF DO TB  Cd RC Data
---- ---- -------- ------  -- -- -- --  -- -- --------------------------------
TLD2 Svc                                    FTOPEN  TEMP
-------------------------------------------- DD=ISP14085  DSN=PRADIER.SC65.SPFT
TLD2 SvcR                                 0 FTOPEN  TEMP
TLD2 Svc                                    FTINCL  SKELLAB
-------------------------------------------- DD=ISP14083  DSN=PRADIER.PRADIER.S
TLD2 Read SKELLAB       1                   >>3>>START>> IF >>
TLD2 Read SKELLAB       2                   )SET RC = 0
TLD2 Read SKELLAB       3                   )IF &RC = 0 THEN )DO
TLD2 Read SKELLAB       4                   >>3>>DO<<<<
TLD2 Read SKELLAB       5                   )SET RC = 4
TLD2 Read SKELLAB       6                   )ENDDO
TLD2 Read SKELLAB       7                   >>3>>END<<<< RC=&RC
TLD2 Read SKELLAB  ------------------------ Total Records=7
TLD2 Src  SKELLAB       1   1               >>3>>START>> IF >>
TLD2 DatR SKELLAB       1   1             0 >3>START> IF >
TLD2 Src  SKELLAB       2   1               )SET RC = 0
TLD2 CtlR SKELLAB       2   1             0 )SET RC = 0
TLD2 Src  SKELLAB       3   1               )IF &RC = 0 THEN )DO
TLD2 CtlR SKELLAB       3   1  1        T   0 )IF 0 = 0 THEN
TLD2 Src  SKELLAB       3   1  1            )DO
TLD2 CtlR SKELLAB       3   1  1  1         0 )DO
TLD2 Src  SKELLAB       4   1  1  1           >>3>>DO<<<<
TLD2 DatR SKELLAB       2   1  1  1         0 >3>DO<<
TLD2 Src  SKELLAB       5   1  1  1           )SET RC = 4
TLD2 CtlR SKELLAB       5   1  1  1         0 )SET RC = 4
TLD2 Src  SKELLAB       6   1  1  1           )ENDDO
TLD2 CtlR SKELLAB       6   1  1        X   0 )ENDDO
TLD2 Src  SKELLAB       7   1               >>3>>END<<<< RC=&RC
TLD2 DatR SKELLAB       3   1             0 >3>END<< RC=4
TLD2 SvcR                                 0 FTINCL  SKELLAB
TLD2 Svc                                    FTCLOSE
-------------------------------------------- DD=ISP14085  DSN=PRADIER.SC65.SPFT
TLD2 SvcR                                 0 FTCLOSE
TLD2 SvcR          ------------------------ Total Records=3
******************************** Bottom of Data ********************************
```

Figure 17-3 Sample skeleton and the trace generated by the **ISPFTTRC** *command*

17.2 Panel trace

In order to help application developers to debug the ISPF panel processing, support has been added to the ISPF Dialog Manager to trace the Panel Service calls (**DISPLAY**, **TBDISPL**, and **PQUERY**) and the processing that occurs within the Dialog Manager Panel code, including the

processing of the statements within the **)ABCINIT**, **ABCPROC**, **)INIT**, **REINIT**, and **)PROC** sections of the panel.

The trace is started and stopped using the new **ISPDPTRC** command, which will do one of the following:

- Start the trace if not active.
- Stop and optionally view or edit the trace output where the trace is active.

The **ISPDPTRC** command can be used to trace panel display services that are invoked from any screen within the current ISPF session. Figure 17-4 shows the syntax of the **ISPDPTRC** command.

```
ISPDPTRC  [END]
          [VIEW]
          [QUIET]
          [DSP|DISPLAY( None | In | Out | Both ) ]
          [PNL|PANEL( * | panel_name | panel_mask ) ]
          [READ( None | Summary | Detail ) ]
          [SCR|SCREEN( 0 | * | screenid )]
          [SECT|SECTION(* | All | None |
                [Init] [Reinit] [Proc] |
                [NOInit] [NOReinit] [NOProc] ) ]
          [SVC|SERVICE( None | Detail ) ]
```

Figure 17-4 The syntax of the ISPDPTRC command

These are the options for the **ISPDPTRC** command:

END — Terminates the trace if active. No attempt is made to edit or view the trace data set.

VIEW — Terminates the trace if active and views the trace data set. If an allocation for the DD ISPDPTRC is present, this data set is viewed. SYSOUT data sets are not supported.

QUIET — Prevents the issuing of the trace initialization and termination messages.

DISPLAY — Controls the generation of trace records resembling the panel as displayed at the terminal. Only the panel for the active screen is shown when a panel is being read into memory.

- None — No trace records are produced during the panel display processing.
- In — Generates trace records showing the panel, including data entered after the user has pressed the enter key or a function key.
- Out — This is the default setting. Generates trace records showing the panel as it is shown on the screen.
- Both — Generates both the In and Out display traces.

PANEL — Controls the generation of trace records based on the panel name.

- * — Generate trace records for all panels. This is the default.
- *panel_name* — Generates trace records only for the panel name as specified.
- *panel_mask* — Generates trace records for panels matching the panel_mask. The mask may contain a percent sign (%) to represent a single character or an asterisk (*) to represent any number of characters.

READ — Controls the generation of trace records when a panel is being read into memory.

- None — No trace records are produced during the read processing.

	`Summary`	Generates summary information, including where the panel was loaded from (either ISPPLIB or LIBDEF data set), and the number of records read, until the)END statement was detected. This is the default setting.
	`Detail`	Generates the same information as the Summary trace, but includes the panel source. Pre-processed panels are not able to be displayed.
`SCREEN`	Controls the generation of trace records based on the screen ID.	
	`0`	Generate trace records for all logical screens. This is the default.
	`*`	Generate trace records for the current screen ID.
	`Screenid`	Generates trace records only for the logical screen ID as specified. The ***`screenid`*** is a single character in the range 1-9, A-W.
`SECTION`	Controls the generation of trace records for the different panel logic sections. The default is all sections.	
	`* | All`	Generates trace records for all panel processing logic sections.
	`None`	Suppresses generation of trace records for any of the panel processing logic sections.
	`Init`	Generates trace records for the)ABCINIT and)INIT sections.
	`Reinit`	Generates trace records for the)REINIT section.
	`Proc`	Generates trace records for the)ABCPROC and)PROC sections.
	`NOInit`	Turns off the generation of trace records for the)ABCINIT and)INIT sections.
	`NOReinit`	Turns off the generation of trace records for the)REINIT section.
	`NOProc`	Turns off the generation of trace records for the)ABCPROC and)PROC sections.
`SERVICE`	Controls the generation of trace records for the panel processing service calls, namely DISPLAY, TBDISPL, and PQUERY.	
	`None`	No trace records are produces during the service call processing.
	`Detail`	Generates trace records for the DISPLAY, TBDISPL and TBQUERY service calls, showing all the parameters. A trace record is produced both before and after the call processing, with the post record showing the return code from the service. This is the default setting.

The output from the trace is written to a dynamically allocated variable blocked data set that has a record length of 255. If the ddname ISPDPTRC is pre-allocated, this data set is used, provided it refers to a sequential, variable blocked data set with a record length that is at least 255.

Figure 17-5 shows an example of the header information in the trace output:

- Date and time (GMT) when the trace was initialized
- ISPF level information as found in dialog variable ZISPFOS
- z/OS level information as found in dialog variable ZOS390RL
- The **`ISPDPTRC`** command with the invocation parameters
- The options that are in effect for the current execution of the file tailoring trace
- Module-level information for each of the ISPF modules associated with panel processing

```
========= ISPF Panel Trace ==================== 2005.124 13:41:02 GMT ==========
     ZISPFOS: ISPF FOR z/OS 01.07.00          ZOS390RL: z/OS   01.07.00
     ISPDPTRC Command: ISPDPTRC
    Options in Effect: PANEL(*) SCREEN(0) SECTION(INIT REINIT PROC)
                       READ(SUMMARY) SERVICE(DETAIL) DISPLAY(BOTH)

     Physical Display: PRI=24x80  ALT=32x80  GUI=OFF

     ISPCDI   Version: ISPCDI 04349-BASE z/17
     ISPDPA   Version: ISPDPA 04349-BASE z/17
     ISPDPE   Version: ISPDPE 04349-BASE z/17
     ISPDPL   Version: ISPDPL 04349-BASE z/17
     ISPDPP   Version: ISPDPP 04349-BASE z/17
     ISPDPR   Version: ISPDPR 04349-BASE z/17
     ISPDPS   Version: ISPDPS 04349-BASE z/17
     ISPDTD   Version: ISPDTD 04349-BASE z/17
     ISPPQR   Version: ISPPQR 04349-BASE z/17
     ISPDPTR0 Version: ISPDPTR0 05017-BASE z/17
================================================================================
TLD# Type  Panel   Section Cd  RC  Data
---- ---- -------- ------- -- ---  ------------------------------------------
```

Figure 17-5 Example of the header information in the trace output

The remainder of the trace is broken into columns to show each trace record. The descriptions of these fields are as follows:

`TLD#` The task or screen identifier where the panel service is being invoked.

`Type` The trace entry type. The valid types are:

- `Svc` These records are generated for calls to the ISPF display services and shows all the call parameters. This is limited to the DISPLAY, TBDISPL, and PQUERY services. The generation of this type of trace record is controlled by the ISPDPTRC SERVICE parameter.
- `SvcR` These records are generated at return from the ISPF display services. The trace includes the return code from the service.
- `Read` The generation of this type of trace record is controlled by the ISPDPTRC READ parameter. A summary trace does not show the panel source records. The source of pre-processed panels cannot be displayed.
- `Dsp0` These records are generated displaying an ISPF panel at the screen. Attribute bytes are also included in the display. The generation of this type of trace record is controlled by the ISPDPTRC DISPLAY parameter.
- `DspI` These records are generated after a user has pressed the Enter key or function key, and show the data displayed on the ISPF panel at that time. Attribute bytes are also included in the display. The generation of this type of trace record is controlled by the ISPDPTRC DISPLAY parameter.
- `PrcR` These records are generated during the processing of the panel logic sections, including)INIT,)REINIT,)PROC,)ABCINIT and)ABCPROC. The data as displayed resembles that of the original panel, but may not be identical to it. Where an assignment statement includes dialog variables or functions an additional record is displayed showing the result of the assignment. The generation of this type of trace record is controlled by the ISPDPTRC SECTION parameter.

Err — These records are generated when a ISPF panel processing error occurs and ISPF issues an error message. The records generated include both the short and long error messages.

Panel — The ISPF panel name associated with the trace record.

Section — The logic section associated with the PrcR type trace record.

Cd — The Condition value returned for IF and ELSE panel statements.

- **T** Indicates a True condition.
- **F** Indicates a False condition.

RC — The Return Code, shown only for SvcR, and PrcR type trace records.

Data — Trace data for the particular trace entry. The trace data will extend the full width of the output file and will wrap if required.

Figure 17-6 shows a display output trace generated for panel ISRUTIL. It includes a scale line across the top and down the side of the panel and includes panel size, cursor position, and an indication of the key or command entered. It also shows a trace generated from the processing of the)PROC section of panel ISRUTIL after the number 4 was entered in the command field. Statements skipped as the result of a false condition on an IF or ELSE statement are never displayed. In addition, the panel trace will always split the value pairs for the TRANS functions onto separate records, making the trace more readable. The result of an assignment statement is only shown when the assignment statement includes a dialog variable, including panel control variable, or a panel function.

A new appendix in the *Interactive System Productivity Facility (ISPF) User's Guide Volume 1*, SC34-4822 describes the new **ISPDPTRC** command and the contents of the panel trace.

```
================================================================================
TLD# Type  Panel    Section Cd  RC  Data
---- ---- -------- ------- -- ---  ------------------------------------------
TLD2 DspI                          0----+----1----+----2----+----3----+----4----+
TLD2 DspI ISRUTIL                  |.. Menu. Help.
TLD2 DspI ISRUTIL                  |-----------------------------------------
TLD2 DspI ISRUTIL                  |.                          .Utility Selection
TLD2 DspI ISRUTIL                  |.Option ===>&4
TLD2 DspI ISRUTIL                  +
TLD2 DspI ISRUTIL                  |.1 .Library.....Compress or print data set.
TLD2 DspI ISRUTIL                  |.               .rename, delete, browse, edi
TLD2 DspI ISRUTIL                  |.2 .Data Set....Allocate, rename, delete, cat
TLD2 DspI ISRUTIL                  |.               .information of an entire da
TLD2 DspI ISRUTIL                  1.3 .Move/Copy...Move, or copy members or data
TLD2 DspI ISRUTIL                  |.4 .Dslist......Print or display (to process)
TLD2 DspI ISRUTIL                  |.               .Print or display VTOC infor
...
...
TLD2 DspI ISRUTIL                  |.15.Search-ForE.Search data sets for strings
TLD2 DspI ISRUTIL                  |.16.Tables......ISPF Table Utility
TLD2 DspI ISRUTIL  --------------- Screen=23x80  Cursor=4/15  Key=ENTER
TLD2 PrcR ISRUTIL  PROC         0  &ZCMDWRK=&Z
TLD2 PrcR ISRUTIL  PROC       ->   &ZCMDWRK=''
TLD2 PrcR ISRUTIL  PROC    T    0  IF(&ZCMD ¬= &Z)
TLD2 PrcR ISRUTIL  PROC         0    &ZCMDWRK=TRUNC(&ZCMD,'.')
TLD2 PrcR ISRUTIL  PROC       ->     &ZCMDWRK=4
TLD2 PrcR ISRUTIL  PROC         0    &ZTRAIL=.TRAIL
TLD2 PrcR ISRUTIL  PROC       ->     &ZTRAIL=''
TLD2 PrcR ISRUTIL  PROC    F    0    IF(&ZCMDWRK = &Z)
TLD2 PrcR ISRUTIL  PROC         0  &ZSEL=TRANS(TRUNC(&ZCMD,'.')
TLD2 PrcR ISRUTIL  PROC    +             1,'PGM(ISRUDA) PARM(ISRUDA1) SCRNAME(LI
TLD2 PrcR ISRUTIL  PROC    +             2,'PGM(ISRUDA) PARM(ISRUDA2) SCRNAME(DS
TLD2 PrcR ISRUTIL  PROC    +             3,'PGM(ISRUMC) SCRNAME(MCOPY)'
TLD2 PrcR ISRUTIL  PROC    +             4,'PGM(ISRUDL) PARM(ISRUDLP) SCRNAME(DS
...
...
TLD2 PrcR ISRUTIL  PROC    +             14,'PGM(ISRSFM) SCRNAME(SRCHFOR)'
TLD2 PrcR ISRUTIL  PROC    +             15,'PGM(ISRSEPRM) PARM(S4) SCRNAME(SRCH
TLD2 PrcR ISRUTIL  PROC    +             16,'PGM(ISRUTABL) NEWPOOL SCRNAME(TBLUT
TLD2 PrcR ISRUTIL  PROC    +             ' ',' '
TLD2 PrcR ISRUTIL  PROC    +             '*','?')
TLD2 PrcR ISRUTIL  PROC       ->   &ZSEL='PGM(ISRUDL) PARM(ISRUDLP) SCRNAME(DSLI
....
```

Figure 17-6 A trace generated by the ISPDPTRC command

17.3 LIBDEF enhancements

The ISPF LIBDEF service provides for the dynamic definition of application data sets, thus allowing application data sets to be specified during an ISPF session. This eliminates the need for allocate statements to define all application data sets before invoking an ISPF session.

Enhancements have been made to the ISPF LIBDEF service to have the STACK option (rather than UNCOND) set as the default for the LIBDEF service in order to avoid situations where nested ISPF applications can inadvertently free data sets LIBDEFed by previous

applications. This can occur when a nested application uses the LIBDEF service and does not specify one of the options COND, UNCOND, STACK, or STKADD, thereby getting the default of UNCOND. This causes the LIBDEF by the nested application to replace the library LIBDEFed by the previous application. After the nested application terminates and control returns to the previous application it is likely to fail because its LIBDEFed library has been inadvertently removed.

The default LIBDEF processing option (COND, UNCOND, STACK, or STKADD) can be set in the ISPF configuration table. Changes to this setting can be made using the ISPF Configuration Utility, which is documented in the ISPF Planning and Customizing manual. The new option is contained in the Miscellaneous DM Settings, under menu item 5, ISPDFLTS, CUA® Colors and other DM Settings, as shown in Figure 17-7.

```
                    Modify ISPDFLTS and Other DM Settings
Command ===>
                                                                More:     +
  ISPDFLTS Settings
                                          Enter "/" to select option
  Number of Rows for TBADD   1              Enable ISPF Exits
  SAS/C TCP/IP Prefix Value  DEFAULT        Use z/OS Unix Sockets
  SAS/C TCP/IP Data Value    DEFAULT

  Command Table Settings
  APPLID for Site Command Table 1 . . .       Site Command Table Search Order
  APPLID for Site Command Table 2 . . .       1  1. Before
  APPLID for Site Command Table 3 . . .          2. After
  APPLID for User Command Table 1 . . .
  APPLID for User Command Table 2 . . .
  APPLID for User Command Table 3 . . .

  Miscellaneous DM Settings
  Maximum Number of Split Screens . . 8
  Year 2000 Sliding Rule  . . . . . 65
  Retrieve Command Stack Size . . . . 512
  TPUT Buffer Size  . . . . . 0
  Default Primary Panel . . . . . ISP@MSTR
  Default LIBDEF Processing Option    STACK     (COND UNCOND STACK or STKADD)
```

Figure 17-7 New default LIBDEF processing option

17.4 LIBDEF service enhancements

The LIBDEF service is changed so the default processing option is no longer UNCOND but is the option specified in the ISPF configuration table. If an option has not been specified in the configuration table then the default will be UNCOND.

The LIBDEF service is also changed so that if the processing option is STKADD (either specified or defaulted) and there is no existing stack for the lib-type specified, a return code of 4 is provided to the application. This allows the application to determine if the lib-type was already stacked prior to the LIBDEF STKADD and therefore can ensure the LIBDEFed libraries are removed before the application terminates. Figure 17-8 shows an example of some REXX code using the new return code to ensure the LIBDEFed ISPPLIB data sets are remove before termination.

These changes are documented in the description of the LIBDEF service in the ISPF Services Guide.

```
/* REXX */
ADDRESS ISPEXEC
 .
 .
"LIBDEF ISPPLIB DATASET ID('dsn') STKADD"
STKADD_RC = RC
 .
 .
IF STKADD_RC <= 4 THEN DO    /* LIBDEF ADDED DSN TO ISPPLIB  */
  "LIBDEF ISPPLIB"           /* REMOVE OUR LIBDEF D ISPPLIB  */
  IF STKADD_RC = 0 THEN      /* ISPPLIB WAS ALREADY LIBDEF D */
    "LIBDEF ISPPLIB"         /* REMOVE THAT ISPPLIB ALSO     */
END
EXIT
```

Figure 17-8 Example of some REXX code using the new LIBDEF return code

17.5 Support for large format sequential data sets

Large format sequential data sets are a new format data set available with z/OS V1R7. Many ISPF functions have been modified in order to allow ISPF users to process large format sequential data sets. Many of these changes will be transparent to the user. The ISPF functions that have been modified to support large format sequential data sets include:

- Browse
- Edit/View
- DSINFO Service
- Data Set Allocate (option 3.2)
- Data Set List (option 3.4)
- Data Set Move/Copy (option 3.3)

Browse, Edit, and View now support large format sequential data sets. DSINFO service also was modified and returns the value LARGE in variable ZDSDSNT for large format sequential data sets.

ISPF option 3.2 supports the creation of large format sequential data sets. This requires a value of LARGE to be entered in the “Data set name type” field on the Allocate New Data Set panel, as shown in Figure 17-9.

```
                          Allocate New Data Set
Command ===>

Data Set Name  . . . : SYS1.LARGE.DATASET

Management class . . .                (Blank for default management class)
Storage class  . . . .                (Blank for default storage class)
 Volume serial . . . . SBOXFC         (Blank for system default volume) **
 Device type . . . . .                (Generic unit or device address) **
Data class . . . . . .                (Blank for default data class)
 Space units . . . . . CYLINDER       (BLKS, TRKS, CYLS, KB, MB, BYTES
                                       or RECORDS)
 Average record unit                  (M, K, or U)
 Primary quantity  . . 10000          (In above units)
 Secondary quantity    0              (In above units)
 Directory blocks  . . 0              (Zero for sequential data set) *
 Record format . . . . FB
 Record length . . . . 80
 Block size  . . . . . 27920
 Data set name type    LARGE          (LIBRARY, HFS, PDS, LARGE, BASIC, *
                                       EXTREQ, EXTPREF or blank)
 Expiration date . . .                (YY/MM/DD, YYYY/MM/DD
Enter "/" to select option             YY.DDD, YYYY.DDD in Julian form
   Allocate Multiple Volumes           DDDD for retention period in days
                                       or blank)
```

Figure 17-9 Allocate New Data Set panel

ISPF option 3.2 supports the display of information for large format sequential data sets. A large format sequential data set is indicated by a value of LARGE in the “Data set name type” field on the Data Set Information panel.

The Attribute and Total views for the ISPF Data Set List utility (option 3.4) identify large format sequential data sets by displaying a value of PS-L in the Dsorg column (Figure 17-10).

```
DSLIST - Data Sets on volume SBOXFC                              Row 1 of 2
Command ===>                                                  Scroll ===> PAGE

Command - Enter "/" to select action               Dsorg  Recfm  Lrecl  Blksz
------------------------------------------------------------------------------
         SYS1.LARGE.DATASET                          PS-L  FB        80  27920
         SYS1.VTOCIX.SBOXFC                          PS    F       2048   2048
***************************** End of Data Set list *****************************
```

Figure 17-10 Data Set List utility panel

17.6 Table utility

The ISPF table utility (option 3.16) provides functions for processing ISPF tables. This option is more user-friendly than ISPF option 7.4 (Dialog Test – Tables), which is limited to working with one table row at a time. The ISPF Table Utility can also be invoked using the Utilities pull-down available on many ISPF panels.

The Table Utility entry panel (Figure 17-11) allows you to specify a table data set or DD, a table name, and an option to be performed.

```
                          ISPF Table Utility
Option ===>

   blank Display table list                 E Edit table
       B Browse table                       I Import table data

Enter one of the parameters below:
   Table Data Set . . _____________________________________________
   or Table DD  . . . ISPTLIB_ (Default is ISPTLIB)

   Table Name . . . . ________ (Blank or pattern for table selection list)

   Import Data Set    _____________________________________________

   Enter "/" to select option
   _  Open table in SHARE mode
```

Figure 17-11 Table Utility entry panel

The Edit and Browse functions allow you to view the data in the rows of an ISPF table in full-screen mode (that is, multiple rows are displayed on a screen). Figure 17-12 shows an example of the browse function screen.

```
BROWSE                  ISPF Table SMPECMDS                   Row 1 to 4 of 4
Command ===>                                                 Scroll ===> PAGE
                                                              Shift ===> PAGE
    ZCTVERB  ZCTTRUNC ZCTACT                                ZCTDESC
    ----+--- ----+--- ----+----1----+----2----+----3----+- ----+----1----+----2
    RFIND    0        PASSTHRU                             ALLOW SMP/E DIALOGS
    CONTINUE 4        PASSTHRU                             ALLOW SMP/E DIALOGS
    TUTORIAL 5        SELECT PGM(ISPTUTOR) PARM(&TUTORPAN) INVOKE TUTORIAL
    START    0                                             ALLOW ISPF TO HANDLE
****************************** Bottom of data ********************************
```

Figure 17-12 Browse function screen

The ISPF table utility edit function allows data in a table to be changed simply by overtyping the displayed column values. Specifying a numeric value immediately after a line command causes the command to process multiple rows. Using the **E** line command against a row displays a panel showing the extension variable values for the row. Here the user can create, modify, or delete extension variables for the row.

The browse and edit **EXPORT** and **FEXPORT** (fast export) primary commands are used to invoke the export function. This function is used to write the data within a table to a sequential data set. The **EXPORT** primary command displays a panel allowing the user to change the format of the data written to the data set. The data set created by the export function can be used as input by the Table Utility import function to create new ISPF tables.

Primary commands are provided to support processing against the entire table, including changing the format of the displayed data, as follows:

STRUCT	Display Table Structure panel
SORT	Display Table Sort Definition panel
FIND	Search for a string in the table rows
RFIND	Repeat the last **FIND** command

EXPORT	Display Table Export Layout panel
FEXPORT	Fast **EXPORT** command (bypass Table Export Layout panel)
SAVE	Save table changes to disk
CANCEL	Terminate Edit without saving changes
STATS	Display Table Statistics panel
EXPAND	Display table variable value in expand popup panel
INSERT	Insert a blank row at the top of the table

Column values are displayed in scrollable fields allowing columns to be scrolled left or right, and individual column values to be expanded and displayed in a pop-up window. These are the line commands that allow you to work with individual or multiple table rows:

I	Insert one or more rows
D	Delete one or more rows
R	Repeat one or more rows

The ISPF table utility import function is invoked through the entry panel. The import function reads the data within a sequential data set and either creates a new ISPF table or replaces an existing table. The data in the sequential data set is required to be in the format generated by the Table utility export function.

A description of the ISPF Table Utility can be found in the *Interactive System Productivity Facility (ISPF) User's Guide Volume II*, SC34-4823.

17.7 HTML and XML highlighting

HILITE is used to control the use of color in the editor by changing the settings for the enhanced color and language-sensitive editing features. In z/OS V1R7, support has been added to ISPF to support HTML and XML highlighting.

Figure 17-13 shows the ISPF editor highlighting for a XML document.

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       PRADIER.PRADIER.SKELS(XML#1) - 01.00            Columns 00001 00072
Command ===>                                                 Scroll ===> HALF
****** ***************************** Top of Data ****************************
000100 <?XML VERSION="1.0" ENCODING="UTF-8"?>
000200 <HELPSESSION>
000300  <APPLYTO LANGUAGE="MUI">
000400   <WINDOWSETTINGS NORESIZE="TRUE">
000500             <TITLE>REMOTE ASSISTANCE</TITLE>
000600    <LEFT>100</LEFT>
000700    <TOP>100</TOP>
000800    <WIDTH>410</WIDTH>
000900    <HEIGHT>135</HEIGHT>
001000    <LAYOUT>KIOSK</LAYOUT>
001100   </WINDOWSETTINGS>
001200  </APPLYTO>
002300 </HELPSESSION>
002400
002500
002600
002700
002800
002900
003000
003100
****** **************************** Bottom of Data ***************************
```

Figure 17-13 ISPF Editor highlighting for a XML document

17.8 UNICODE viewing

The ISPF browse function has been enhanced to support viewing data sets that contain both Unicode and ASCII data. The browse **DISPLAY** command has been enhanced to allow the user to view Unicode and ASCII data in a data set. The Browse **FIND** command has been enhanced to allow users to find Unicode and ASCII strings in a data set.

With DB2 Version 8 the DB2 catalog data is stored in Unicode format. Therefore, data sets created using DB2 extraction tools can contain a mix of EBCDIC and Unicode data. This presents a problem for users who want to view these types of data sets from z/OS since there are no generally available tools that support viewing both EBCDIC and Unicode data.

The browse **DISPLAY** primary command has been modified to support the syntax shown in Figure 17-14.

```
DISPLAY CCSID ccsid_number | ASCII | USASCII | EBCDIC | UCS2 | UTF8 | UTF16 | UTF32
        [LINE start_line [end_line] ]
        [ COLS start_col [end_col] ]

DISPLAY RESET
```

Figure 17-14 Syntax added for the browse Display command

Multiple **DISPLAY** commands can be issued and the specifications are merged, with later specifications taking precedence over earlier specifications when there is a conflict. The **LINE**, **COLS**, and **CCSID** parameters can be specified in any order. The **LINE** and **COLS** parameters are

optional, and if omitted, the command applies to the full line or column range. The following describes the parameters:

LINE — Identifies the lines on the display that contain Unicode or ASCII data. If only one start_line is specified, the command only applies to that line.

COLS — Identifies the columns on the display that contain Unicode or ASCII data. If only one start_col is specified, the command only applies to that column.

