z/OS Version 1 Release 5 Implementation

BCP, JES3, JES2, SDSF, RMF, Communications Server, Consoles

Infoprint Server, ISPF, WLM, PSF for z/OS

z/OS UNIX, RACF, SMP/E, ServerPac

Paul Rogers
Patrick Bruinsma
Olivier Daurces
Robert Kohler
Meganen Naidoo
Miha Petric
Natabar Sahoo

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/OS Version 1 Release 5 Implementation

November 2004

SG24-6326-00

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

First Edition (November 2004)

This edition applies to Version 1, Release 5 of z/OS (5694-A01), 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®
ibm.com®
z/Architecture™
z/OS®
zSeries®
Advanced Function Presentation™
Advanced Function Printing™
AFP™
AIX®
BCOCA™
CICS®
DB2 Universal Database™
DB2®
DFS™
DFSMSdfp™
DFSMSdss™
DFSMShsm™
DFSMSrmm™
DFSORT™
DPI®
Enterprise Storage Server®
ESCON®
FICON®
Geographically Dispersed Parallel Sysplex™
GDPS®
HiperSockets™
HyperSwap™
Infoprint®
Intelligent Printer Data Stream™
IBM®
IMS™
IMS/ESA®
IP PrintWay™
IPDS™
Language Environment®
Lotus®
Multiprise®
MO:DCA™
MQSeries®
MVS™
MVS/ESA™
NetSpool™
NetView®
Open Class®
OS/2®
OS/390®
OS/400®
Parallel Sysplex®
Print Services Facility™
PrintWay™
PR/SM™
Redbooks™
Redbooks (logo) ™
Resource Link™
RACF®
RMF™
S/390®
Tivoli®
VTAM®
WebSphere®
Workplace™
1-2-3®

The following terms are trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered 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

z/OS® Version 1 Release 5 offers a number of enhancements that improve availability, scalability and performance, application flexibility, and ease of use. In this IBM® Redbook, we describe these functional enhancements and provide information to help you install, tailor, and configure this release.

After giving an overview of this release, we cover the enhancements made to the following components:

- ServerPac
- Base Control Program (BCP)
- JES3
- JES2
- SDSF
- Infoprint® Server
- ISPF
- Workload Manager (WLM)
- Console restructure
- RMF™
- SMP/E for z/OS and OS/390®
- UNIX® System Services (USS)
- z/OS Security Server RACF®
- PSF 3.4.0 for z/OS
- Communication Server for z/OS V1R5

We also provide RMF Performance Monitor metrics and describe the system trace entry created in the system trace table for the high virtual storage service IARV64.

This redbook is intended for systems programmers and administrators responsible for customizing, installing, and migrating to these newest levels of z/OS.

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, and z/OS UNIX. Before joining the ITSO 16 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.

Patrick Bruinsma is an Advisory IT Specialist working for IBM Global Services in the Netherlands. He has six years of experience on z/OS, DB2®, MQSeries®, Websphere MQ

Workflow, Blaze Advisor, CICS®, and UNIX System Services. He particpated in a previous ITSO residency writing about UNIX-related topics.

Olivier Daurces is an Advisory IT specialist working for IBM Technical Support in France. He has five years of experience in the z/OS field. His areas of expertise include RACF and Parallel Sysplex®.

Robert Kohler is a Certified Consulting Systems Products IT Specialist working for IBM US in Technical Sales Support - Americas Techline. He has more than 23 years of systems programming experience in mainframe environments on MVS™, OS/390, and z/OS platforms. His expertise covers a wide range of hardware and software products and he specializes in installation, implementation, migration, performance tuning, and capacity planning.

Meganen Naidoo is a Technical Architect working for Comparex Africa, the leading provider of competitive, innovative and practical business solutions, based in South Africa. He has more than 20 years of mainframe experience, working on VM, OS/390, z/OS, and Linux® system platforms. His areas of expertise include a variety of technical topics on z/OS, CICS, and Storage Management. He specializes in research and development, system installations and migrations, and problem determination and resolutions.

Miha Petric is a System programmer from Slovenia working as an IBM subcontractor. He has worked in the MVS field since 1978. His areas of expertise include MVS systems and subsystems. He is an IBM business partner for education and teaches IBM classes.

Natabar Sahoo is an Advisory IT Specialist working for IBM Singapore. He has been part of the Integrated Technology Services, zSeries® team since 1995. He has 22 years of experience in the IT field, including 14 years in Large Systems. He holds a degree in Electrical Engineering from University College of Engineering, Burla, Orissa, India. His areas of expertise include z/OS, Parallel Sysplex, WLM, TCPIP, RACF, problem diagnosis, teaching, system administration, and implementation and migration of z/OS and SW products.

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 Internet note 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 5 overview

In this chapter we describe the functional enhancements available in z/OS Version 1 Release 5. The changes we introduce include the following:

- In the base control program (BCP):
  - WLM dynamic application environment API
  - AutoPDAS (HyperSwap™)
  - Outage avoidance logger service tasks hang relief
  - Logstream data set deletion management
- In UNIX System Services:
  - BPXPRMxx creates mountpoints within the file system
  - BPXCOPY extensions for UNIX file to file copy
  - Sysplex verification shutdown versus hard wait
  - Sysplex remount
  - New symlink symbolics
- In Security server with RACF:
  - Dynamic templates
  - Multilevel security (MLS)
- In DFSMS:
  - RMM backup at any time
  - RMM duplicate VOLSER support
  - Re-index of online volumes
  - DFSMS enhancements

1.1 z/OS Version 1 Release 5

z/OS V1R5 offers a number of enhancements that provide improvements in availability, scalability and performance, application flexibility, and ease-of-use. z/OS V1R5 and all the features and Web deliverables available for this release run on the following IBM servers:

- zSeries z990 or equivalent
- zSeries z900 or equivalent
- zSeries z800 or equivalent
- S/390® Parallel Enterprise Servers — Generation 5 (G5) and Generation 6 (G6) models or equivalent

  > **Note:** Driver 26, licensed internal code V1.6.2, or later, is required on a G5 or G6 server to support architectural enhancements required by z/OS.

- Multiprise® 3000 Enterprise Server or equivalent

> **Attention:** zSeries file system (zFS) is the strategic UNIX Systems Services file system for z/OS. The Hierarchical File System (HFS) functionality has been stabilized. HFS is expected to continue shipping as part of the operating system and will be supported in accordance with the terms of a customer's applicable support agreement. IBM intends to continue enhancing zFS functionality, including RAS and performance capabilities, in future z/OS releases. All requirements for UNIX file services are expected to be addressed in the context of zFS only.

1.1.1 Functional enhancements with z/OS V1R5

The functional changes in z/OS V1R5 that may be of particular interest to system programmers are the following:

- BCP enhancements
  - WLM dynamic application environment API
  - AutoPDAS (HyperSwap)
  - Outage avoidance - logger service tasks hang relief
  - Logstream data set deletion management
  - System-managed Coupling Facility (CF) structure duplexing
  - Consoles enhancements feature
  - 64-bit shared virtual storage
- UNIX System Services enhancements
  - BPXPRMxx creates mountpoints within the file system
  - BPXCOPY extensions for z/OS UNIX file-to-file copy
  - Sysplex verification shutdown versus hard wait
  - Sysplex remount
  - New symlink symbolics
- z/OS Security server with RACF enhancements
  - Dynamic template
  - Multi-level security
- DFSMS enhancements
  - RMM backup at any time
  - RMM duplicate VOLSER support

- Re-index of online volumes
- SMS availability and usability enhancements

1.2 z/OS V1R5 BCP enhancements

The Base Control Program (BCP) provides essential operating system services. The BCP includes the I/O configuration program (IOCP), the workload manager (WLM), system management facilities (SMF), the z/OS UNIX System Services (z/OS UNIX) kernel, and support for Unicode. 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.

The BCP for z/OS V1R5 changes are briefly described in this section; complete details on the function and implementation are presented in later chapters.

1.2.1 WLM dynamic application environment API

Prior to z/OS V1R5 it was necessary to define one or multiple application environments through WLM's administrative application when using WLM's Queue and Server management capabilities. This application environment definition tells WLM how it can start server address spaces in order to manage the number of servers for the environment. This configuration step had to be done manually for various applications like DB2, scalable webserver, MQSeries Workflow, and Websphere Enterprise Edition.

In z/OS V1R5, this support addresses the customization and replication problems of application environments. There is a new service which allows an application to define and remove application environments as well as enhancements to allow multiple application environments dynamically. This eliminates the manual customization steps and reduces the chance of introducing configuration errors. This new support does the following:

- Enables dynamic definition of WLM application environments dynamically
- Eliminates traditional manual customization steps using the WLM administrative application
- Reduces the chance of configuration errors
- Simplifies installation, especially in WebSphere® applications

1.2.2 AutoPDAS (HyperSwap)

PPRC Dynamic Address Switching (P/DAS) provides the ability to redirect all application I/O from one PPRC volume to another PPRC volume with minimal application impact. P/DAS operations are based on the Peer-to-Peer Remote Copy (PPRC) functions of the 3990 Model 6 Storage Control, and can be used in shared-DASD environments. P/DAS allows application-transparent switching of I/O to support the following tasks:

- Planned outages (device or subsystem)
- Device migration
- Workload movement

HyperSwap is a combination of z/OS IOS code, DFSMS device support code, ESS u-code, and Geographically Dispersed Parallel Sysplex™ (GDPS®) code that will allow PPRC devices to be swapped (from the primary device to the secondary device) in parallel and non-disruptively to the applications. Contrary to today's Peer-to-Peer Dynamic Address Switching (P/DAS) continuous availability solution, HyperSwap was designed to minimize the time it takes to swap a single device pair.

HyperSwap Stage I support provided the functionality that allowed GDPS to swap, in parallel, many PPRC pair devices at a time, thus minimizing the overall time it takes to swap an installation's DASD devices. Initially, GDPS provided support for “planned outages” to occur, for example, to provide maintenance or migration of an installation's devices. The ultimate goal, however, is to allow GDPS to be used to provide “unplanned” swaps so that a complete installation site can be swapped to a secondary site during a disaster recovery scenario non-disruptively.

HyperSwap Stage II support will provide the system triggers necessary for GDPS to initiate a HyperSwap operation for “unplanned outages.” For these cases, HyperSwap may be able to provide continuous availability in the event of a logical subsystem control unit error or in a true disaster recovery scenario.

1.2.3 System Logger service tasks hang relief

This support in z/OS V1R5 is a follow-on from the Logger Monitor support introduced in z/OS V1R4, as well as through the PTFs for Logger APAR OW51854. Logger now monitors its allocation and HSM recall service tasks for delays and provides a mechanism using WTORs to interrupt these delayed requests. Logger handles the interruption as an error condition for the current request. However, by removing the delayed request, other log stream resource requests can then be processed. New monitoring messages, IXG271I and IXG272E (similar to IXG311I and IXG312E), can be issued if Logger detects that a delay of a service request is inhibiting other log stream resource requests.

Also being introduced in this support is new Logger message IXG314I, which indicates when Offload monitoring has been terminated on a system via an EXIT response to an IXG312E action message. Existing message IXG066I now only indicates when there is no Logger event monitoring in effect on the system. IXG066I is no longer issued following the EXIT response to an IXG312E message.

1.2.4 Logger data set deletion management

If the physical deletion housekeeping for offload data sets exceeds 60 seconds and other log stream requests are being delayed, Logger issues a new message IXG266I and quiesces the data set deletion activity to allow other log stream requests to be processed. There is no expected system management response to this action other than to check if any remaining data sets that might still need to be deleted are eventually deleted on subsequent log stream offloads.

In addition to managing the amount of data sets that are deleted, the initial log stream offload data set will be deleted at the end of the DEFINE LOGSTREAM operation. This support definitely minimizes bottlenecks in a sysplex to avoid system contentions.

1.2.5 System-Managed Coupling Facility (CF) Structure Duplexing

System-Managed CF Structure Duplexing was available exclusively through a readiness review program since April 8, 2003. System-Managed CF Structure Duplexing is now generally available and the readiness review program has ended. Customers who are already participating in the readiness review program continue to receive support for their System-Managed CF Structure Duplexing implementation. Customers who want to begin using System-Managed CF Structure Duplexing may do so at any time and are encouraged to take advantage of a self-assessment questionnaire available on Resource Link™.

1.2.6 Consoles Enhancements feature

The z/OS V1R4 Consoles Enhancements feature was the first deliverable in the new zSeries console strategy, which is intended to enhance the operator messaging architecture of z/OS. The feature focuses on minimizing the possibility of outages due to exhaustion of system resources used for messaging. This capability was an optional feature in z/OS V1R4; as of z/OS V1R5, the function is rolled into the base product. Several tasks that were formerly best practices are now required, and some functions are no longer relevant.

The overall objective of the z/OS Consoles Enhancements feature is to improve system availability by enhancing the capacity and reliability of message delivery. To accomplish this, major changes to the message production and consumption flow help reduce the possibility of bottlenecks which can cause a backlog of undelivered messages.

In a future release of z/OS, IBM plans to eliminate the one-byte console ID interface. With the advent of four-byte console IDs (in MVS SP V4.1.0), customers and vendors have been encouraged to migrate away from the use of one-byte interfaces. Details of the one-byte console ID interface elimination are planned to be communicated in a future z/OS announcement. To help prepare for the removal of this interface, tools have been provided in the z/OS Consoles Enhancements feature that will identify uses of the one-byte console ID interface in the environment.

More information about the Consoles Enhancements and the one-byte console ID interface can be found in *z/OS MVS Planning: Operations*, SA22-7601.

1.2.7 64-bit shared virtual storage

z/OS V1R5 delivers the 64-bit shared memory support to enable middleware for sharing a large amount of 64-bit virtual storage among multiple address spaces. This is a significant capacity enhancement for relieving shared virtual storage constraints.

1.3 z/OS V1R5 UNIX System Services enhancements

Changes to UNIX System Services for z/OS V1R5 are briefly described here; more details on the function and implementation are presented in later chapters in this redbook.

1.3.1 Create mountpoints within the file system

BPXPRMxx USS parmlib member BPXPRMxx supports a new keyword on the existing ROOT and MOUNT statements named the MKDIR keyword. This allows one or more directories to be created in the mounted file system as part of mount processing during USS initialization. The MKDIR keyword is an optional keyword, and there is no default regardless of whether it is specified or not.

In addition, the SETOMVS SYNTAXCHECK= parmlib member is enhanced to check the MVS catalog for the existence of the HFS or zFS data set names listed on each ROOT and MOUNT statement in the specified parmlib member. This helps to ensure that the MOUNTs will succeed.

1.3.2 BPXCOPY extensions for z/OS UNIX

The BPXCOPY program provides the ability to copy a sequential data set or a partitioned data set or a PDSE member into a hierarchical file system (HFS) for use in the UNIX System Services environment. The functionality of BPXCOPY is extended in z/OS V1R5 to support the copying of an HFS file to an HFS file. Currently, BPXCOPY supports copying data from a data set to a file in a directory. After copying the file, it also sets any specified attributes and path mode, and creates links and symlinks for the file. There is a need for a method to copy a file from one directory into another directory, and also set its associated attributes and path mode, and create links and symlinks.

OPUT provides the ability to copy a file to another file, but lacks the other tasks that can be done as part of the copy. The syntax for specifying an HFS file on a DD statement already exists since it is already used for the output file. There is no need for new syntax on the input DD statement. Instead, the description of the input DD statement for BPXCOPY will need to include information about specifying HFS files. A DD statement allocates a data set or file and sets up a ddname. The input ddname can specify an MVS data set (either a sequential data set or a member of a partitioned data set or PDSE) or the input ddname can be the full pathname of the HFS file. When you invoke BPXCOPY from JCL, you must use SYSUT1 as the input ddname. If BPXCOPY is invoked from LINK, XCTL, or ATTACH, a TSO/E **CALL** command with the asis option, or by a call after a LOAD, you can specify an alternative ddname.

1.3.3 Sysplex shutdown verification

The current OMVS kernel and file system initialization sequence remains unchanged up to the point where the LFS version is verified. The OMVS mainline initialization/termination routine, BPXINIT, invokes the file system subcomponent to initialize once sufficient OMVS kernel resources are built, but prior to actually building the INIT process. The file system initialization builds its required infrastructure and initializes all physical file systems as specified in the associated BPXPRMxx parmlib members. Prior to actually mounting the ROOT file system, the check for LFS Version compatibility is made.

1.3.4 Sysplex remount

Currently, in a shared HFS environment, in order to change the mount mode (read-only or read-write) of a mounted file system, it must be unmounted, and then mounted again in the desired mode. But in order to unmount a file system, all file systems mounted under it must be unmounted first. Remount allows changing the mount mode of a mounted file system without the requirement to explicitly unmount and mount it again. Remount is currently supported in non-sysplex mode only. Remount is now e supported in a sysplex as long as all sysplex members are at the z/OS V1R5 level. If one or more sysplex members are downlevel, the following message appears:

```
errno EINVAL and errnojr JrNotSupInSysplex
```

Remount is an option on unmount. The syntax for remount in a sysplex is the same as it currently is for non-sysplex. The externals for remount are the same for sysplex except that the remount will no longer be rejected as long as all sysplex members are at least at the z/OS V1R5 level. The remount can be requested from the server or from any of the client systems. When remount successfully completes, the mode of the file system will be changed on all systems in the sysplex.

1.3.5 New symlink symbolics

This new support provides the capability to mount different file systems at a logical mount point that resolves to a different path name on different systems.

On a pathname lookup, when a component of the pathname is a symlink that begins with one of two new identifiers, the MVS static symbols in the template are replaced with the resolved substitution text using the ASASYMBM system service. For pathname lookup, $SYSSYMR/

results in a relative pathname, that is, the lookup proceeds from its current position in the pathname. $SYSSYMA/ results in an absolute pathname, that is, the lookup starts over at the root.

1.4 z/OS Security Server RACF enhancements

Changes to the Security Server RACF for z/OS V1R5 are briefly described here; more details on the function and implementation are presented in later chapters in this redbook.

1.4.1 Dynamic templates

This function supports the ability to refresh RACF templates without IPLing. In previous releases, when changes are made to the templates (via service or product upgrade) and the customers do not update them before an IPL, they have to do an additional IPL to fix the problem because the RACF database and the templates are out of sync.

In z/OS V1R5, this line item allows template changes to be done dynamically, eliminates the need for customer intervention, and avoids system errors and outages.

1.4.2 Multilevel security

z/OS is the first and only IBM operating system to provide multilevel security (MLS) support. z/OS on zSeries is a great platform for organizations and agencies that want to take advantage of the benefits of MLS.