CCSID *ccsid_number* | ASCII | USASCII | EBCDIC | UCS2 | UTF8 | UTF16 | UTF32 — Identifies the CCSID for the Unicode or ASCII data. ccsid_number may be specified using the acronyms UTF8, UTF16, UTF32, UCS2, UNICODE, ASCII, USASCII, or EBCDIC. UTF8 represents CCSID 1208, UTF16 - CCSID 1200, UTF32 - CCSID 1232, ASCII - CCSID 850, USASCII - CCSID 819, UCS2 & UNICODE - CCSID 17584, and EBCDIC - CCSID 1047.

RESET — Resets all definitions made with the **DISPLAY** command as described previously.

All definitions are also reset when the user leaves the current BROWSE session.

Figure 17-15 shows a data set browsed with the default EBCDIC option. Figure 17-16 shows the same data set after the **display line 2 ascii** command has been issued. This command will display the second line using ASCII format.

```
 BROWSE    PRADIER.ALOC.TEST                          Line 00000000 Col 001 080
 Command ===>                                                Scroll ===> CSR
******************************** Top of Data ********************************
/ÂÄÀÁÃÅÇÑ¦,%_>?øÉÊËÈÍÎÏÌ`:                                        00000010
/ÂÄÀÁÃÅÇÑ¦,%_>?øÉÊËÈÍÎÏÌ`:                                        00000011
******************************** Bottom of Data ********************************
```

Figure 17-15 Data set displayed with the default EBCDIC option

```
 BROWSE    PRADIER.ALOC.TEST                            Converted data shown
 Command ===> display line 2 ascii                           Scroll ===> CSR
******************************** Top of Data ********************************
/ÂÄÀÁÃÅÇÑ¦,%_>?øÉÊËÈÍÎÏÌ`:                                        00000010
abcdefghijklmnopqrstuvwxyz@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@------±±
******************************** Bottom of Data ********************************
```

Figure 17-16 Data set after the **Display line 2 ascii** *command*

Figure 17-17 displays the hex format of the data set after the **display line 2 ascii** command has been issued in order to show that only the display format was changed, leaving the data set without modification.

```
 BROWSE    PRADIER.ALOC.TEST                          Line 00000000 Col 001 080
 Command ===> hex vert                                       Scroll ===> CSR
******************************** Top of Data ********************************
 ------------------------------------------------------------------------------
/ÂÄÃÁÀÅÇÑ¦,%_>?øÉÊËÈÍÎÏÌ`:                                          00000010
6666666666666667777777777744444444444444444444444444444444444444444444FFFFFFFF
123456789ABCDEF0123456789A000000000000000000000000000000000000000000000000000010
 ------------------------------------------------------------------------------
abcdefghijklmnopqrstuvwxyz@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@------±±
6666666666666667777777777744444444444444444444444444444444444444444444FFFFFFFF
123456789ABCDEF0123456789A000000000000000000000000000000000000000000000000000011
 ------------------------------------------------------------------------------
```

Figure 17-17 Hex format of the data set after the display command have been issued

The browse **FIND** primary command has also been modified to support the searching for Unicode or ASCII character strings in the data set being browsed, as shown in Figure 17-18.

```
FIND    C'mystring' UTF8|ASCII|USASCII
```

Figure 17-18 Syntax added for the find display command

This form of the **FIND** command will find occurrences of ***mystring*** within the data being browsed, where ***mystring*** has been stored using in UTF8, ASCII or USASCII CCSID format. This form of the **FIND** command is only available for the character FIND command and only for UTF8, ASCII, or USASCII strings. The case of the string is important since no case conversion is done for either the string being searched for or the data being searched.

17.9 DSINFO - APF and LINKLST status

DSINFO service has been enhanced to return APF and LINKLST status. Values in implicit variable ZDSAPF contain APF status, as follows:

YES Data set in APF list

NO Data set not in APF list

ERR Unable to determine APF status

Values in implicit variable ZDSLNK contain LINKLIST status, as follows:

YES Data set in link list

NO Data set not in link list

ERR Unable to determine LINKLST status

Figure 17-19 shows a piece of code using ZDSAPF and ZDSLNK variables.

```
EDIT       SYS1.OS390.CLIST(PRADIER9) - 01.01             Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 PROC 0
000500    SET DSNAME = SYS1.LINKLIB
000600    ISPEXEC DSINFO DATASET('&DSNAME')
000700
000750    WRITE   &DSNAME ZDSAPF   -> &ZDSAPF
000751    WRITE   &DSNAME ZDSLNK   -> &ZDSLNK
****** **************************** Bottom of Data ****************************

  SYS1.LINKLIB ZDSAPF   -> YES
  SYS1.LINKLIB ZDSLNK   -> YES
***
```

Figure 17-19 Piece of code using ZDSPF and ZDSLNK variables

17.10 Enhanced sort

ISPF Member and Data Set lists were able to be sorted on fields such as Name, Size, Creation Date, and so forth, but the sort direction was pre-set depending on the field chosen. In general, character fields (for example, Name, DSORG, and Userid) were sorted in ascending order while numeric fields (such as Size, Changed Date) were sorted in descending order.

The Member and Data Set list **SORT** commands have been enhanced to allow an (optional) direction to be specified on both the major and minor sort fields. An "A" or "D" character immediately following a field name indicates that the field is to be sorted in an ascending or descending direction, respectively. If a direction is not specified then the default (or preset) sort direction is used. Figure 17-20 shows the new syntax for the sort command.

```
SORT    major-field [A|D] [minor-field] [A|D]
```

Figure 17-20 Syntax for the **SORT** *command*

17.11 Display system and user ID

Two new commands, **SYSNAME** and **USERID**, have been provided to allow the user to display the system name and user ID on all ISPF panels. This data is displayed at the top left-hand corner of a panel in the panel information area used by the **PANELID** and **SCRNAME** commands.

These commands allow users who have to log on to multiple z/OS systems to quickly recognize which particular system a terminal emulator session is for.

The syntax for the commands is:

SYSNAME[ON|OFF] Display | Remove system name

USERID[ON|OFF] Display | Remove user ID

```
  Menu  List  Mode  Functions  Utilities  Help
_______________________________________________________________________________
SC65 PRADIER                  ISPF Command Shell
Enter TSO or Workstation commands below:

===> sysname;userid
```

Figure 17-21 Example of sysname and userid commands

18

Device allocation and commands

This chapter describes the relationship between device allocation and **VARY OFFLINE** or **UNLOAD** commands, and the enhancements made in this area.

18.1 Device allocation overview

Device allocation, which selects devices for each request, must examine all eligible devices to determine which is the best and it needs a static view of the devices from which to select. This means that devices cannot be changing state from the time when allocation gathers its list of eligible devices until it actually selects a device. Meanwhile, it is possible to take devices offline or bring devices online with a **VARY OFFLINE/ONLINE** command, or an **UNLOAD** command can be issued to unload a device. Because of that, a serialization process must be implemented to ensure a proper system execution.

18.1.1 Device allocation serialization

To provide a serialization between the **VARY** or **UNLOAD** commands and device allocation, three resources are used:

SYSIEFSD.Q4 This resource is used to serialize changes to the unit control block (UCB) by allocation and **VARY** command processing. Allocation obtains the resource as SHARED while the **VARY** command obtains it EXCLUSIVELY. If a **VARY** command hangs while holding this resource all allocations will also hang.

SYSIEFSD.CHNGDEVS This resource is used in the processing of **UNLOAD** commands and allocation. Allocation obtains the resource as SHARED while the **UNLOAD** command obtains it EXCLUSIVELY. But since **VARY OFFLINE** may need to unload a tape device before varying it offline, **VARY** processing gets the CHNGDEVS resource exclusive. This serializes the **UNLOAD** and **VARY** command processing, as well as allocation. If the resource is hung, then other **UNLOAD** commands will be hung behind it and allocations may also hang.

SYSIEFSD.VARYDEV This resource is used in the processing of **VARY** commands. If the resource is hung, then all **VARY** commands will hang behind it.

The **VARY** command and the **UNLOAD** command hold the resources SYSIEFSD.Q4 or SYSIEFSD.CHNGDEVS exclusively while processing their respective commands. Device allocation waits for the ENQs to become free before continuing allocating devices to batch jobs. Since these commands require that I/O be done to the device, the delay can be substantial. When the device is broken (non-responsive) and an attempt is made to remove this device from the system, all allocations are held up while the **VARY** command waits for the device time out specified as a missing interrupt handler (MIH) value in the IECIOSxx parmlib member.

18.1.2 Device allocation

Device allocation is the assignment of input/output devices and volumes to job steps. Requests for device allocation come from data definition (DD) statements and dynamic device allocation requests. Data definition (DD) statements can be entered into the system by:

- Job input to the JES reader
- Jobs submitted through the TSO **SUBMIT** command
- Started tasks
- The **MOUNT** command
- TSO logons

APPC transactions and dynamic device allocation or unallocation requests, in contrast, originate within executing programs. While performing device allocations, the system might ask you to:

- Mount or dismount volumes
- Make decisions (for example, to bring a device online immediately or to wait)

To control the amount of work you have to do related to device allocation, you might want to restrict device allocation requests. To control device allocation requests from data definition (DD) statements, you might restrict each of the forms of input for these statements (for example, by holding the reader, or by setting a maximum LOGON count). Because they originate within executing programs, however, you cannot control dynamic device allocation/unallocation requests.

18.1.3 Device assignment

Operationally, the assignment of devices is influenced by:

- The online or offline status of the device

  Generally, to be allocated to job steps, devices must be online. The following are exceptions:

  - When the online test executive program (OLTEP) or a similar testing program is running.
  - When teleprocessing devices are allocated.

  You can bring offline devices online with the **VARY** command or in response to any allocation recovery message, IEF238D.

- The MOUNT attribute

  The MOUNT attribute, which applies only to tape or DASD devices, is influenced by the **MOUNT** and **UNLOAD** system commands, and, during initialization, by entries in the VATLSTxx parmlib member. Allocation requests that can be satisfied by mounted devices are processed quickly and without intervention.

- The USE attribute

  A parameter of the **MOUNT** command, the USE attribute affects the type of data sets that can be allocated on a tape or DASD volume. The USE attribute can also be set during initialization by entries in the VATLSTxx member of parmlib. Having a proper mix of volumes with various USE attributes reduces the amount of volume mounting.

In Figure 18-1 on page 406, on the right, observe that the **VARY** process gets serialized and processes each requested device. The ENQ serialization, obtained with exclusive control, prevents allocation from getting the ENQ with shared control. Any job that needs to allocate a device must wait for the SYSIEFSD.Q4/SYSIEFSD.CHNGDEVS ENQs to be released.

Important: While the **VARY** command executes, any job that needs to allocate a device must wait for the SYSIEFSD.Q4 and SYSIEFSD.CHNGDEVS ENQs to be released. This can cause many problems.

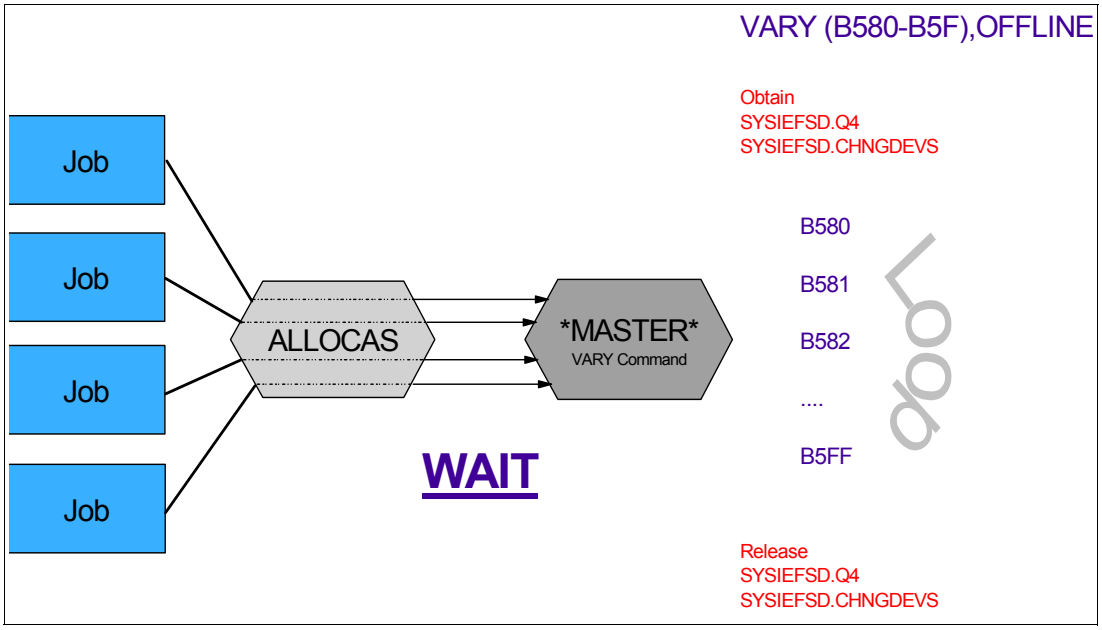


Figure 18-1 The process prior z/OS V1R7

18.2 Enhancements in z/OS V1R7

With z/OS V1R7, an enhancement makes all **VARY OFFLINE** and **UNLOAD** command requests go in a pending state. The **VARY OFFLINE** and **UNLOAD** commands already had an existing process for delaying processing. When an allocated device was the target of a **VARY OFFLINE** or **UNLOAD** command, that command would go in a pending state. That is, it would not be processed until the device was no longer allocated. Later, when another allocation request was processed, allocation would check to see if it could complete any pending processes.

In z/OS V1R7, the enhancement always makes every **VARY OFFLINE** and **UNLOAD** request go into a pending state. This removes the I/O out of the common allocation.

Important: Now, the SYSIEFSD.Q4 and SYSIEFSD.CHNGDEVS resources are obtained exclusively only while marking the devices pending offline or pending unload. No I/O is required for this process. Allocation will not select such a pending device, so there is no requirement to ensure pending devices are offline before allocation processes a request.

18.2.1 Using an additional task in the allocation address space

In this release a new task in the allocation address space (ALLOCAS) will be used to check periodically (when posted, otherwise once per second) if pending OFFLINE or pending UNLOAD devices have become unallocated. This task will obtain the resource SYSIEFSD.VARYDEV to serialize the changes to the device and will not holding the resources SYSIEFSD.Q4 or SYSIEFSD.CHNGDEVS.

18.2.2 Parallelize VARY OFFLINE and UNLOAD commands

As of z/OS V1R7, all the processing in the allocation address space (ALLOCAS) is centralized. Now it is possible to parallelize the processing, as follows:

- One new task continues to monitor pending OFFLINE/UNLOAD devices. It periodically checks the pending OFFLINE/UNLOAD devices, when posted or once per second, while the requests are pending.
- in prior releases there was only one task to process the requests. Now up to 32 tasks are attached to process the requests. The tasks will be attached as needed but they will not be detached when they are no longer needed.
- Serialization is modified, as follows:
  - Each task will obtain the resource SYSIEFSD.VARYDEV shared, to prevent other processes (such as **VARY ONLINE**, recovery allocation **VARY** processing, or ACTIVATEs) from changing the device status.
  - A new resource, SYSIEFSD.VARYDEV_0000*xxxx* (*xxxx* = device number), was created to prevent multiple tasks (which are sharing VARYDEV) from processing the same device.

All other processes have been modified to ensure they follow the same serialization hierarchy. As always, the front-end processing of the **VARY OFFLINE** command will get the necessary resources and mark the devices as pending offline, and create the control block for the back-end process to work against. This queue is shown in Figure 18-2, in the lower left. Unlike prior releases, the pending device processor will not be called in the **VARY** invoker's address space. All calls to this routine will be done from the ALLOCAS address space, centralizing the processing for consistency of processing and better understandability.

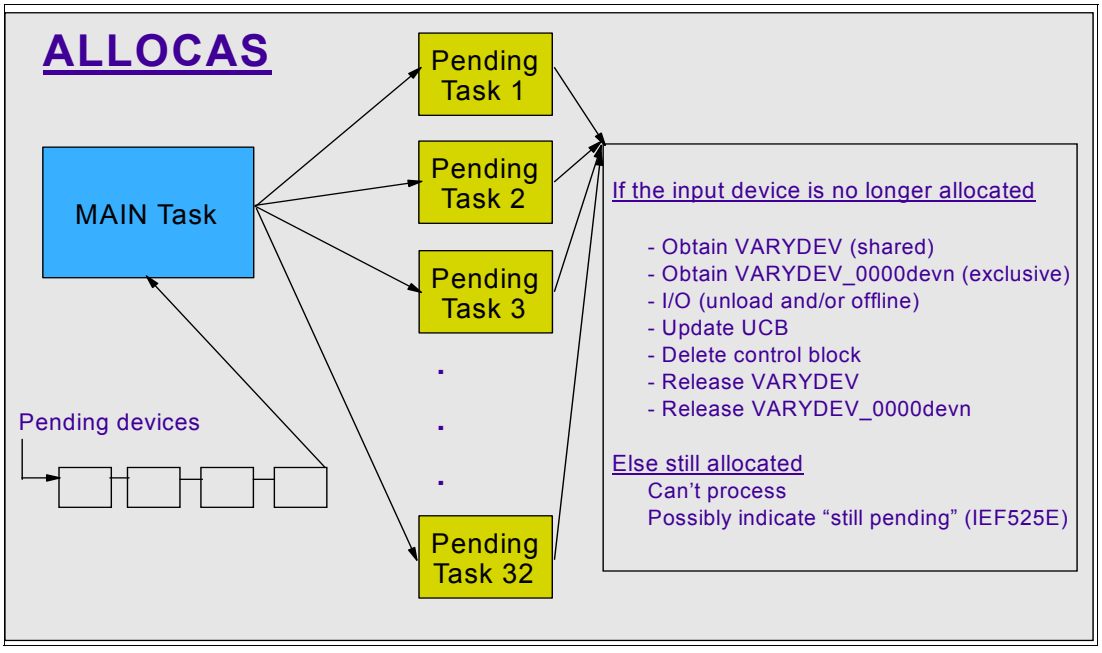


Figure 18-2 Illustration how the requests now will be processed

The backend processing in the allocation address space will consist of one main task which will queue work to the other subtasks. The subtasks will process the requests by obtaining the SYSIEFSD.VARYDEV resource (shared) and a device-specific version of the SYSIEFSD.VARYDEV ENQ, doing the I/O, updating the UCB to make the device offline, and releasing the ENQ. Any request that cannot be immediately processed (for example, it has

become allocated again, or has been marked “in use by system”) will be requeued back to the main task for later retry.

Note: The tasks will not be attached until needed, but once attached, will remain for the life of the system. There are 32 tasks; this value is not tunable.

18.2.3 Updating other processes to ensure same serialization

Recovery allocation (message IEF238D) may drive allocation recursive retry and bring devices online holding SYSIEFSD.VARYDEV instead of holding the resources SYSIEFSD.Q4/SYSIEFSD.CHNGDEVS. The IOS **VARY PATH** and **CF CHP** commands are updated to hold SYSIEFSD.Q4/SYSIEFSD.CHNGDEVS only while marking devices pending OFFLINE or UNLOAD. I/O for these commands now uses SYSIEFSD.VARYDEV, the same as allocation.

The **UNLOAD** command and internal unload module used by IOS are updated to serialize tape devices using the same serialization.

18.3 Changed messages

The following message is deleted:

```
IEE734I dev NOW UNLOADED [ - DEVICE IS BOXED]
```

The message is replaced by the following message:

```
IEF282I dev NOW UNLOADED [ -DEVICE IS BOXED]
```

Message IEF030I replaces the following message when issued because the operator replied with a device that had a volume that did not match the volume on the allocation request:

```
IEF490I INVALID REPLY
```

Message IEF031I is issued in place of the following message when issued due to an unexpected response:

```
IEF490I INVALID REPLY
```

Messages IEF231I and IEF881I are issued by module recovery.

Message IEF414I is issued in place of the following message:

```
IEE314I UNIT NOT AVAILABLE – UNLOAD ATTEMPTED
```

18.4 Migration considerations

In the past, a device marked “pending unload” by an **UNLOAD** operator command would continue to be pending unload until the command completed. With this enhancement, that will no longer be true. A message is issued by the **UNLOAD** command if it finds that the device it was to unload is no longer pending unload. The message ID is IEF415I.

18.4.1 DEALLOC procedure in SYS1.PROCLIB

The **START DEALLOC** command is a procedure in SYS1.PROCLIB that is used to redrive allocation to process pending offline and unload devices. Since the offline and unload

processes are not driven out of mainline allocation any longer, **START DEALLOC** is no longer needed.

18.4.2 Recovery allocation

Recovery allocation is the process that issues message IEF238D, allowing an operator to get a device online for a job that would fail allocation otherwise. Retry allocation is a recursive process that retries a given allocation from the beginning. When an operator replies to IEF238D with a device for allocation to bring online, allocation must now retry its processing from the beginning due to serialization hierarchy requirements. This may result in different messages being issued than in prior releases.

18.4.3 Order of offline messages

Now that devices are processed by many tasks in parallel, the offline messages for a given command (such as **V (5b0-5bF),OFFLINE**) are unlikely to be issued in the order that they were in past releases.

18.5 Device allocation examples

Following are examples to illustrate the new enhancements in z/OS V1R7.

Assume the following conditions:

- A VTS with 128 devices, 64 devices have volumes mounted
- A broken control unit
- Operator needs to **VARY OFFLIN**E to repair, as follows:

```
VARY (B580-B5FF),OFFLINE
```

- An MIH value of 3 minutes is defined for each device.

For these conditions, the following must be done:

- 64 unloads
- 128 offline operations

Table 18-1 displays the conditions if each operation times out.

Table 18-1 Results, if each operation times out

Release	(MIH * no. of unloads) +	(MIH * no. of devices) / no. of tasks	Result
Prior releases	(3 * 64) +	(3 * 128) / 1	576 minutes
z/OS V1R7	(3 * 64) +	(3 * 128) / 32	18 minutes

Note: In prior releases, the result would be 576 minutes. During this time no jobs will run. With z/OS V1R7, the processing would take 18 minutes, and during this time jobs will continue to run.

19

Security Server RACF

This chapter discusses the enhancements and changes to RACF, including:

- Mixed-case password support
- Miscellaneous RACF enhancements
- Nested ACEE support

19.1 Mixed-case password support

With z/OS V1R7, installations can now optionally use mixed-case passwords. This dramatically increases the maximum number of possible passwords and reduces the probability of being able to guess passwords.

Support for mixed-case passwords is enabled by the following RACF command:

```
SETROPTS PASSWORD(MIXEDCASE)
```

Setting this option allows lower-case characters in both commands and macros. By default the setting in effect is NOMIXEDCASE, indicating that all passwords are “folded” to uppercase.

Important: Once set to MIXEDCASE, you should be aware that setting it back to NOMIXEDCASE will mean that anyone that has signed on with a mixed case password while MIXEDCASE was in effect will not be able to log on until their password is reset.

Password rules

The new rules regarding passwords and mixed case are shown in the RACF panel in Figure 19-1. The following new character types are provided for specifying in the password syntax rules:

- NATIONAL - #, $, and @
- MIXEDCONSONANT – Upper and lower case CONSONANT
- MIXEDVOWEL – Upper and lower case VOWEL
- MIXEDNUM – Upper and lower case ALPHANUM

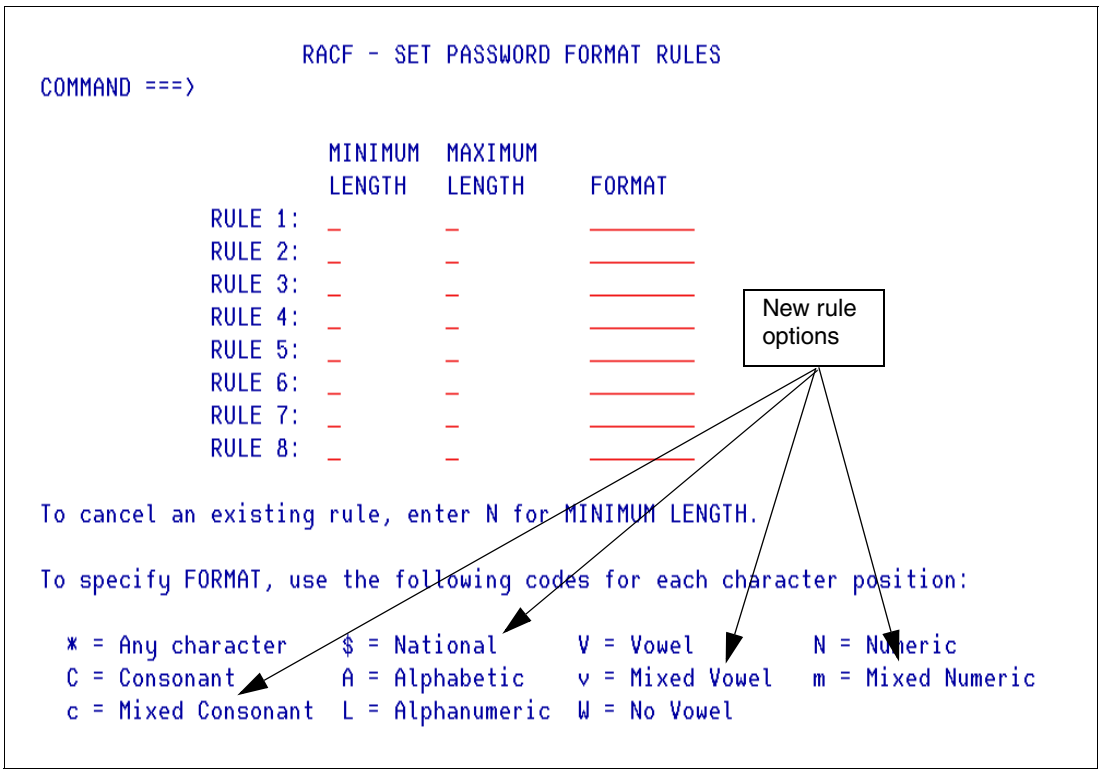


Figure 19-1 New password format rules

19.1.1 Migration and coexistence considerations

Before using **SETROPTS PASSWORD(MIXEDCASE)** you must check that all applications supplying passwords do not automatically fold passwords to uppercase. If you want to allow (or enforce) mixed-case passwords then you may need to alter your password syntax rules.

When sharing a RACF database, all systems using mixed case passwords must be at a level that supports **SETROPTS PASSWORD(MIXEDCASE)**.

The user community must be educated in the use of (or requirement to use) mixed case passwords.

Note: You cannot specify a new password that is different only in the case of the letters. For example you cannot replace NEW0Ne1 with newone1.

19.2 Miscellaneous RACF enhancements

There are several changes in RACF of a smaller nature.

Password logging

The audit class of USER will now log all password changes including successful password changes.

```
SETROPTS AUDIT(USER)
```

Minimum password change interval

The new MINCHANGE parameter of **SETROPTS PASSWORD** command allows an installation to specify a minimum number of days before a user can change their password. This option is set as shown in Figure 19-2 and as follows:

```
SETROPTS PASSWORD(MINCHANGE(nnn))
```

Here, *nnn* is the number of days before a password change is allowed. The default value of MINCHANGE(0) allows users to change their passwords as many times as they want. This change ensures that the password history is valid and will help enforce more secure passwords by preventing users from reusing recent passwords.

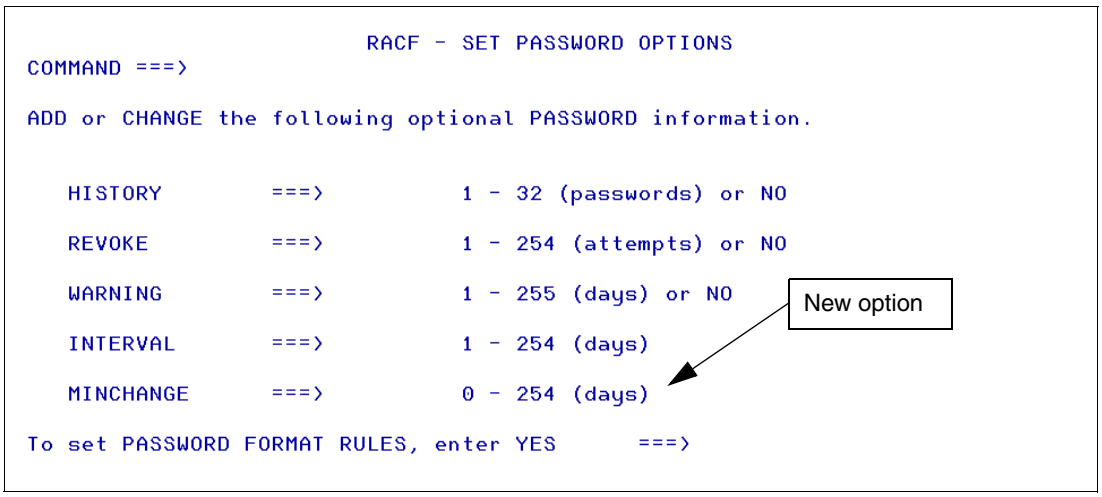


Figure 19-2 New PASSWORD option MINCHANGE

Users who attempt to change the password before the minimum change period has expired will receive a failure message but not the exact reason for the failure.

Note: CONTROL access to IRR.PASSWORD.RESET grants the authority to override the minimum period.

Revoke and resume enhancements

The revoke and resume processing has been enhanced so that when a user or connection is resumed or revoked, the RESUME and REVOKE dates will remain in the profile.

The RESUME and REVOKE dates can be cleared by specifying NOREVOKE or NORESUME on an **ALTUSER** or **CONNECT** command.

Inactivity checking to start at user creation

The **ADDUSER** command now stores the userid creation date; this is used for inactivity checking.

When you define a new userid, the userid's last access date is set to the creation date. If the userid is not used within the number of days specified by **SETROPTS INACTIVE**, the user ID will be revoked. When you issue the **LISTUSER** command for a new user ID that has never been used, the last access date will be listed as UNKNOWN.

19.3 Nested ACEE support

Some applications and daemons initiate requests that require access to protected resources to which the client who invoked the daemon may not otherwise need access. For example, if you are using TLS (encryption) with FTP, then the FTP client end-user will need access to controlled ICSF resources.

The FTP client address space, running trusted code, needs access to protected ICSF keys and services, as follows:

- The client is running under the identity of the end-user, not the FTP server.
- The client user ID must be permitted to access protected ICSF resources.
- The client user ID could then access those resources in a context other than as an FTP client.

The solution to this problem in z/OS V1R7 is to introduce the concept of a nested ACEE and a new type of resource called a *delegated resource*.

In the FTP example, shown in Figure 19-3, the ACEE of the FTP server is nested under the ACEE of the FTP client. The nested ACEE is used for authorization checks when the resource being checked is defined as a delegated resource.

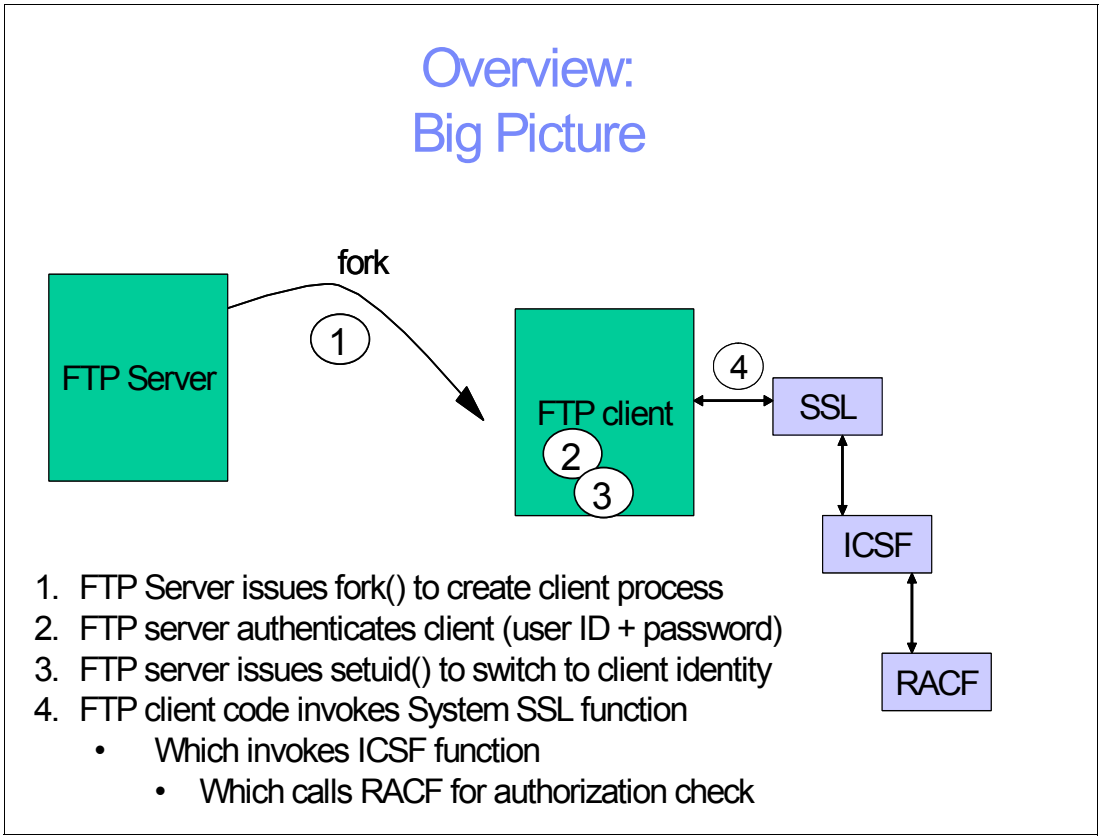


Figure 19-3 FTP client executes under authority of the end-user

Delegated resources

Delegated resources are general resources that are eligible to be accessed by specially programmed applications that request RACF to check the application or daemon's authority for a resource when the client's authority is insufficient. Applications programmed in this way, such as the FTP daemon, are said to contain support for nested ACEEs because the identity of the application or daemon is nested beneath the identity of the client.

A resource is defined to be delegated by adding the text string RACF-DELEGATED to the APPLDATA field of the profile protecting the resource. This string may appear anywhere within the APPLDATA field. Note that this is only applicable to classes that are RACLISTed.

For example:

```
RDEFINE CSFSERV CSFENC APPLDATA('RACF-DELEGATED')
```

A new z/OS UNIX environment variable _BPXK_DAEMON_ATTACH is used to direct that the setuid() family of functions is to preserve the invoking identity by using the new nested ACEE support. In z/OS V1R7, the FTP Server code can exploit the new environment variable and use nested ACEEs.

Usage

Create a nested ACEE, as follows:

```
RACROUTE REQUEST=VERIFY, USERID=CLIENT, PASSWRD=, … , NESTED=YES
```

For a FASTAUTH check, specify as follows:

```
RACROUTE REQUEST=FASTAUTH, ACEE=....
```

If the client (or *primary*) identity fails the authorization check, FASTAUTH will re-drive the authorization check using the embedded identity (the nested ACEE) if the following conditions are met:

- The caller is supervisor state or system key
- The resource is defined as delegated

19.4 RACF R_admin callable service enhancements

R_admin (IRRSEQ00) is a RACF callable service that provides an interface with which to manage and retrieve RACF profile and SETROPTS data.

To use the service applications, create a parameter list, and call the R_admin service, R_admin builds a RACF command image and sends it to the RACF address space for processing. The output from the command is returned to the caller.

In order to retrieve information from RACF using the original implementation, applications have to parse the returned output from list commands such as **LISTUSER**. This output was limited to a maximum of 4096 lines of output. In addition, since the commands are processed by the RACF address space, it is possible to overload the ARCF address space with too many requests.

In z/OS V1R7 new functionality has been added to the R_admin callable service to allow callers to extract User, Group, and Connect information from the RACF database in a tokenized form that is more easily manipulated by a program.The new R_admin extract function runs in the caller's address space, and is not sent to the RACF address space for processing. This will improve performance, and eliminates the problem of overloading the RACF address space with requests. Also, there is no limit to the amount of returned data so that all data in the profile is returned.

The new extract function is limited to User, Group, and Connect information.

20

IBM ported tools: OpenSSH enhancements

The OpenSSH program product can be installed on z/OS V1R4 and later. This chapter describes the changes for version 3.8.1p1 and covers the following topics:

- OpenSSH overview
- New OpenSSH functions
- New keywords
- Migration and installation

20.1 OpenSSH overview

OpenSSH is a suite of network connectivity tools that provide secure encrypted communications between two untrusted hosts over an unsecured network. Some of the utilities that it includes are:

`ssh` z/OS client program for logging into a z/OS shell. It can also be used to log into other platform's UNIX shells; it is an alternative to `rlogin`.

`scp` Used to copy files between networks, it is an alternative to `rcp`.

`sftp` Provides file transfers over an encrypted ssh transport, it is an interactive file transfer program similar to `ftp`.

`sshd` A daemon program for ssh that listens for connections from clients. It supports both SSH protocol versions 1 and 2 simultaneously.

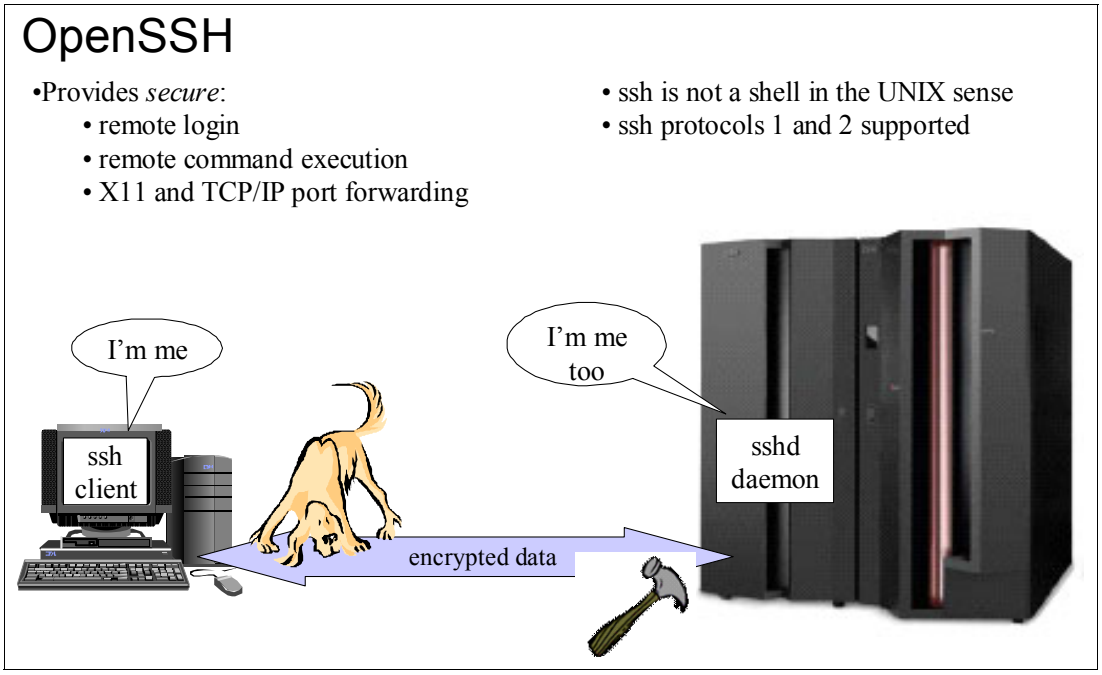


Figure 20-1 OpenSSH overview

20.2 New OpenSSH functions

The following is new for OpenSSH 3.8.1p1:

- Multilevel security support

  The OpenSSH daemon supports assigning a security label (SECLABEL) to a user based on the user's port of entry.

- Password reset capability

  If a user's password expires while attempting login, it can now be reset using OpenSSH.

- Daemon restart capability

  The OpenSSH daemon is now tolerant of TCP/IP stack changes, as follows:

  – If TCP/IP is recycled, sshd will not exit, but will wait and then re-initialize when TCP/IP returns.

– If sshd is started from /etc/rc but TCP/IP has not been started yet, sshd will wait for TCP/IP to come up.
– In a common INET (CINET) environment, a new stack will automatically be recognized by the daemon. Sending a SIGHUP signal to sshd to recognize the new stack is no longer required.

20.3 New keywords

For OpenSSH 3.8.1p1, certain configuration keywords were changed. The keywords that were used in OpenSSH 3.5p1 are still supported on IBM z/OS, but not by the OpenSSH base distribution. To start using the new keywords, all systems that share a configuration must be in version 3.8.1p1.

Table 20-1 OpenSSH keywords changed

File	OpenSSH 3.5p1	OpenSSH 3.8.1p1
daemon (sshd_config)	KeepAlive	TCPKeepAlive
	VerifyReverseMapping	UseDNS
client (ssh_config)	KeepAlive	TCPKeepAlive

New ssh_config keywords are the following:

`AddressFamily` Specifies which address family to use when connecting. Valid arguments are `any`, `inet` (for IPv4 only) or `inet6` (for IPv6 only).

`ConnectTimeout` Specifies the time out (in seconds) used when connecting to the ssh server, instead of using the default system's TCP time out. This value is used only when the target is down or is unreachable, not when it refuses the connection.

`EnableSSHKeysign` Setting this option to `yes` in the global client configuration file /etc/ssh/ssh_config enables the use of the helper program ssh-keysign during HostbasedAuthentication. The argument must be `yes` or `no`. The default is `no`.

`ForwardX11Trusted` If this option is set to `yes`, then remote X11 clients will have full access to the original X11 display. If this option is set to `no`, then remote X11 clients are considered untrusted and will be prevented from stealing or tampering with data belonging to trusted X11 clients. The default is `no`.

`IdentitiesOnly` Specifies that ssh should only use the authentication identity files configured in the ssh_config files, even if the ssh-agent offers more identities. The argument to this keyword must be `yes` or `no`. The default is `no`.

`ServerAliveInterval` Sets a time out interval in seconds after which, if no data has been received from the server, ssh sends a message through the encrypted channel to request a response from the server. The default is `0`, indicating that these messages are not sent to the server. Restriction: This option applies to protocol version 2 only.

`ServerAliveCountMax` Sets the number of server alive messages that can be sent without ssh receiving any messages back from the server. If this threshold is reached while server alive messages are being sent, ssh disconnects from the server, thus ending the session. The default value is `3`.

`TCPKeepAlive` Specifies whether the system should send TCP keepalive messages to the other side. If they are sent, a lost network connection or stopping of one of the machines will be properly noticed. However, this means that OpenSSH connections will end if the route is down temporarily. The default is `yes` (to send TCP keepalive messages), and the client will notice if the network goes down or the remote host dies. This is important in scripts as well as to many users.

`VerifyHostKeyDNSNew` Specifies whether to verify the remote key using DNS and SSHFP (SSH fingerprint) resource records. If this option is set to `yes`, the client will implicitly trust keys that match a secure fingerprint from DNS. Insecure fingerprints will be handled as if this option was set to `ask`. If this option is set to `ask`, information on fingerprint match is displayed, but the user will still need to confirm new host keys according to the StrictHostKeyChecking option. The argument must be `yes`, `no` or `ask`. The default is `no`. This option applies to protocol version 2 only.

New sshd_config keywords:

`TCPKeepAlive` Specifies whether the system should send TCP keepalive messages to the other side. If they are sent, a lost network connection or stopping of one of the machines will be properly noticed. However, this means that connections will die if the route is down temporarily, and some people find it annoying. On the other hand, if keepalives are not sent, sessions may hang indefinitely on the server, leaving ghost users and consuming server resources. The default is `yes` (to send TCP keepalive messages), and the server will notice if the network goes down or the client host crashes. This option avoids infinitely hanging sessions. To disable TCP keepalive messages, set the value to `no`.

`UseDNS` Specifies whether sshd should look up the remote host name and check that the resolved host name for the remote IP address maps back to the same IP address. The default is `yes`.

The config files are full described at *IBM Ported Tools for z/OS User's Guide*, SA22-7985.

20.4 Migration and installation

APAR OA10315 upgrades the OpenSSH and OpenSSL functionality to OpenSSH 3.8.1p1 and OpenSSL 0.9.7d.

IBM Ported Tools for z/OS User's Guide, SA22-7985 and *IBM Ported Tools for z/OS Program Directory*, GI11-2847 provide information about prerequisites, installation, and migration for OpenSSH 3.8.1p1.

21

XES locking constraint relief

Cross-system extended services (XES) allow subsystems, system products, and authorized applications running in a sysplex to use a coupling facility for high performance, high availability data sharing.

In this chapter we introduce the changes made in z/OS V1R7, as follows:

- Data sharing concepts
- Lock structures
- Migration considerations

21.1 Data sharing concepts

Data sharing in a sysplex gives applications the opportunity to access directly and change the same data by using a coupling facility.As illustrated in Figure 21-1, the user can store and access data in the coupling facility in any of the following three types of structures:

List structure Enables users to share information organized as entries on a set of lists or queues. Connections could use a list structure, for example, to distribute work or maintain shared status information. List structure services are accessed through the IXLLIST, IXLLSTC, IXLLSTE, and IXLLSTM macros.

Cache structure Allows high performance sharing of frequently referenced data. Cache structure services are accessed through the IXLCACHE macro.

Lock structure Allows users to create a customized set of locks and locking protocols for serializing user-defined resources, including list or cache structure data. You can implement a serialization mechanism with any scope you require, thereby reducing contention for resources. For instance, rather than serializing at a data set level, you can use the lock structure to serialize access at the record or field level. Lock structure services are accessed through the IXLLOCK macro.

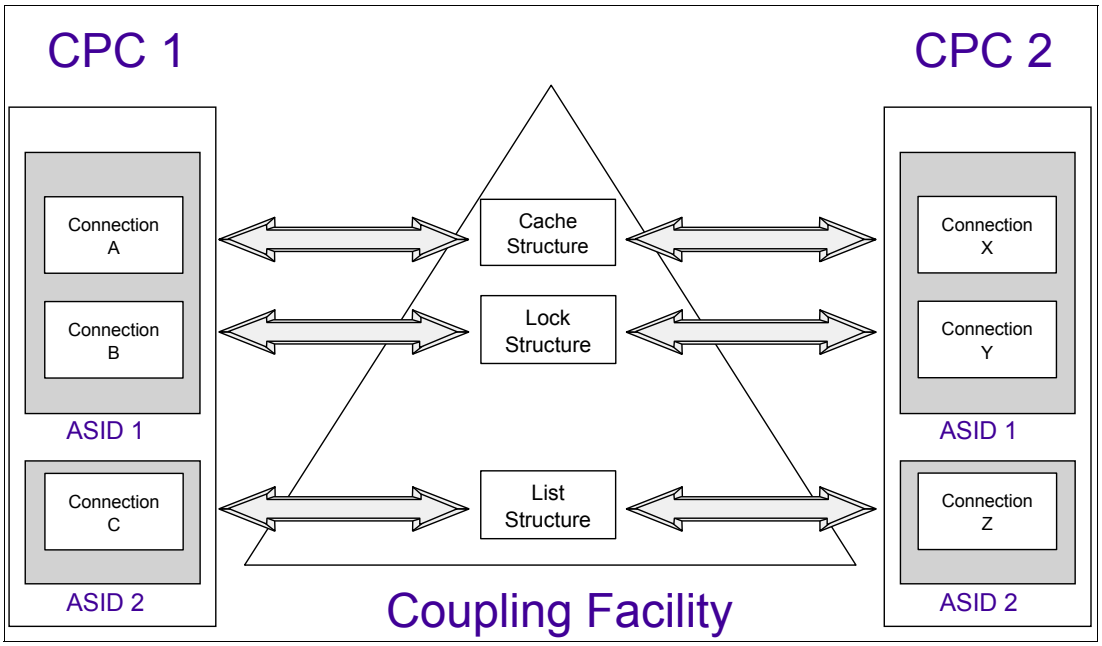


Figure 21-1 Example of how structures could be connected

21.2 Lock structures

To use a lock structure, authorized applications need to connect to the structure by invoking the IXLCONN macro. After a successful connection the IXLLOCK macro enables obtaining shared or exclusive serialization on user-defined resources. The IXLLOCK macro supports the following requests:

`OBTAIN` Obtain either a shared or exclusive ownership of a resource.

`ALTER` Alter the attributes of an owned resource.

RELEASE	Release the ownership of an owned resource.
PROCESSMULT	Process multiple requests with one single IXLLOCK invocation.

These requests cause data space storage used to manage the resources to be used. Some locking information is stored in different control blocks which reside in data spaces. The current limitation of concurrently held or requested locks lies with the size (2 GB) of the data space used to manage local resources. For more information about using lock services, see *z/OS MVS Programming Sysplex Services Reference*, SA22-7618 and *z/OS MVS Programming Sysplex Services Guide*, SA22-7617.

21.3 Enhancements in z/OS V1R7

In prior releases the number of concurrently held or requested locks (IXLLOCK requests) was limited due to the size of only one local data space (2 Gigabyte). z/OS V1R7 addresses this constraint by allocating more additional local data spaces.

21.3.1 Data space limit removal

In z/OS V1R7, an allocation of 16 additional local data spaces is done whenever a connection (IXLCONN) to a lock structure is made. This reduces the likelihood of encountering out of data space conditions. As of this release the following data spaces exist:

- 1 base local data space
- 1 global data space
- 16 additional local data spaces

The new data spaces have the same characteristics as the existing data spaces. The naming convention is xxxxxIXL. The hash value specified on IXLLOCK requests is now also used to determine which local data space is used while processing the request.

Once the connection (IXLCONN) to the lock structure is established the 16 new local data spaces are created.

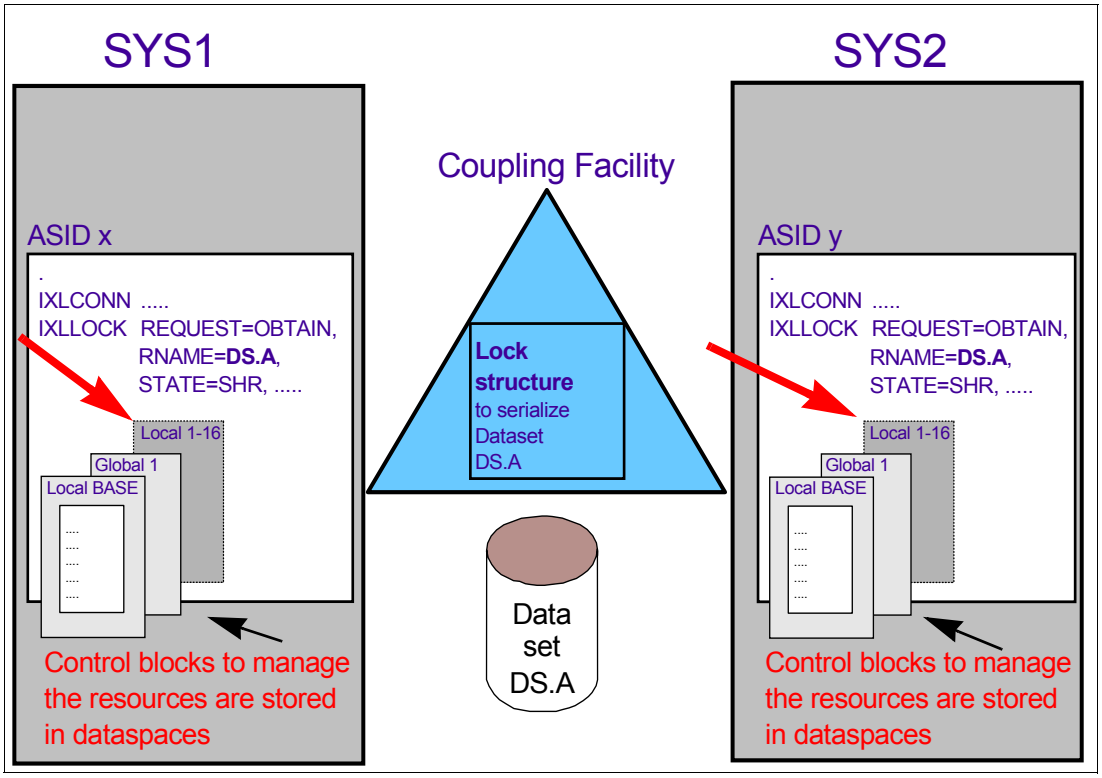


Figure 21-2 Overview of the data spaces

21.3.2 Monitoring of XES data space storage

z/OS V1R7 provides new internal pre-established thresholds which allow monitoring of data space usage. The application can activate the monitoring by specifying the new keyword MONITORSTORAGE=1 in the IXLCONN macro. The default is MONITORSTORAGE=0. When the amount of in-use storage reaches the threshold, incoming requests will be rejected with a new return code and a new reason code. This gives the user an additional opportunity to react to data space storage situations (ABEND x'026').

The new return code is IXLRETCODEENVERROR and the new reason code is ISLRSNCODERESOURCESCONSTRAINED. However, the MONITORSTORAGE keyword can be overridden with a new keyword in the IXLLOCK macro (CRITICALREQUEST).

At run time, applications can use the IXCQUERY REQUEST=FEATURES request to determine if support for monitoring of data space storage is activated.

Table 21-1 Relationship between IXLCONN and IXLLOCK settings

IXLCONN MONITORSTORAGE=	IXLLOCK CRITICALREQUEST=	Monitoring of data space usage activated?
1	1	NO
1	0	YES
0	0	NO
0	1	NO

Note: The CRITICALREQUEST option only has meaning when MONITORSTORAGE=1 is set.

For more information about these new keywords see *z/OS MVS Programming Sysplex Services Reference*, SA22-7618.

21.3.3 Reclaim unused storage faster

The process to reclaim unused data space storage is timer-driven and occurs normally once per minute as long as there is storage to be reclaimed. With assistance of the new pre-established thresholds this process occurs more frequently in an aggressive threshold situation (about once every 15 seconds) to reclaim unused storage faster. Once enough storage has been reclaimed to reach a low threshold, the process occurs at the normal interval (once per minute).

21.4 Migration and coexistence

For coexistence considerations with other releases see APARs OA01511, OA03194 and for IRLM 2.1 see APAR PQ83320.

22

HCD/HCM enhancements

The new and enhanced functions in HCD/HCM, which is shipped with z/OS V1R7, are described in this chapter. In addition, the migration steps to IODF version 5 are also described. The following enhancements are covered:

- Local IOCDS download
- Enhanced CHPID aggregate
- Count of filtered list elements
- Enhanced OS group change
- Improved PFSHOW handling
- IODF list sort function
- Unused device number prompt
- Point-to-point CTC connection report
- Automated IODF check
- CSS/OS Compare report enhancements
- HCM check configuration file utility
- HCM general box
- View HCD report from HCM

22.1 Local IOCDS download

If a processor definition has an SNA address (network name, CPC name) specified, HCD always performs a remote IOCDS download regardless of whether the IOCDS download is for the local processor or not. If the HMC is currently not available, the remote IOCDS download fails. In such a case, an IOCP input data set has to be produced from HCD, and the IOCP program has to be directly invoked in order to write the IOCDS to the local CPC.

HCD now provides an option to perform a local IOCDS download to the local CPC even if the processor has an SNA address defined and the HMC function is currently not available. The modified BUILD IOCDS dialog screen is shown in Figure 22-1.

This panel includes the new option Remote Write. This option is initialized with a value of `Yes` when an SNA address is defined to the selected processor. This setting can be changed to `No` for a local IOCDS write. Then, the IOCDS download is performed locally, as if the processor did not have an SNA address defined.

If an SNA address is not available in the processor definition, the panel displays `Remote Write = No`; this setting cannot be changed.

The syntax of the batch invocation of a Build IOCDS task is correspondingly enhanced by the new parameter LOCALWRT. Specifying this parameter forces a local write independent of whether an SNA address has been defined or not.