Multilevel security addresses government requirements for highly secure data which can be shared between agencies on demand. New security features in DB2 V8 and z/OS V1R5 enable customers to have a single repository of data which can be accessed by different agencies, by people with different need-to-know authority. This access is managed at the row/column level in DB2 to provide the granularity that is required.

A multilevel security system has two primary goals. First, the controls are intended to prevent unauthorized individuals from accessing information at a higher classification than their authorization. Second, the controls are intended to prevent individuals from declassifying information. Multilevel security function will allow customers more stringent access control to resources than is provided by user permissions.

Earlier in OS/390, labeling was provided and OS/390 received B1 security certification. In z/OS V1R5, multilevel security will extend the labeled security protection of z/OS to include TCP/IP and UNIX System Services, and provide enhancements to Security Server, JES2, SDSF, and others. This will enable z/OS to meet the stringent requirements for multilevel security.

DB2 V8 UDB for z/OS

DB2 UDB for z/OS V8 provides the support for multilevel security. Each row of a DB2 database can now contain a DB2 security label. This label is checked against the security label of the requester. Only rows containing matching security labels can be accessed by the requester. The security of a row or column is maintained at a Database Administration or Security Administration level. An application developer need not worry about adding security sensitive code to a Query or Insert.

1.5 z/OS V1R5 DFSMS enhancements

Changes to the DFSMS for z/OS V1R5 are briefly described here; more details on the function and implementation are presented in later chapters in this redbook.

1.5.1 DFSORT™

Memory object sorting is a new DFSORT capability that uses a memory object on 64-bit real architecture to reduce I/O processing, elapsed time, execute channel programs (EXCPs), and channel usage for selected sort applications. A memory object is a data area in virtual storage that is allocated above the bar and backed by central storage. With memory object sorting, a memory object can be used exclusively, or along with disk space, for temporary storage of records. DFSORT uses memory object sorting automatically when it offers better performance than hipersorting or dataspace sorting.

1.5.2 DFSMSrmm™ backup at any time

Prior to this support, if inventory management was already running, you could not start a backup, or vice versa.

In z/OS V1R5, backup can now run at the same time as inventory management or vice versa. When backup runs at the same time, the inventory management processing tolerates the backup and waits until updates can be made to the control data set (CDS). Only CDS updates may have to wait since processing can continue for volumes and data sets that need no updates until the point an update is required.

If non-intrusive backup (concurrent) is used, CDS updates do not have to wait, and journal threshold can start backup even if inventory management is running. There is the option to Backup and Clear Journal without CDS backup; you will need to adjust RESTORE jobs to use additional journal backups.

Message EDG0123D can be issued twice at startup time, once for backup in progress and once for inventory management.

1.5.3 DFSMSrmm duplicate VOLSER support

Prior to this support, duplicate volumes were processed outside of DFSMSrmm management by using ignore processing via EDGUX100.

In z/OS V1R5, for private physical volumes you can define the VOL1 label parameter in addition to the unique volser. DFSMSrmm open processing has been updated to detect duplicate volumes and use the correct CDS volume information to manage the volume. It is possible to continue to ignore duplicate volumes but it is recommended to add them to the DFSMSrmm CDS and manage them. Conversion extract records are updated to include the VOL1 label volser and EDGCNVT converts this new field into the volume record.

This new support does not have to be enabled: you can use ADDVOLUME or CHANGEVOLUME command with VOL1(volser) to define duplicate volser.

Note: An NL tape cannot be a duplicate volume. Such a volume can be defined using any unique value for the volser and no special processing is required by DFSMSrmm to distinguish the volume from another identical volume.

1.5.4 Re-index online volumes

In z/OS V1R5, DFSMS provides the support which allows the re-indexing of a volume which is online to more than one system in a shared volume environment. This change eliminates the need to vary a volume offline to all shared systems prior to rebuild a VTOC index.

1.5.5 SMS availability and usability enhancements

There are five separate DFSMS enhancements made available in z/OS V1R5, as follows:

GDS reclaim processing

In prior releases, GDS reclaim processing caused generation data set (GDS) data overlay or corruption due to system failure. With this new enhancement, it allows an installation to turn off automatic GDS reclaim processing by specifying in the IGDSMSxx parmlib member a new keyword, as follows:

```
GDS_RECLAIM {YES| NO}
```

If NO is specified, manual means must be used to either delete the generation, rename it, or roll it in.

Use the **SET SMS=xx** command to change the IGDSMSxx member being used or use the **SETSMS GDS_RECLAIM {YES|NO}** command to change the value.

> **Note:** When using the command, the changed value is not retained across an IPL. Installations that have different levels of DFSMS or different settings should not share GDSes across all of the systems.

Save ACDS as the SCDS

This enhancement enables users to save ACDS as the SCDS, which eliminates the extra work of recreating the SCDS when the source SCDS and its backup are lost. The following new command is added in this support:

```
 SETSMS SAVESCDS(scds_dsname)
```

DFSMS will verify the 'scds_dsname' is not the active ACDS or COMMDS.

High threshold messages

In prior releases, no alert is issued to users when available space in a pool storage group is above its high allocation threshold. With this enhancement, you now receive a high-allocation-threshold-exceeded message in the printed log when a pool storage group has exceeded its high allocation threshold after a data set is successfully allocated to it. This message can be the signal, for you as the storage administrator or for an automated program, to perform space management. The cumulative space in a storage group includes all the volumes that are online and enabled or quiesced to DFSMS. Messages are not issued until utilized space falls below 80% of the high threshold or if it exceeds the high threshold again.

Multi-tiered storage groups

The best volume from any of the supplied storage group will currently be selected for allocation. With the multi-tiered groups function enabled, users are now able to specify a storage group order for pool storage groups. A new field is added to the storage class definition panel, as follows:

```
'Multi-Tiered SG' {YES | NO}
```

The process honors storage group sequence order specified in ACS storage group selection routines. The volumes that are defined in the first storage group are eligible to be primary volume candidates.

End of volume (EOV) failure messages

This enhancement ensures that DFSMS EOV generates messages to the JOBLOG and hardcopy log regardless of whether the caller requested the messages or not.

1.6 z/OS Communications Server - Network management

Most z/OS application workloads depend on reliable network communications to meet business and end-user performance objectives. As a result, many users rely on network management and monitoring applications to track utilization of critical z/OS network resources and to detect disruptions in z/OS network communications.

In z/OS V1R5, the ability to perform these network management and monitoring functions is significantly enhanced by the introduction of several new programming interfaces provided by z/OS Communications Server. These new interfaces allow applications to efficiently obtain detailed statistics and information related to native TCP/IP workloads and SNA workloads using Enterprise Extender (EE) and High Performance Routing (HPR).

1.7 IBM Infoprint Server

Infoprint Server is an optional feature of z/OS that uses z/OS UNIX System Services. This feature is the basis for a total print serving solution for the z/OS environment. It lets you consolidate your print workload from many servers onto a central z/OS print server. Infoprint Server delivers improved efficiency and lower overall printing cost with the flexibility for high-volume, high-speed printing from anywhere in the network. With Infoprint Server, you can reduce the overall cost of printing while improving manageability, data retrievability, and usability.

z/OS V1R5 provides the following new functions:

- Infoprint Central
- IP PrintWay™ extended mode
- E-mail JCL parameters
- NetSpool™ enhancements
- Security enhancements for Infoprint Central
- Common message log
- IBM Infoprint XT Extender

1.7.1 Infoprint Central

Continuing the IBM commitment to help lower customers' overall cost of distributed print operations, Infoprint Server includes a new component called Infoprint Central. Infoprint Central is a Web-based GUI for managing print jobs and printers throughout the enterprise from anywhere in the enterprise using a Web browser. Intended primarily for help desk operators, it lets users query the status of jobs and printers, see job and printer messages, stop and start printers, move jobs from one printer to another, cancel or hold jobs, and many other functions. Infoprint Central can use integrated z/OS security services so that users can be authorized to perform only certain tasks, or to perform tasks only on designated devices.

1.7.2 IP PrintWay extended mode

Infoprint Central is backed by a new architecture in the component that delivers print or e-mail output to printers, servers or users over TCP/IP or Internet Printing Protocol (IPP). IP PrintWay extended mode uses the Sysout Application Programming Interface (SAPI) to access print jobs and job information from the JES spool. The advantage of this change can be higher availability and throughput, more flexibility for handling print-related tasks, and scalability of Infoprint Server for very large distributed print environments.

1.7.3 Common message log

A new common message log helps to improve productivity of help desk operators for print problem diagnosis and resolution, thus helping to increase system availability and user satisfaction. Messages can easily be accessed from Infoprint Central for a particular job or printer. A utility is also provided that enables an administrator to see all messages for a particular time period, for example. Messages can be retained for a user-specified period of time, including messages for “historical” jobs which have already been processed by the system.

1.8 Migrating applications, compilers, and libraries

Changes that affect applications, compilers, and libraries with z/OS V1R5 are briefly described here; more details on the function and implementation are presented in later chapters in this redbook.

1.8.1 C/C++ enhancements

The C/C++ compiler feature introduces several new enhancements including the following:

- Support for 64-bit compiles

  The compiler has been enhanced to generate z/Architecture™ instructions which include utilizing the 64-bit general purpose registers. This new 64-bit support will enable C and C++ developers to recompile existing 32-bit C/C++ applications into 64-bit code and to compile new 64-bit C/C++ code. The WARN64 compiler option will help developers detect possible portability errors when moving code from 32-bit to 64-bit. The LP64 compiler option can be used to identify compile-time problems when moving code to the 64-bit virtual environment. Object code is not generated in this release.

- Performance enhancements

  A new higher optimization level, OPTIMIZE(3), provides the compiler's highest and most aggressive level of optimization. OPTIMIZE(3) is recommended when the desire for run-time improvement outweighs the concern for minimizing compilation resources. Profile-directed feedback, invoked by the IPA (PDF) suboption, can be used to collect execution profile information on an application and this information is then used to further tune compiler optimizations near conditional branches and in frequently executed code sections. Additional options and pragmas are introduced to help the developer to improve their application performance including Loop Unrolling option and pragmas, additional ARCH/TUNE options, and new built-in functions.

- DB2 preprocessor integration

  The C/C++ compiler has been enhanced to integrate the functionality of the DB2 precompiler. A new SQL compiler option enables the compiler to process embedded SQL statements.

- DEBUG option

  The DEBUG option is introduced to generate debug information based on the Debug with Arbitrary Record Format (DWARF) Version 3 debugging information format, which was developed by the UNIX International Programming Languages Special Interest Group (SIG) and is an industry standard format. The compiler is now capable of generating two different formats of debug information including the existing In-Store-Debug (ISD) format information.

1.8.2 C/C++ IBM Open Class® Library

The base element C/C++ IBM Open Class Library is removed from z/OS V1V5. It includes development support for the Application Support Class and Collection Class libraries, which is withdrawn in z/OS V1R5. For information about migrating, see IBM Open Class Library Transition Guide.

Note: IBM will standardize on the Standard C++ Library, including the Standard Template Library (STL) and other features of the ISO C++ 1998 Standard.

Run-time support for these libraries is available in the new base element Run-Time Library Extensions. This new base element is an extension of the run-time support provided by the Language Environment® element. Specifically, it includes:

- UNIX System Laboratories (USL) I/O Stream Library and USL Complex Mathematics Library, previously included in the C/C++ IBM Open Class Library element.
- IBM Open Class DLLs, previously included in the C/C++ IBM Open Class Library element.
- Common Debug Architecture (CDA) libraries and utilities, which are new in z/OS V1V5.

Support withdrawn: The application development support (that is, the headers, source, sidedecks, objects, and samples from the Application Support Class and Collection Class libraries) is withdrawn from the C/C++ IBM Open Class Library (IOC) in z/OS V1V5. Applications that use these IOC libraries cannot be compiled nor linked using z/OS V1V5. Run-time support for the execution of existing applications which use IOC libraries is provided with z/OS V1R5, but is planned to be removed in a future release.

1.8.3 C/C++ with Debug Tool

As of z/OS V1Vr, optional feature C/C++ with Debug Tool is no longer in z/OS. For debugging tools, see the following Web site:

```
http://www-3.ibm.com/software/awdtools/debugtool/
```

1.9 Migration, coexistence, and planning

z/OS and z/OS.e use the same operating system software (their code is identical). Upon IPL, custom parameters invoke an operating environment that is comparable to z/OS in all aspects of operation, service, management, reporting, and zSeries hardware functionality. No new skills are required for z/OS.e.

Install coexistence and fallback PTFs on your z/OS V1R2, z/OS VR.3 or z/OS V1R4 systems to allow those systems to coexist with z/OS V1R5 systems during your migration, and allow backout from z/OS V1R5 to your current z/OS releases if necessary.

1.9.1 Changes in base elements and priced features

Following are the changes in base elements and pricing for z/OS V1R5 and z/OS.e V1R5, as shown in Figure 1-1 on page 14.

Cryptographic Services

Cryptography is the transformation of data to conceal its meaning. In z/OS V1R5 and z/OS.e V1R5, the base element Cryptographic Services, introduced in OS/390 V2R7, provides the following base cryptographic functions:

- Data secrecy
- Data integrity
- Personal identification
- Digital signatures
- Management of cryptographic keys

  Keys as long as 56 bits are supported by this base element. Keys longer than 56 bits are supported by the related optional feature z/OS Security Level 3 (see Figure 1-2 on page 15).

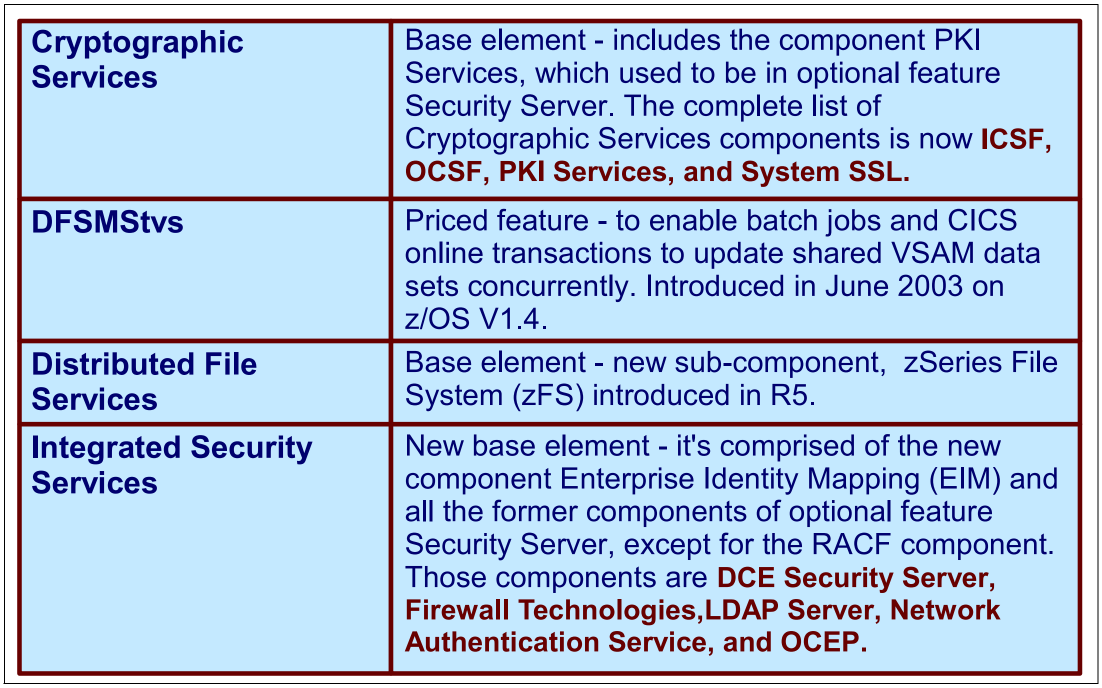
Cryptographic Services consists of the components shown in Figure 1-1 on page 14; the following are changed in z/OS V1R5 and z/OS.e V1R5:

- Integrated Cryptographic Service Facility (ICSF)
- Public Key Infrastructure (PKI) Services

> **Note:** Prior to z/OS V1R5, this component was in the optional feature Security Server, although it was licensed with the base operating system and could be used without ordering or enabling Security Server. This component uses RACF, the OCSF component of base element Cryptographic Services, and the ICSF component of base element Cryptographic Services for encryption.

Integrated Security Services

Prior to z/OS V1R5, the five moved components (DCE Security Server, Firewall Technologies, LDAP Server, Network Authentication Service, and OCEP), although packaged with the priced feature Security Server, were unpriced (that is, were licensed with the base operating system and could be used without ordering or enabling Security Server). Now that these five components are in the base element, Integrated Security Services, the packaging lines up with the pricing.



Cryptographic Services	Base element - includes the component PKI Services, which used to be in optional feature Security Server. The complete list of Cryptographic Services components is now **ICSF, OCSF, PKI Services, and System SSL.**
DFSMStvs	Priced feature - to enable batch jobs and CICS online transactions to update shared VSAM data sets concurrently. Introduced in June 2003 on z/OS V1.4.
Distributed File Services	Base element - new sub-component, zSeries File System (zFS) introduced in R5.
Integrated Security Services	New base element - it's comprised of the new component Enterprise Identity Mapping (EIM) and all the former components of optional feature Security Server, except for the RACF component. Those components are **DCE Security Server, Firewall Technologies,LDAP Server, Network Authentication Service, and OCEP.**

Figure 1-1 New base elements and priced features in z/OS V1R5

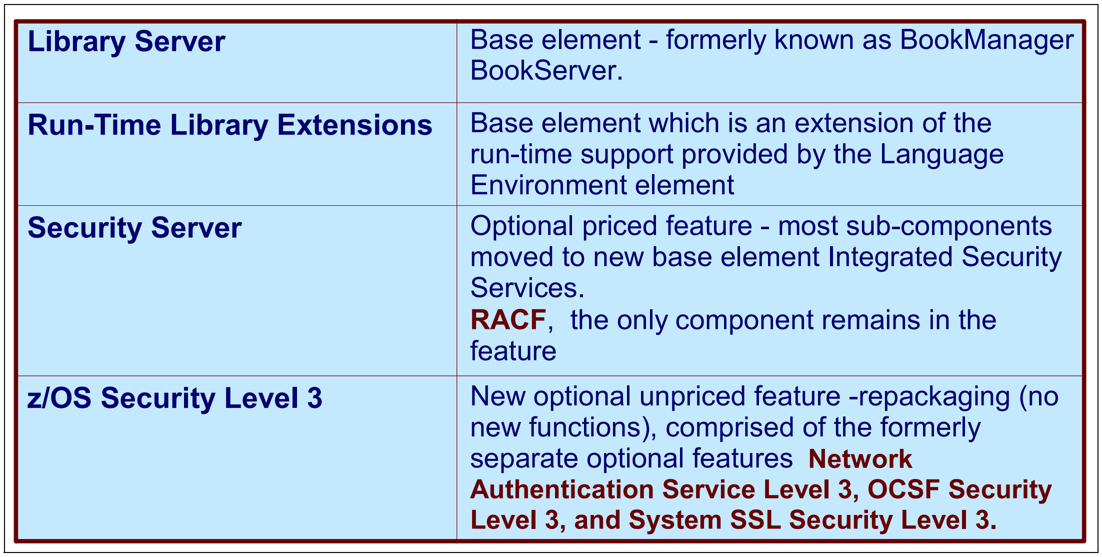
Be aware of the following changes as well:

Security Server

This optional feature had seven components but now has only one: RACF. The PKI Services component moved to the Cryptographic Services base element. The other five components moved to new base element Integrated Security Services. They are DCE Security Server, Firewall Technologies, LDAP Server, Network Authentication Service, and OCEP.

z/OS Security Level 3

This former optional feature is now a component of the new optional feature z/OS Security Level 3. Also, it has a new name: Network Authentication Service Level 3 (“Security Server” was removed from the name).