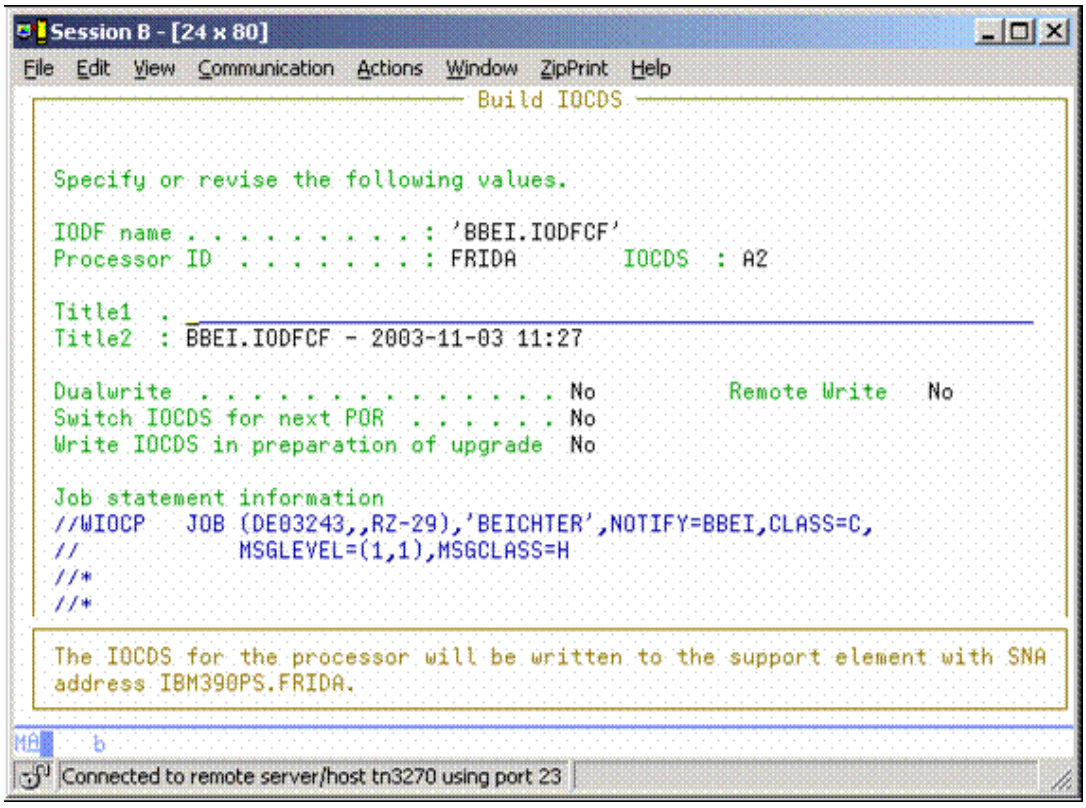


Figure 22-1 The modified Build IOCDS screen

22.2 Enhanced CHPID aggregate

The existing aggregate CHPID action copies all the I/O attachment (control units and devices) of a source CHPID to a target CHPID, provided that the following prerequisites are fulfilled:

- Source and target CHPIDs are defined to the same CSS and are different.
- Source and target CHPIDs are compatible regarding their control unit connectivity. All control unit types connectable to the source CHPID must be connectable to the target CHPID and vice versa.
- The source CHPID must not be connected to a control unit that is already connected to the target CHPID. Also, a link address/unit address/CUADD combination used by a control unit connected to the source CHPID must not be used by a control unit connected to the target CHPID.
- Either the source CHPID has the same CHPID mode as the target CHPID or all devices accessible by the source CHPID must be connected to only one CHPID.
- Source and target CHPIDs must have the same dynamic switch defined. A dynamic switch is required.
- Source and target CHPIDs must be connected to the same entry switch or both must have no entry switch defined.
- The user must not lose connectivity by a CHPID aggregate action. The source CHPID access and candidate lists must be equal to or a subset of the access and candidate lists of the target CHPID.
- Connecting all control units and devices from the source CHPID to the target CHPID will not cause any defined limit for the target CHPID (for example, maximum number of unit addresses) to be exceeded.

These conditions are very restrictive and do not support tasks like moving a subset of control unit attachments from one CHPID to another one, for example; moving all DASD attachments from an ESCON CHPID to a FICON CHPID, and leaving the tape attachments at the source CHPID.

This function is now enhanced such that the user can select the control units to be moved from a source CHPID to a target CHPID. The source CHPID also may be connected to a different switch and even switch architecture than the source CHPID.

22.2.1 Select Control Units to be Aggregated panel

When the HCD user performs an aggregate CHPID action, the new panel Select Control Units to be Aggregated is displayed, as shown in Figure 22-2 on page 430.

This panel lists all control units that are currently attached to the source CHPID. Each control unit shows the source switch port to which it is connected if this is known to HCD, along with the target switch port and the target link address after the aggregate if they can be determined by HCD.

The user can select a subset (or all) of the control units to be aggregated to the target CHPID. Then only the selected control units and their attached I/O devices are disconnected from the source CHPID and connected to the target CHPID. The target CHPID may be connected to a different switch than the source CHPID. The control unit selection panel allows changing the control unit port and link address for the move to the target CHPID.

If no selection is done, no aggregation is performed.

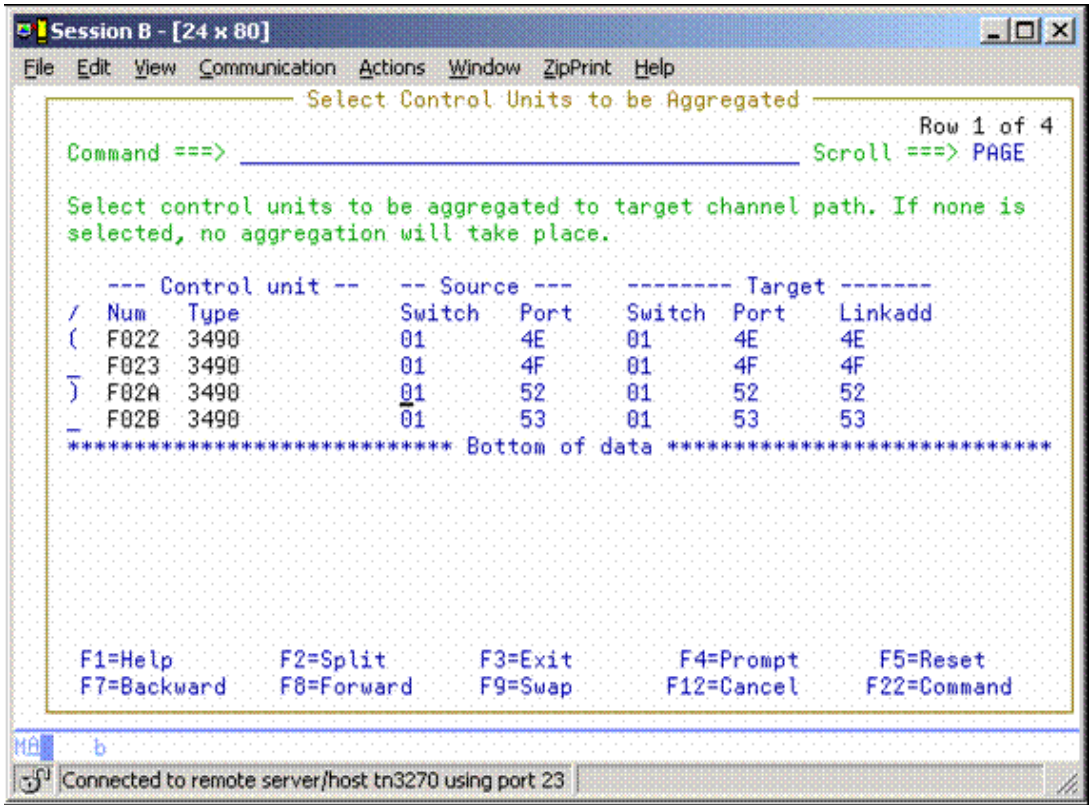


Figure 22-2 Sample enhanced CHPID aggregate display

22.3 Count of filtered list elements

A filtered list in HCD does not show the number of displayed rows. This makes it difficult for the HCD user to get an overview of the number of displayed objects.

The reasons for not showing the number of filtered objects in the upper part of a list panel are the following:

- For the complete list, HCD keeps counters in the IODF. For a filtered list, these counts have to be determined dynamically. With an update action, the number of objects in the filtered list may change, meaning the count would have to be determined after each change.
- For good performance, HCD lists a panel as soon as it is filled up before processing all objects according to the filter criteria. To get the total count, all objects would have to be processed before a list panel could be displayed.

HCD now provides the capability to determine the total count of rows in a filtered list upon user request. Figure 22-3 on page 431 provides an example.

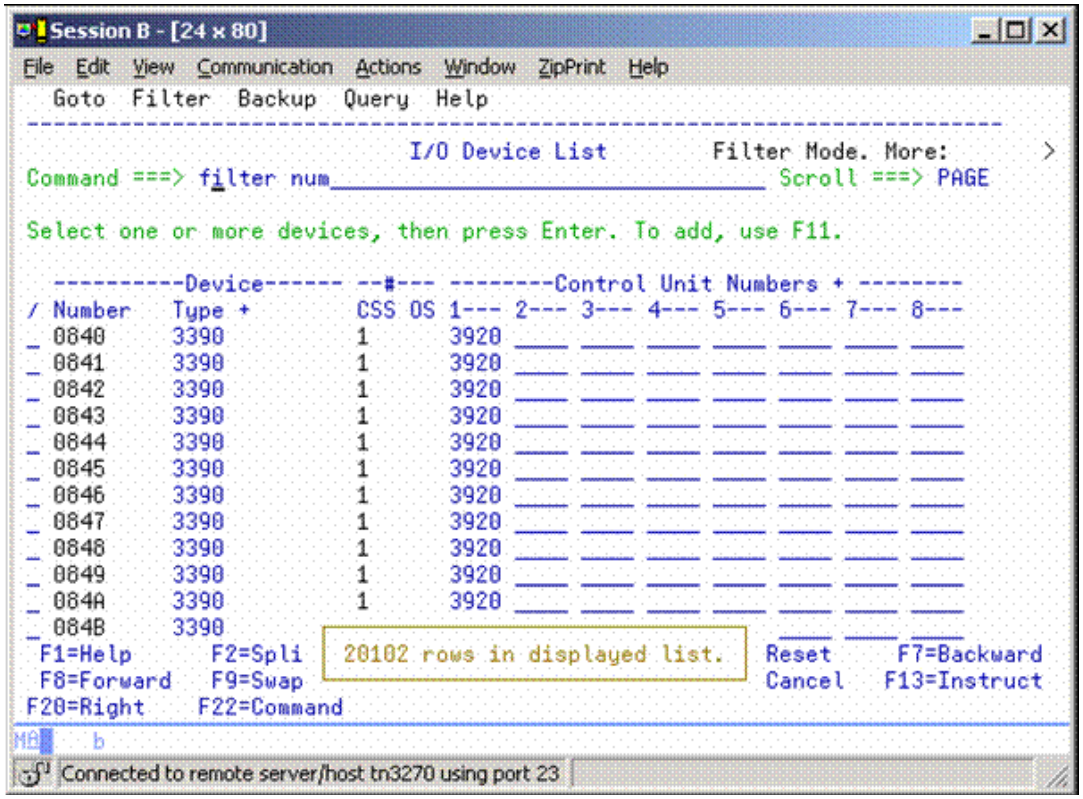


Figure 22-3 sample screen for filter option

22.4 Enhanced OS group change

When an OS group change is performed on a set of selected devices, all devices are updated with the OS parameters that are displayed on the Define Device Group Parameters/Features panel. The listed attributes show the settings from the first selected device in the group. There is no indication whether all selected devices had the same settings. Therefore, there is some risk of making attribute settings changes that are not intended.

HCD now warns the user if all selected devices for the OS group change do not have the same attribute settings. In such a cause, warning message CBDA374I is given.

22.5 Improved PFSHOW handling

HCD did not retain the PFSHOW setting of the user that was in effect outside of HCD. After exiting the HCD dialog, the PFSHOW setting of the HCD task remained active.

To support an application-specific PFSHOW setting, the HCD behavior is changed as follows:

- When HCD is invoked for the very first time, PFSHOW is set to ON.
- The PFSHOW setting used during the HCD session is saved in the ISPF profile pool before leaving HCD.
- HCD is always invoked with the saved setting of PFSHOW.
- On entry of HCD, the actual setting of PFSHOW is saved. Upon exit of HCD, it is restored to its original value.

22.6 IODF list sort function

When prompting for an IODF, navigation on the resulting IODF list to a specific IODF may be cumbersome if there is a large list of IODFs.

To facilitate navigation to a specific IODF on the Available IODFs panel a sort function is now provided, allowing the user to specify sorting by IODF name, size, creation date. As illustrated in Figure 22-4, the following sort options appear in the status part of the screen:

- F14 sorts the displayed IODF list according to IODF name, in increasing alphabetical order (default).
- F15 sorts the displayed IODF list according to IODF size, in decreasing numbers of allocated size.
- F16 sorts the displayed IODF list according to decreasing IODF creation dates.

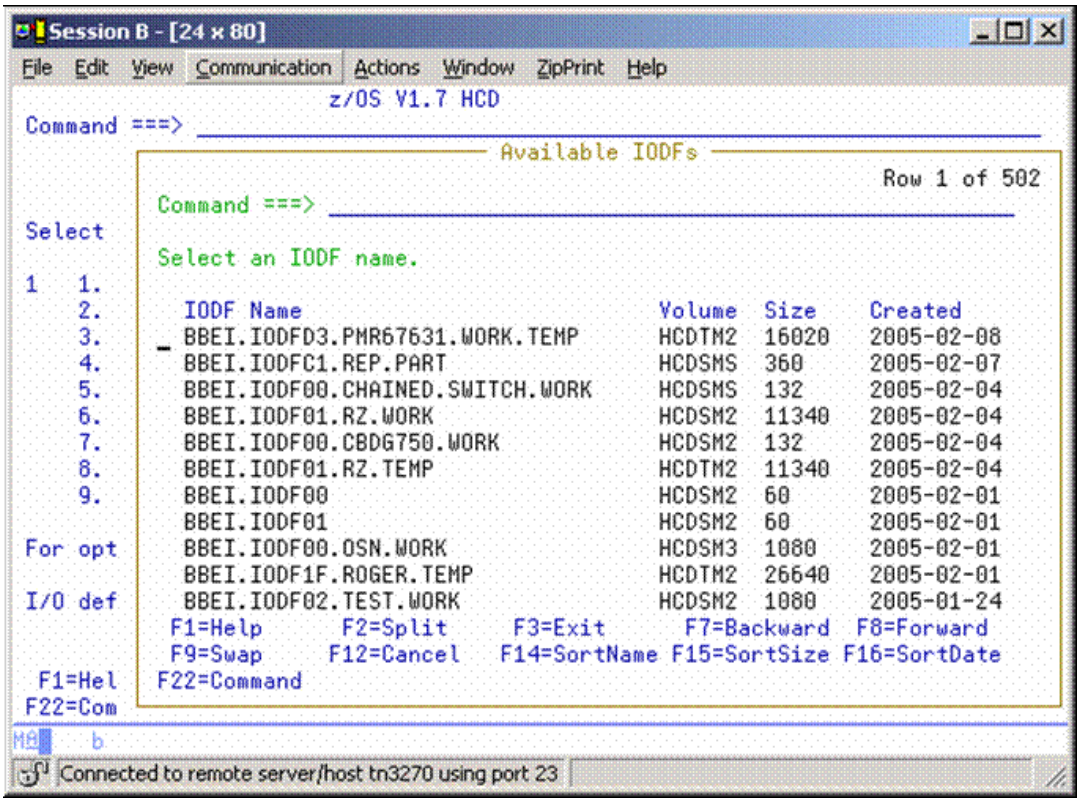


Figure 22-4 sample IODF selection panel

22.7 Unused device number prompt

In order to avoid duplicate device numbers, a list of unused device numbers would ease the task of selecting new device numbers.

HCD has been changed to provide a prompt on the Add Device panel to show the device ranges that have not been used yet in the current IODF. Thus, the user can easily select a new device number (range).

All device numbers which are still unused in the IODF are displayed as ranges on the new panel Unused Device Number Ranges, as shown in Figure 22-5 on page 433. Free device

numbers can be identified and then entered on the Device number/ Number of devices fields of the Add Device panel either manually or via selection from the prompt panel. In the latter case, the selected device number and range are saved in the entry fields. (Device ranges of more than 4 digits are not saved.). An example is shown in Figure 22-6 on page 434.

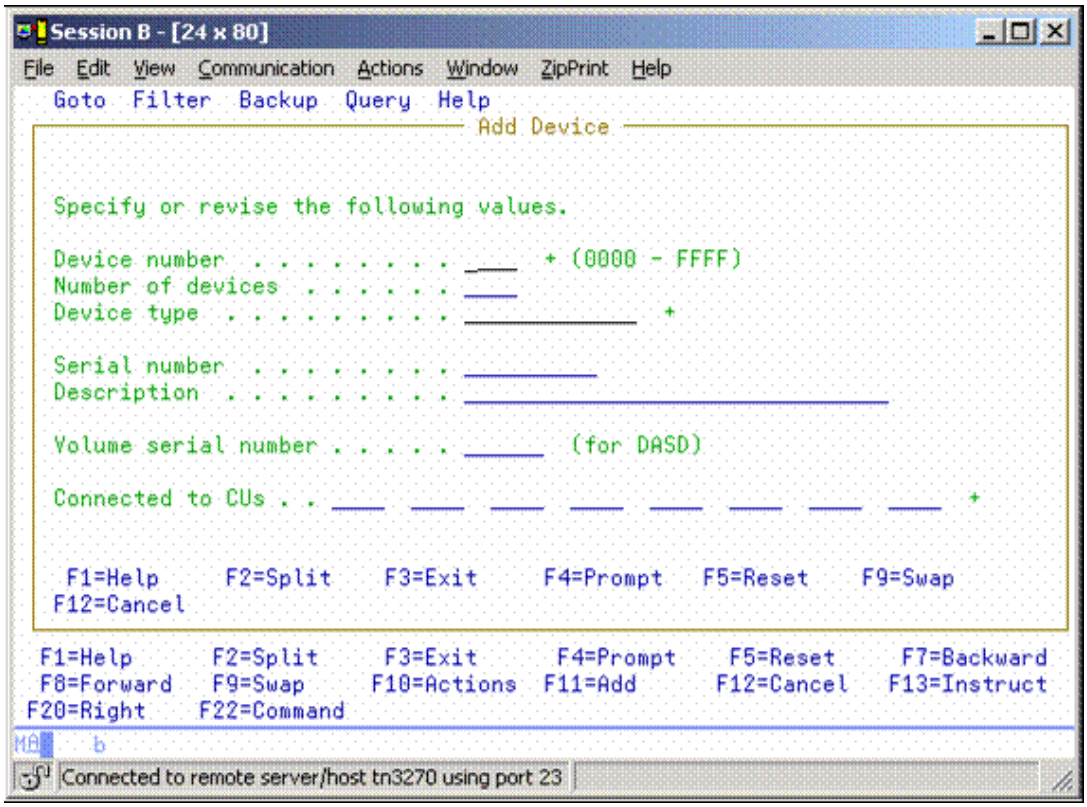


Figure 22-5 Sample ADD DEVICE screen

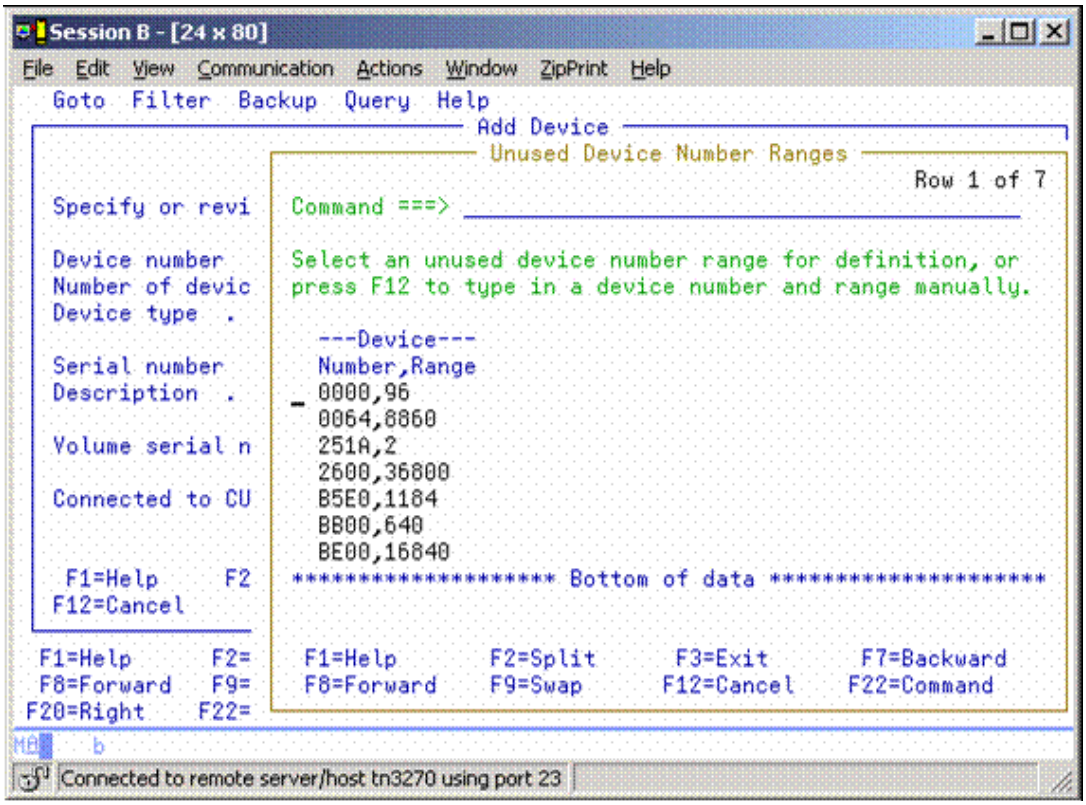


Figure 22-6 Unused Device Number Ranges screen

22.8 Point-to-point CTC connection report

The CTC Connection Report in HCD validates only CTC connections that go through a dynamic switch. For point-to-point CTC connections, HCD is not able to determine the two sides of the CTC connection. This forces customers to either omit the HCD CTC validation, or to temporarily define the CTC connection via a dynamic switch and remove the switch definitions after the CTC validation shows no error. To improve this situation, HCD now supports point-to-point ESCON or FICON CTC connections in its CTC Connection List. In order to allow HCD to determine the communicating channels, all the control units that are involved in a point-to-point CTC connection of two CHPIDs have to specify the same unique serial number.

HCD uses the CTC control units that specify the same serial numbers to match corresponding CHPIDs of a CTC connection. By doing this, the CTC Connection List/report is now able to show validated CTC point-to-point connections. If more than two point-to-point connected CHPIDs use control units with the same serial numbers (ambiguous definition), message CBDG758I is displayed:

```
CBDG758I HCD cannot determine unique point-to-point connection. CHPID xx of processor
nnnn is connected to more than one target CHPID
```

The screen in Figure 22-7 on page 435 shows the validation of the defined CTC connections.

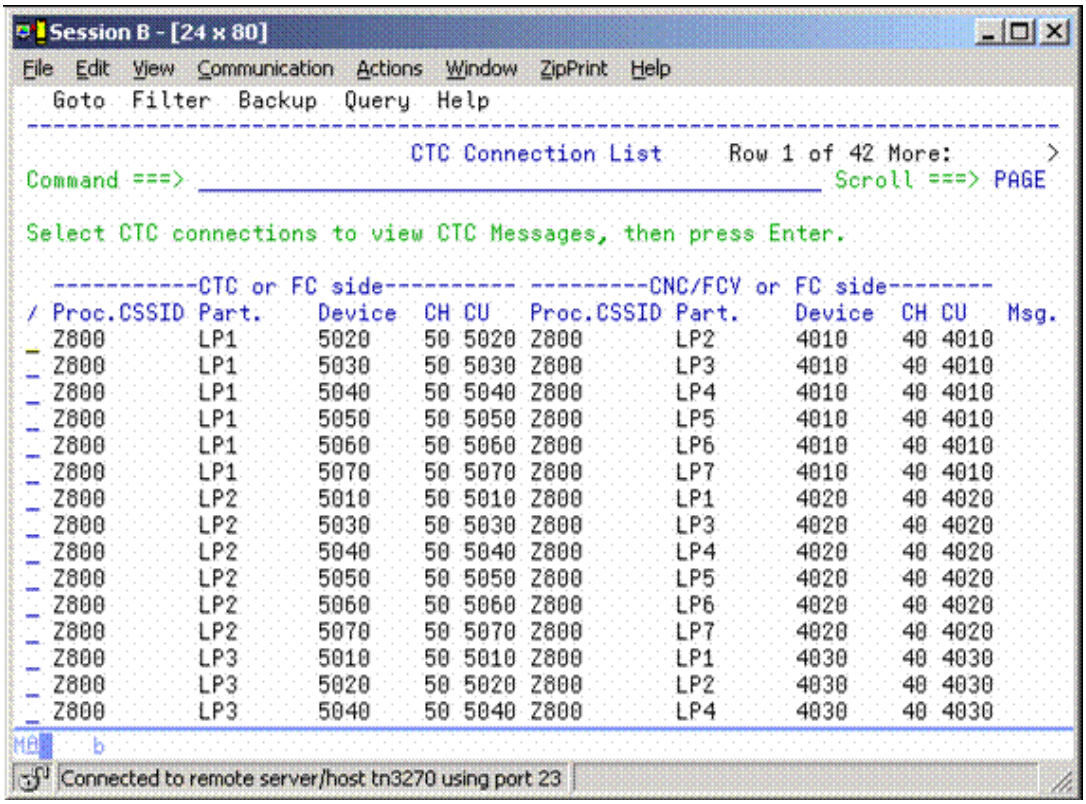


Figure 22-7 sample report

22.9 Automated IODF check

The HCD **TRACE ID=IODF** command provides a means to check an IODF to determine whether it is consistent and its structure is defect-free. This is a manual process. However, often a defect in the IODF may be dormant for a long time until it shows up due to a change in the affected IODF records. By then, the reconstruction of the problem's cause may be very difficult, and sometimes no longer possible. In order to catch IODF defects as early as possible, HCD provides a profile option (CHECK_IODF = YES) that enables an automatic IODF check each time an IODF in update mode is being closed. Depending on the size of the IODF, this additional check extends the response time when closing the IODF (for example, when leaving HCD, or switching IODFs). Using this profile option, (known) defects that are introduced in the IODF are detected as early as possible.

22.10 CSS/OS Compare report enhancements

Up to now, devices that are excluded from a partition via an explicit device candidate list show up in the CSS/OS Compare report, although the devices are absent from the CSS. The report, therefore, had to be used by manually cross-referencing with the explicit device candidate list information from the CSS Device Detail report. The CSS/OS Device Compare report now gets an indication for devices that belong to the limiting partition via CHPIDs but are excluded from the CSS via explicit device candidate lists.

Important: Users no longer need to additionally use the CSS Device Detail report to get this information.

22.11 HCM check configuration file utility

In the past, a corrupted HCM configuration file showed up only when the corrupted data was accessed. This made it hard to reproduce the cause of the defect. In order to improve this situation, an HCM configuration checker function is required. The HCM Check Configuration File utility allows HCM users to check their configuration files for structural consistency. Thus, HCM users can get an early warning if there is a defect in either the IODF or the HCM configuration file that might eventually lead to functional failures. If necessary, the resulting file can be sent to IBM support to help diagnose structural consistency issues.

The HCM menu selection **Utilities/Check Configuration File** opens the Check Configuration File dialog.

The check is limited to the currently open HCM configuration file or the IODF it is associated with. Consistency checking of the IODF will be performed by HCD (this corresponds to the HCD TRACE ID=IODF command), and messages resulting form the IODF check will be displayed in the HCM dialog's output window. You can use the Save As button to save the messages written to the dialog's output window into a file. Additional diagnostic output can be written to a file named eeqccf.txt, in the directory where HCM is installed.

You can select the **Repair** option to attempt a repair of any structural inconsistencies found in the IODF. The Repair option cannot be used when checking a production IODF. Figure 22-8 on page 437 shows sample output of a configuration file check.

Important: If the check was invoked in HCM standalone mode, IODF check is disabled and some HCM configuration file checks are skipped.

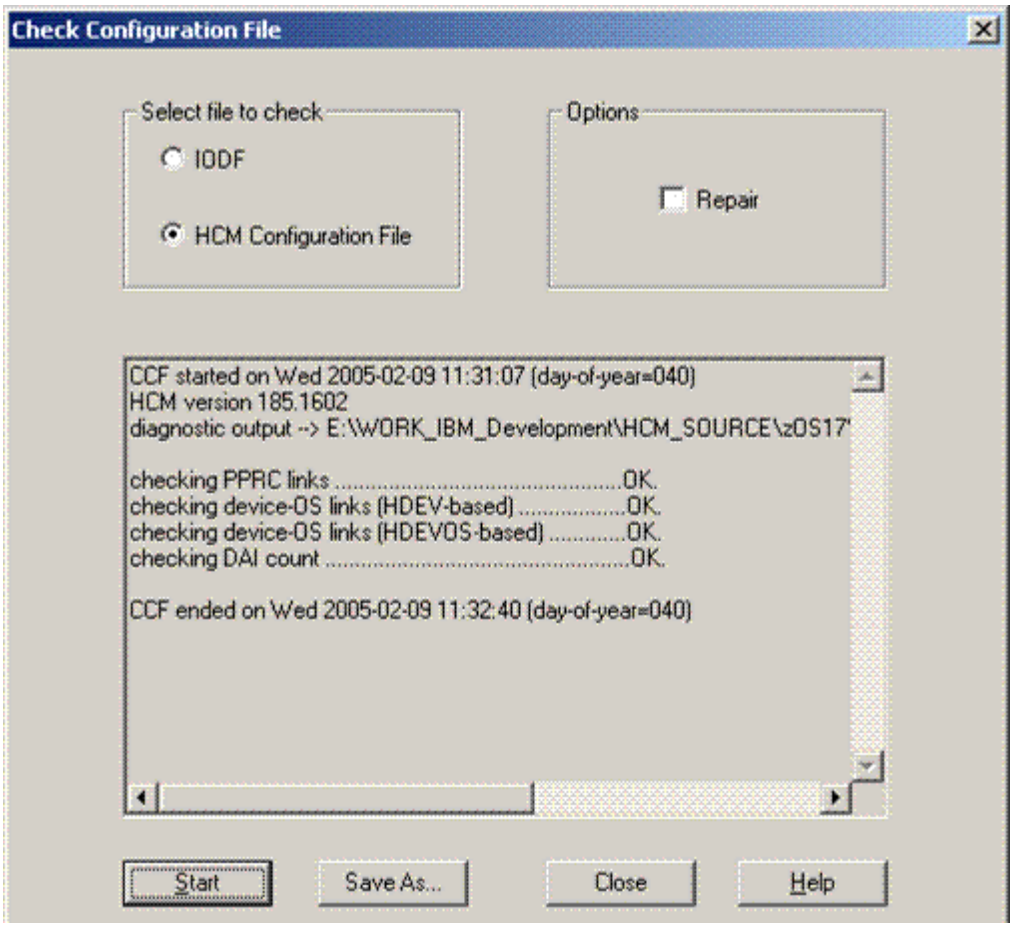


Figure 22-8 Sample output of check configuration file

22.12 HCM general box

The physical objects that can be defined in HCM are related to the logical I/O configuration that needs to be defined for zSeries machines. This excludes the documentation of those physical hardware boxes and their interconnection cables that are not involved in a logical I/O configuration definition like HMC, channel extenders, non-zSeries hardware, and so forth. This is one reason why HCM cannot be used as a general cable management tool. HCM now introduces the concept of a *general box* that can be used for the documentation of any sort of physical objects. The general box is similar to the *cabinet object*, meaning it consists of port definitions that can be used to document (cable) connections. In contrast to a cabinet, there is no restriction on where to define a general box. Thus, customers are now able to document all their physical hardware and cables using general boxes. This positions HCM not only as a zSeries configuration management tool, but as a data-center-wide documentation tool.

A new report functionality is included in the HCM tool beginning with z/OS V1R7. To use the report function you must install the new HCM manager on your workstation.

Defining physical objects in HCM

The new Create General Box dialog shown in Figure 22-9 on page 438 allows you to define a new General Box object. Enter all the required information. The Description and Comments fields are optional, while the fields Name (the ID) and Graphical Position are mandatory fields. Press Enter if you only want to create an empty General Box object.

To create the panels and general box ports as well as the General Box itself, select the checkbox labeled Create Panel, which is located in the Panel section of the dialog. Once this box is selected, ID becomes a mandatory field, and you must specify reasonable values for Rows and Columns as well as for General Box Port Naming Format.

Press the Create and Connect button to create the General Box and the specified panel and general box ports.

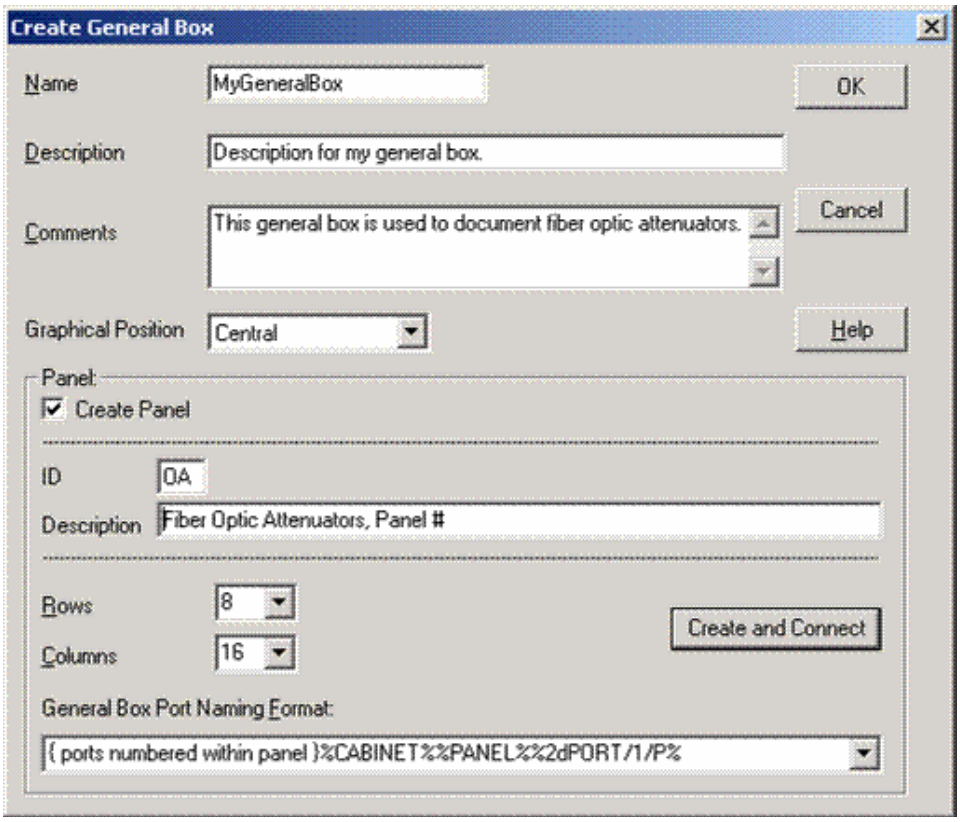


Figure 22-9 Example of creating a general box

General Box dialog

The General Box dialog shown in Figure 22-10 is very similar to the Cabinet dialog. Use this dialog to work with the general box itself, with the panels defined to the general box, and with the general box ports that are defined for the selected panel. As for patch ports, you can change the general box port attributes, connect and disconnect the general box ports, edit a link of which they are part, and you can locate a general box port directly via the General Box dialog.

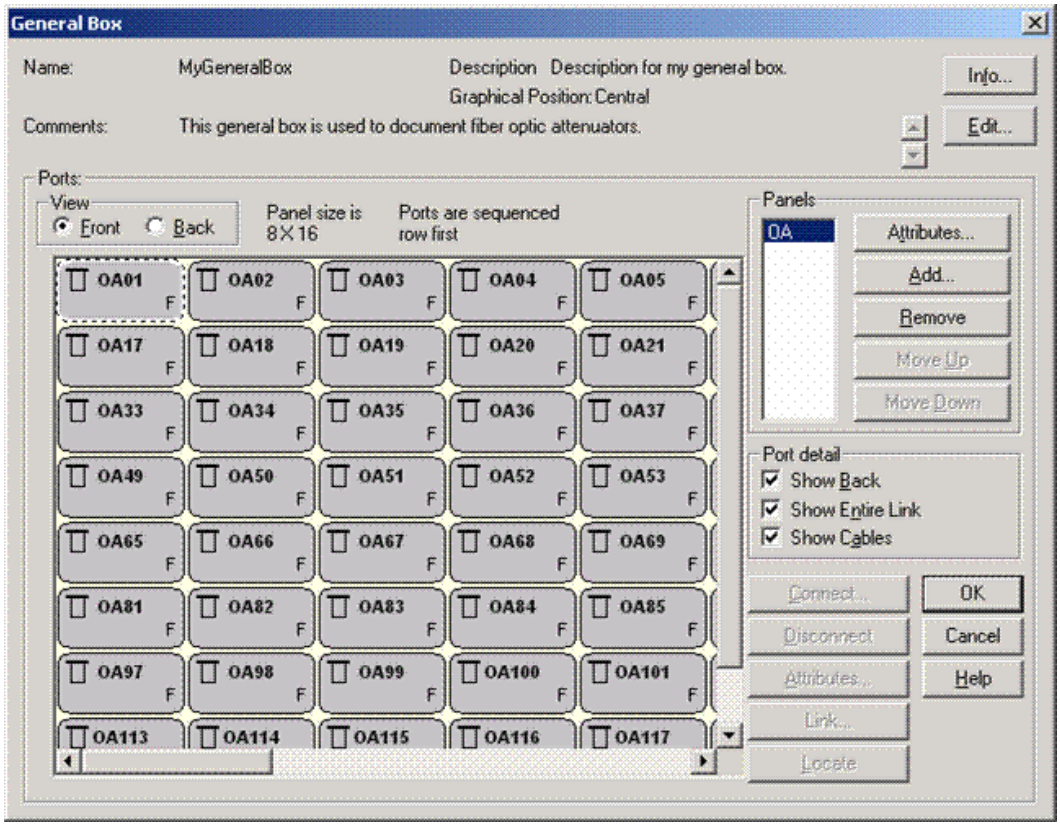


Figure 22-10 The General Box dialog

In addition, you can add or remove panels for a general box, change the attributes of a panel, and modify the order of the panels within the general box. Finally, you can edit the general box itself. The general box also contains 20 user fields like other physical objects in HCM. You can use General Box objects, therefore, to document your physical environment with greater accuracy.

22.13 View HCD reports from HCM

Whereas current HCM reports have the focus on physical objects and their connectivity, customers also need the logical I/O configuration view of the HCD reports. The HCD reports, currently, can only be obtained directly from HCD. Now, HCM provides a dialog to get an HCD textual report directly into an HCM window and allows saving that data on a PC file. Thus, the generation of HCD reports for an HCM user has been facilitated.

The function IODF Reports is invoked by selecting **File** → **View IODF Reports**. This submenu is disabled when running HCM in stand-alone mode. A sample screen is shown in Figure 22-11 on page 440. HCM offers this function via a new dialog. The same report types (CSS report, Switch report, and OS report) and report limitations (for example, specific processor or operating system) as under HCD can be selected from HCM. The request is sent to HCD. HCD generates the report and provides the output back to HCM, where it is displayed in a separate window.

The report has the same layout as under HCD, and can be saved in a file on the workstation. HCM does not validate the report input parameters or the report output. All validity checks are done by HCD. Any error messages from HCD are displayed in the Message List window of

HCM. The report produced based on the selections made in Figure 22-11 is shown in Figure 22-12.

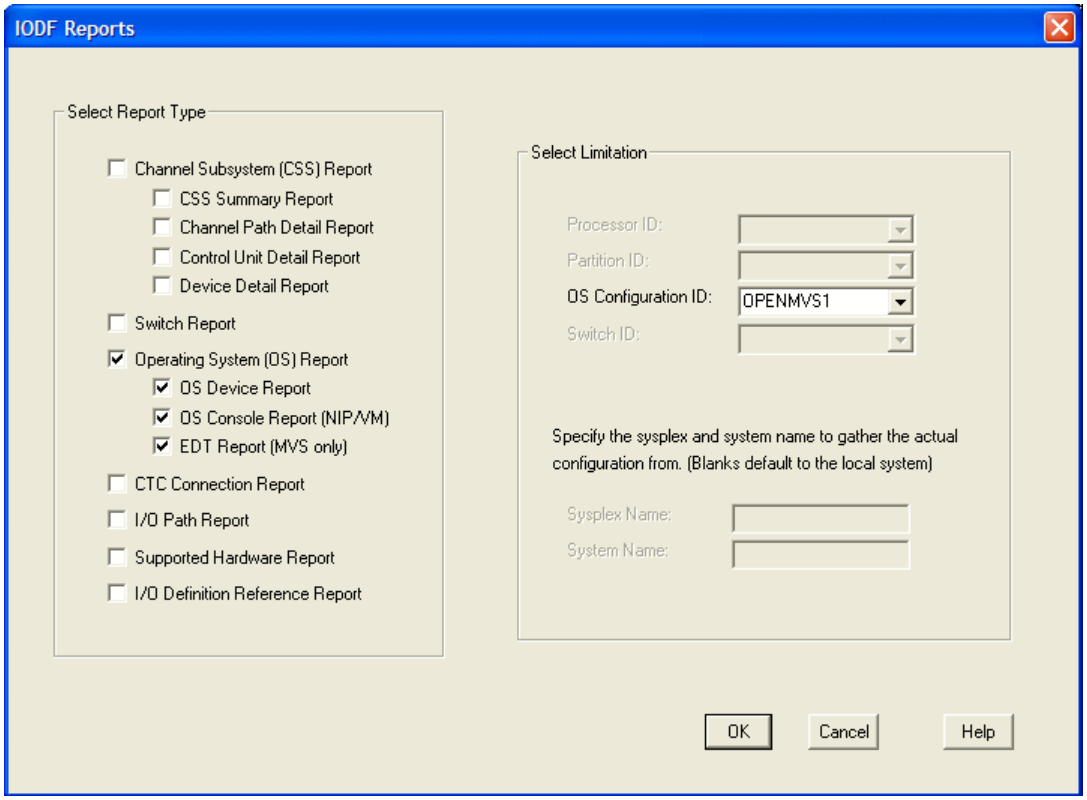


Figure 22-11 Report selection dialog

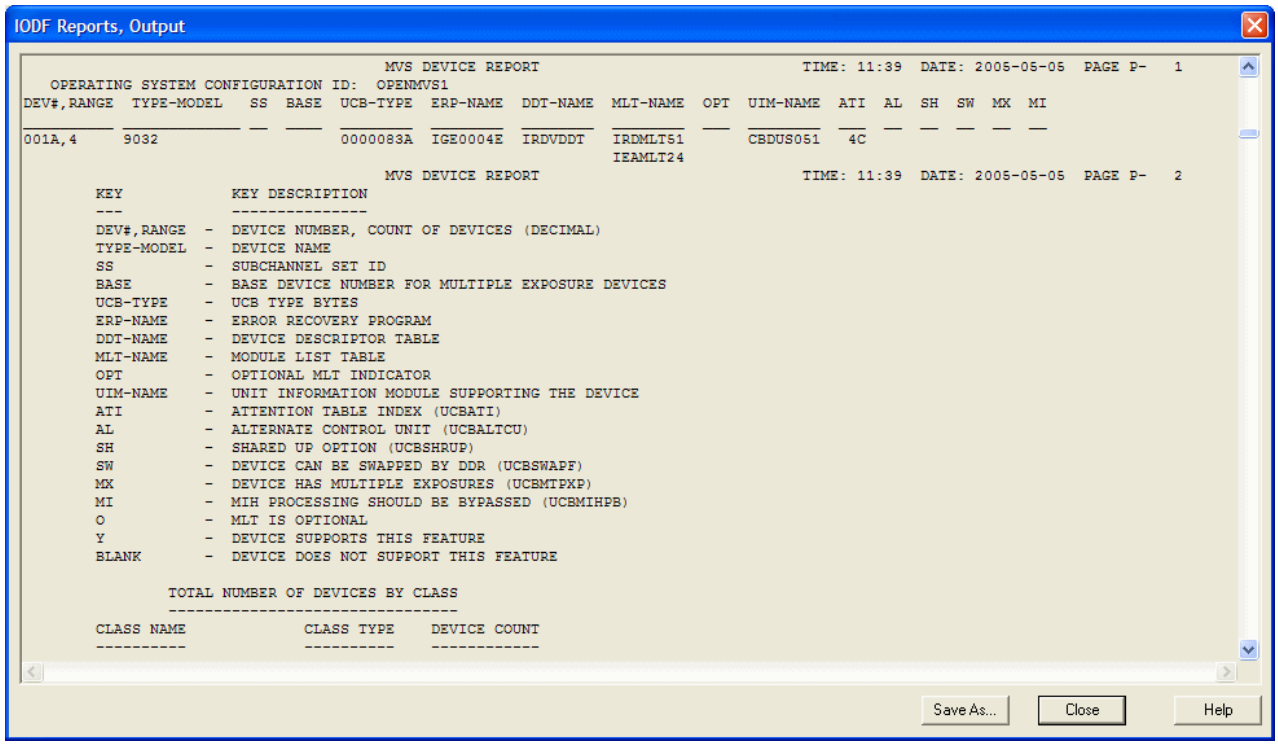


Figure 22-12 Sample report output

22.14 Migration to IODF V5

z/OS V1R7 HCD starts using the new V5 IODF format, representing devices in device groups rather than containing individual device definition records. z/OS V1R7 HCD can access an IODF from previous HCD releases and can perform view/read functions (for example, copy, activate) without permanently upgrading the IODF. In this case you will see an message box like the one shown in Figure 22-13. However, when you try to change the configuration contained in a lower version IODF, upgrading is required before the change. A message will inform you about the necessity of a permanent upgrade. Earlier HCD releases provide limited read access to V5 IODFs. There is coexistence support for V5 IODFs on back-level HCD releases for read-only functions like the ACTIVATE function. You need to install the corresponding PTF.

If you need to update a V5 IODF using an HCD version earlier than z/OS V1R7 HCD, you can use a STEPLIB or JOBLIB allocation for the z/OS V1R7 libraries (SYS1.LINKLIB, SYS1.NUCLEUS and SYS1.SCBDHENU). If you want to share an IODF among multiple z/OS or OS/390 systems that are at different release levels, you have to consider some restrictions concerning IPL, IODF usage, and dynamic reconfiguration.

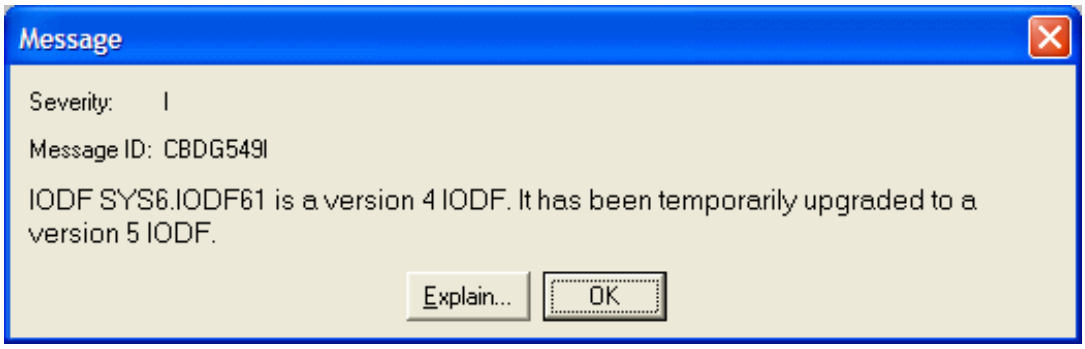


Figure 22-13 Migration of the IODF

IODF format conversions between version 4 and version 5

Consider the following IODF upgrade functions for migration of z/OS releases V1R4 to V1R7:

z/OS V1R7: IODF upgrade function available to migrate from version 4 to version 5 IODF.

z/OS V1R4 - V1R6: Fall-back solution to downgrade from version 5 IODF to version 4 IODF via Export I/O definitions (Build I/O configuration statements) and Import I/O definitions (Migrate I/O configuration statements).

23

RMF enhancements in z/OS V1R7

This chapter describes how the System Address Space Analysis section of the CPU Activity Postprocessor report is changed to reflect the higher number of CPUs available per MVS image.

23.1 CPU activity report

The System Address Space Analysis section of the CPU activity report provides overall address space information. In z/OS V1R7, this report is redesigned to reflect the higher number of CPUs available per MVS image.

The upper boundaries of 14 and 35 for the queue length distribution are no longer up-to-date. With the increasing numbers of CPUs per MVS image, the following problems arise:

- The upper boundary of 14 for the InReady queue was meaningful when MVS images used to have three to five processors, but nowadays it makes less sense.
- With WLM CPU management, the number of online CPs can change during a reporting interval. For this reason, it is difficult to determine whether and how many address spaces were actually waiting for a processor.
- The upper boundary of 35 for queue types In, OutReady, OutWait, LogicalOutReady, and LogicalOutWait, as well as the address space types, is no longer meaningful. With current z/OS releases, the In and LogicalOutWait queue, as well as the STC address spaces, always show 100% in the 35+ class. Actual distribution is not available.

Therefore, distribution of all other queue types and address space types, apart from the InReady queue, is removed from the report, leaving the minimum, maximum, and average numbers.

The InReady queue distribution in this version of z/OS is oriented towards the maximum number of online CPs. This approach considers the fact that the number of online CPs may vary from sample to sample. The base of each distribution bucket is always the current number of standard CPs being online at that point of time when the sample is taken. The “N” in the report represents the total number of online CPs when the sample is taken.

The number of address spaces in the InReady queue is not distributed into fixed buckets anymore. As can be seen in Figure 23-1, the first bucket (B1) reflects the percentage of samples when all jobs could be dispatched. The 2nd bucket (B2) reflects the percentage of samples when one job could not be dispatched, and so on. The last bucket (B13) reflects the percentage of samples when more than 80 jobs could not be dispatched.

New InReady queue distribution

Distribution buckets	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
Number of Address Spaces	<= N	N+1	N+2	N+3	<= N+5	<= N+10	<= N+15	<= N+20	<= N+30	<= N+40	<= N+60	<= N+80	> N+80

N = Number of CPs online when sample is taken

Figure 23-1 New InReady queue distribution

Figure 23-2 presents an example of samples taken with different numbers of online CPs and address spaces on the InReady queue, in order to show which bucket is incremented.

Sample	Time when sample taken	Address spaces on InReady queue	Number of CPs currently online	Bucket being incremented
1	08:10:00	9	10	B1 (<= N)
2	08:10:01	10	9	B2 (N+1)
3	08:10:02	15	8	B6 (<= N+10)
4	08:10:03	25	8	B8 (<= N+20)
5	08:10:04	18	8	B6 (<= N+10)
6	08:10:05	20	10	B6 (<= N+10)
7	08:10:06	15	10	B5 (<= N+5)
8	08:10:07	17	10	B6 (<= N+10)
9	08:10:08	10	9	B2 (= N+2)
10	08:10:09	8	9	B1 (<= N)

Figure 23-2 Samples of online CPs and address spaces on the InReady queue

The new design of the System Address Space Analysis section of the CPU Activity report can be seen in Figure 23-3 on page 446.

```
                              C P U  A C T I V I T Y  R E P O R T

SYSTEM ADDRESS SPACE ANALYSIS            SAMPLES =  3,600

NUMBER OF ADDRESS SPACES

        ------------------ QUEUE TYPES --------------------     --------- ADDRESS SPACE TYPES ---------
        IN              OUT       OUT    LOGICAL   LOGICAL      BATCH      STC     TSO    ASCH     OMVS
        READY      IN   READY     WAIT   OUT RDY   OUT WAIT

 MIN        1      54       0        0         0         83         5      119      10       0        3
 MAX       10      85       1        0         0        118        19      154      12       0       23
 AVG      2.4    56.4     0.0      0.0       0.0      105.4       6.1    141.7    10.5     0.0      3.5

DISTRIBUTION OF IN-READY QUEUE

 NUMBER OF             0    10   20   30   40   50   60   70   80   90   100
 ADDR SPACES    (%)    |....|....|....|....|....|....|....|....|....|....|

 <=  N         82.1    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  =  N +  1    12.9    >>>>>>>>
  =  N +  2     3.6    >>>
  =  N +  3     1.1    >
 <=  N +  5     0.0
 <=  N + 10     0.0
 <=  N + 15     0.0
 <=  N + 20     0.0
 <=  N + 30     0.0
 <=  N + 40     0.0
 <=  N + 60     0.0
 <=  N + 80     0.0
 >   N + 80     0.0

 N = NUMBER OF PROCESSORS ONLINE    (  3.0 ON AVG )
```

Figure 23-3 CPU activity report

23.2 Monitoring zFS file system activity

Two new Monitor III reports provide overview and performance information about the zFS activity:

- The zFS Summary Report helps to control the zFS environment and I/O balancing.
- The zFS Activity Report measures zFS activity on the basis of single file systems – for example, a file system's performance and DASD utilization.

See “RMF support for zFS” on page 363 for details about this new support.

23.3 Monitoring storage group and disk space

Two new Monitor III reports support disk space monitoring at the storage group level as well as at the volume level:

- The Storage Space Report displays information about capacity and available space for defined storage groups.

- The Disk Space Report provides capacity and space information for volumes belonging to the defined storage groups.

23.4 Support for IBM System z9 processors

Starting with IBM System z9 (z9-109) processors, the Integrated Cryptographic Service Facility (ICSF) exploits SHA-256 hashing. RMF enhances the Crypto Hardware Activity report to provide measurement data for SHA-1 and SHA-256. In addition, new overview conditions are available for the postprocessor.

24

SDSF enhancements

This chapter provides information about new facilities included in z/OS V1R7 SDSF. The enhancements and changes in SDSF are as follows:

- A new panel is used to manage health checking. This feature is described in “Using SDSF to manage checks” on page 145.
- Monitoring JES2 resources
- Other SDSF enhancements

24.1 Monitoring JES2 resources

The JES2 monitor (RM) panel allows authorized users to display information about JES2 resources such as JOEs, JQEs, and BERTs. From this panel, users can identify resource shortages and view the history for resources. Access the SDSF Resource Monitor panel with the **RM** command, as shown in Figure 24-1.

```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF RESOURCE MONITOR DISPLAY  SC70                    LINE 1-17 (17)
COMMAND INPUT ===>                                        SCROLL ===> CSR
NP   RESOURCE SysId Status     Limit  InUse  InUse% Warn% IntAvg IntHigh IntLow
     BERT     SC70             20000    542   2.71    80    542     544    541
     BSCB     SC70                10      0   0.00    80      0       0      0
     BUFX     SC70               200      0   0.00    80      0       3      0
     CKVR     SC70                17      0   0.00    80      0       1      0
     CMBS     SC70               208      0   0.00    80      0       0      0
     CMDS     SC70               200      0   0.00    80      0       0      0
     ICES     SC70                33      0   0.00    80      0       0      0
     JNUM     SC70             29001   5936  20.46    80   5936    5938   5934
     JOES     SC70             30000   6507  21.69    80   6507    6509   6506
     JQES     SC70             20000   5937  29.68    80   5937    5939   5935
     LBUF     SC70               120      0   0.00    80      0       0      0
     NHBS     SC70               100      0   0.00    80      0       0      0
     SMFB     SC70               102      0   0.00    80      0       0      0
     TBUF     SC70               106      0   0.00     0      0       0      0
     TGS      SC70             52045  25272  48.55    80  25271   25277  25265
     TTAB     SC70                 3      0   0.00    80      0       0      0
     VTMB     SC70                50      0   0.00    80      0       0      0
```

Figure 24-1 SDSF JES2 monitor display

You can view all intervals available in JES2 monitor using **RM ALL** in the SDSF command input and using the **S** command. In Figure 24-2 we show an example of the results of the **S TGS*** command. In this example we used the **ARRANGE TIME A RESOURCE** command to display column time after resource and the **SORT TIME A** command to do an ascending sort by time. Reset the resources shown by typing **S** without parameters.

```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF RESOURCE MONITOR DISPLAY  SC70                    LINE 1-26 (506)
COMMAND INPUT ===>                                        SCROLL ===> CSR
NP   RESOURCE Time     SysId Limit  InUse  InUse% Warn% IntAvg IntHigh Status
     TGS       9:54:33 SC70   52045  23841  45.80    80  23761   23841
     TGS      10:00:00 SC70   52045  23852  45.82    80  23847   23852
     TGS      11:00:00 SC70   52045  23860  45.84    80  23858   23860
     TGS      12:00:01 SC70   52045  23861  45.84    80  23861   23861
     TGS      13:00:00 SC70   52045  23863  45.85    80  23863   23863
     TGS      14:00:00 SC70   52045  23868  45.86    80  23867   23869
     TGS      15:00:00 SC70   52045  23868  45.86    80  23868   23868
     TGS      16:00:00 SC70   52045  23878  45.87    80  23875   23879
     TGS      17:00:00 SC70   52045  23881  45.88    80  23882   23883
     TGS      18:00:00 SC70   52045  23881  45.88    80  23881   23882
     TGS      19:00:00 SC70   52045  23882  45.88    80  23882   23882
     TGS      20:00:00 SC70   52045  23884  45.89    80  23884   23884
     TGS      21:00:00 SC70   52045  23885  45.89    80  23885   23885
     TGS      22:00:00 SC70   52045  23886  45.89    80  23886   23887
     TGS      23:00:00 SC70   52045  23888  45.89    80  23887   23888
     TGS       0:00:00 SC70   52045  23889  45.90    80  23889   23889
...
```

Figure 24-2 SDSF Resource Monitor Display panel for RM ALL

24.2 Other SDSF enhancements

This section describes the following additional enhancements to SDSF in this release:

- Support for NJE connections over TCP/IP in JES2
- Console restructure support
- Default browse action
- Cursor placement
- Unconditional wait on / command
- New parameters on the SR command
- New columns and action characters

24.2.1 Support for NJE connections over TCP/IP in JES2

The Lines and Nodes panels are enhanced to exploit new support in z/OS V1R7 JES2 for NJE connections over TCP/IP.