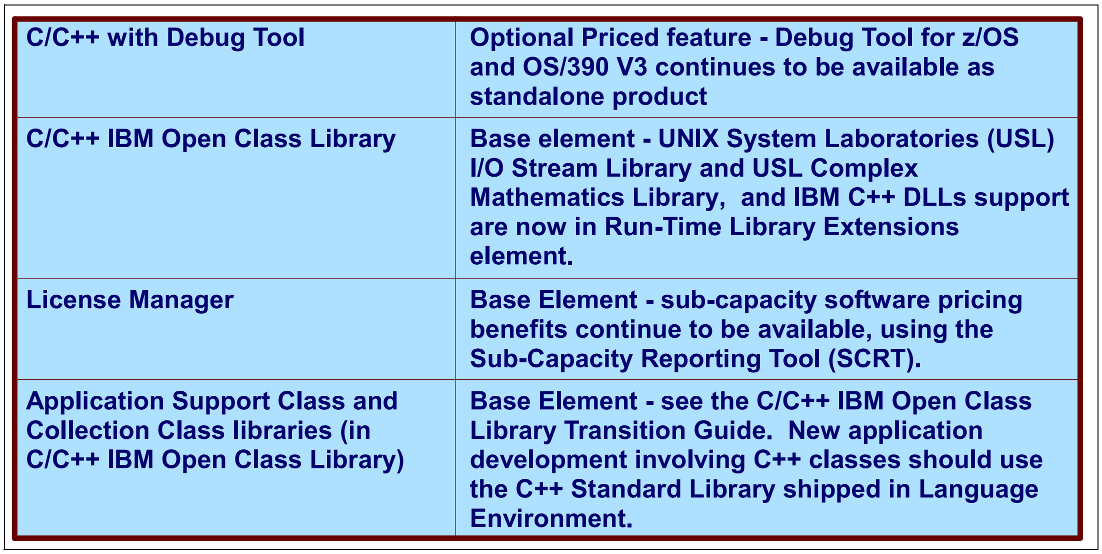


Library Server	Base element - formerly known as BookManager BookServer.
Run-Time Library Extensions	Base element which is an extension of the run-time support provided by the Language Environment element
Security Server	Optional priced feature - most sub-components moved to new base element Integrated Security Services. **RACF**, the only component remains in the feature
z/OS Security Level 3	New optional unpriced feature -repackaging (no new functions), comprised of the formerly separate optional features **Network Authentication Service Level 3, OCSF Security Level 3, and System SSL Security Level 3.**

Figure 1-2 Base elements and pricing changes

1.9.2 Functions withdrawn with z/OS V1R5

The elements identified in Figure 1-3 have been removed from the system in z/OS V1R5.



C/C++ with Debug Tool	**Optional Priced feature - Debug Tool for z/OS and OS/390 V3 continues to be available as standalone product**
C/C++ IBM Open Class Library	**Base element - UNIX System Laboratories (USL) I/O Stream Library and USL Complex Mathematics Library, and IBM C++ DLLs support are now in Run-Time Library Extensions element.**
License Manager	**Base Element - sub-capacity software pricing benefits continue to be available, using the Sub-Capacity Reporting Tool (SCRT).**
Application Support Class and Collection Class libraries (in C/C++ IBM Open Class Library)	**Base Element - see the C/C++ IBM Open Class Library Transition Guide. New application development involving C++ classes should use the C++ Standard Library shipped in Language Environment.**

Figure 1-3 Elements that have been withdrawn in z/OS V1R5

2

ServerPac enhancements for z/OS V1R5

Your ServerPac order includes the CustomPac installation dialog, an Interactive System Productivity Facility (ISPF) dialog that you use to install the order.

This chapter presents some important new concepts that you should understand before using the dialog to install your order.

This chapter also describes the enhancements and changes that have been incorporated into ServerPac. The following topics are discussed:

- Automatic block sizes
- zFS support
- Large volume support
- Pre-RECEIVEd service tapes
- Various panel changes to improve usability
- Reference information

2.1 z/OS V1R5 ServerPac

There are a number of significant changes to ServerPac in this release. This chapter describes many of these changes. If you install z/OS V1R5 with a ServerPac, an installation dialog job is provided to perform this action. If you install z/OS V1R5 with a CBPDO, instructions to perform this action are provided in the z/OS Program Directory.

For additional information on the changes to the ServerPac, see the following documentation:

- *ServerPac: Using The Installation Dialog,* SA22-7815
- *ServerPac: Installing Your Order* (shipped with your order)
- *z/OS and z/OS.e Planning for Installation,* GA22-7504

2.1.1 Tivoli® NetView® and System Automation ordering considerations

Parts of two stand-alone products are included in the z/OS V1R5 msys for Operations base element:

- Tivoli NetView for OS/390 V1R4 (5697-B82)
- System Automation for OS/390 V2R2 (5645-006)

If you already have these stand-alone products installed (at the V1R4 and V2R2 levels respectively), you can install z/OS V1R5 (including msys for Operations) in the same SMP/E zone as the stand-alone products.

Note: In this case, it is recommended that you order these stand-alone products in your z/OS V1R5 ServerPac. They will be installed in the same zones as z/OS V1R5, and will not require separate maintenance and duplication of service work.

2.1.2 ServerPac enhancements

This chapter describes the following enhancements in this ServerPac release:

- Automatic block sizes
- CustomPac dialog panel changes
- zFS support
- Large volume support
- Service tapes
- Usability enhancements

2.2 Automatic block sizes

In previous ServerPac releases, there were default block sizes set in each ServerPac order. Although you could make changes and save them, and they would be carried forward if you merged configurations, this work had to be repeated for each new data set.

In the z/OS V1R5 ServerPac, the best block size for every data set is set automatically. For some data sets, the block sizes are set to specific values. For others, a block size must be determined based on the device characteristics, the record format, and the record length. DFSMSdfp system-determined block sizes (SDB) is used to do this. As a result, you will not know what the block size for many data sets will be in advance. Therefore, except for data sets you define yourself, the block sizes are no longer displayed in the dialog, and you also cannot change them.

2.2.1 Block size enhancements

The following enhancements were made:

- The ALLOCDS job uses DCB BLKSIZE=0 for most data sets (system-determined block size) and 32760 for RECFM=U data sets.
- Data set space is still allocated using block allocation units, but they are *only* used for space allocation. An exception has been made for unreblockable data sets.
- The dialog space displays are changed to show *tracks* or percentage of the volume size, depending on the displayed panel. Figure 2-1 on page 20 shows data set space in tracks. Figure 2-3 on page 21 shows displayed space as a percentage of the volume size.
- Data set bytes, block sizes are *not* shown in the dialogs. An exception has been made for user-defined data sets.
- The `CH`ange `OPTIBLOCK` command is also removed because block sizes are automatically set.

Block size allocations

The data set block sizes are set according to the following rules:

BLKSIZE 0 BLKSIZE is always zero for:

- RECFM=F (Fixed)
- RECFM=FB (Fixed Blocked)
- RECFM=V (Variable, including VB, VS and VBS)

BLKSIZE 32760 Always 32760 for undefined load libraries

RECFM=U

Font data sets Always 12288 for font data sets

Products They will be set as specified by product owner, as follows:

- RECFM=D (Direct, including DB and DBS)
- Data sets that are marked as “unreblockable”

UADS For UADS, the right block size differs by installation, so it will model this from your existing data set. The UADS block size is optimized when the average user ID entry is not split into multiple members.

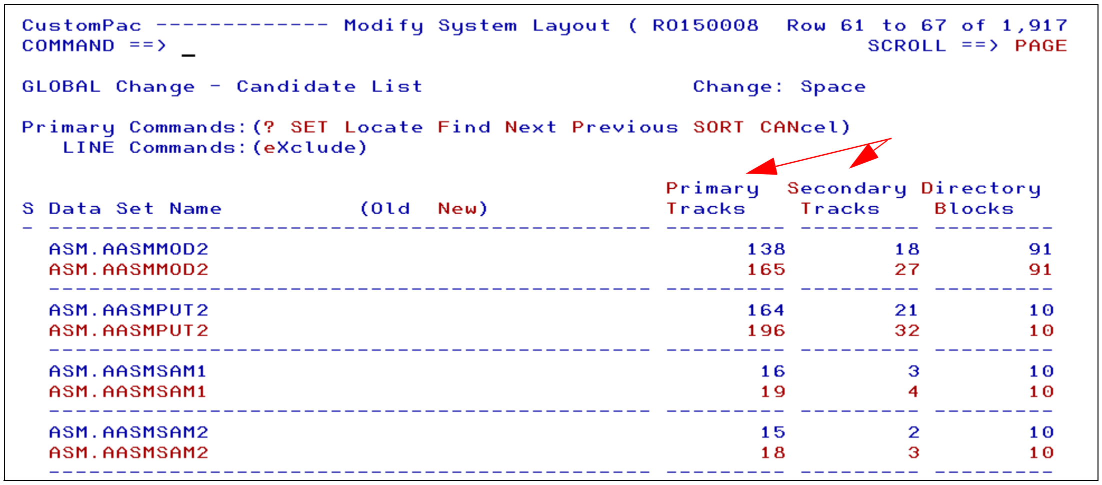
2.3 CustomPac dialog panel changes

The three big changes to panels are:

- BLKSIZE is not displayed except for user-defined data sets.
- Space is shown in tracks or as a percentage of the volume size.
- Panel displays now show “Data Set Type” instead of the previously displayed “DSNTYPE” or “DSORG.” Figure 2-4 on page 22 shows the data set type field.

2.3.1 Global change - Candidate List panel

Figure 2-1 on page 20 is the panel you get when you issue the change command `CH` on a data set list panel. In this case, the change space command `CH S` was issued. The space display fields used to display the block size and number of blocks. Now, they show primary tracks, secondary tracks, and directory blocks.



```
CustomPac ------------- Modify System Layout ( R0150008  Row 61 to 67 of 1,917
COMMAND ==> _                                                 SCROLL ==> PAGE

GLOBAL Change - Candidate List                       Change: Space

Primary Commands:(? SET Locate Find Next Previous SORT CANcel)
   LINE Commands:(eXclude)

                                                Primary  Secondary Directory
S Data Set Name          (Old  New)             Tracks     Tracks    Blocks
- ---------------------------------------------  --------- --------- ---------
  ASM.AASMMOD2                                        138        18        91
  ASM.AASMMOD2                                        165        27        91
  ---------------------------------------------  --------- --------- ---------
  ASM.AASMPUT2                                        164        21        10
  ASM.AASMPUT2                                        196        32        10
  ---------------------------------------------  --------- --------- ---------
  ASM.AASMSAM1                                         16         3        10
  ASM.AASMSAM1                                         19         4        10
  ---------------------------------------------  --------- --------- ---------
  ASM.AASMSAM2                                         15         2        10
  ASM.AASMSAM2                                         18         3        10
  ---------------------------------------------  --------- --------- ---------
```

Figure 2-1 Data set space shown in tracks

2.3.2 SUMMARY of Physical Volumes panel

On the SUMMARY of Physical Volumes panel, shown in Figure 2-2 on page 21, volume space in cylinders is now shown. The new fields on this panel display the following information:

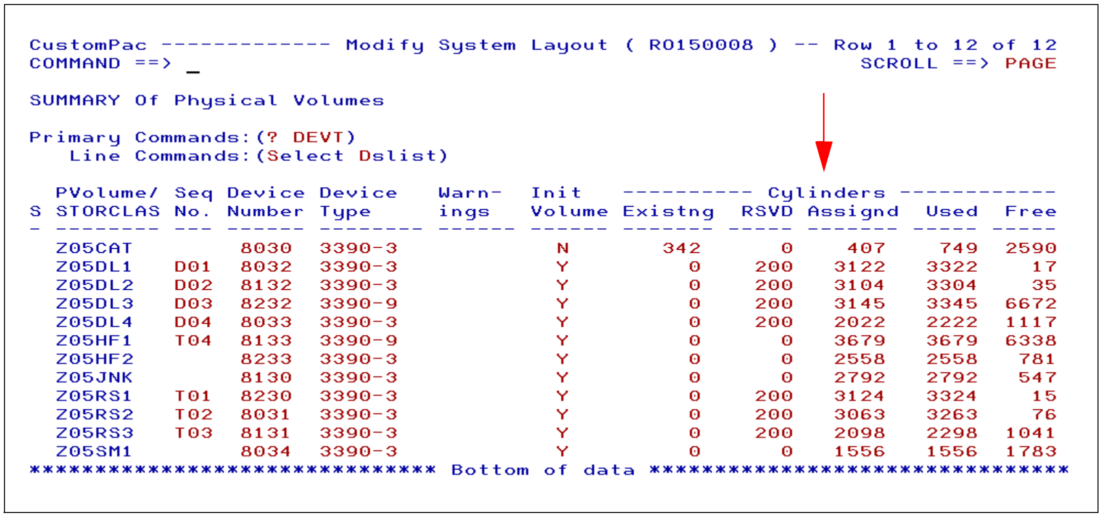
Existing This column used to say “Y” or “N” and now it shows the number of cylinders used by existing data on the volume.

Assignd The total number of cylinders used by the data sets you assigned to the volume in the dialog.

Used The sum of existing data, reserved space, and assigned space.

Device Number The “CCUU” heading is changed to “Device Number” for consistent use of terms.

On the action character line for each volume, shown in Figure 2-2 on page 21, you can now use Select (s), which displays the panel Display and Change Volume Attributes, where you change the attributes of the physical volume that you selected.



```
CustomPac ------------- Modify System Layout ( R0150008 ) -- Row 1 to 12 of 12
COMMAND ==> _                                                  SCROLL ==> PAGE

SUMMARY Of Physical Volumes

Primary Commands:(? DEVT)
   Line Commands:(Select Dslist)

  PVolume/ Seq Device Device    Warn- Init   ---------- Cylinders ------------
S STORCLAS No. Number Type      ings  Volume Existng RSVD Assignd  Used  Free
- -------- --- ------ --------- ------ ------ ------- ----- ------- ----- -----
  Z05CAT        8030  3390-3              N       342     0    407   749  2590
  Z05DL1   D01  8032  3390-3              Y         0   200   3122  3322    17
  Z05DL2   D02  8132  3390-3              Y         0   200   3104  3304    35
  Z05DL3   D03  8232  3390-9              Y         0   200   3145  3345  6672
  Z05DL4   D04  8033  3390-3              Y         0   200   2022  2222  1117
  Z05HF1   T04  8133  3390-9              Y         0     0   3679  3679  6338
  Z05HF2        8233  3390-3              Y         0     0   2558  2558   781
  Z05JNK        8130  3390-3              Y         0     0   2792  2792   547
  Z05RS1   T01  8230  3390-3              Y         0   200   3124  3324    15
  Z05RS2   T02  8031  3390-3              Y         0   200   3063  3263    76
  Z05RS3   T03  8131  3390-3              Y         0   200   2098  2298  1041
  Z05SM1        8034  3390-3              Y         0     0   1556  1556  1783
******************************* Bottom of data ********************************
```

Figure 2-2 Volume space shown in cylinders

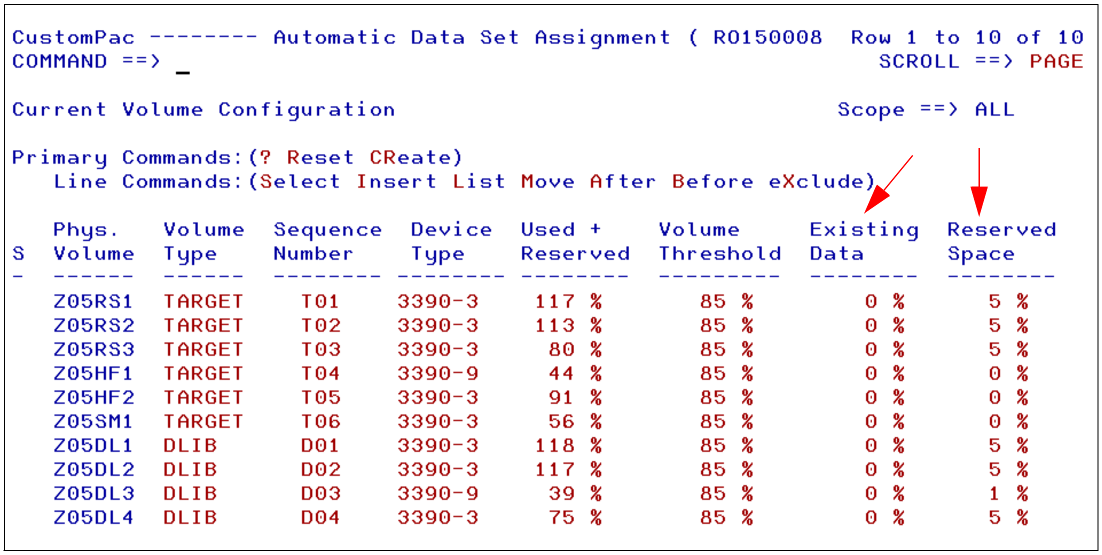
2.3.3 Current Volume Configuration panel

In Figure 2-3, the Existing Data and Reserved Space fields are changed so that they show the percentage of the volume used by data sets assigned to the volume, existing data on the volume, and space reserved on the volume. The fields are changed as follows:

Existing Data If the field Init Volume is set to "Y" for a volume, as shown in Figure 2-2, then the value of Existing Data displayed for that volume is 0%.

Used + Reserved This field shows the sum of assigned space, reserved space, and the space taken up by existing data sets on the volume.