SDSF adds columns for IP address, IP name, port name and number, and network server name to the Lines panel. The over-typeable unit column accepts a new value, TCPIP.

SDSF adds a column for network server number to the Nodes panel. The column for network server number is over-typeable.

New column names in the Lines and Nodes panels are described in the SDSF Operation and Customization for z/OS V1R7 manual.

24.2.2 Console restructure support

SDSF includes several changes in response to z/OS consoles changes in z/OS V1R7:

- Support for migration IDs is removed. Users can no longer request a migration ID with the **SET CONSOLE** command.
- The master console is no longer used to issue system commands. The M prefix can no longer be used with the / command to request a master console.
- Data for the SR display, and WTORs for the log display, are obtained with a new service provided by consoles.
- New parameters in ISFPARMS provide control over whether an EMCS console is required to issue system commands, and control the authority used with the EMCS console, such as:
  - EMCSAUTH, which indicates the authority that will be used with the EMCS console.
  - EMCSREQ, which indicates whether SDSF must use the EMCS support for system commands.

Console query service CNZQUERY

SDSF uses a new console query service CNZQUERY to be able to display the information shown in Figure 24-3 on the SDSF SYSTEM REQUESTS line, where fields have the following meanings:

- (CEM) – Critical eventual action messages
- (EM) – Eventual action messages
- (IM) – Immediate action messages
- (REPLIES|R|RM) – Reply messages

```
Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF SYSTEM REQUESTS  RM 2    IM 6    CEM 0    EM 0    LINE 1-8 (8)
COMMAND INPUT ===>                                        SCROLL ===> CSR
PREFIX=AP*  DEST=(ALL)  OWNER=*  SYSNAME=
ACTION=//-Block,=-Repeat,+-Extend,C-Remove,D-Display,R-Reply
NP      REPLYID    SysName  JobName  Message-Text
        29         SC63     IMS71OG  *029 DFS996I *IMS READY*  IMSG
        524007     SC63     MERONIR  *WLMREG-- IWMSRSRG: Register sample server
        95014      SC65     JES3     *IAT7921 ISSUE START/CANCEL/RESTART DC REQ
        176014     SC65     IEESYSAS *IAZ0537I JES3NS NJETCP SERVER WAITING FOR
        168015     SC70     JESASO01 *IAZ0537I NETSRV1 NJETCP SERVER WAITING FO
        178014     SC65     JESBSO01 *IAZ0537I NETSRV1 NJETCP SERVER WAITING FO
        58013      SC64     TWSC     *IEF099I JOB TWSC     WAITING FOR DATA SET
        521        SC64     AOFAPPL  *521 DSI802A SC64N    REPLY WITH VALID NCC
```

Figure 24-3 SDSF SYSTEM REQUESTS panel with new information

24.2.3 Default browse action

SDSF adds function to allow for a default browse action (S, SB or SE) on the job, output, and CK panels. The default browse action is invoked when you select a row, that is, place the cursor in the NP column and press Enter.

For the job and output panels, these commands support the browse action (DA, H, I, JDS, O, OD, and ST).

Users can set the default browse action character with a new SDSF **SET BROWSE ?** command, as shown in Figure 24-4. The initial value can be set in ISFPARMS with the new BROWSE parameter. In addition, the related CURSOR parameter in ISFPARMS now accepts a value of TOP.

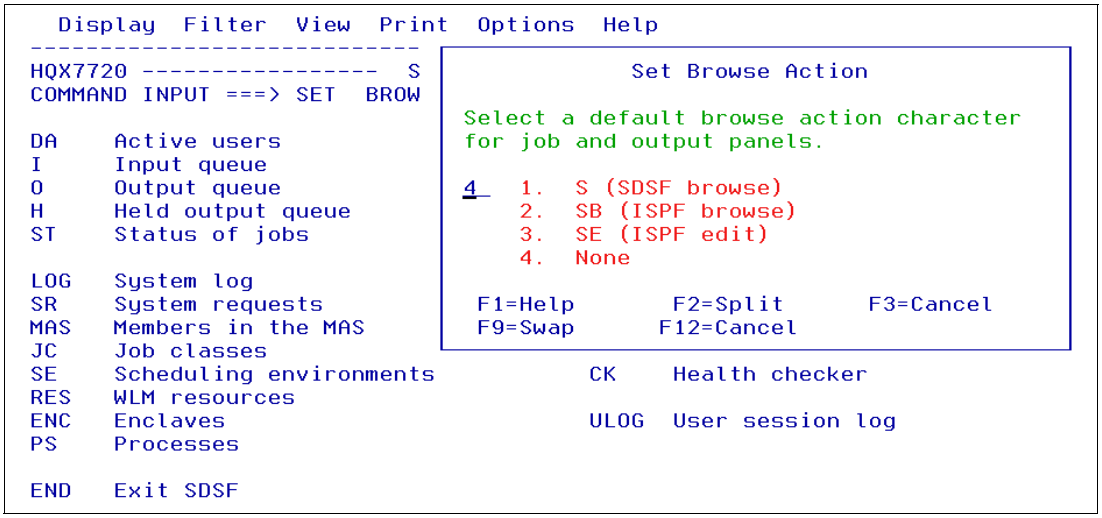


Figure 24-4 Set browse action default

24.2.4 Cursor placement

A new option for placement of the cursor on tabular panels scrolls the last row you worked with to the top of the panel and places the cursor on the command line. This is useful when you define a default browse action character with **SET BROWSE**.

You control the placement of the cursor with the **SET CURSOR ?** command or the set cursor choice from the options pull-down, as shown in Figure 24-5.

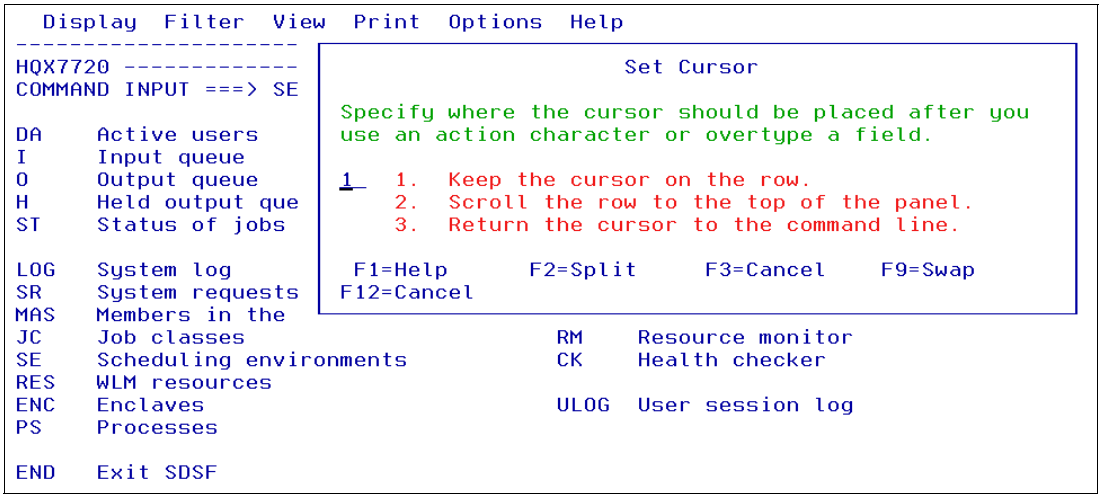


Figure 24-5 Set cursor pull-down

24.2.5 Unconditional wait on / command

SDSF provides additional user control over how long it waits before displaying messages issued in response to the / command, as follows:

```
W/D A,L
```

Users can force an unconditional wait that is not ended by the receipt of messages; SDSF waits the entire timeout interval that is specified with **SET DELAY**. You request an unconditional wait with a new W prefix on the / command.

24.2.6 New parameters on the SR command

SDSF adds parameters to the **SR** command that provide additional filters by message type. The parameters for the filter are: CEM (critical eventual action), EM (eventual action), and IM (immediate action). RM (replies) is added as a synonym for the existing R parameter. This parameters are provided in the queue field, as shown in Figure 24-6.

```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF SYSTEM REQUESTS  RM 1    IM 7    CEM 0    EM 0    LINE 1-8 (8)
COMMAND INPUT ===>                                      SCROLL ===> CSR
NP      REPLYID    obID    Date     Time     Console  RouteCd Desc Type   Queue
        21121      ES3     2005.131 08:08:44 VAINI    0000000 4200 ACTION IM
        592        TC25249 2005.131 08:30:05          4200000 0200 REPLY  RM
        86121      SO      2005.133 10:08:50          8100000 4000 ACTION IM
        427121     ES3     2005.136 03:13:21          0200000 4200 ACTION IM
        5082120            2005.139 10:40:59 INTERNAL 0120000 4300 ACTION IM
        480121     EESYSAS 2005.139 10:43:30 INTERNAL 0120000 4300 ACTION IM
        482121             2005.139 10:43:30 INTERNAL 0120000 4300 ACTION IM
        490121     ES3     2005.140 07:44:08          0200000 4200 ACTION IM
```

Figure 24-6 IQueue field in SDSF system requests panel

24.2.7 New columns and action characters

SDSF adds an over-typeable USERDATA column to the OD (output descriptors) and JDS (job data set) panels. Up to 16 values can be specified.

When accessing the OD panel and moving over to the USERDATA column, type a **+** in the USERDATA column and the panel shown in Figure 24-7 is displayed. You can type in your userdata here.

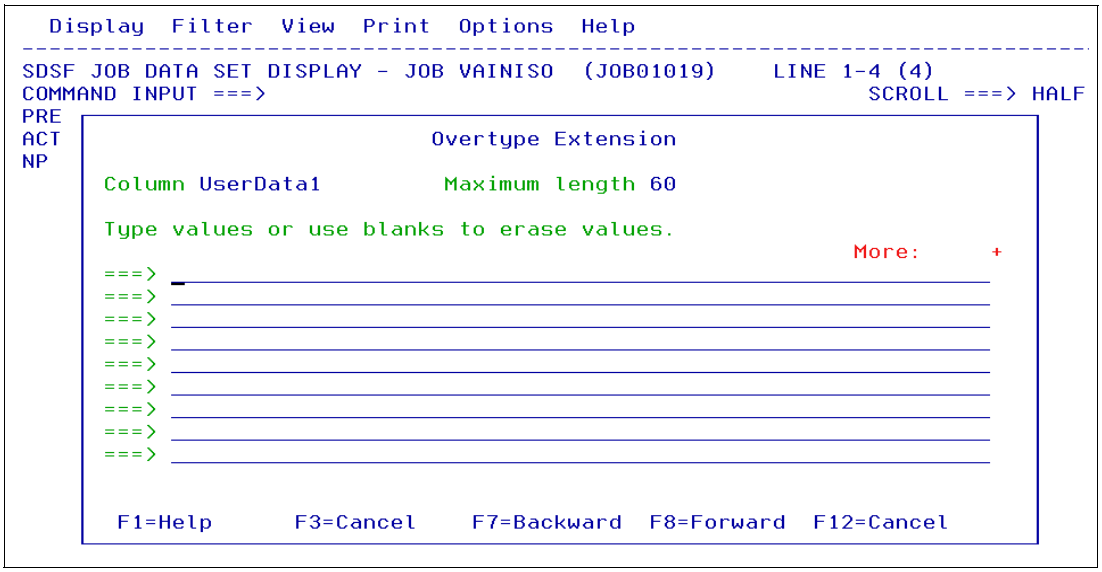


Figure 24-7 Panel to type in userdata

When you type in your userdata, for example the word TEST1 for a data set, the output descriptors panel with a Q action character in the NP column (shown in Figure 24-8) shows the use of the userdata column.

```
 Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------------
SDSF OUTPUT DESCRIPTORS - JOB USER1    (TSU24965)      LINE 183-208 (312)
COMMAND INPUT ===>                                       SCROLL ===> CSR
NP  DDNAME   Output Descriptors
    JESJCL   PrtQueue



    JESJCL   IP Destination
             LOCAL


    JESJCL   UserData
             TEST1
```

Figure 24-8 USERDATA column on OD panel

24.2.8 Specifying a disposition of KEEP

The **0** action on the H panel is enhanced to allow users to request an outdisp of KEEP. You have to type **0K** for this request. Previously, an **0** action always resulted in an outdisp of WRITE.

24.2.9 New support for zAAPs

Support is provided for the zSeries Application Assist Processor (zAAP) on the DA panel, including usage information at the system and address space level. This information was introduced in APARs PK06616 and PQ93310, which have been included in z/OS V1R7 SDSF.

The following information is added:

- Changes to the title line to show a zAAP view of CPU utilization. The zAAP view compliments the MVS and LPAR views already shown. A new column also shows the zAAP view of CPU utilization, and is useful for systems other than the one the user is logged on to. These changes were introduced in APAR PK06616, which has been incorporated into z/OS V1R7 SDSF.
- New columns to show zAAP utilization for an address space. These columns were introduced in APAR PQ93310, which has been incorporated into z/OS V1R7 SDSF.

New columns

The new columns are as follows:

ZAAPTIME Accumulated zAAP service time in seconds

ZAAPCPTM Accumulated general processor service time that was eligible for a zAAP, in seconds

ZAAPUSE Percent of the total zAAP time used by the address space in the most recent interval

SZAAP zAAP view of CPU utilization for the zAAP that is processing the job in the most recent interval

24.2.10 New action character on MAS panel

SDSF adds action characters on the MAS panel to display information about the JES2 monitor in the Log, and to stop the monitor. The new action characters are **J** and **ZM**.

```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF MAS DISPLAY SC65 XCFJESBB    1% SPOOL              COMMAND ISSUED
COMMAND INPUT ===>                                        SCROLL ===> HALF
RESPONSE=SC65
 @HASP9100 D MONITOR
 NAME     STATUS       ALERTS
 -------- ------------ ------------------------
 MAINTASK ACTIVE
 SAMPLER  ACTIVE
 COMMANDS ACTIVE
 PROBE    ACTIVE
 @HASP9102 MONITOR MODULE INFORMATION
 NAME     ADDRESS  LENGTH   ASSEMBLY DATE  LASTAPAR LASTPTF
 -------- -------- -------- -------------- -------- --------
 HASJMON  2455A000 000010B8 05/09/05 10.22 OA11475  UA18216
 HASJSPLR 2455F290 00002DA0 04/04/05 11.08 NONE     NONE
 HASJCMDS 2455B0B8 000041D8 08/12/05 08.26 OA12794  UA20508
```

Figure 24-9 Using action character J on MAS panel

Start or stop the monitor

The **ZM** action character is used to either start or stop the JES2 monitor.

24.3 New action characters on PS panel

SDSF adds action characters on the PS panel to provide new control of UNIX System Services processes, as follows:

T T generates the command **`F BPXOINIT,TERM=pid`**. This is equivalent to sending a SIGTERM to the process using the shell **`kill -s term pid`** command.

K K generates the command **`F BPXOINIT,FORCE=pid`** and is equivalent to the shell **`kill -s kill pid`** command. This sends a SIGKILL to the process.

25

z/OS V1R7 I/O supervisor enhancements

This chapter describes enhancements to the I/O supervisor component of z/OS V1R7, as follows:

- Captured UCB overlay protection
- IOS control blocks above the line
- New device support:
  - IBM TotalStorage DS8000
  - IBM TotalStorage DS6000

25.1 Captured UCB overlay protection

Previously it has been possible to define Unit Control Blocks (UCBs) to reside above the line (in 31-bit addressable storage) by coding LOCANY=YES in the hardware configuration for these devices. Other MVS control blocks continue to use a 3-byte field (24-bit address) to point to the UCB; therefore, a service was provided to allow UCBs to be "captured." This capture capability utilized storage sharing services to create a view of the UCB that resides below the line in 24-bit addressable storage.

Programs that manipulate captured UCBs can inadvertently overlay storage in the captured view of the UCB, which can result in a storage overlay of the area around the actual UCB. A storage overlay such as this can result in an unscheduled IPL.

In order to provide better protection for the operating system, in z/OS V1R7 you can now make the captured view of the UCB read only.

In z/OS V1R7, it is recommended that captured UCB overlay protection be enabled to prevent inadvertent overlay of UCBs, and thus improve system availability. To enable this protection, the IECIOSxx parmlib member should be updated to include the CAPTUCB,PROTECT=YES statement. Although this is the default value, it is sensible to highlight selection of this option by explicitly coding it in your IECIOSxx options.

This value can also be dynamically changed by the **SETIOS** command, as follows:

```
SETIOS CAPTUCB,PROTECT=YES | NO
```

```
SETIOS CAPTUCB,PROTECT=YES
IOS090I SETIOS. CAPTUCB UPDATE(S) COMPLETE
```

Figure 25-1 Using the SETIOS command to enable captured UCB protection

Alternatively, you can update your IECIOSxx parmlib member and issue the **SET IOS=xx** command.

You can display the current system setting by using the **D IOS,CAPTUCB** command.

```
D IOS,CAPTUCB
IOS088I 16.10.52 CAPTURED UCB DATA 849
CAPTURED UCB PROTECTION IS ENABLED
```

Figure 25-2 Display status of captured UCB protection

Interactions and dependencies

Prior to enabling captured UCB overlay protection, software could successfully make updates to a UCB in the captured view. Once UCB overlay protection is enabled, updates to the 24-bit view of the UCB will fail with a protection exception.

Any software that updates UCBs directly will need to use the real address of the UCB instead of using the captured UCB directly.

Implementation

Check with vendors that your ISV products will work correctly with captured UCB protection enabled. This support is enabled by default, but add CAPTUCB PROTECT=YES to your IECIOSxx parmlib member.

25.2 IOS control blocks above the line

In z/OS V1R7, the IOS control blocks (IOQs, IOSB, SRB and EWA) can be defined to reside above the line in 31-bit addressable storage, providing virtual storage constraint relief for storage below the line.

In order to enable this protection, the IECIOSxx parmlib member should be updated to include the STORAGE IOSBLKS=31 statement. Although this is the default value, it is sensible to highlight selection of this option by explicitly coding it in your IECIOSxx options.

This value can also be dynamically changed by the **SETIOS** command, as follows:

```
SETIOS STORAGE,IOSBLKS=24 | 31
```

Figure 25-3 shows the setting of the default and the response from the issued command.

```
SETIOS STORAGE,IOSBLKS=31
IOS090I SETIOS. STORAGE UPDATE(S) COMPLETE
```

Figure 25-3 Using the SETIOS command to set the storage attribute for IOS control blocks

Alternatively, you can update your IECIOSxx parmlib member and issue the following command specifying the parmlib member:

```
SET IOS=xx command
```

You can display the current system setting by using the **D IOS,STORAGE** command, as shown in Figure 25-4.

```
D IOS,STORAGE
IOS089I 10.38.09 STORAGE DATA 968
IOS BLOCKS RESIDE IN 31 BIT STORAGE
```

Figure 25-4 Display residency of IOS control blocks

25.2.1 Interactions and dependencies

UIMs, services, and I/O drivers will determine whether control blocks above the line can be used for processing. The main consideration when moving these control blocks and work areas above the line is whether the vendor products that are used in your installation will tolerate the 31-bit residency.

25.2.2 Implementation considerations

Check with vendors that your ISV products will work correctly with IOS control blocks in 31-bit storage. This function is enabled by default, but you can add the STORAGE IOSBLKS=31statement to your IECIOSxx parmlib member.

25.3 IBM TotalStorage DS8000

The IBM TotalStorage DS8000 series is intended for medium and large enterprises, and offers a high-capacity storage system designed to provide exceptional performance while adding virtualization capabilities that can help you allocate system resources more effectively

and better control application quality of service. The DS8000 sets a new standard in cost effectiveness and includes the following functional features:

- Up to 6 times faster than the ESS Model 800
- Scalability from 1 TB up to 192 TB
- Virtualization – logical partitioning
- Addressing enhancements
- Designed to add/adapt new technologies
- New management tools
- Designed for high availability 24 X 7 environments
- Flashcopy and mirroring capability

For further information about the IBM TotalStorage DS8000 see the following Web site:

`http://www.ibm.com/servers/storage/disk`

25.4 IBM TotalStorage DS6000

The IBM TotalStorage DS6000 series is designed to deliver many of the key features of the Enterprise Storage Server in an amazingly small, modular package.

The DS6000:

- Has a new control unit type - the D/T 1750
- Maintains ESS code structures and function
- Is attachable via FICON and FCP paths only
- Not all CHPIDs have equal access to the devices

  Introduces the concept of preferred paths. This is a performance enhancement that will ensure that I/O operations use the quickest path to a device.

For further information about the IBM TotalStorage DS6000 see the following Web site:

`http://www.ibm.com/servers/storage/disk`

Preferred path support

For the DS6000, z/OS V1R7 recognizes path attributes and does the following:

- Directs I/O down the set of preferred paths first
- Responds to loss of last preferred path by switching over and directing I/O down the set of non-preferred paths

Operator commands

The MVS commands **`D M=DEV`** and **`DEVSERV PATH`** have been updated to display information on preferred paths if they exist for the device. Examples are shown in Figure 25-5 and Figure 25-6.

```
d m=dev(980)
IEE174I 00.25.48 DISPLAY M 702
DEVICE 0980   STATUS=ONLINE
CHP                  87   D4   D8   0D
DEST LINK ADDRESS    00   00   00   00
PATH ONLINE          Y    Y    Y    N
CHP PHYSICALLY ONLINE Y   Y    Y    Y
PATH OPERATIONAL     Y    Y    Y    N
PATH ATTRIBUTES      PF   PF   NP   NS
MANAGED              N    N    N    N
MAXIMUM MANAGED CHPID(S) ALLOWED:  0
DESTINATION CU LOGICAL ADDRESS = 00
CU ND      = NOT AVAILABLE
DEVICE NED = 001750.000.IBM.13.000000015894.0E3F
*************** SYMBOL EXPLANATIONS **************
PF=PREFERRED   NP=NON-PREFERRED   NS=NOT SPECIFIED
```

Figure 25-5 D M=DEV display showing preferred pathing

```
DS PATH,8000
IEE459I 15.51.44 DEVSERV PATHS 431
 UNIT DTYPE  M CNT VOLSER  CHPID=PATH STATUS
      RTYPE   SSID CFW TC   DFW   PIN  DC-STATE CCA  DDC   ALT  CU-TYPE
 8000,33903 ,0,000,RF8000,B0=< B1=+ B2=+ B3=+ 95=+
      PATH ATTRIBUTES     NS   PF   NP   NP   PF
      1750    2240  Y  YY.  YY.    N   SIMPLEX   00   00        2107
 ************************ SYMBOL DEFINITIONS ************************
 O = ONLINE                          + = PATH AVAILABLE
 < = PHYSICALLY UNAVAILABLE          NS = NOT SPECIFIED
 PF = PREFERRED                      NP = NON-PREFERRED
```

Figure 25-6 DEVSERV PATH display showing preferred pathing

New messages for preferred pathing are issued as a result of VARY PATH and CF CHP commands:

```
IOS165I DEVICE dddd. PREFERRED PATHING NOW IN USE
IOS165I DEVICE dddd. PREFERRED PATHING NO LONGER IN USE
```

RMF support

RMF is updated to include information about preferred pathing where applicable, as follows:

- SMF record 78 (RMF Virtual Storage and I/O Queuing Activity) subtype 3
- SMF record 79 (RMF Monitor II Activity) subtype 14
- RMF post-processor I/O Queuing Activity report
- RMF Monitor II and III IOQUEUE I/O Queuing Activity report

26

Language Environment enhancements

IBM z/OS Language Environment provides common services and language-specific routines in a single run-time environment for C, C++, COBOL, Fortran (z/OS only; no support for z/OS UNIX System Services, or CICS), PL/I, and assembler applications. It offers consistent and predictable results for language applications, independent of the language in which they are written.

This chapter describes the enhancements that are introduced with z/OS Version 1 Release 7, as follows:

- Language Environment changes
- Language Environment options
- Migration to Release 7
- Behavior of applications

26.1 Language Environment enhancements

The following list shows you the major changes that have been made in z/OS V1R7.

- Support for RTLS has been removed.
- Language Environment, along with z/OS, provides another level of run-time options through a CEEPRMxx parmlib member in the system parmlib. The member is identified during IPL by a CEE=xx statement, either in the IEASYSxx data set or during the IPL.
- Language Environment supports the ability to provide additional run-time options at invocation through a DD statement named CEEOPTS.
- Parameter updates have been made to the STACK64 and THREADSTACK64 run-time options.
- The USRHDLR option has been updated.
- Hexadecimal floating point support for AMODE 64 C/C++ applications is provided.
- New SUSv3 application programming interfaces (APIs) are provided.
- There are updates to the abnormal termination exit information.
- ANSI C99 compliance is provided and the C/C++ run-time library supports the latest level of the standard.
- fork() in a multi-threaded environment is supported.
- Pre-initialized environments for authorized programs.
- XPCFTCH enhancements.
- IPv6 advanced socket API functions support.

26.1.1 Removal of RTLS

The RTLS function is no longer used by the Language Environment and the SCEERTLS data set is no longer shipped and should be removed from your installation. The CEEPRMxx member that resides in your parmlib replaces the CEECOPT, CEEDOPT, and CELQDOPT exits.

In z/OS V1R6, Language Environment no longer used the RTLS services provided by the operating system, which was previously used to assist with run-time migration. This includes removal of the RTLS initialization paths and all descriptions of RTLS in the Language Environment publications. The SCEERTLS library is no longer shipped. The following run-time options are no longer supported:

- LIBRARY
- RTLS
- VERSIONT

These run-time options are removed from the options reports generated by RPTOPTS(ON), CEEDUMP, and the IPCS verb exit. The CEEXOPT macro has been updated to prevent the use of these run-time options when building new CEEDOPT, CEECOPT, CEEROPT or CEEUOPT CSECTs. Existing CEECOPT and CEEDOPT members that contain these run-time options must be modified to remove them.

Attention: If these run-time options are encountered in existing CEEROPT or CEEUOPT CSECTs, Language Environment issues CEE3611I informational messages.

26.2 CEEPRMxx parmlib member

Starting in z/OS V1R7, IBM introduced a new parmlib member that controls the Language Environment default options in a system. You can find a sample member in the SCEESAMP library shipped with the operating system.

The parmlib values control such features as:

- The national language in which messages appear
- How a debug tool is invoked
- When condition handling is invoked
- How storage is allocated to the heap and stack
- How much storage is allocated
- The format of the program invocation character parameter
- Creation of a storage and/or run-time options report
- Shared storage allocations

Note: If you do not want to customize Language Environment now, you can put it into production using the IBM-supplied defaults. Or, you can use the instructions in this book to customize Language Environment later if you choose.

26.2.1 Application programmer users

For many of the run-time options, application programmers can override the installation defaults in their code. Application programmers at your site will be the primary users of Language Environment. Ask them what defaults they prefer for run-time options and user exits, which affect their work directly. Doing so will ensure that the modifications you make will best support the application programs being developed at your site.

26.2.2 Activating CEEPRMxx parmlib member

The CEEPRMxx parmlib member can be used for specifying default run-time options in Language Environment. The member is identified during IPL by a CEE=xx statement, either in the IEASYSxx data set or in the IPL parms. You can also do the following:

- Change the member after IPL with the SET CEE=xx command
- Change individual options using the SETCEE command
- Display current option settings with the D CEE command

Note: Using this support is not required, so the default IEASYS00 member does not specify a CEEPRMxx member. If you want to use this support, a sample CEEPRM00 member is included in SCEESAMP.

Attention: If you do not specify a CEEPRMxx parmlib member in the system, the default Language Environment options are provided by the CEECOPT, CEEDOPT, and CELQDOPT assembler exits.

26.2.3 Structure of the CEEPRMxx parmlib member

The CEEPRMxx parmlib member can have up to three sections, as shown in Table 26-1. In each section you can code all available options to configure the destination environment. The next section describes how to dynamically activate the changed CEEPRMxx parmlib member.

Table 26-1 Supported sections in the CEEPRMxx parmlib member

Section	Explanation
CEECOPT	31 bit CICS option group
CEEDOPT	31 bit non-CICS option group
CELQDOPT	64 bit options group

Figure 26-1 shows an example section you can specify in the CEEPRMxx parmlib member.

```
/*********************************************************************/
/* 64 bit options group                                              */
/*********************************************************************/
CELQDOPT(
         ENVAR(''),
         FILETAG(NOAUTOCVT,NOAUTOTAG),
         HEAPCHK(OFF,1,0,0,0),
         HEAPPOOLS64(OFF,8,4000,32,2000,128,700,256,350,
         1024,100,2048,50,3072,50,4096,50,8192,25,16384,10,
         32768,5,65536,5),
         HEAP64(1M,1M,KEEP,32K,32K,KEEP,4K,4K,FREE),
         INFOMSGFILTER(OFF,,,,),
         IOHEAP64(1M,1M,FREE,12K,8K,FREE,4K,4K,FREE),
         LIBHEAP64(1M,1M,FREE,16K,8K,FREE,8K,4K,FREE),
         NATLANG(ENU),
         NOTEST(ALL,*,PROMPT,INSPPREF),
         POSIX(OFF),
         PROFILE(OFF,''),
         RPTOPTS(OFF),
         RPTSTG(OFF),
         STACK64(1M,1M,128M),
         STORAGE(NONE,NONE,NONE,),
         THREADSTACK64(OFF,1M,1M,128M),
         TERMTHDACT(TRACE,,96),
         TRACE(OFF,,DUMP,LE=0),
         TRAP(ON,SPIE)
        )
```

Figure 26-1 Example of the CELQDOPT section in CEEPRMxx parmlib member

26.2.4 Changing the active CEEPRMxx member

To activate a whole new CEEPRMxx parmlib member after an IPL of the system, use the **SET CEE=xx** command. In Figure 26-2 you can see the successful activation of the CEEPRM00 parmlib member.

```
SET CEE=00
CEE3742I THE SET CEE COMMAND HAS COMPLETED.
```

Figure 26-2 Example of the SET CEE=00 command

After the activation of a new CEEPRMxx parmlib member, all values replaced by the new member come from the new member.

26.2.5 Changing individual options

To change a parameter in a specified group you can use the **SETCEE** command. The command is limited to 126 characters. Figure 26-3 shows the syntax of the command.

```
SETCEE [CEEDOPT,opt,opt,...]
       [CEECOPT,opt,opt,...]
       [CEELQDOPT,opt,opt,...]
```

Figure 26-3 Syntax of the SETCEE command

Figure 26-4 shows how you can set the Language Environment option POSIX to ON.

```
 SETCEE CEEDOPT,POSIX(ON)
 CEE3743I THE SETCEE COMMAND HAS COMPLETED.
```

Figure 26-4 Example of the SETCEE command

26.2.6 Display current settings

To display the current active member and its settings, use the **D CEE** command. The format of the command is:

```
D CEE [{,CEEDOPT} ] |,{CEECOPT} |,{CELQDOPT} [,L={a|name|name-a}]
```

If you enter the **D CEE** command only, you will see the active CEEPRMxx parmlib member suffixes. Table 26-2 provides a description of all valid option for the D CEE command.

Table 26-2 D CEE options

Option	Description
CEEDOPT	Displays all options for the 31 bit CICS option group
CEECOPT	Displays all options for the 31 bit non-CICS option group
CELWDOPT	Displays all options for the 64 bit options group
L=	You specify an output console by name or number for the output area

Sample output from the command is shown in Figure 26-5 on page 470.

```
CEE3745I 16.37.57 DISPLAY CEEDOPT
CEE=(01)
 LAST WHERE SET                OPTION
 ------------------------------------------------------------------
 CEEPRM01                       ABPERC(NONE)
 CEEPRM01                       ABTERMENC(ABEND)
 CEEPRM01                       ALL31(ON)
 CEEPRM01                       ANYHEAP(16384,8192,ANYWHERE,FREE)
 CEEPRM01                       BELOWHEAP(8192,4096,FREE)
 CEEPRM01                       CBLOPTS(ON)
 CEEPRM01                       CBLPSHPOP(ON)
 CEEPRM01                       CBLQDA(OFF)
 CEEPRM01                       CHECK(ON)
 CEEPRM01                       COUNTRY(US)
 CEEPRM01                       DEPTHCONDLMT(10)
 CEEPRM01                       ENVAR(("")
 CEEPRM01                       ERRCOUNT(0)
 CEEPRM01                       ERRUNIT(6)
 CEEPRM01                       FILEHIST
 CEEPRM01                       FILETAG(NOAUTOCVT,NOAUTOTAG)
 CEEPRM01                       HEAP(32768,32768,ANYWHERE,KEEP,8192
                                  4096)
.....................................................................
```

Figure 26-5 Sample output produced by the D CEE,CEEDO command

26.3 Using the CEEOPTS DD statement

Language Environment allows you to provide additional invocation-level run-time options using the CEEOPTS DD statement. The CEEOPTS DD can refer to:

- An in-stream data set
- A regular sequential data set
- A member of a regular or extended partitioned data set

If specified, the data set must be available during initialization of the enclave so the options can be merged.

The default Language Environment settings provided by CEEPRMxx parmlib member can be overridden using a CEEOPTS DD statement in your invoking JCL. At this time four possibilities exist to do that. They are described in the following sections.

CEEOPTS DD statement examples

To specify the CEEOPTS DD statement, use the following syntax:

- For in-stream JCL:

```
//CEEOPTS DD *
  ALL31(OFF),STACK(,,BELOW)
```

- For a sequential data set:

```
//CEEOPTS DD DSN=LUTZ.CEEOPTS.DATASET,DISP=SHR
```

- For a partitioned data set:

```
//CEEOPTS DD DSN=LUTZ.CEEOPTS.DATASET(MYOPTS), // DISP=SHR
```

- To ignore the DD statement:

```
//CEEOPTS DD DUMMY
```

Important: Consider the following restrictions:

1. The CEEOPTS DD supports only DASD data sets that can be read with QSAM. An informational message is issued when an unsupported data set type is used.
2. Only the first 3K (excluding comment lines) of the CEEOPTS file are read. All other information is ignored.
3. The file must be in fixed block or fixed format. Variable block format is not supported.
4. The CEEOPTS DD is ignored under CICS, LRR, SPC, and for an exec() program.

```
//CEEIVP   JOB ,'KUEHNER',NOTIFY=&SYSUID,
//    CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),TIME=1440
//*====================================================================*/
//* sample that invokes the Language Environment IVP program.        */
//* Language Environment defaults will be override by the CEEOPTS    */
//* options.                                                          */
//*====================================================================*/
//IVP     EXEC PGM=CEEIVP,PARM='RPTOPTS(ON)/'
//STEPLIB  DD DISP=SHR,DSN=LUTZ.LOADLIB
//CEEOPTS  DD *
 ALL31(OFF),STACK(,,BELOW)
/*
//SYSPRINT DD SYSOUT=*
//CEEMOUT  DD SYSOUT=*
//CEEOUT   DD SYSOUT=*
//
```

Figure 26-6 Sample JCL to invoke the LE IVP program

Figure 26-7 shows part of the sample IVP job output to demonstrate the use of the CEEOPTS DD statement.

```
Options Report for Enclave CEEIVP 05/03/05 6:21:55 PM
Language Environment V01 R07.00

LAST WHERE SET                OPTION
------------------------------------------------------------------------------
Installation default            ABPERC(NONE)
Installation default            ABTERMENC(ABEND)
Installation default          NOAIXBLD
DD:CEEOPTS                      ALL31(OFF)
Installation default            ANYHEAP(16384,8192,ANYWHERE,FREE)
Installation default          NOAUTOTASK
Installation default            BELOWHEAP(8192,4096,FREE)
```

Figure 26-7 Part of the IVP job output

26.3.1 Language Environment run-time options

The default Language Environment run-time options shipped by IBM where changed as follows:

- Parameter update for STACK64

- Parameter update for THREADSTACK64
- USRHDLR was changed

STACK64 changes

The STACK64 option controls the allocation of the threads stack storage for AMODE64 applications.

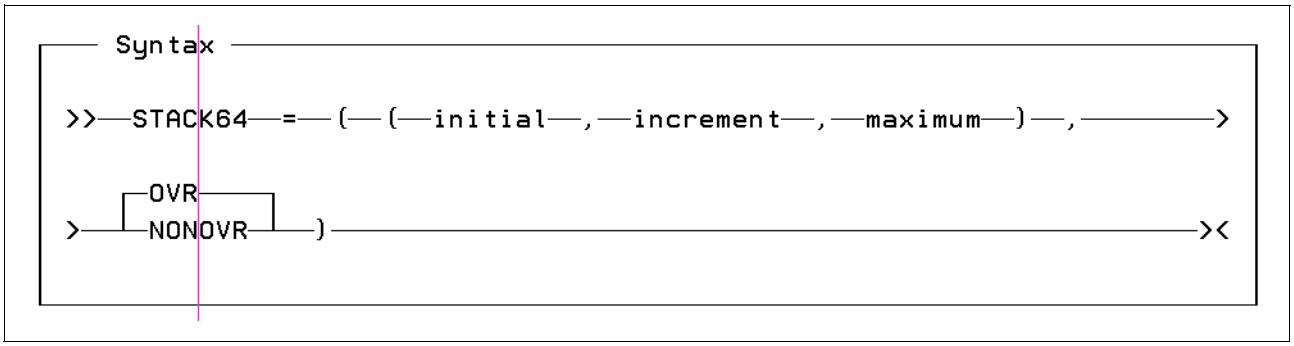


Figure 26-8 Syntax of the STACK64 parameter

Figure 26-8 shows the syntax of the STACK64 parameter. The changes are:

- When the maximum parameter size is less than the initial size, *initial* is used as the maximum stack size.
- The maximum stack segment is the maximum of STACK64 initial and maximum sizes.

THREADSTACK64 changes

The THREADSTACK64 option controls the allocation of the thread stack storage for AMODE64 applications, except for the initial thread in a multi-threaded environment.

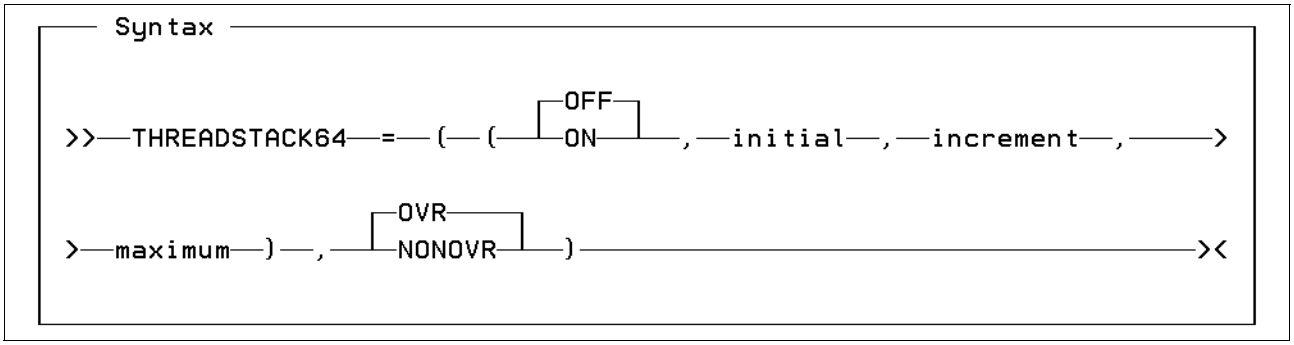


Figure 26-9 Syntax of the THREADSTACK64 parameter

Figure 26-9 shows the syntax of the THREADSTACK64 parameter. The changes are:

- When the maximum parameter size is less than the initial size, *initial* is used as the maximum stack size.
- The maximum thread stack segment is the maximum of THREADSTACK64 initial and maximum sizes.

USRHDLR changes

USRHDLR registers a user condition handler at stack frame 0, allowing you to register a user condition handler without having to include a call to CEEHDLR in your application and then recompile the application.

The following usages option was added:

- To turn off one of the sub-options previously specified by USRHDLR (lmname or lmname2), specify the option with either empty single quotes or empty double quotes. For example, to turn off the lmname2 sub-option after it had been previously specified, use either USRHDLR(lmname,'') or USRHDLR(lmname,"").

26.3.2 Customizing user exits

Restriction: Only the abnormal user exit supports AMODE 64 applications.

Because of the removal of some parameters (described in 26.1.1, “Removal of RTLS” on page 466) your user exits may be recompiled. Otherwise, you will receive an informational message that has no impact on the operation.

Change: AMODE 64 abnormal termination exit routines should specify the __FETCHABLE=RENT__ option on the CELQPRLG MACRO.

26.4 Migration to Release 7

This section describes Language Environment migration actions that you can perform on your current (old) system. You don’t need the z/OS V1R7 level of code to make these changes, and the changes don’t require the z/OS V1R7 level of code to run once they are made.

26.4.1 Update the CSD based on the newest CEECCSD

In z/OS V1R7 IBM provides an update of the CICS CSD file. The new definitions for supporting Language Environment in CICS are in the hlq.SCEESAMP library. Install the new definition against your active CSD file in the appropriate group.

Note: The group containing the Language Environment definition must be in the group listed in the CICS startup group.

IBM provides an additional member in hlq.SCEESAMP called CEECCSDX. This member contains the CICS CSD definition for use of the XPLINK option in applications programs.

Note: This job does not need to be run for CICS releases earlier than CICS TS 3.1.

26.4.2 Update Language Environment load modules in the LPA

Basically, IBM recommends placing the whole hlq.SCEELPA in your LPA concatenation. The SCEELPA contains a subset of the modules from the SCEERUN library and load modules are reentrant and heavily used by z/OS.

Attention: If you add modules from the SCEERUN library to LPA or LINKLST, be sure that all applications are able to work with the Language Environment. Otherwise, you should use the STEPLIB mechanism until the application is Language Environment ready.

Figure 26-10 shows sample statements placed in the PROGxx member to load Language Environment modules to the LPA.

```
LPA ADD DSNAME(CEE.SCEERUN) MODNAME(CEEBLRR,CEELRRIN,CEELRRTR,
              CEEMENU0,CEEMENU2,CEEMENU3,CEEMENU4,CEEMENU5,
              CEEMUEN0,CEEMUEN2,CEEMUEN3,CEEMUEN4,CEEMUEN5)

LPA ADD DSNAME(CEE.SCEERUN) MODNAME(CEEBINIT,
               CEEBLIBM,CEEBLIIA,CEEBPICI,CEEPIPI)
```

Figure 26-10 Add modules to LPA

Figure 26-11 shows the eligible message modules for Japanese language only.

```
LPA ADD DSNAME(CEE.SCEERUN) MODNAME(CEEMJPN0,
               CEEMJPN2,CEEMJPN3,CEEMJPN4,CEEMJPN5)
```

Figure 26-11 Add additional Japanese modules

Attention: The use of dynamic LPA will decrease the amount of available CSA/ECSA storage.

Furthermore, you can place a lot of language-dependent modules in the LPA or ELPA, specifically:

- C/C++ component modules
- PL/I Component modules
- FORTRAN component modules
- COBOL component modules

Table 26-3 provides a list of sample jobs that reside in the SCEESAMP library for each language.

Table 26-3 sample jobs for LPA usage

Language	sample job
C/C++	EDCWLPA
PL/I	IBMALLP2
FORTRAN	AFHWMLP2
COBOL	IGZWMLP4

26.4.3 Update Language Environment load modules in the LNKLST

Depending on your installation, we recommend placing the SCEERUN and SCEERUN2 into the LNKLST concatenation and removing any references in STEPLIBs. If you have application programs that are unable to run with the language environment you should add the pre-LE runtime libraries.

Attention: If you add the SCEERUN library LNKLST, be sure that all applications are able to work with the Language Environment. Otherwise you should use the STEPLIB mechanism until the application is Language Environment ready.

A

Sample code for the Subsystem Interface (SSI)

This appendix provides sample code related to the enhancements made in the Subsystem Interface (SSI):

- Change the SYSOUT priority with a SAPI call.
- Issue a Job Verbose call.

A.1 Change the SYSOUT priority with a SAPI call

The following is an example of how the SYSOUT priority can be changed. The program must run in an authorized library.

Example 26-1 Issue a SAPI Call

```
SAPICALL TITLE  'Sample SAPI Call (change the OUTPUT Priority)'
SAPICALL CSECT  ,
SAPICALL AMODE  31
SAPICALL RMODE  ANY
         USING  STATWORK,R10         Est work area addressability
         USING  STATMAIN,R12         Est base addressability
STATMAIN STM    R14,R12,12(R13)      Save callers registers
         LR     R12,R15              Set base register
         STORAGE OBTAIN,LENGTH=STATWLEN,ADDR=(R10),LOC=ANY             C
                                     Obtain local work area

         LR     R0,R10               Zero the
         LA     R1,STATWLEN           work area
         SLR    R15,R15                that was
         MVCL   R0,R14                  just obtained

         ST     R13,SAVEAREA+4       Chain
         LA     R15,SAVEAREA          in
         ST     R15,8(R13)             new
         LR     R13,R15                 save area
*********************************************************************
*        Process the SSOB
*********************************************************************
         USING  SSOB,WK_SSOB
         XC     SSOBEGIN(SSOBHSIZ),SSOBEGIN  clear the ssob
         MVC    SSOBID,=C'SSOB'      set eyecatcher
         MVC    SSOBLEN,=Y(SSOBHSIZ) set size
         MVC    SSOBFUNC,=AL2(SSOBSOU2) set function (SAPI 79)
         LA     R0,WK_SSS2           R0 -> WK_SSS2
         ST     R0,SSOBINDV          save SSS2 address in SSOB
*********************************************************************
*        Process the SSS2
*********************************************************************
         USING  SSS2,WK_SSS2
         MVI    SSS2TYPE,SSS2PUGE    Request type is PUT/GET
         MVI    SSS2VER,SSS2CVER     set current version of SSS2
         MVC    SSS2LEN,=Y(SSS2SIZE) set size of SSS2
         MVC    SSS2EYE,=C'SSS2'     set eyecatcher
         MVI    SSS2SEL1,SSS2SCLS    use output class as filter
         OI     SSS2SEL1,SSS2SJBI    use job id as filter
         MVI    SSS2CLSL,C' '        blank out the
         MVC    SSS2CLSL+1(L'SSS2CLSL-1),SSS2CLSL class selection field
         MVI    SSS2CLSL,C'A'        set output class filter
         MVC    SSS2JBIL,=Cl8'JOB24897' we only want
         MVC    SSS2JBIH,=Cl8'JOB24897'  to process this job id
*********************************************************************
*        IEFSSREQ Call
*********************************************************************
         MODESET MODE=SUP            Supervisor state for SSI funct.
         LA     R1,WK_SSOB           Get SSOB address
         O      R1,=X'80000000'      Indicate last SSOB
         ST     R1,PARMPTR           Set parm pointer
         LA     R1,PARMPTR           Get R1 for IEFSSREQ
```

```
         IEFSSREQ    ,               Issue SAPI call
         LTR     R15,R15             Any SSI errors ?
         BNZ     ERROR               yes, go process errors
         MODESET MODE=PROB           Return to problem program state
         CLC     SSOBRETN,=F'0'      Check return code from subsystem
         BH      SUBERROR            go process subsystem errors
*********************************************************************
*        Modify SYSOUT priority, then return to caller
*********************************************************************
         MVI     SSS2DSP2,SSS2RPRI   modify SYSOUT priority
         MVI     SSS2DPRI,X'8A'      set new SYSOUT priority
         MVI     SSS2DSP1,SSS2DKPE   keep the dataset, do not delete
         OI      SSS2DSP1,SSS2RNPT   finished with the dataset
         MVI     SSS2MSC1,SSS2CTRL   end of processing
         MODESET MODE=SUP            Supervisor state for SSI funct.
         LA      R1,WK_SSOB          Get SSOB address
         O       R1,=X'80000000'     Indicate last SSOB
         ST      R1,PARMPTR          Set parm pointer
         LA      R1,PARMPTR          Get R1 for IEFSSREQ
         IEFSSREQ    ,               Issue SAPI call
         LTR     R15,R15             Any SSI errors ?
         BNZ     ERROR               yes, go process errors
         MODESET MODE=PROB           Return to problem program state
         CLC     SSOBRETN,=F'0'      Check return code from subsystem
         BH      SUBERROR            go process subsystem errors
         B       EXIT

*********************************************************************
*        Error processing
*********************************************************************
ERROR    DS      0H
         LR      R2,R15              Save return code
         MODESET MODE=PROB           Return to problem program state
*        ....
*        ....
         B       EXIT
*********************************************************************
*        Errors from the subsystem interface
*********************************************************************
SUBERROR DS      0H
*        inspect Returncode SSOBRETN
*        inspect Reasoncode SSS2REAS
*********************************************************************
*        Return to the caller
*********************************************************************
EXIT     DS      0H
         L       R13,SAVEAREA+4      Get callers save area
         STORAGE RELEASE,LENGTH=STATWLEN,ADDR=(R10)                  C
                                     Return local work area
         L       R14,12(R13)         Restore callers
         LM      R0,R12,20(R13)      registers
         SLR     R15,R15             Set a zero return code
         BR      R14                 Return to caller
         DROP    R10,R12

         LTORG ,

*********************************************************************
*        Work area DSECT
*********************************************************************
```

```
STATWORK DSECT ,
SAVEAREA DS   18F                    Save area
PARMPTR  DS   A                      Pointer for MVS calls
WK_SSOB  DS   CL(SSOBHSIZ)
WK_SSS2  DS   CL(SSS2SIZE)
STATWLEN EQU  *-STATWORK             Length of local storage area
*********************************************************************
*        Equates *
*********************************************************************
R0       EQU  0
R1       EQU  1
R2       EQU  2
R3       EQU  3
R4       EQU  4
R5       EQU  5
R6       EQU  6
R7       EQU  7
R8       EQU  8
R9       EQU  9
R10      EQU  10
R11      EQU  11
R12      EQU  12
R13      EQU  13
R14      EQU  14
R15      EQU  15
*********************************************************************
*        DSECTS
*********************************************************************
         CVT      DSECT=YES,LIST=NO
         IEFJESCT TYPE=DSECT
         IEFJSSIB
         IEFSSOBH
         IAZSSS2  DSECT=YES
         END
```

A.2 Issue a Job Verbose call

Here is an example of how the Job Verbose Call can be called. The program must run in an authorized library.

Example 26-2 Issue a Job Verbose Call

```
JOBVERB  TITLE  'Sample SSI Job Verbose Call'
JOBVERB  CSECT  ,
JOBVERB  AMODE  31
JOBVERB  RMODE  ANY
         USING  STATWORK,R10         Est work area addressability
         USING  STATMAIN,R12         Est base addressability
STATMAIN STM    R14,R12,12(R13)      Save callers registers
         LR     R12,R15              Set base register
         STORAGE OBTAIN,LENGTH=STATWLEN,ADDR=(R10),LOC=ANY             C
                                     Obtain local work area
         LR     R0,R10               Zero the
         LA     R1,STATWLEN           work area
         SLR    R15,R15                that was
         MVCL   R0,R14                  just obtained

         ST     R13,SAVEAREA+4       Chain
         LA     R15,SAVEAREA          in
         ST     R15,8(R13)             new
         LR     R13,R15                 save area
*********************************************************************
*        Set up basic extended status SSOB *
*********************************************************************
         USING  SSOB,STSSOB          Est SSOB addressability
         LA     R0,STSSOB            Ensure that
         LA     R1,L'STSSOB           the SSOB
         SLR    R15,R15                area is
         MVCL   R0,R14                  all zero

         MVC    SSOBID,=C'SSOB'      Set SSOB eyecatcher
         MVC    SSOBLEN,=Y(SSOBHSIZ) Set length of SSOB header
         MVC    SSOBFUNC,=Y(SSOBESTA) Set status 2 function code
         MVC    SSOBSSIB,=F'0'        Use LOJ SSIB
         LA     R0,SSOB+SSOBHSIZ      Point to STAT extension
         ST     R0,SSOBINDV          Point base to extension

         USING  STAT,SSOB+SSOBHSIZ   Est STAT extension addr'blty
         MVC    STATEYE,=C'STAT'     Move in the eyecatcher
         MVC    STATLEN,=Y(STATSIZE) Set length of extension
         MVC    STATVER,=AL1(STATCVRL,STATCVRM) Set current version
         MVI    STATTYPE,STATVRBO    Set Job verbose request
*********************************************************************
*        Make only filter this Job ID
*********************************************************************
         OI     STATSEL1,STATSJBI    Use STATJBIL, STATJBIH as filter
         MVC    STATJBIL,=C'JOB24692'  USE THIS
         MVC    STATJBIH,=C'JOB24692'   JOB ID
*********************************************************************
*        Call the subsystem
*********************************************************************
         MODESET MODE=SUP            Supervisor state for SSI function
         LA     R1,STSSOB            Get SSOB address
         O      R1,=X'80000000'      Indicate last SSOB
         ST     R1,PARMPTR           Set parm pointer
```

```
         LA     R1,PARMPTR           Get R1 for IEFSSREQ
         IEFSSREQ ,                  Issue extended status SSI call
         LTR    R15,R15              Any SSI errors?
         BNZ    ERROR                Yes, go process errors
         MODESET MODE=PROB           Return to problem program state
*********************************************************************
*        Process results for IEFSSREQ
*********************************************************************
         USING  STATJQ,R4            Est STATJQ addressability
         L      R4,STATJOBF          R4 -> SJQE (STATJQ)
         L      R5,STJQVRBO          R5 -> SJVE (STATVE)
*********************************************************************
*        Processing of SJVE (STATVE)
*********************************************************************
         XR     R6,R6                clear R6
         LH     R6,STVEOHDR-STATVE(,R5) offset to first header section
         AR     R5,R6                R5 -> Job verbose data header
         LA     R6,STJVSIZE          R6 = offset to Job QUEUE element
         AR     R5,R6                R5 -> Job QUEUE element section
         USING  STATJQVB,R5          Est STATJQVB addressability
*********************************************************************
*        Processing of Job Queue Element Verbose Section
*********************************************************************
*        ....
*        ....
         DROP   R5
*********************************************************************
*        Finished, return data area passed
*********************************************************************
DONESJQE DS     0H
         MODESET MODE=SUP            Supervisor state for SSI function
         MVI    STATTYPE,STATMEM     Set memory management call
         LA     R1,STSSOB            Get SSOB address
         O      R1,=X'80000000'      Indicate last SSOB
         ST     R1,PARMPTR           Set parm pointer
         LA     R1,PARMPTR           Get R1 for IEFSSREQ
         IEFSSREQ  ,                 Issue extended status SSI call
         MODESET MODE=PROB           Return to problem program state
         B      EXIT                 Go exit the command processor
*********************************************************************
*        Error processing
*********************************************************************
ERROR    LR     R2,R15               Save return code
         MODESET MODE=PROB           Return to problem program state
*        ....
*        ....
*********************************************************************
*        Return to the caller
*********************************************************************
EXIT     L      R13,SAVEAREA+4       Get callers save area
         STORAGE RELEASE,LENGTH=STATWLEN,ADDR=(R10)                  C
                                     Return local work area
         L      R14,12(R13)          Restore callers
         LM     R0,R12,20(R13)       registers
         SLR    R15,R15              Set a zero return code
         BR     R14                  Return to caller
         DROP   R10,R12

         LTORG ,
```

```
*************************************************************************
*        Work area DSECT
*************************************************************************
STATWORK DSECT ,
SAVEAREA DS   18F                    Save area
PARMPTR  DS   A                      Pointer for MVS calls
STSSOB   DS   XL(SSSTLEN8)           Enhanced status SSOB
STATWLEN EQU  *-STATWORK             Length of local storage area
*************************************************************************
*        Equates *
*************************************************************************
R0       EQU  0
R1       EQU  1
R2       EQU  2
R3       EQU  3
R4       EQU  4
R5       EQU  5
R6       EQU  6
R7       EQU  7
R8       EQU  8
R9       EQU  9
R10      EQU  10
R11      EQU  11
R12      EQU  12
R13      EQU  13
R14      EQU  14
R15      EQU  15
*************************************************************************
*        DSECTs
*************************************************************************
         IEFJESCT ,
         IEFJSSOB ,
         IAZSSST DSECT=YES
         IAZJSAB ,
         CVT DSECT=YES
JOBVERB  CSECT ,
         END ,
```

B

Sample code for an IBM Healthchecker for z/OS check

This appendix provides an example of the following components:

- An HZSADDCHECK exit routine
- Message input for creating a message table
- An example of a simple check routine

B.1 An HZSADDCHECK exit routine

The following is an example of an HZSADDCHECK exit routine. The routine must be reeantrant and linked into an APF-authorized library.