Because these two columns are changed to display percentages rather than alphabetic characters, the columns are rearranged for better readability.



```
CustomPac -------- Automatic Data Set Assignment ( R0150008  Row 1 to 10 of 10
COMMAND ==> _                                                    SCROLL ==> PAGE

Current Volume Configuration                                   Scope ==> ALL

Primary Commands:(? Reset CReate)
   Line Commands:(Select Insert List Move After Before eXclude)

   Phys.  Volume  Sequence  Device    Used +    Volume     Existing  Reserved
S  Volume Type    Number     Type     Reserved  Threshold  Data      Space
-  ------ ------  --------  --------  --------  ---------  --------  --------
   Z05RS1 TARGET    T01     3390-3    117 %        85 %       0 %       5 %
   Z05RS2 TARGET    T02     3390-3    113 %        85 %       0 %       5 %
   Z05RS3 TARGET    T03     3390-3     80 %        85 %       0 %       5 %
   Z05HF1 TARGET    T04     3390-9     44 %        85 %       0 %       0 %
   Z05HF2 TARGET    T05     3390-3     91 %        85 %       0 %       0 %
   Z05SM1 TARGET    T06     3390-3     56 %        85 %       0 %       0 %
   Z05DL1 DLIB      D01     3390-3    118 %        85 %       0 %       5 %
   Z05DL2 DLIB      D02     3390-3    117 %        85 %       0 %       5 %
   Z05DL3 DLIB      D03     3390-9     39 %        85 %       0 %       1 %
   Z05DL4 DLIB      D04     3390-3     75 %        85 %       0 %       5 %
```

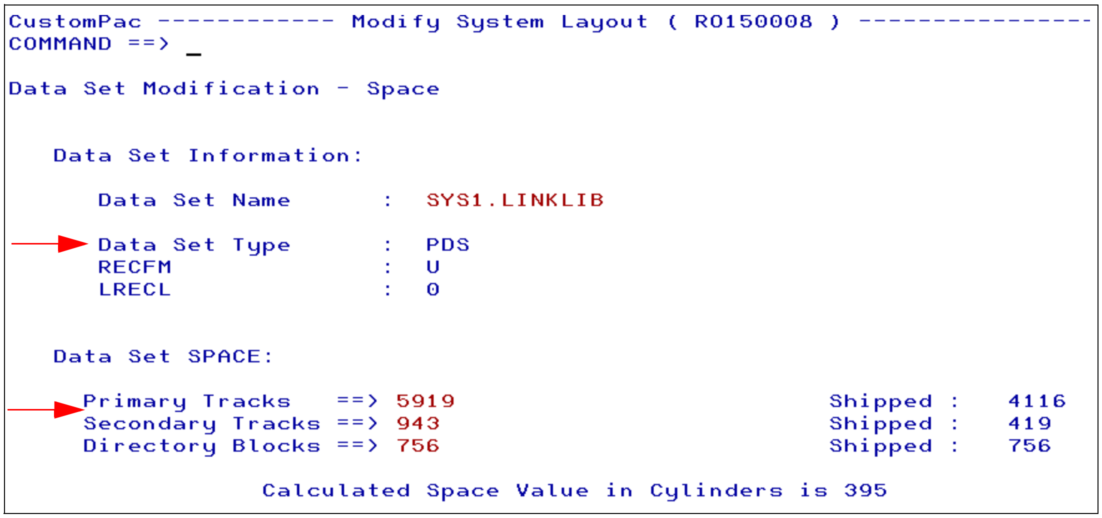
Figure 2-3 Space shown as a percentage of volume size

2.3.4 Data Set Modification - Space panel

When you select a data set from a data set list to modify its space, the panel displayed by the dialog is shown in Figure 2-4. The changes make the space displays consistent and more readable. For example, sequential data sets formerly shown as “PS” are now shown as “SEQ” and VSAM data sets are shown as “VSAM.”

On the bottom of the panel, you can see that space is displayed and specified in tracks, and is also shown by calculating and displaying the space in cylinders.

In the middle, the dialog shows the data set type rather than the DSORG. So you see “PDS” rather than “PO,” “PDSE” instead of “PO-E,” and so on.



```
CustomPac ------------ Modify System Layout ( R0150008 ) ----------------
COMMAND ==> _

Data Set Modification - Space


   Data Set Information:

      Data Set Name       :   SYS1.LINKLIB

      Data Set Type       :   PDS
      RECFM               :   U
      LRECL               :   0


   Data Set SPACE:

     Primary Tracks    ==> 5919                          Shipped :    4116
     Secondary Tracks  ==> 943                           Shipped :    419
     Directory Blocks  ==> 756                           Shipped :    756

                 Calculated Space Value in Cylinders is 395
```

Figure 2-4 The Data Set Type field

2.4 zFS support overview

zSeries File System (zFS) was introduced in z/OS V1R2, support was also rolled back to OS/390 R10 and z/OS R1. All file system data sets are currently shipped as HFS data sets. However, zFS can be used instead of HFS for all file systems except the root file system. Installations may convert their HFS root to a zFS root, it is just not supported by service.

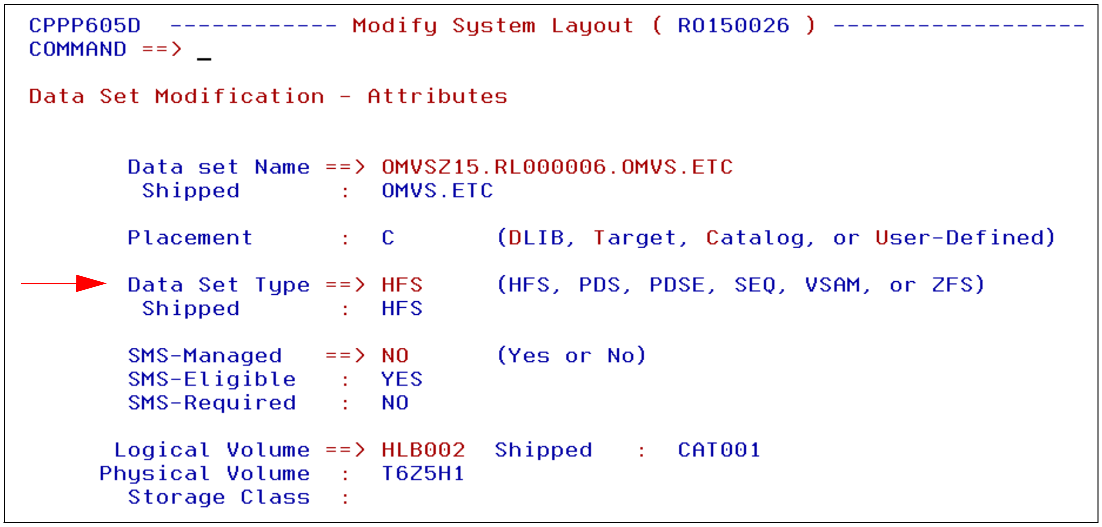
2.4.1 Data Set Modification - Attributes panel

zFS has been added as a data set type in the dialog along with PDS, PDSE, SEQ, and VSAM, making displays of data set types consistent with commonly-used terms.

The dialog previously displayed DSNTYPE and DSORG. The installation dialog panels now allow you to specify either HFS or ZFS for a file system data set type as follows:

- The data set attributes panel, as shown in Figure 2-5 on page 23.
- Using the CHange DSNTYPE command, as shown in Figure 2-6 on page 23.

In both cases, the switchable attribute (displayed in the View and Change option of the Modify System Layout) governs the ability to change from HFS to zFS.

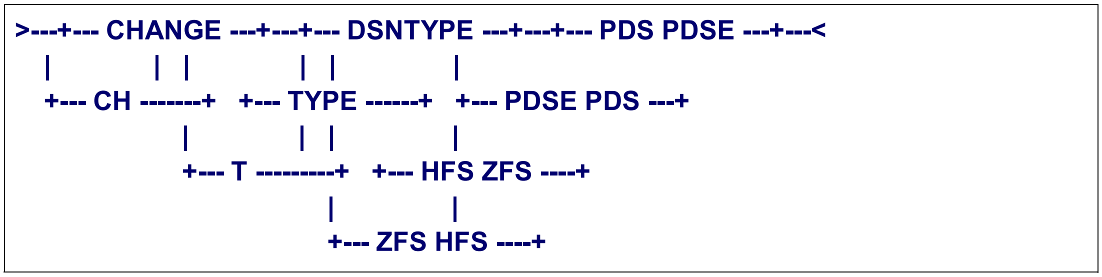


```
CPPP605D   ------------ Modify System Layout ( RO150026 ) -----------------
COMMAND ==> _

Data Set Modification - Attributes


       Data set Name ==> OMVSZ15.RL000006.OMVS.ETC
        Shipped        :  OMVS.ETC

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

       Data Set Type ==> HFS      (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 2-5 Set the data set type panel

CH TYPE command

The change type command **CH TYPE** now allows you to change between HFS and zFS. All UNIX System Services file system data sets are shipped in ServerPac as HFS data sets. Those eligible to be zFS data sets can be changed to zFSs on the panels, and by using the new operands of the **CH** command.



```
>---+--- CHANGE ---+---+--- DSNTYPE ---+---+--- PDS PDSE ---+---<
    |              |   |               |   |
    +--- CH -------+   +--- TYPE ------+   +--- PDSE PDS ---+
                       |               |   |
                       +--- T ---------+   +--- HFS ZFS ----+
                                           |
                                           +--- ZFS HFS ----+
```

Figure 2-6 CH command to set the data set type

The command syntax is as follows:

CH TYPE source target

source The current data set type (DSNTYPE)

target The new data set type

You can change PDS data sets to PDSEs (or change PDSEs to PDSs):

- **CH TYPE PDS PDSE**
- **CH TYPE PDSE PDS**

You can also change HFS data sets to zFS data sets, or change zFS data sets to HFS data sets:

- **CH TYPE HFS ZFS**
- **CH TYPE ZFS HFS**

Note: zFS can be used for all file systems except the root file system, which must be HFS.

2.4.2 Using zFS in the installation jobs

Using zFS file systems in place of HFS is transparent to applications. The same utilities (like pax) work for both, but there are some differences in the installation jobs:

- An HFS data set is a non-VSAM data set, while a zFS lives in a VSAM Linear Data Set (LDS). Using zFS in place of an HFS is transparent to applications. The same utilities work for both.
- zFS data sets must be preformatted with the IOEAGFMT program. A format step will be added to the ALLOCDS job when zFS is used in the configuration.
- The RESTFS job is not added unconditionally because there is a significant amount of free ECSA required to start the ZFS address space. When ZFS is selected the RESTFS job will:
  - Put the appropriate TYPE on MOUNT parameters in the BPXPRMFS parmlib member.
  - Add a FILESYSTYPE for zFS to BPXPRMFS. See “New started task procedure and parmlib changes” on page 24.
- The RACF setup for zFS will be done unconditionally, based on the assumption that zFS will eventually be used. The RACFDRV and the RACFTGT jobs are updated to:
  - Add a group (DFSGRP) for the DFS™ setup
  - Add a user ID (ZFS) for the zFS address space
- Because different programs are used by the system to process HFS and zFS files, a different FILESYSTYPE statement is required in BPXPRMxx.
- Additional security system setup is needed to use zFS.
- zFS has a VSAM catalog entry, which cannot be indirect.

Note: CICS, DB2, and IMS™ ServerPacs assume that any necessary zFS setup has been done. Also, additional ECSA is required to use zFS data sets. Before using a zFS, you should review your virtual storage map and ensure there is at least 70MB of ECSA available in addition to the ECSA normally required to run your system.

New started task procedure and parmlib changes

The ZFS procedure is copied from IOE.SIOEPROC to the CPAC.PROCLIB data set in z/OS orders. You can then add this procedure to your JES procedure library concatenation.

The following statement is added to BPXPRMFS to enable zFS file system support:

```
FILESYSTEM TYPE(ZFS) ENTRYPOINT(IOEFSCM) ASNAME(ZFS)
```

Where:

`ZFS` Identifies an entry for the ZFS type of file system.

`IOEFSCM` Identifies the program that knows how to process the file system type.

`ZFS` Specifies the address space name and the procedure name to start. This name must correspond to the user ID created in RACFDRV and RACFTGT and the name of the procedure created in CPAC.PROCLIB.

RACF job definitions

Figure 2-7 on page 25 is a RACF profile example to start the z/OS address space. Note that there is no STARTED profile for ZFS. This is because ServerPac uses a STARTED profile that assigns the started task a user ID that is the same as the started task name. If you do not use such a profile, you need to create a STARTED profile for the started task, or specify the user ID for the task in the started procedures table (ICHRIN03).

```
In RACFDRV and RACFTGT jobs:

  ADDGROUP         +
  DFSGRP         +
  SUPGROUP(SYS1) +
  OMVS(GID(2))

ADDUSER          +
  ZFS            +
  DFLTGRP(STC)   +
  OMVS(HOME(/opt/dfslocal/home/dfscntl) UID(0))

CONNECT          +
  ZFS            +
  GROUP(DFSGRP)  +
  AUTH(CREATE)
```

Figure 2-7 RACF profile example

> **Note:** The ADDUSER user ID (ZFS) must match the ASNAME in the FILESYSTYPE statement in BPXPRMxx.

2.5 Large volume support

The Enterprise Storage Server® (ESS) 2105 DASD and DFSMS introduced large volume support. You can configure IBM 3390 Model 9 devices on the ESS to have up to 32,760 cylinders per device. The system now allows volume sizes as large as 24 GB and 32,760 cylinders, allowing you to access a larger amount of data. All functions currently supported for 3390 Model 9 continue to be supported. A large volume is a volume that is larger than a “real” 3390-9 volume. The volume capacity would be almost 28 GB. The DFSMS components enhanced for this function are DFSMSdfp, DFSMSdss™, and DFSMShsm™.

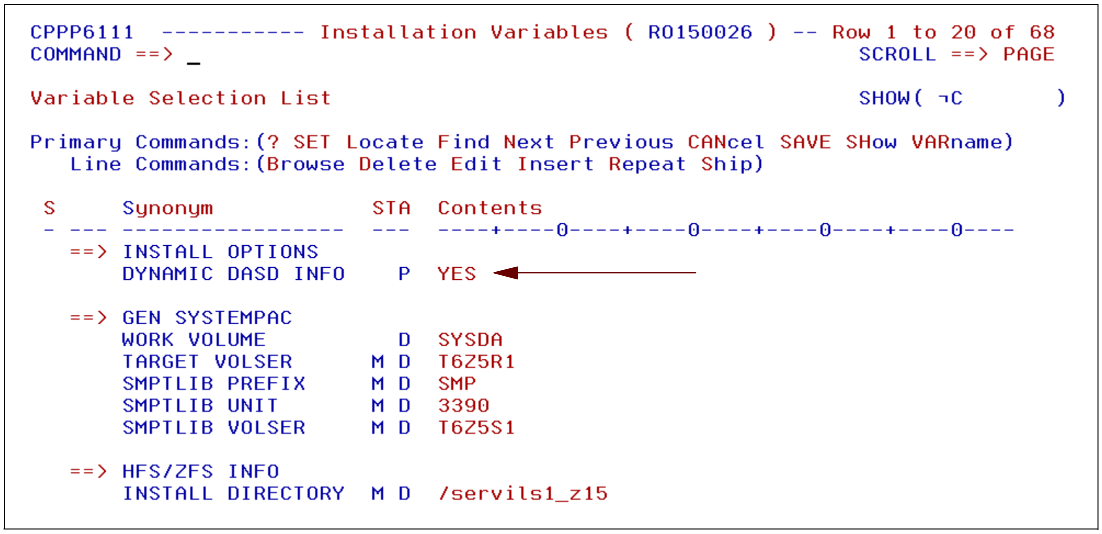
With the large volume support feature of DFSMS you can define any 3390 volume size ranging from 1024 cylinders to 32760 cylinders on 2105 DASD.

> **Note:** When you use large volume storage, it is important to use parallel access volumes (PAV) to achieve the highest level of I/O performance from Enterprise Storage Server (ESS). If you define large volume sizes on storage that does not use PAV, it takes longer for the system to access these volumes. You should define an adequate number of alias addresses on ESS to get better throughput.

2.5.1 Dynamic DASD information

When you INIT a DASD volume to be a large volume, and if you do it in advance and the volume is online, the system can be asked by ServerPac all about the device. But ServerPac needs to know whether to ask. So the new DYNAMIC DASD INFO variable is used for this purpose.

The new variable called DYNAMIC DASD INFO is shown under the INSTALL OPTIONS heading in the Installation Variables List panel in Figure 2-8 on page 26.