Example: B-1 An HZSADDCHECK exit routine

```
REDAC001 TITLE  'My own Healthchecker Checks'
REDAC001 CSECT  ,
REDAC001 AMODE  31
REDAC001 RMODE  ANY
         USING  STATWORK,R10         Est work area addressability
         USING  STATMAIN,R12         Est base addressability
STATMAIN STM    R14,R12,12(R13)      Save callers registers
         LR     R12,R15              Set base register
         STORAGE OBTAIN,LENGTH=STATWLEN,ADDR=(R10),LOC=ANY            C
                                     Obtain local work area
         LR     R0,R10               Zero the
         LA     R1,STATWLEN           work area
         SLR    R15,R15                that was
         MVCL   R0,R14                  just obtained

         ST     R13,SAVEAREA+4       Chain
         LA     R15,SAVEAREA          in
         ST     R15,8(R13)             new
         LR     R13,R15                 save area
*********************************************************************
*        Processing
*********************************************************************
ADDCH    DS     0H
         HZSADDCK OWNER==CL16'REDBOOK',                               X
               NAME==CL32'RED_CHECK_LOGON',                           X
               CHECKROUTINE==CL8'REDCHK01',                           X
               MSGTBL==CL8'REDMSG1',                                  X
               EXITRTN==CL8'REDAC001',                                X
               DATE==CL8'20050518',                                   X
               REASON==CL126'Playing with z/OS V1R7',                 X
               REASONLEN==F'126',                                     X
               SEVERITY=HI,                                           X
               INTERVAL=TIMER,MINUTES==H'5',                          X
               PARMS==CL8'GABERT',PARMSLEN=8,                         X
               MF=(E,WHZSADDCK)

*********************************************************************
*        Return to the caller
*********************************************************************
EXIT     L      R13,SAVEAREA+4       Get callers save area
         STORAGE RELEASE,LENGTH=STATWLEN,ADDR=(R10)                   C
                                     Return local work area
         L      R14,12(R13)          Restore callers
         LM     R0,R12,20(R13)       registers
         SLR    R15,R15              Set a zero return code
         BR     R14                  Return to caller
         DROP   R10,R12

         LTORG ,

*********************************************************************
*        Work area DSECT
*********************************************************************
```

```
STATWORK DSECT ,
SAVEAREA DS   18F                    Save area
         HZSADDCK MF=(L,WHZSADDCK)
STATWLEN EQU  *-STATWORK             Length of local storage area
*********************************************************************
*        Equates *
*********************************************************************
R0       EQU  0
R1       EQU  1
R2       EQU  2
R3       EQU  3
R4       EQU  4
R5       EQU  5
R6       EQU  6
R7       EQU  7
R8       EQU  8
R9       EQU  9
R10      EQU  10
R11      EQU  11
R12      EQU  12
R13      EQU  13
R14      EQU  14
R15      EQU  15
REDAC001 CSECT ,
         END ,
```

B.2 Message input for creating a message table

Here is an example of a message input and setup data set. We used the member HZSMSGNJ in SYS1.SAMPLIB to create the message table.

Example: B-2

```
//GABERT1 JOB  GABERT,NOTIFY=GABERT
//        SET  SYSPROC=SYS1.SBLSCLIO
//        SET  HZSMDSN=*
//        SET  HZSADSN=GABERT.ASM.SRC(REDMSG1)
//        SET  HZSSDSN=*
//*
//* ************************************************************
//* *                                                          *
//* * $MAC(HZSMSGNJ) COMP(SCHZS) PROD(HZS7720):                 *
//* *     HCHECKER SAMPLE MESSGAGE GENERATION JOB              *
//* *                                                          *
//* * PROPRIETARY STATEMENT:                                   *
//* *                                                          *
//* *     LICENSED MATERIALS - PROPERTY OF IBM                 *
//* *     5694-A01                                             *
//* *     (C) COPYRIGHT IBM CORP. 2005                         *
//* *     STATUS = HZS7720                                     *
//* *                                                          *
//* *  HZSMSGEN - Z/OS HEALTH CHECKER MESSAGE GENERATION       *
//* *  -------------                                           *
//* *                                                          *
//* *  FUNCTION - GENERATE A AN ASSEMBLER CSECT USING A        *
//* *             STRUCTURE MESSAGE SCRIPT                     *
//* *                                                          *
//* *                                                          *
//* *                                                          *
//* *  INSTRUCTIONS                                            *
//* *                                                          *
//* *  1. Customize the job card                               *
//* *                                                          *
//* *  2. Set variable SYSPROC to the name of the library      *
//* *     containing the message generation exec HZSMSGEN.     *
//* *                                                          *
//* *  3. Set variable HZSMDSN to the name of the library      *
//* *     containing the input message script. A fixed record  *
//* *     length of 80 is recommended for this data set.       *
//* *                                                          *
//* *                                                          *
//* *  4. Set variable HZSADSN to the name of the library      *
//* *     to contain the output assembler CSECT. A fixed record *
//* *     length of 80 is required for this data set.          *
//* *                                                          *
//* *     This job will fail with a RC = 20 when a DUMMY       *
//* *     data set is used.                                    *
//* *                                                          *
//* *  5. Set variable HZSSDSN to the name of a sequential     *
//* *     data set or a member of a PDS.  This data set        *
//* *     contains symbol <entity> statements.                 *
//* *                                                          *
//* *  NOTE: Errors are reported by the SYSTSPRT DD            *
//* *                                                          *
//* *                                                          *
//* * CHANGE-ACTIVITY:                                         *
```

```
//* *  $LO=HCHECK  HZS7720,040821, PDZJ: HEALTH CHECKER         *
//* *  $L1=ME01048 HZS7720,040901, PDZJ: PROLOGUE               *
//* *                                                           *
//* *                                                           *
//* ***********************************************************
//*
//CONV    EXEC PGM=IKJEFT01,REGION=32M,PARM='%HZSMSGEN'
//SYSTSPRT DD  SYSOUT=*
//SYSPROC  DD  DISP=SHR,DSN=&SYSPROC
//SYSTSIN  DD  DUMMY
//HZSADSN  DD  DISP=SHR,DSN=&HZSADSN
//HZSSDSN  DD  DATA
<!--                                                                -->
<!-- HZSSDSN is used by the check writer to define symbol           -->
<!-- definitions that are resovled during message generation        -->
<!--                                                                -->
<!ENTITY book1 "z/OS MVS System Commands">
<!ENTITY book2 "z/OS MVS Initialization and Tuning Guide">
<!ENTITY book3 "z/OS MVS Initialization and Tuning Reference">
/*
/*
/*
/* Input messages are written in SGML
/*
/*
/*
//HZSMDSN  DD  DATA
<lines props="copyright" id="REDBOOK">
*    This is my copyright information
</lines>
<msglist xreftext=REDMSG1>
<msg class=exception>
<msgnum xreftext=001>RED0010E</msgnum>
<msgtext>
The System detected that the user <mv xreftext=OUTLEN(8)>userid</mv>
is still logged on.
</msgtext>
<msgitem class="EXPLANATION">
<p>Although it is very late the user is still logged on to the
system. This might have a bad impact on the batch workload running
during the night</p>
</msgitem>
<msgitem class="SYSACT">
<p>The system knows how to handle this situation and continues
to run.</p>
</msgitem>
<msgitem class="ORESP"><p>
Call the user. Ask him to leave the office or cancel the user.</p>
</msgitem>
<msgitem class="SPRESP"><p>
Go for a coffee with the user.  </p>
</msgitem>
<msgitem class="PROBD"><p>n/a</p></msgitem>
<msgitem class="SOURCE"><p>REDBOOK checks</p>
</msgitem>
<msgitem class="REFDOC"><p>see &book1;</p>
</msgitem>
<msgitem class="AUTOMATION"><p>n/a</p>
</msgitem>
<msgitem class="MODULE">
```

```
<p>The name of the check routine and message table</p>
</msgitem>
<msgitem class="RCODE"><p>n/a</p>
</msgitem>
<msgitem class="DCODE"><p>n/a</p>
</msgitem>
</msg>
<msg class=information>
<msgnum xreftext=002>REDO010I</msgnum>
<msgtext>
Init Processing encountered
</msgtext>
<msgitem class="EXPLANATION"> <p>n/a</p>
</msgitem>
<msgitem class="SYSACT"> <p>n/a</p>
</msgitem>
<msgitem class="ORESP"> <p>n/a</p>
</msgitem>
<msgitem class="SPRESP"> <p>n/a</p>
</msgitem>
<msgitem class="PROBD"> <p>n/a</p>
</msgitem>
<msgitem class="SOURCE"><p> The owning product</p>
</msgitem>
<msgitem class="REFDOC"> <p>n/a</p>
</msgitem>
<msgitem class="AUTOMATION"> <p>n/a</p>
</msgitem>
<msgitem class="MODULE"><p>
The name of the check routine and message table</p>
</msgitem>
<msgitem class="RCODE"> <p>n/a</p>
</msgitem>
<msgitem class="DCODE"> <p>n/a</p>
</msgitem>
</msg>
<msg class=information>
<msgnum xreftext=003>REDO011I</msgnum>
<msgtext>
Cleanup Processing encountered
</msgtext>
<msgitem class="EXPLANATION"> <p>n/a</p>
</msgitem>
<msgitem class="SYSACT"> <p>n/a</p>
</msgitem>
<msgitem class="ORESP"> <p>n/a</p>
</msgitem>
<msgitem class="SPRESP"> <p>n/a</p>
</msgitem>
<msgitem class="PROBD"> <p>n/a</p>
</msgitem>
<msgitem class="SOURCE"><p> The owning product</p>
</msgitem>
<msgitem class="REFDOC"> <p>n/a</p>
</msgitem>
<msgitem class="AUTOMATION"> <p>n/a</p>
</msgitem>
<msgitem class="MODULE"><p>
The name of the check routine and message table</p>
</msgitem>
```

```
<msgitem class="RCODE"> <p>n/a</p>
</msgitem>
<msgitem class="DCODE"> <p>n/a</p>
</msgitem>
</msg>
<msg class=information>
<msgnum xreftext=004>REDO013I</msgnum>
<msgtext>
Check Processing encountered
</msgtext>
<msgitem class="EXPLANATION"> <p>n/a</p>
</msgitem>
<msgitem class="SYSACT"> <p>n/a</p>
</msgitem>
<msgitem class="ORESP"> <p>n/a</p>
</msgitem>
<msgitem class="SPRESP"> <p>n/a</p>
</msgitem>
<msgitem class="PROBD"> <p>n/a</p>
</msgitem>
<msgitem class="SOURCE"><p> The owning product</p>
</msgitem>
<msgitem class="REFDOC"> <p>n/a</p>
</msgitem>
<msgitem class="AUTOMATION"> <p>n/a</p>
</msgitem>
<msgitem class="MODULE"><p>
The name of the check routine and message table</p>
</msgitem>
<msgitem class="RCODE"> <p>n/a</p>
</msgitem>
<msgitem class="DCODE"> <p>n/a</p>
</msgitem>
</msg>
<msg class=information>
<msgnum xreftext=005>REDO014I</msgnum>
<msgtext>
Delete Processing encountered
</msgtext>
<msgitem class="EXPLANATION"> <p>n/a</p>
</msgitem>
<msgitem class="SYSACT"> <p>n/a</p>
</msgitem>
<msgitem class="ORESP"> <p>n/a</p>
</msgitem>
<msgitem class="SPRESP"> <p>n/a</p>
</msgitem>
<msgitem class="PROBD"> <p>n/a</p>
</msgitem>
<msgitem class="SOURCE"><p> The owning product</p>
</msgitem>
<msgitem class="REFDOC"> <p>n/a</p>
</msgitem>
<msgitem class="AUTOMATION"> <p>n/a</p>
</msgitem>
<msgitem class="MODULE"><p>
The name of the check routine and message table</p>
</msgitem>
<msgitem class="RCODE"> <p>n/a</p>
</msgitem>
```

```
<msgitem class="DCODE"> <p>n/a</p>
</msgitem>
</msg>
<msg class=information>
<msgnum xreftext=006>RED0100I</msgnum>
<msgtext>
The user <mv>userid</mv> is not logged on to the system.
</msgtext>
<msgitem class="EXPLANATION"> <p>n/a</p>
</msgitem>
<msgitem class="SYSACT"> <p>n/a</p>
</msgitem>
<msgitem class="ORESP"> <p>n/a</p>
</msgitem>
<msgitem class="SPRESP"> <p>n/a</p>
</msgitem>
<msgitem class="PROBD"> <p>n/a</p>
</msgitem>
<msgitem class="SOURCE"><p> The owning product</p>
</msgitem>
<msgitem class="REFDOC"> <p>n/a</p>
</msgitem>
<msgitem class="AUTOMATION"> <p>n/a</p>
</msgitem>
<msgitem class="MODULE"><p>
The name of the check routine and message table</p>
</msgitem>
<msgitem class="RCODE"> <p>n/a</p>
</msgitem>
<msgitem class="DCODE"> <p>n/a</p>
</msgitem>
</msg>
</msglist>
```

B.3 An example of a simple check routine

Here is an example of a simple check routine. If a particular user, provided as a parameter to the check routine, is logged on to the system, the check routine issues message RED0010E.

Example: B-3

```
REDCHK01 TITLE  'My Healthchecker Check'
REDCHK01 CSECT  ,
REDCHK01 AMODE  31
REDCHK01 RMODE  ANY
         USING  STATWORK,R10         Est work area addressability
         USING  STATMAIN,R12         Est base addressability
*********************************************************************
*
*        Housekeeping
*
*********************************************************************
STATMAIN DS     0H
         STM    R14,R12,12(R13)      Save callers registers
         LR     R12,R15              Set base register
         LR     R9,R1                Save parm
         STORAGE OBTAIN,LENGTH=STATWLEN,ADDR=(R10),LOC=ANY            C
                                     Obtain local work area
         LR     R0,R10               Zero the
         LA     R1,STATWLEN           work area
         SLR    R15,R15                that was
         MVCL   R0,R14                  just obtained

         ST     R13,SAVEAREA+4       Chain
         LA     R15,SAVEAREA          in
         ST     R15,8(R13)             new
         LR     R13,R15                 save area

*********************************************************************
*
*        Check why we're called
*
*********************************************************************
         LR     R1,R9                Restore parm address
         L      R9,0(,R1)            Get address of HZSPQE from parm
         USING  HZSPQE,R9            Est. addressability HZSPQE
         USING  HZSMGB,WHZSMGB       Est. addressability HZSMGB

         CLC    PQE_FUNCTION_CODE,=F'1'  Init Processing ?
         BE     FUNC_INIT

         CLC    PQE_FUNCTION_CODE,=F'2'  Check Processing ?
         BE     FUNC_CHECK

         CLC    PQE_FUNCTION_CODE,=F'3'  Cleanup Processing ?
         BE     FUNC_CLUP

         B      FUNC_DEL                 Delete Processing

FUNC_INIT   DS  0H
*********************************************************************
*
*        Initialize Processing
*
```

```
*********************************************************************
         MVC    MGB_ID,=F'2'         set id
         LA     R0,WHZSMGB           get address from parm area
         ST     R0,PHZSMGB             and store it into pointer
         HZSFMSG REQUEST=CHECKMSG,MGBADDR=PHZSMGB,MF=(E,WHZSFMSG)

         B      EXIT

FUNC_CHECK  DS  0H
*********************************************************************
*
*        Check Processing
*
*********************************************************************
         MVC    MGB_ID,=F'4'         set id
         LA     R0,WHZSMGB           get address from parm area
         ST     R0,PHZSMGB             and store it into pointer
         HZSFMSG REQUEST=CHECKMSG,MGBADDR=PHZSMGB,MF=(E,WHZSFMSG)

         CLC    PQE_PARMLEN,=H'0'    Parm found ?
         BNH    FUNC_CHECK_STOP      No, then leave and stop this check

*********************************************************************
*        Prepare variable part of message (userid)
*********************************************************************
         MVC    MGB_INSERT_CNT,=F'1' 1 insert
         LA     R0,V1_LENGTH         Get address of the variable array
         ST     R0,MGB_INSERTADDR     and store it in HZSMGB
         MVC    V1_LENGTH,=H'8'      Set length of variable text
         MVC    V1_VALUE(8),PQE_PARMAREA move variable

*********************************************************************
*        Is User logged on to the system ?
*********************************************************************
         L      R1,CVTPTR               R1 -> CVT
         L      R2,CVTASVT-CVT(,R1)     R2 -> ASVT
         L      R3,ASVTMAXU-ASVT(,R2)   R3 = Max. number of ASCBs
         LA     R4,ASVTENTY-ASVT(,R2)   R4 -> 1st. ASCB
ASIDLOOP DS     0H
         ICM    R1,15,0(R4)             Get ASCB
         BM     ASIDNEXT                ASCB invalid X'80', Skip
         ICM    R2,15,ASCBJBNS-ASCB(R1) Pointer to JOBNAME Field
         BZ     ASIDNEXT                nothing there, skip
         CLC    0(8,R2),PQE_PARMAREA    is it the right user ?
         BNE    ASIDNEXT                no, then skip

*********************************************************************
*        User logged on, write exception message and leave
*********************************************************************
         MVC    MGB_ID,=F'1'         set id
         LA     R0,WHZSMGB           get address of parm area
         ST     R0,PHZSMGB            and store it into pointer
         HZSFMSG REQUEST=CHECKMSG,MGBADDR=PHZSMGB,MF=(E,WHZSFMSG)
         B      EXIT                 leave

*********************************************************************
*        Loop control
*********************************************************************
ASIDNEXT DS     0H
         LA     R4,4(,R4)            point to next ASCB
```

```
         BCT    R3,ASIDLOOP          so many times...

*********************************************************************
*        User not logged on, write information message
*********************************************************************
         MVC    MGB_ID,=F'6'         set id
         LA     R0,WHZSMGB           get address of parm area
         ST     R0,PHZSMGB            and store it into pointer
         HZSFMSG REQUEST=CHECKMSG,MGBADDR=PHZSMGB,MF=(E,WHZSFMSG)
         B      EXIT                 leave

*********************************************************************
*        Stop this check if no parm was found
*********************************************************************
FUNC_CHECK_STOP ds 0H
         HZSFMSG REQUEST=STOP,REASON=BADPARM,MF=(E,WHZSFMSG)
         B      EXIT

FUNC_CLUP   DS  0H
*********************************************************************
*
*        Cleanup Processing
*
*********************************************************************
         MVC    MGB_ID,=F'3'         set id
         LA     R0,WHZSMGB           get address from parm area
         ST     R0,PHZSMGB             and store it into pointer
         HZSFMSG REQUEST=CHECKMSG,MGBADDR=PHZSMGB,MF=(E,WHZSFMSG)

         B      EXIT

FUNC_DEL    DS  0H
*********************************************************************
*
*        Delete Processing
*
*********************************************************************
         MVC    MGB_ID,=F'5'         set id
         LA     R0,WHZSMGB           get address from parm area
         ST     R0,PHZSMGB             and store it into pointer
         HZSFMSG REQUEST=CHECKMSG,MGBADDR=PHZSMGB,MF=(E,WHZSFMSG)

         B      EXIT

*********************************************************************
*
*        Finished, return to Health Checker
*
*********************************************************************
EXIT     DS     0H
         L      R13,SAVEAREA+4       Get callers save area
         STORAGE RELEASE,LENGTH=STATWLEN,ADDR=(R10)                    C
                                     Return local work area
         L      R14,12(R13)          Restore callers
         LM     R0,R12,20(R13)       registers
         SLR    R15,R15              Set a zero return code
         BR     R14                  Return to caller
         DROP   R10,R12

         LTORG ,
```

```
*********************************************************************
*
*        WORK AREA DSECT
*
*********************************************************************
STATWORK DSECT ,
SAVEAREA DS   18F                    SAVE AREA
         HZSFMSG MF=(L,WHZSFMSG)
WHZSMGB  DS   CL(HZSMGB_LEN)         control block for HZSFMSG macro
V1_LENGTH DS  H'0'
V1_VALUE  DS  CL256' '
PHZSMGB  DS   F'0'                   Pointer
STATWLEN EQU  *-STATWORK             LENGTH OF LOCAL STORAGE AREA

*********************************************************************
*
*        EQUATES *
*
*********************************************************************
R0       EQU  0
R1       EQU  1
R2       EQU  2
R3       EQU  3
R4       EQU  4
R5       EQU  5
R6       EQU  6
R7       EQU  7
R8       EQU  8
R9       EQU  9
R10      EQU  10
R11      EQU  11
R12      EQU  12
R13      EQU  13
R14      EQU  14
R15      EQU  15

*********************************************************************
*
*        DSECTS
*
*********************************************************************
         HZSPQE DSECT=YES
         HZSMGB DSECT=YES
         CVT   DSECT=YES
         IHAASCB LIST=YES
         IHAASVT LIST=YES
REDCHK01 CSECT ,
         END ,
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 497. Note that some of the documents referenced here may be available in softcopy only.

- *z/OS Version 1 Release 2 Implementation*, SG24-6235
- *z/OS Version 1 Release 3 and 4 Implementation*, SG24-6581
- *z/OS Version 1 Release 5 and Version 1 Release 6 Implementation*, SG24-6326

Other publications

These publications are also relevant as further information sources:

- *z/Architecture Principles of Operation,* SA22-7832
- *z/OS and z/OS.e Planning for Installation*, GA22-7504
- *z/OS Introduction and Release Guide*, GA22-7502
- *z/OS License Program Specifications*, GA22-7503
- *z/OS Migration*, GA22-7499
- *z/OS MVS Migration*, GA22-7580
- *z/OS MVS Planning: Operation*, SA22-7601
- *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- *z/OS MVS Using the Subsystem Interface*, SA22-7642
- *z/OS MVS Programming: Workload Management Services*, SA22-7619
- *z/OS MVS Planning: Global Resource Serialization*, SA22-7600
- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- *z/OS MVS System Management Facitlies*, SA22-7630
- *z/OS MVS System Commands*, SA22-7627
- *z/OS UNIX System Services Planning*, GA22-7800
- *z/OS Security Server RACF Command Language Reference*, SA22-7687
- *z/OS Security Server RACF Security Administrator’s Guide*, SA22-7683
- *z/OS MVS System Messages Volume 8 (IEF-IGD),* SA22-7638
- *ServerPac: Using the Installation Dialog*, SA22-7815
- *Tivoli OS/390 Installation: Migration Guide Version 1 Release 4*, SC31-8768
- *z/OS Network Job Entry (NJE) Formats and Protocols*, SA22-7539
- *IBM Health Checker for z/OS User’s Guide*, SA22-7994
- *Device Support Facilities User's Guide and Reference, Release 17*, GC35-0033

- *z/OS Communications Server IP User's Guide and Commands*, SC31-8780
- *z/OS MVS Interactive Problem Control System (IPCS) Customization*, SA22-7595
- *z/OS Communications Server New Function Summary*, GC31-8771
- *z/OS Distributed File Service zSeries File System Administration*, SC24-5989
- *Interactive System Productivity Facility (ISPF) User's Guide Volume 1*, SC34-4822
- *Interactive System Productivity Facility (ISPF) User's Guide Volume II*, SC34-4823
- *z/OS DFSMS: Using the New Functions,* SC26-7473
- *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426
- *z/OS Communications Server IP User's Guide and Commands*, SC31-8780
- *z/OS Communications Server IP Configuration Reference*, SC31-8776
- *z/OS MVS Programming Sysplex Services Reference*, SA22-7618
- *z/OS MVS Programming Sysplex Services Guide*, SA22-7617
- *IBM Ported Tools for z/OS User's Guide*, SA22-7985
- *IBM Ported Tools for z/OS Program Directory*, GI11-2847
- *z/OS JES2 Initialization and Tuning Reference,* SA22-7533
- *SMP/E User's Guide,* SA22-7773
- *z/OS System Codes,* SA22-7626
- *z/OS MVS System Messages Volume 8 (IEF-IGD),* SA22-7638
- *DFSMSdfp Diagnosis*, GY27-7618
- *z/OS V1.R7 MVS System Commands,* SA27-7627
- *DFSMSrmm Application Programming Interface,* SG26-7403
- *DFSMSrmm Implementation and Customization Guide,* SC26-7405
- *z/OS V1R7.0 UNIX System Services Command Reference*, SA22-7802
- *z/OS V1R7.0 JES2 Initialization and Tuning Reference*, SA22-7533
- *z/OS DFSMS: Using Data Sets*, SC26-7410
- *z/OS C/C++ Compiler and Run-Time Migration Guide for the Application Programmer*, GC09-4913

Online resources

These Web sites and URLs are also relevant as further information sources:

- Health Checker for z/OS

  `http://www.ibm.com/servers/eserver/zseries/zos/downloads/`

- Parallel Sysplex availability checklist

  `http://www.ibm.com/servers/eserver/zseries/pso/`

- ITSO Redbooks

  `http://www.redbooks.ibm.com/`

- zSeries Platform Test Report

  `http://www.ibm.com/servers/eserver/zseries/zos/integtst/`

- Washington System Center Flashes

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

- Enhanced Preventive Service Planning tool

  http://techsupport.services.ibm.com/390/psp_main.html

- SCRT is available as a download from the zSeries pricing web site :

  http://www.ibm.com./zseries/swprice

- Information about the IBM TotalStorage DS8000 and IBM TotalStorage DS6000, see the following website:

  http://www.ibm.com/servers/storage/disk.

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

z/OS Version 1 Release 7 Implementation

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

z/OS Version 1 Release 7 Implementation

Redbooks

ServerPac, SMP/E, Installation, BCP, JES2, ISPF

Health Checker, Security, Language Environment

zFS, z/OS UNIX, Comm Server, DFSMS

This IBM Redbook describes the new functions of z/OS Version 1 Release 7. These functions further strengthen the zSeries platform with enhancements designed to deliver increased availability of z/OS UNIX System Services (z/OS UNIX), to support new security standards, and to improve enterprise-wide workload management. Enhancements are made to the following areas:

- Application integration with enhancements to the zSeries File System (zFS)
- Enhanced security with Communications Server support for IP filtering, Internet Key Exchange (IKE), and Virtual Private network (VPN), without requiring the use of the Integrated Security Services Firewall Technologies
- Security Server (RACF) support for mixed-case passwords
- Improved availability with TCP/IP sysplex autonomics
- Greater scalability with support for sequential data sets larger than 65,535 tracks
- System Logger and XRC+ optimization of duplexing log data
- Greater ease of use with the IBM Health Checker for z/OS
- z/OS UNIX dynamic service activation and a new dynamic service activation function
- JES2 has checkpoint problem recovery and supports TCP/IP for NJE networking
- A new direct access device space management/Common VTOC Access Facility (DADSM/CVAF) device support address space starts during a system IPL
- IDCAMS FROMKEY/TOKEY support in REPRO MERGECAT is planned to help make it easier to move catalog entries from one catalog to another
- VSAM RLS 64-bit virtual can allow continued growth for applications using VSAM RLS
- DFSMShsm extends TTOC records to support more than 330,000 data sets per volume and support DFSMShsm journal data sets larger than 65,535 tracks
- DFSMSrmm supports data sets larger than 65,535 tracks for the journal, journal backup, and certain temporary data sets
- DFSMSdss exploits Format-1 Channel Command Words (CCWs) so that it can use storage obtained above the 16-MB line during EXCP processing
- Hardware Configuration Definition (HCD) Input Output Definition File (IODF) improvements

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6755-00 ISBN 0738494089