```
CPPP6111  ----------- Installation Variables ( R0150026 ) -- Row 1 to 20 of 68
COMMAND ==> _                                                 SCROLL ==> PAGE

Variable Selection List                                       SHOW( ¬C       )

Primary Commands:(? SET Locate Find Next Previous CANcel SAVE SHow VARname)
   Line Commands:(Browse Delete Edit Insert Repeat Ship)

 S     Synonym            STA  Contents
 - --- ------------------ ---  ----+----0----+----0----+----0----+----0----
   ==> INSTALL OPTIONS
       DYNAMIC DASD INFO    P  YES  <----------------

   ==> GEN SYSTEMPAC
       WORK VOLUME          D  SYSDA
       TARGET VOLSER      M D  T6Z5R1
       SMPTLIB PREFIX     M D  SMP
       SMPTLIB UNIT       M D  3390
       SMPTLIB VOLSER     M D  T6Z5S1

   ==> HFS/ZFS INFO
       INSTALL DIRECTORY  M D  /servils1_z15
```

Figure 2-8 Installation variables showing new DYNAMIC DASD INFO

Note: The **DYNAMIC DASD INFO** default is **YES**

Setting DYNAMIC DASD INFO

- If DYNAMIC DASD INFO is set to YES:

  For online volumes, the dialog will get the device information needed to define the device by itself in most instances.

  – If it succeeds, the Device Type field is updated automatically.

  – If it can't get the device information, you can tell the dialog to ignore the problem on a volume-by-volume basis, but then you have to define the device manually.

- If you change the setting to NO, the dialog acts like it did before.

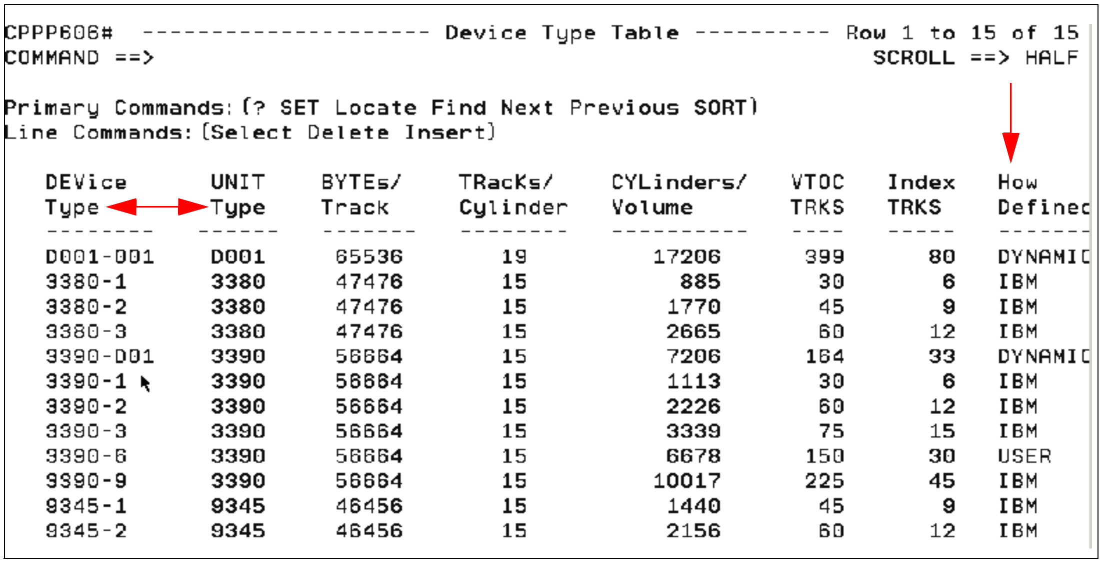
Setting the device type, model and unit

One of the objectives of this function was to make it possible to use any DASD with the dialog, as follows:

- If the device type matches an existing one, then the same device type is used. (Example: a 3390 is always a “3390”)
- If the device model matches an existing one, then the same device model is used. (Example: a 3390-9 is always a “3390-9”)
- Otherwise, the dialog generates a device type (Dnnn), model (Dnn) or both, as shown in Figure 2-9 on page 27. (Example: 3390-D001 or D001-D01 for dynamically defined or nonstandard devices.)
- When a device type is generated by a dialog, you will need to specify the UNIT to be used for allocation.
- When a device model is generated by the dialog, you do not need to do anything else.

Device type table

Figure 2-9 shows the direct access storage devices (DASDs) that are defined to the dialog, including user-defined devices if any. You cannot edit or delete the IBM-supplied devices in this list. You can, however, add entries for real or emulated devices, and edit or delete those entries. Use the insert action character (**i**) to add a user-defined device; Figure 2-11 on page 28 shows the result when you have added in the field information.



```
CPPP606#  --------------------- Device Type Table ---------- Row 1 to 15 of 15
COMMAND ==>                                                    SCROLL ==> HALF

Primary Commands:(? SET Locate Find Next Previous SORT)
Line Commands:(Select Delete Insert)

   DEVice      UNIT    BYTEs/   TRacKs/   CYLinders/  VTOC   Index   How
   Type        Type    Track    Cylinder  Volume      TRKS   TRKS    Definec
   --------   ------   -------  --------  ----------  ----   -----   -------
   D001-D01    D001     65536      19        17206     399      80   DYNAMIC
   3380-1      3380     47476      15          885      30       6   IBM
   3380-2      3380     47476      15         1770      45       9   IBM
   3380-3      3380     47476      15         2665      60      12   IBM
   3390-D01    3390     56664      15         7206     184      33   DYNAMIC
   3390-1      3390     56664      15         1113      30       6   IBM
   3390-2      3390     56664      15         2226      60      12   IBM
   3390-3      3390     56664      15         3339      75      15   IBM
   3390-6      3390     56664      15         6678     150      30   USER
   3390-9      3390     56664      15        10017     225      45   IBM
   9345-1      9345     46456      15         1440      45       9   IBM
   9345-2      9345     46456      15         2156      60      12   IBM
```

Figure 2-9 Device Type Table

Dynamically defined device

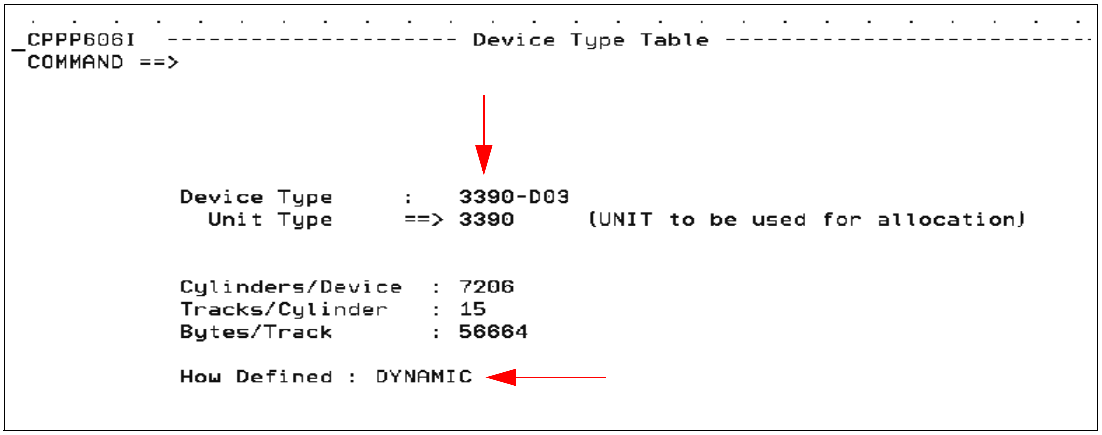
Figure 2-10 shows the panel you see when you select a volume to change its attributes. Because the dialog found out about the device's characteristics, you cannot change them here (and if you could, the dialog would correct them anyway the next time it looked at the device).

Nonstandard device

If the device type is nonstandard, all you need to enter here is the UNIT name to be used within JCL for this device type. For IBM devices, UNIT defaults to the generic unit name, so you should only need to do this for non-IBM, dynamically-defined device types, not dynamically-defined device models for which the device type is an IBM device type (like 3380 or 3390).

For any device, UNIT defaults to the first one found for a matching device type in the table.

.



```
 CPPP606I  --------------------- Device Type Table --------------------------
 COMMAND ==>



           Device Type     :   3390-D03
             Unit Type     ==> 3390      (UNIT to be used for allocation)


           Cylinders/Device  : 7206
           Tracks/Cylinder   : 15
           Bytes/Track       : 56664

           How Defined : DYNAMIC
```

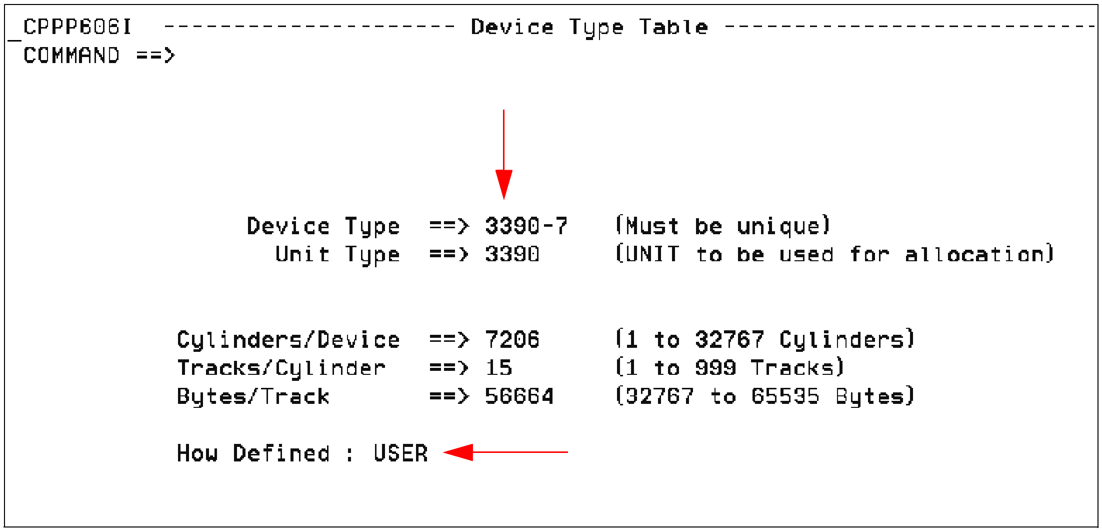
Figure 2-10 Dynamically defined device

User-defined device

Figure 2-11 shows the Device Type Table panel that has been defined by a user. When a device is user-defined, all data entry fields can be modified as shown. This panel now contains some changed fields for consistency with other displays as follows:

- Device Name changed to Device Type
- Device Type changed to Unit Type

This panel is never displayed for IBM-defined devices.



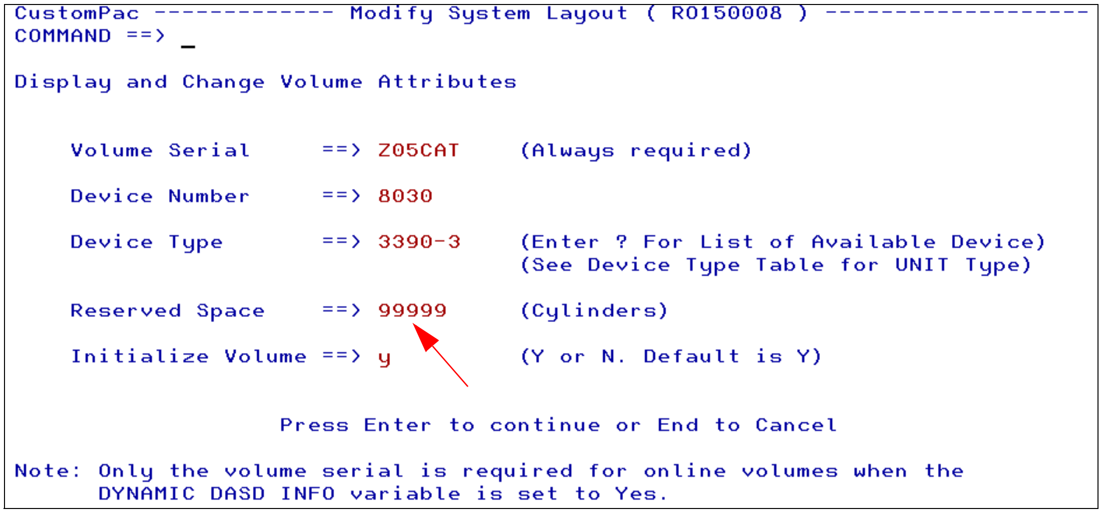
```
_CPPP606I  --------------------- Device Type Table ---------------------------
 COMMAND ==>




                 Device Type  ==> 3390-7   (Must be unique)
                   Unit Type  ==> 3390     (UNIT to be used for allocation)


            Cylinders/Device  ==> 7206     (1 to 32767 Cylinders)
            Tracks/Cylinder   ==> 15       (1 to 999 Tracks)
            Bytes/Track       ==> 56664    (32767 to 65535 Bytes)

            How Defined :  USER
```

Figure 2-11 User-defined device

2.5.2 More reserved space

The Reserved Space field in the Display and Change Volume Attributes panel allowed only 999 cylinders to be reserved on a volume. But that is only 3% of a large 32760 cylinder volume. This was addressed by making all reserved space input fields 5 digits long. You can reserve all but 1 allocable cylinder.

Displays of reserved space will show it as a percentage of the volume size or as a number of reserved cylinders. Figure 2-12 on page 29 shows the five digit reserved space field.



```
CustomPac ------------- Modify System Layout ( R0150008 ) -------------------
COMMAND ==> _

Display and Change Volume Attributes


    Volume Serial      ==> Z05CAT    (Always required)

    Device Number      ==> 8030

    Device Type        ==> 3390-3    (Enter ? For List of Available Device)
                                     (See Device Type Table for UNIT Type)

    Reserved Space     ==> 99999     (Cylinders)

    Initialize Volume ==> y          (Y or N. Default is Y)


                   Press Enter to continue or End to Cancel

Note: Only the volume serial is required for online volumes when the
      DYNAMIC DASD INFO variable is set to Yes.
```

Figure 2-12 Five digit reserved space field

2.5.3 Automatic VTOC and index sizing

Given the proliferation of possible volume sizes, this feature was added to avoid making you define the VTOC and index sizes. The VTOC and VTOC index sizes are now calculated by the dialog.

VTOC size The VTOC size will be set to about 1 track for every 45 cylinders on the volume, which has proven to be enough for system software volumes.

Index size The index size will be set to about 1/5 of the VTOC size.

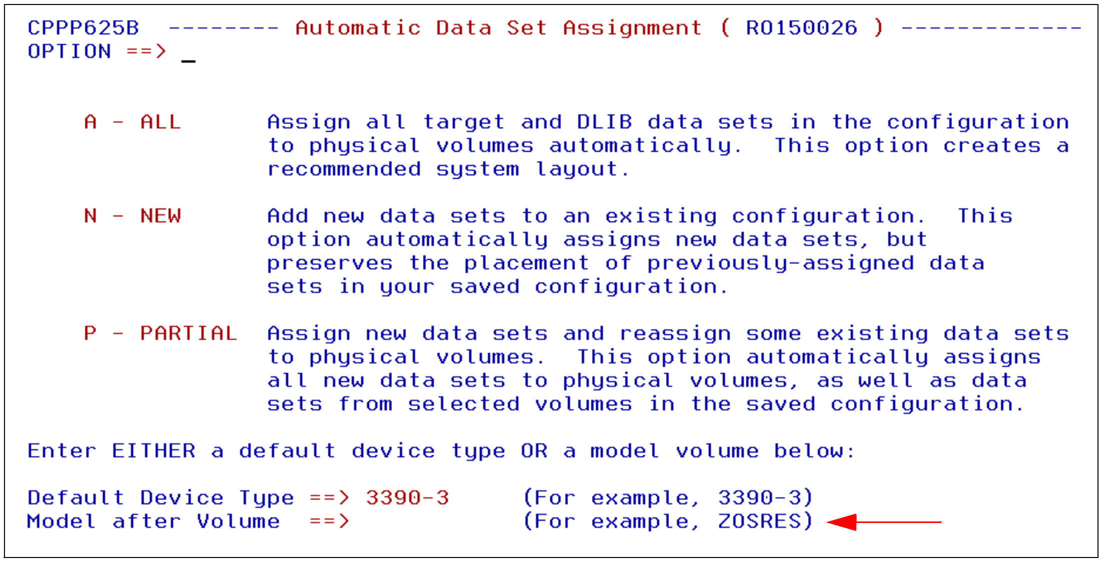
Note: The new sizes are consistent with the values that have been set for fixed volume sizes in the past, and therefore these calculations work with volumes of varying size.

2.5.4 Automatic data set assignment

The first panel in the Recommended System Layout option is Automatic Data Set Assignment, as shown in Figure 2-13. Since there are a lot of potential device types now, the dialog is changed so you can specify either of the following:

Model volume When you specify a model volume, the dialog retrieves the characteristics of the volume's device and uses them if any new volumes need to be created. This lets you avoid defining devices if you are using non-IBM DASD or user-defined volume sizes before using the automatic data set assignment function in the dialog.

Default device type The default device or model-after volume serial you enter is stored in your ISPF profile data set (ISPPROF). Therefore, everyone can have their own default device type. This is especially handy if people use different volume sizes for subsystem volumes than for z/OS volumes.



```
CPPP625B  -------- Automatic Data Set Assignment ( R0150026 ) -------------
OPTION ==> _


    A - ALL       Assign all target and DLIB data sets in the configuration
                  to physical volumes automatically.  This option creates a
                  recommended system layout.

    N - NEW       Add new data sets to an existing configuration.  This
                  option automatically assigns new data sets, but
                  preserves the placement of previously-assigned data
                  sets in your saved configuration.

    P - PARTIAL   Assign new data sets and reassign some existing data sets
                  to physical volumes.  This option automatically assigns
                  all new data sets to physical volumes, as well as data
                  sets from selected volumes in the saved configuration.

Enter EITHER a default device type OR a model volume below:

Default Device Type ==> 3390-3     (For example, 3390-3)
Model after Volume  ==>            (For example, ZOSRES)
```

Figure 2-13 Automatic data set assignment

2.5.5 JES2 updates for large volumes

JES2 uses a TTR pointer to mark the start and end of its spool extents. This means that a spool extent can't be larger than 64K tracks. This also meant that something had to be done to JES2 to support spool extents that extended past the 64K track boundary. This was accomplished by using relative track addressing in place of absolute track addressing. To avoid introducing an incompatibility with usermods and exits, the RELADDR parameter was added to the SPOOLDEF statement that controls the track addressing mode.

The JES2 spool data sets must fall entirely within the first 64K tracks, unless the RELADDR parameter of SPOOLDEF specifies either:

ASNEEDED If a SPOOL data set resides within the first 64K track, then JES2 uses absolute track addressing. If the SPOOL data set crosses beyond the first 64K tracks, then JES2 uses relative track addressing.

ALLWAYS JES2 always uses relative track addressing regardless of where the SPOOL data set is located on a volume. This is useful when testing to ensure that applications and exits function correctly with relative track addressing.

NEVER JES2 never uses relative addressing. If a SPOOL data set is not contained within the first 64K tracks on a volume, then the volume cannot be used for SPOOL and the start of that volume will fail.

ASNEEDED is specified in ServerPac to avoid incompatibilities when spool extents end beyond the 64K track boundary.

Note: RELADDR=NEVER is the JES default.

2.6 Pre-received service tapes

Service tapes have been eliminated from the ServerPac order. All service will now be RECEIVED already, and loaded to your new SMPPTS data set. The SMPPTS is compacted, so it should not grow much, if at all. It is allocated with approximately 10 to 15% free space. It might be larger than it is now just before PUTnnnn PTFs are marked as RSUnnnn. PTFs are compacted in the SMPPTS, so the space requirement for this data set should be close to its old default size. The SMPPTS data set is loaded during the RESTORE job.

> **Note:** Make sure you place the SMPPTS data set on a volume where it has room to extend, or allocate an SMPPTS spill data set, if you plan to RECEIVE a lot of service before applying and accepting the service that is already received.

2.7 Various panel changes to improve usability

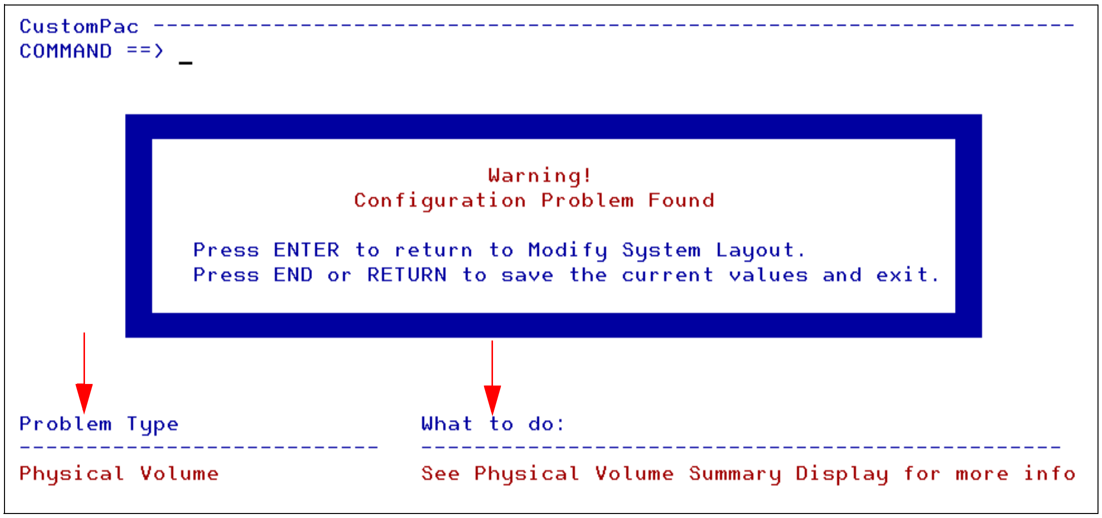
The Modify System Layout warnings panel has been reworked to offer some guidance and to add the capability to check the validity of many variables at the time you enter them.

Minor usability changes include:

- Spelling out commands on the primary and line command lists
- Removing clutter from the displays
- Using correct terms

The Change Volume Attributes panel field “Existing Data” has been changed to “Initialize Volume” to make the meaning more explicit.

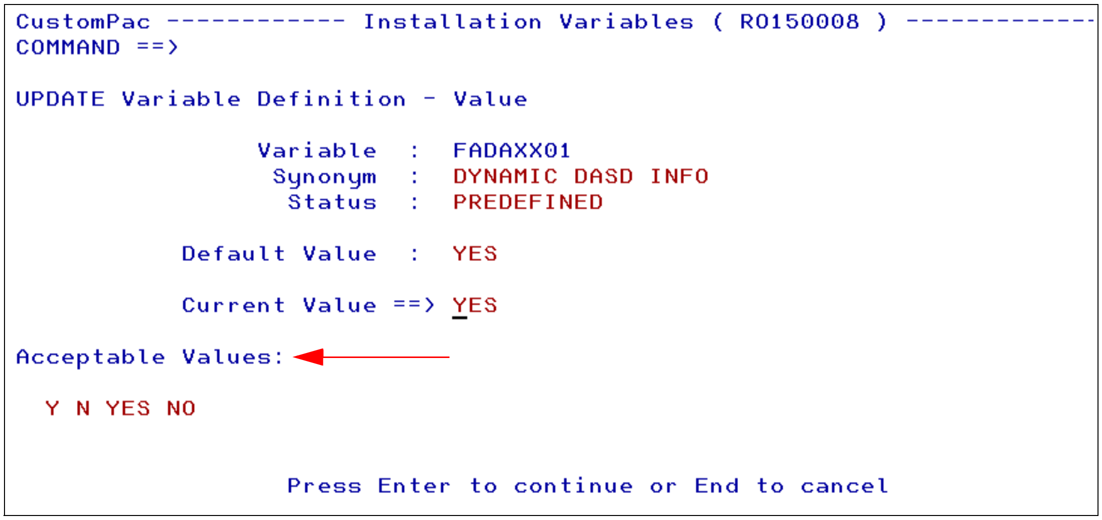
Figure 2-14 on page 31 shows the added guidance to the Modify System Layout panel.



```
CustomPac -------------------------------------------------------------------
COMMAND ==> _

                         Warning!
                 Configuration Problem Found

      Press ENTER to return to Modify System Layout.
      Press END or RETURN to save the current values and exit.

Problem Type                  What to do:
---------------------------   ------------------------------------------------
Physical Volume               See Physical Volume Summary Display for more info
```

Figure 2-14   Added guidance in the Modify System Layout panel

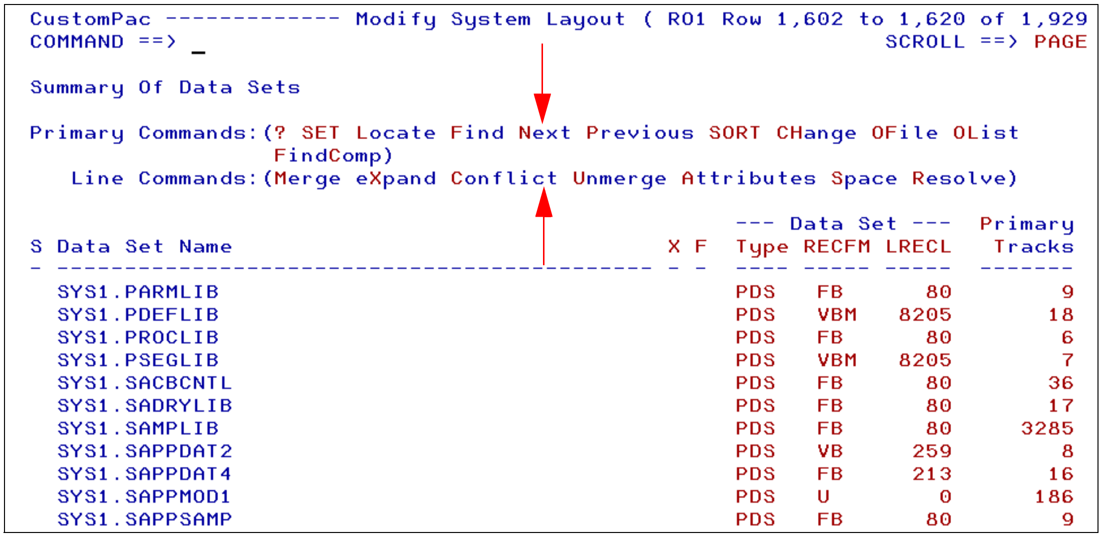
Figure 2-15 shows the added capability to check variables at installation time.



```
CustomPac ------------ Installation Variables ( R0150008 ) -------------
COMMAND ==>

UPDATE Variable Definition - Value

                Variable  :  FADAXX01
                 Synonym  :  DYNAMIC DASD INFO
                  Status  :  PREDEFINED

           Default Value  :  YES

           Current Value ==> YES

Acceptable Values:

  Y N YES NO


                  Press Enter to continue or End to cancel
```

Figure 2-15 Variable checking at installation time

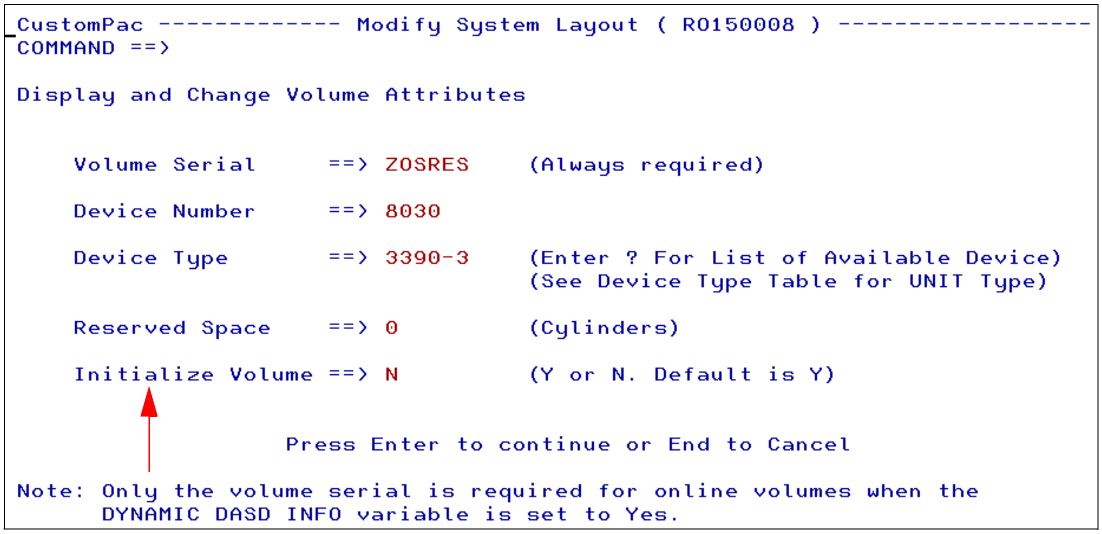
Figure 2-16 on page 32 shows the spelling out of commands on the primary and line command lists.



```
CustomPac ------------- Modify System Layout ( R01 Row 1,602 to 1,620 of 1,929
COMMAND ==> _                                                    SCROLL ==> PAGE

Summary Of Data Sets

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

                                                   --- Data Set ---   Primary
S Data Set Name                               X F  Type RECFM LRECL   Tracks
- ------------------------------------------- - -  ---- ----- -----  -------
  SYS1.PARMLIB                                     PDS  FB       80        9
  SYS1.PDEFLIB                                     PDS  VBM    8205       18
  SYS1.PROCLIB                                     PDS  FB       80        6
  SYS1.PSEGLIB                                     PDS  VBM    8205        7
  SYS1.SACBCNTL                                    PDS  FB       80       36
  SYS1.SADRYLIB                                    PDS  FB       80       17
  SYS1.SAMPLIB                                     PDS  FB       80     3285
  SYS1.SAPPDAT2                                    PDS  VB      259        8
  SYS1.SAPPDAT4                                    PDS  FB      213       16
  SYS1.SAPPMOD1                                    PDS  U         0      186
  SYS1.SAPPSAMP                                    PDS  FB       80        9
```

Figure 2-16 Primary and line command lists

Figure 2-17 shows the “Initialize Volume” field in the Change Volume Attributes panel.



```
CustomPac ------------- Modify System Layout ( R0150008 ) -----------------
COMMAND ==>

Display and Change Volume Attributes


    Volume Serial     ==> ZOSRES    (Always required)

    Device Number     ==> 8030

    Device Type       ==> 3390-3    (Enter ? For List of Available Device)
                                    (See Device Type Table for UNIT Type)

    Reserved Space    ==> 0         (Cylinders)

    Initialize Volume ==> N         (Y or N. Default is Y)


                   Press Enter to continue or End to Cancel

Note: Only the volume serial is required for online volumes when the
      DYNAMIC DASD INFO variable is set to Yes.
```

Figure 2-17 New "initialize Volume" field

3

z/OS V1R5 BCP enhancements

This chapter describes enhancements made to the z/OS V1R5 base control program (BCP), specifically the following:

- 64-bit virtual storage enhancements
- IPCS enhancements
- Program management enhancements
- JCL enhancements
- SMF record type 6 enhancements
- 3590 model H tape drive enhancement
- File sequence number >9999 support
- Unicode enhancements
- 2 GB FICON® channel support
- IARV64 system trace support
- System Logger

3.1 64-bit virtual storage enhancements

The basic 64-bit virtual storage management support is introduced in z/OS 1.2 and lays the foundation for a 64-bit operating system infrastructure.

In 64-bit addressing mode, the address space begins at address 0 and ends at 16 exabytes, an incomprehensibly high address. The address space structure below the 2 gigabyte address has not changed; all programs in AMODE 24 and AMODE 31 continue to run without change.

In the 31-bit address space, a virtual *line* marks the 16-megabyte address. The 64-bit address space also includes the virtual line at the 16-megabyte address; additionally, it includes a second virtual line called *the bar* that marks the 2-gigabyte address. The bar separates storage below the 2-gigabyte address, called *below the bar*, from storage above the 2-gigabyte address, called *above the bar*. The area above the bar is intended for data; no programs run above the bar. There is no area above the bar that is common to all address spaces, and no system control blocks exist above the bar. IBM reserves an area of storage above the bar for special uses to be developed in the future.

3.1.1 MEMLIMIT support

You can set a limit on how much virtual storage above the bar each address space can use. This limit is called the *MEMLIMIT.* If you do not set a MEMLIMIT, the system default is 0, meaning that no address space can use virtual storage above the bar. If you want to use virtual storage above the bar, you need to set the MEMLIMIT explicitly. You can set an installation default MEMLIMIT through System Management Facility (SMF). You can also set a MEMLIMIT for a specific address space in the job control language (JCL) that creates the address space or by using SMF exit IEFUSI.

3.1.2 Using storage

Before z/OS V1R2, a program which needed storage beyond 2G was generally met by creating one or more data spaces or hiperspaces, and keeping track of the data was not always easy. Each dataspace or hiperspace needed to be managed separately and uniquely.

With the introduction of 64-bit virtual in z/OS V1R2, programs requiring a very large buffer pool take advantage by using the area above the 2G bar. The area above the bar is intended to be used for data only, not for executing programs. Programs use the IARV64 macro to obtain storage above the bar in “chunks” of virtual storage called memory objects.

Database subsystems like DB2 and other middleware can greatly benefit from the increased capacity of 64-bit virtual storage by supporting a large number of concurrent users and transactions.

3.1.3 z/OS V1R5 virtual storage enhancements

With z/OS V1R5, the following enhancements for 64-bit virtual storage have been added:

- 64-bit shared memory addressing area between 2 terabytes and 512 terabytes
- Multiple guard area support for private high virtual storage above the bar

3.2 64-bit shared virtual storage support

Starting with z/OS V1R5, address spaces can now share data above the bar, shown in Figure 3-1 on page 38, in addition to having private data. The shared storage is an area which is shared between multiple address spaces or multiple tasks in the same address space.

Each address space can logically be 16 exabytes (2 **64) in size. The area below 2 GB is mapped as before and is totally compatible. The area above the bar is for application data. There are no common areas, system areas, or programs above the bar.

The shared storage appears at the same address in each address space. The default shared area range is between 2T and 512T, as shown in the 64-bit address space memory map in Figure 3-1.

3.2.1 64-bit address space memory map

In a 16 exabyte address space with 64-bit virtual storage addressing, there are three additional levels of translation tables, called *region* tables. They are called region third table (R3T), region second table (R2T), and region first table (R1T), as shown in Figure 3-1. The region tables are 16 KB in length, and there are 2048 entries per table. When the first storage is created above the bar, RSM creates the R3T. The R3T table has 2048 segment table pointers, and provides addressability to 4 TB. When virtual storage greater than 4 TB is allocated, an R2T is created. An R2T has 2048 R3T table pointers and provides addressability to 8 PB. An R1T is created when virtual storage greater then 8 petabytes is allocated. The R1T has 2048 R2T table pointers and provides addressability to 16 exabytes.

As users allocate private storage above the bar, it is first allocated from the Low non-shared area. Similarly, as shared area storage is allocated, it is allocated from the bottom up. This is done to allow applications to have both private and shared memory above the bar and only need a region 3rd table.

If users allocate large private or shared areas above the bar, then Real Storage Manager (RSM) needs to construct region second tables or region first tables at the time of the allocation to manage the virtual storage.

Address memory map

The address memory map in Figure 3-1 shows the following:

0 - 16M 16M is still referred to as the line. Below the line can be addressed with a 24-bit address.

0 - 231** Above the line requires a 31-bit address and does not usually refer to storage beyond 2G.

231 - 2**32** From 2G to 4G is considered the bar. Below the bar can be addressed with a 31-bit address. Above the bar requires a 64-bit address.

The bar

Just as the system does not back the page at 7FFFF000 in order to protect programs from addresses which can wrap to 0, the system does not back the virtual area between 2G and 4G. That means that a 31-bit address with the high bit on will always program check if used in AMODE 64. Above the bar is divided into 3 areas as follows:

231 - 2**41** The low non-shared area starts at 4G and goes to 2T (2**41).

241 - 2**50** The shared area starts at 2T (2**41) and goes to 512T (2**50) or higher if requested.

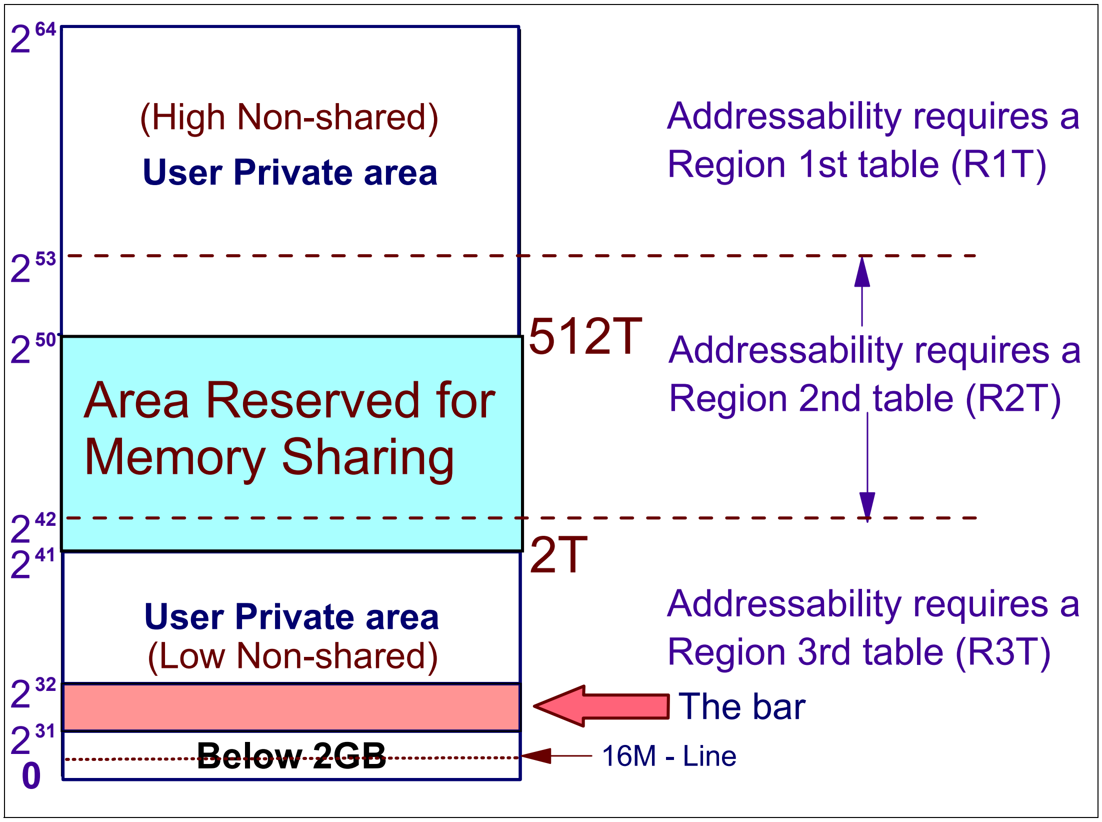

2 64
(High Non-shared)
User Private area
Addressability requires a Region 1st table (R1T)
2 53
2 50
512T
Addressability requires a Region 2nd table (R2T)
Area Reserved for Memory Sharing
2 42
2 41
2T
User Private area
(Low Non-shared)
Addressability requires a Region 3rd table (R3T)
2 32
The bar
2 31
Below 2GB
16M - Line
0

Figure 3-1 z/OS V1R5 address space

3.2.2 64-bit shared virtual storage requirements

To exploit the 64-bit shared virtual storage, the following environment is required:

Hardware z900, z800, z890, or z990

Software z/OS V1R5 or higher running in z/Architecture mode.

New options on the IARV64 macro allow address spaces to share storage above the bar.

Shared Area size can be specified via the HVSHARE keyword in IEASYSxx parmlib member, or by responding to message IEA101A during system IPL. The characteristics of HVSHARE are the following:

- It is specified as: HVSHARE=xxxxxxxxxxxG, or xxxxxxxxxT, or xxxxxP.
- Default shared area starts at 2TB and ends at 512TB.
- The minimum size is zero and the maximum size is 1 exabyte (which is 1048576 terabytes or 1024 petabytes).

HVSHARE value

The size of the HVSHARE value determines where the system places the shared virtual area. If you specify a value less than 2 terabytes (2T), the system obtains storage that straddles the 4 terabyte line as follows:

- Half of the storage comes from above the 4 terabyte line.
- Half of the storage comes from below the 4 terabyte line.

If you specify a value larger than 2T, the system obtains storage starting at the 2 terabyte line. The value you specify is rounded up to a 64 gigabyte boundary.

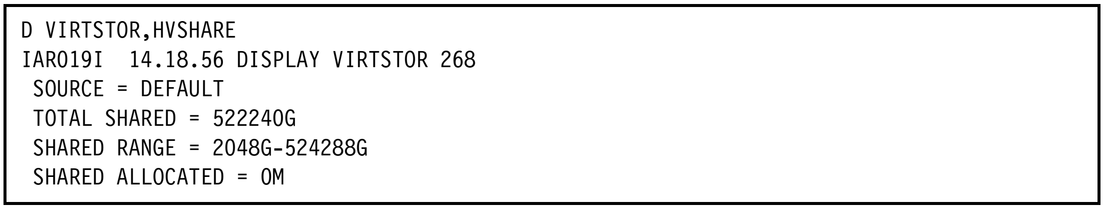
Displaying virtual storage information

Use the **DISPLAY VIRTSTOR** command to identify the virtual storage configuration. The result of this command is displayed in message IAR019I, shown in Figure 3-2. It contains the following information:

- Source of HVSHARE parameter.
  - Parmlib member (xx), that is IEASYSxx.
  - Operator supplied (OP), that is reply to message IEA101A during system IPL.
  - Default (DEFAULT), that is from 2T to 512T.
- The size of the high virtual shared area in gigabytes, in decimal.
- The range of the high virtual shared area in gigabytes, in decimal.
- The amount of shared storage allocated into memory objects in megabytes, in decimal.

The command syntax is:

```
D {VIRTSTOR | VS},HVSHARE
```



```
D VIRTSTOR,HVSHARE
IAR019I  14.18.56 DISPLAY VIRTSTOR 268
 SOURCE = DEFAULT
 TOTAL SHARED = 522240G
 SHARED RANGE = 2048G-524288G
 SHARED ALLOCATED = 0M
```

Figure 3-2 Command D VIRTSTOR

3.2.3 Memory management above the bar

The memory management above 2G is totally new beginning with z/OS V1R2. Above the bar, there are no MVS subpools and no common area. The virtual storage above the bar is organized as memory objects. Programs obtain storage above the bar in chunks of virtual storage called memory objects. The system allocates a memory object as a number of virtual segments; each segment is a megabyte in size and begins on a megabyte boundary. A memory object can be as large as the memory limits set by your installation and as small as one megabyte. To use the storage in a memory object, the program must be in AMODE 64.

Memory objects have a range and a size. The size can be increased or decreased within the range. Only the size is charged against the MEMLIMIT. Characteristics of a memory object, like storage protection key and fetch protection attribute, apply to the entire storage range of the memory object and cannot be changed after the memory object is allocated. The memory object is freed in its entirety; partial freeing is not allowed. Part or all of the memory in the range covered by a private memory object is usable virtual storage and the remainder is not usable and is called the guard area. The extent of the usable virtual storage can be changed, with a compensatory change in the extent of the guard area.

> **Important:** The entire range of virtual storage represented by a shared memory object is usable storage. In other words, there are no guard areas in a shared memory object.

Private memory objects are owned by a task. The shared memory object is shared at the same address in every address space and is owned by the system.

IARV64 macro

A new system service, IARV64, was introduced in z/OS V1R2 to allocate and manage storage above the bar. IARV64 is the only external programming interface provided to allocate and manage storage above the bar. Only data can be stored in virtual storage above the bar. You use the z/Architecture instructions that handle 64-bit registers to access data above the bar. Programs are still loaded below the bar, but they must be running in 64-bit addressing mode (AMODE 64) to address the data in virtual storage above the bar. Virtual storage above the bar cannot be used for loading and executing programs; they must be loaded and executed below the bar. This support does not keep any control information or control blocks above the bar. This area is just for data.

Guard area

When a program creates a memory object, it can specify that the memory object is to consist of two different areas: the usable area and the guard area. The guard area is an optional area within a memory object, sometimes also referred to as a hidden area. It is a number of reserved pages, in multiples of megabytes, starting on a megabyte boundary. The guard area is allocated either from the low virtual address of the memory object and upwards, or from the high virtual address and downwards, as shown in Figure 3-3 on page 41.

The guard area is introduced so that you can reserve an area for future utilization of space for your application. Another way you can use the guard area is to protect yourself from accidentally referencing virtual storage beyond the end of a memory object, and overlaying data in an adjacent memory object. In this case, you should allocate a guard area at high virtual address in your memory object. Similarly, if a program wants to protect its memory object from another program using a memory object at a lower virtual address, it will create a guard area at the lower end of its memory object.

Guard areas cannot be referenced by your program. If you reference the guard area, you will get a program check. The size of the guard area does not count when the system is performing the MEMLIMIT checking, only the usable area counts. The size of the guard area can be changed—for example, to get more usable virtual storage within your memory object. You can increase or decrease the size by using the CHANGEGUARD service. The amount of change is in multiples of one megabyte segments.

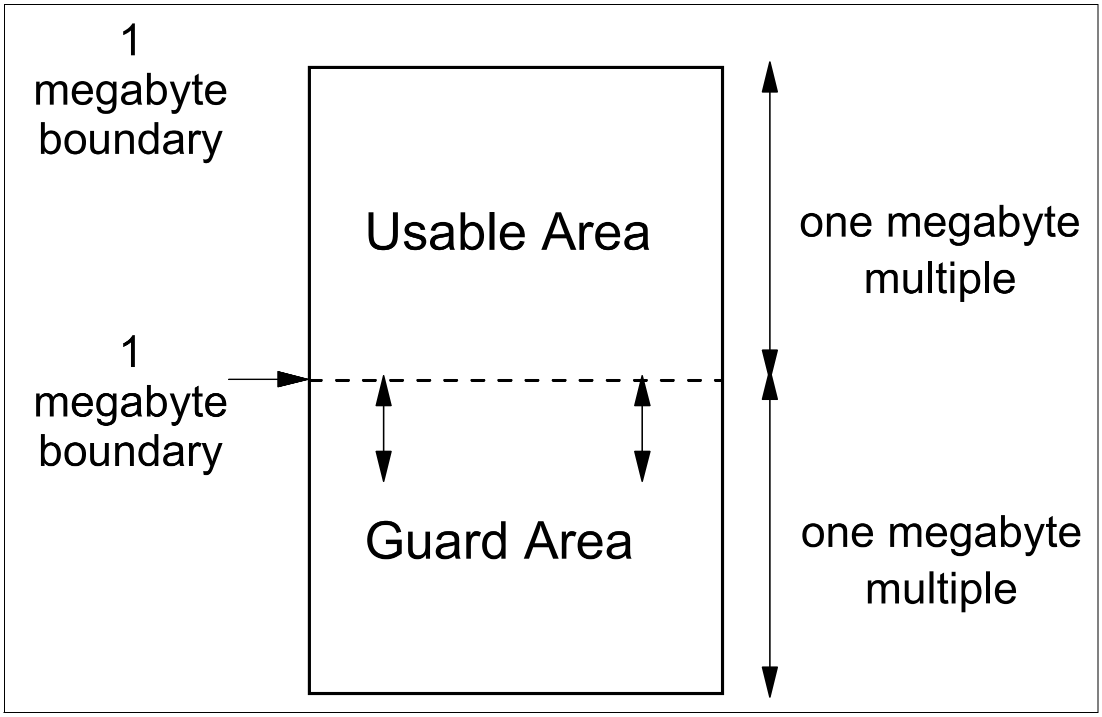


Figure 3-3 A memory object with a guard area

3.2.4 Memory object operations

The IARV64 macro provides all the virtual storage services that a program needs to work with the memory objects. Functions provided by the IARV64 macro are the following:

- Managing memory objects
- Changing the status of pages in the memory object

Managing memory objects

To create a shared memory object, use the IARV64 GETSHARED service. While shared memory storage is like common storage in that it is shared across address spaces, it differs because there is not automatic addressability or access to it. To mange the memory objects, the following IARV64 request types are used:

GETSTOR Create a memory object (only for private memory objects).

CHANGEGUARD Increase or decrease the amount of usable memory in a memory object (only for private memory objects).

GETSHARED Create a shared memory object (only for shared memory objects).

SHAREMEMOBJ Allows an address space to access shared memory objects (only for shared memory objects).

CHANGEACCESS Manages the type of access an address space has to the shared virtual storage (only for shared memory objects).

DETACH Delete memory objects.

LIST Request a list of the addresses of memory objects.

Changing status of pages within a memory object

Authorized programs can use IARV64 macro requests to change the status of memory objects as follows:

PAGEFIX Fix virtual pages in central storage (only for private memory objects).

PAGEUNFIX Undo a Pagefix (only for private memory objects).

DISCARDDATA Discard the data and free the frames.

PAGEOUT Notify the system that pages will probably not be referenced again soon.

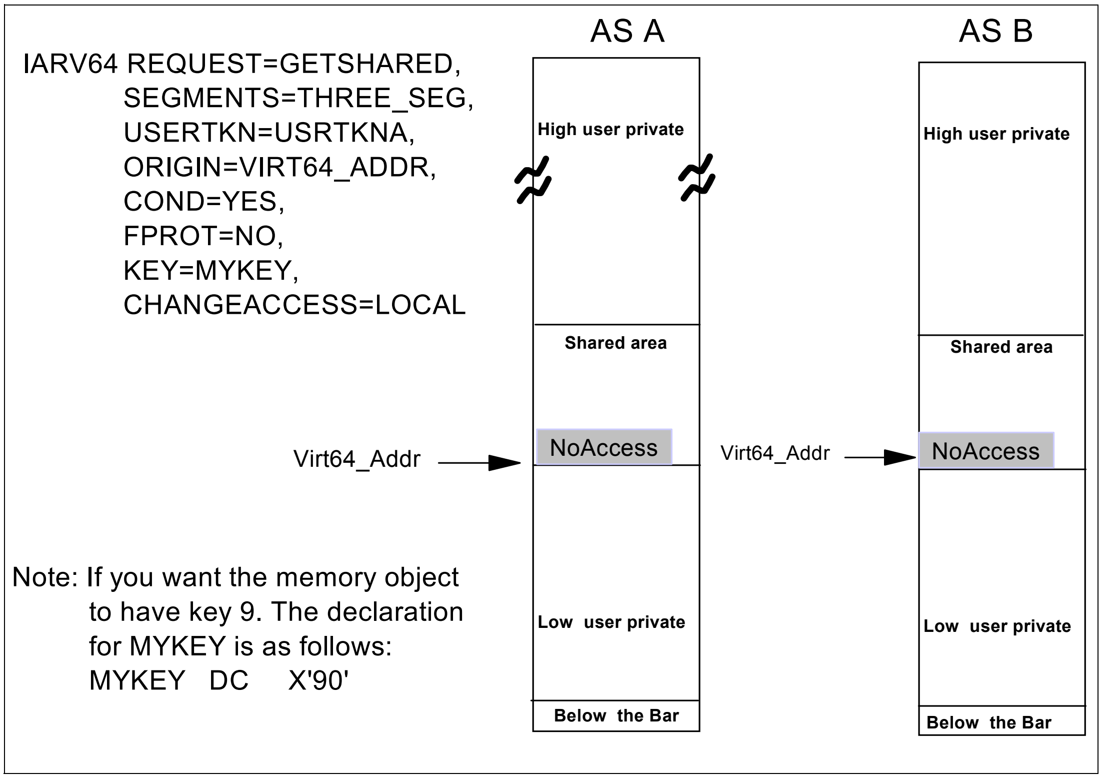
PAGEIN Notify the system that pages will be referenced soon.

> **Note:** For all operations, a system ABEND DC2 (along with the appropriate reason code) is issued for invalid requests, or when a specific request cannot be satisfied. Here is a brief explanation of ABEND DC2. A DC2 ABEND is issued for an invalid request and also can be issued for a valid request which cannot be successfully processed by RSM. In other words, an IARV64 macro invocation to manage storage above 2 gigabytes failed. A hexadecimal reason code returned in the middle two bytes of register 0 describes the error.

3.2.5 Creating shared memory objects

Shared memory objects are allocated (created) by programs using the service IARV64 REQUEST=GETSHARED. When the object is allocated, no one has access to the memory object. GETSHARED creates a system interest in the object.

Figure 3-4 shows an example of creating a shared memory object.



```
IARV64 REQUEST=GETSHARED,
       SEGMENTS=THREE_SEG,
       USERTKN=USRTKNA,
       ORIGIN=VIRT64_ADDR,
       COND=YES,
       FPROT=NO,
       KEY=MYKEY,
       CHANGEACCESS=LOCAL
```

Note: If you want the memory object to have key 9. The declaration for MYKEY is as follows:

```
MYKEY   DC     X'90'
```

Figure 3-4 Creating shared memory objects

The operands on the IARV64 macro in Figure 3-4 are as follows:

CHANGEACCESS Specifies whether subsequent CHANGEACCESS requests are treated as local or global in scope. When local is specified, subsequent CHANGEACCESS requests change access only for the specified address space. When global is specified, subsequent CHANGEACCESS requests change access for ALL address spaces that share the memory object and any new address space that will subsequently share the object.

SEGMENTS Specifies a total size in megabytes.

COND=YES/NO Helps to make the request conditional to avoid ABEND.

USERTKN Specifies an 8-byte User Token which is used to group memory objects for later Detach requests. The left word of the user token must be non-zero. While your program can obtain only 1 memory object at a single invocation of IARV64, it can, for management purposes, relate 2 or more objects to each other by specifying a USERTKN value. A program can delete all memory objects that have the same USERTKN value.

ORIGIN The address of the virtual storage associated with the lowest memory object which is returned to you.

FPROT=YES/NO Specifies whether the memory object is fetch protected or not. The default is NO.

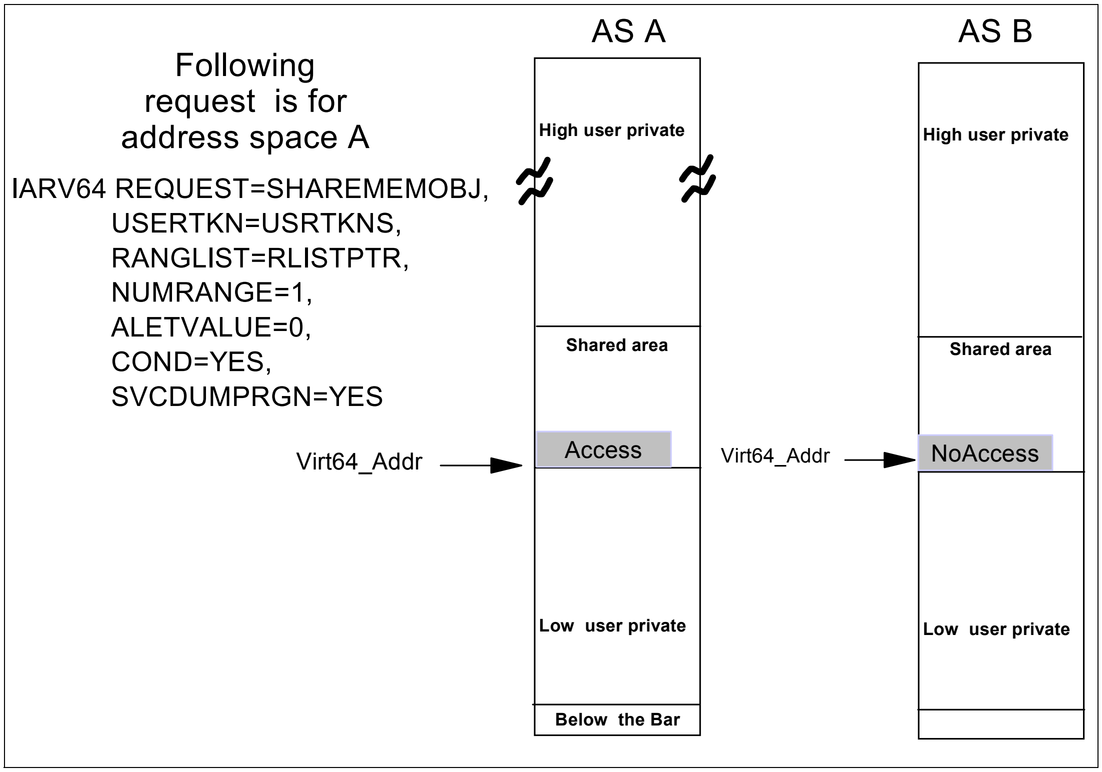
KEY Specifies the storage key for the memory object. The default is the caller's key.

3.2.6 Accessing shared memory objects

To get access to the shared memory objects, programs use the service IARV64 REQUEST=SHAREMEMOBJ. An address space can issue more than one SHAREMEMOBJ request for the same memory object.

Important: To separate each of the requests for the same memory object you need to specify a different user token.

Figure 3-5 on page 44 shows an example of accessing shared memory objects.



```
IARV64 REQUEST=SHAREMEMOBJ,
       USERTKN=USRTKNS,
       RANGLIST=RLISTPTR,
       NUMRANGE=1,
       ALETVALUE=0,
       COND=YES,
       SVCDUMPRGN=YES
```

Figure 3-5 Accessing shared memory objects

The parameters on the IARV64 macro in Figure 3-5 are as follows:

COND=YES/NO COND=YES helps to make the request conditional to avoid having the program abend because it asked to free a memory object that does not exist.

USERTKN A required 8-byte token that relates two or more memory objects to each other. Associate the user token with the memory object, so later you can free several shared memory objects at one time.

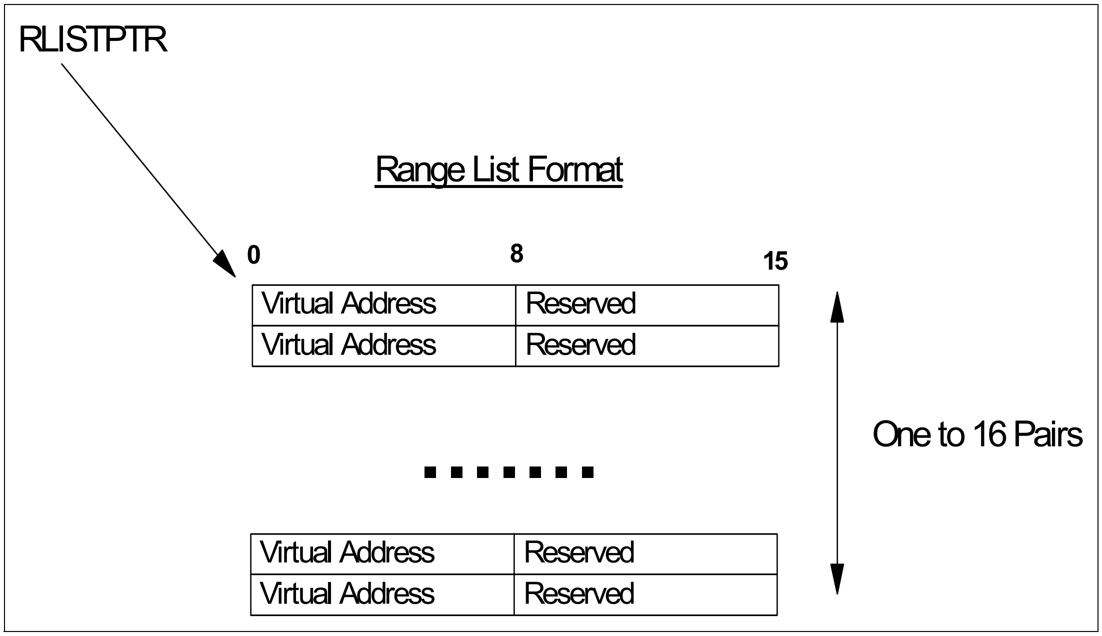
RANGLIST Specifies a list of memory objects that you want to share. Together with the NUMRANGE parameter, the RANGLIST parameter allows you to make multiple storage ranges eligible to be assigned to subspaces. The RANGLIST parameter specifies a fullword that contains an address of a list of ranges, or specifies a register that contains the address of the fullword pointer to the range list that you created when you allocated the storage. The number of entries in the range list is specified on the NUMRANGE parameter.

NUMRANGE Specifies a list of supplied ranges pointed to by RANGLIST and can contain 1 to 16 ranges. The default is 1.

ALETVALUE Designates the address space which is given access to the memory objects by specifying an ALET value. Supported address spaces are Primary (0) and Home (2). The default is 0.

SVCDUMPRGN Specifies whether the object is included in an SVC dump when region is requested as part of SDATA. The default is SVCDUMPRGN=YES.

Figure 3-6 on page 45 shows the format of the RANGLIST.



Virtual Address	Reserved
Virtual Address	Reserved
Virtual Address	Reserved
Virtual Address	Reserved

Figure 3-6 RANGLIST format

Virtual Address can be any address within the shared memory object that you want to be given access to. The second double word of the entry is reserved and must be binary zeros.

3.2.7 Changing shared memory object storage access

The CHANGEACCESS request of the IARV64 macro is used to request a change to the type of access to the specified virtual storage. The three types of access specified using the VIEW= parameter are as follows:

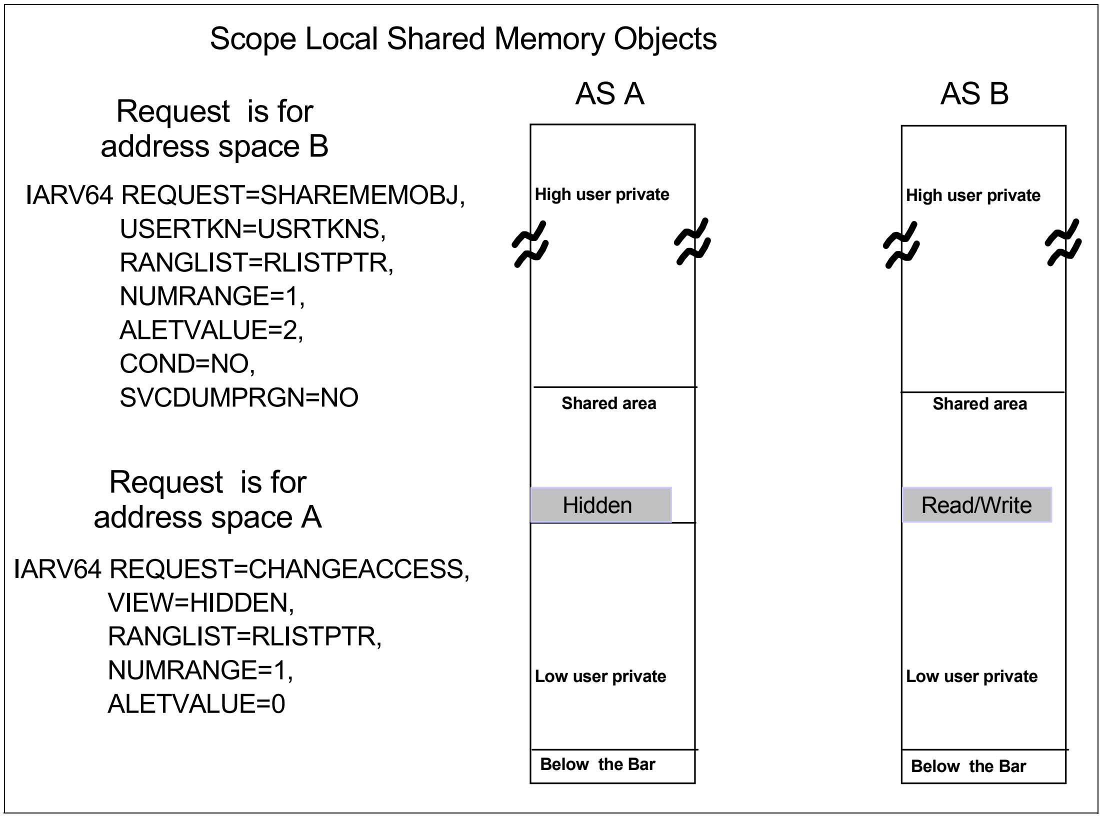
READONLY The area can only be used to read data. Any attempt to alter data by writing onto the area results in a program check.

SHAREDWRITE The area can be used to read or update data.

HIDDEN The data within the area cannot be accessed until its view type is changed to READONLY or SHAREDWRITE. Any attempt to access a hidden area results in a program check.

You can request that the type of access to the specified virtual storage be changed. The scope of the change is determined by the specification of local versus global on the GETSHARED request. When *local* is specified, only the address space specified on the request is affected. If no SHAREMEMOBJ object is done from that address space then the request fails. When *global* is specified, all address spaces currently sharing the memory object are affected, and any subsequent address spaces that share the memory object will see the changed access. For global requests, no prior SHAREMEMOBJ needs to be done. For local requests, the target address space must have done a SHAREMEMOBJ before requesting a change of access.

Figure 3-7 on page 46 shows an example of changing the local storage access to the shared memory object.


Scope Local Shared Memory Objects
Request is for address space B
Request is for address space A
AS A
AS B
High user private
Shared area
Hidden
Read/Write
Low user private
Below the Bar

```
IARV64 REQUEST=SHAREMEMOBJ,
        USERTKN=USRTKNS,
        RANGLIST=RLISTPTR,
        NUMRANGE=1,
        ALETVALUE=2,
        COND=NO,
        SVCDUMPRGN=NO
```

```
IARV64 REQUEST=CHANGEACCESS,
        VIEW=HIDDEN,
        RANGLIST=RLISTPTR,
        NUMRANGE=1,
        ALETVALUE=0
```

Figure 3-7 Changing local storage access

The parameters on the IARV64 macro in Figure 3-7 that have not previously been defined are as follows:

VIEW The type of access you want to have to the virtual storage. The three types of access are READONLY, SHAREDWRITE, or HIDDEN.

RANGLIST Specifies the virtual address, which can be anywhere in the shared memory object. The following rules apply:

- The starting address must be on a segment boundary.
- The starting address must be within a memory object returned by a GETSHARED request.
- A single range must be contained within a single memory object.

Figure 3-8 shows the RANGLIST format for changing storage access.

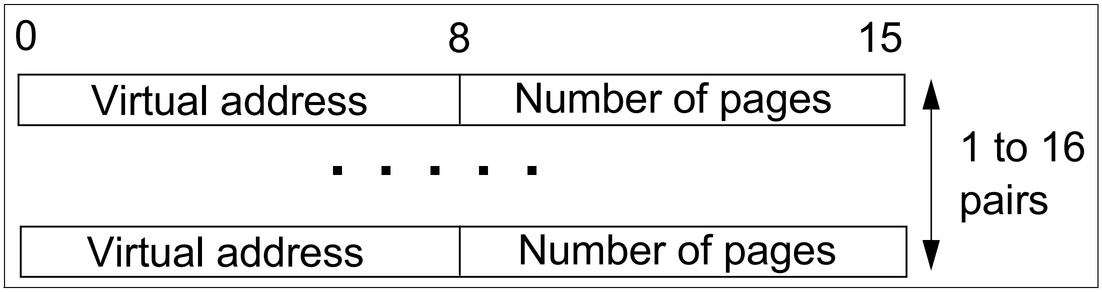


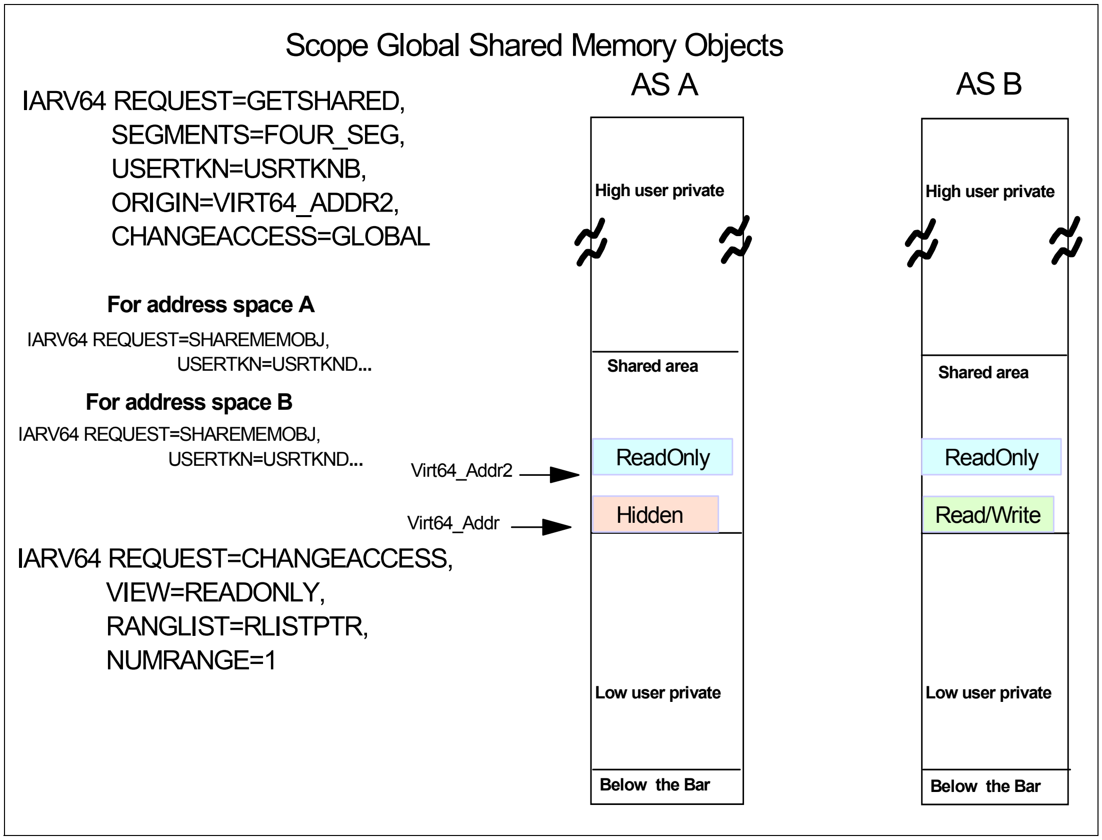
Figure 3-8 RANGLIST format for changing storage access

3.2.8 Changing storage access

As a general rule, the starting virtual address (Virt64_Addr) must be on a segment boundary and must be within a memory object returned by a GETSHARED request. A pair must be contained within a single memory object.

Figure 3-9 shows the following:

- When CHANGEACCESS=GLOBAL is specified, subsequent CHANGEACCESS requests will change access for ALL address spaces that share the memory object and any new address space that will subsequently share the object.
- A GETSHARED request that established a starting virtual address (Virt64_Addr2).
- A REQUEST=SHAREMEMOBJ for each address space.
- An example of changing the global storage access to VIEW=READONLY to the shared memory object to ORIGIN (Virt64_Addr2).



Scope Global Shared Memory Objects

```
IARV64 REQUEST=GETSHARED,
          SEGMENTS=FOUR_SEG,
          USERTKN=USRTKNB,
          ORIGIN=VIRT64_ADDR2,
          CHANGEACCESS=GLOBAL
```

For address space A

```
IARV64 REQUEST=SHAREMEMOBJ,
                 USERTKN=USRTKND...
```

For address space B

```
IARV64 REQUEST=SHAREMEMOBJ,
                 USERTKN=USRTKND...
```

```
IARV64 REQUEST=CHANGEACCESS,
          VIEW=READONLY,
          RANGLIST=RLISTPTR,
          NUMRANGE=1
```

Figure 3-9 Changing global storage access

The parameter on the IARV64 macro in Figure 3-9 that has not previously been defined is as follows:

CHANGEACCESS When CHANGEACCESS=GLOBAL is specified on the CHANGEACCESS parameter of the GETSHARED service, all address spaces currently sharing the memory object are affected, so all address spaces will get the same view. Subsequent IARV64 SHAREMEMOBJ requests for this memory object will also be affected until the next CHANGEACCESS invocation. Memory objects with CHANGEACCESS=GLOBAL support CHANGEACCESS requests without prior SHAREMEMOBJ requests.

3.2.9 Freeing a shared memory object

Virtual storage above the bar is freed using IARV64 REQUEST=DETACH. The IARV64 REQUEST=DETACH service is used to free a shared memory object also. To remove all access to the shared memory objects, issue the following requests:

- All address spaces have to remove interest from the memory object by issuing a DETACH AFFINITY=LOCAL request.
- The system interest is removed from the memory object by issuing a DETACH AFFINITY=SYSTEM request.

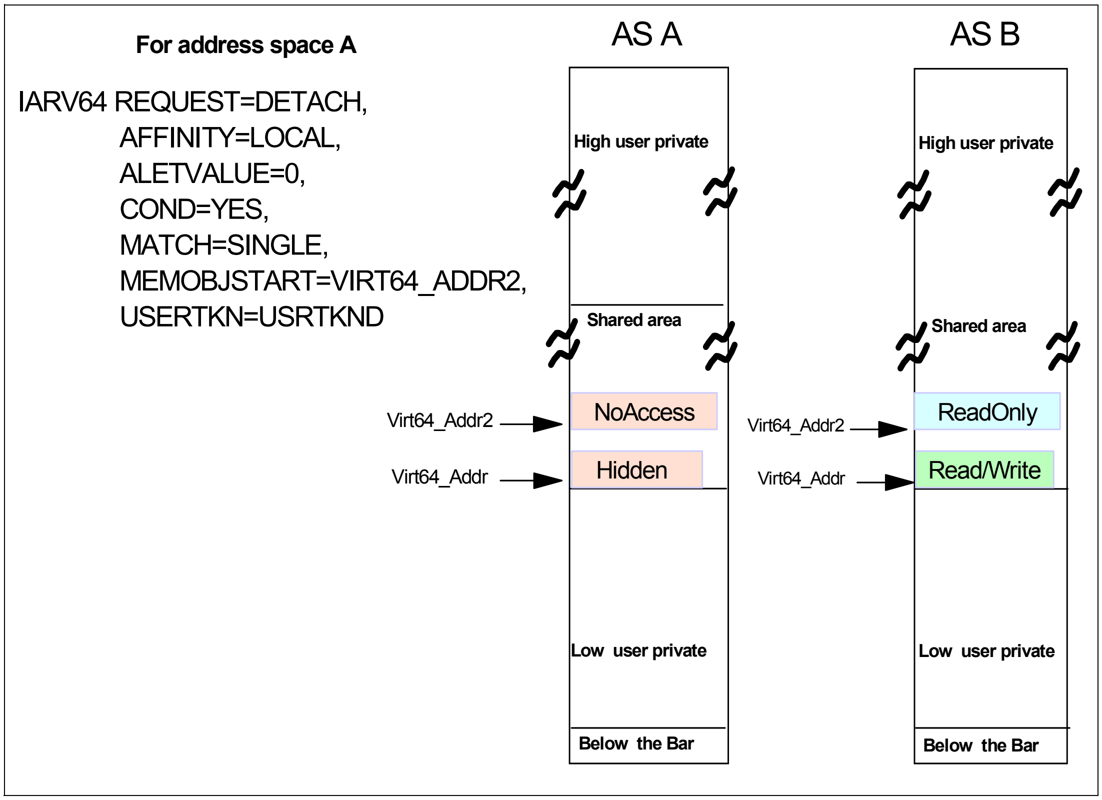
The memory object to be detached can be identified in the following ways:

- By its origin address, MATCH=SINGLE,MEMOBJSTART= .
- A related set of memory objects can be detached by providing the User Token specified when they were created. MATCH=USERTOKEN, USERTKN= .
- You can make the request conditional to avoid an ABEND, COND=YES/NO.

AFFINITY=LOCAL

When AFFINITY=LOCAL is specified against a shared memory object, the shared interest is removed from the specified address space. If the address space has no further shared interest in the object, detach also removes addressability for the specified address space.

Figure 3-10 shows an example of deleting a shared memory object from “AS A,” as specified by the ALETVALUE=0.



```
IARV64 REQUEST=DETACH,
          AFFINITY=LOCAL,
          ALETVALUE=0,
          COND=YES,
          MATCH=SINGLE,
          MEMOBJSTART=VIRT64_ADDR2,
          USERTKN=USRTKND
```

Figure 3-10 Freeing the shared memory object from AS A with AFFINITY=LOCAL

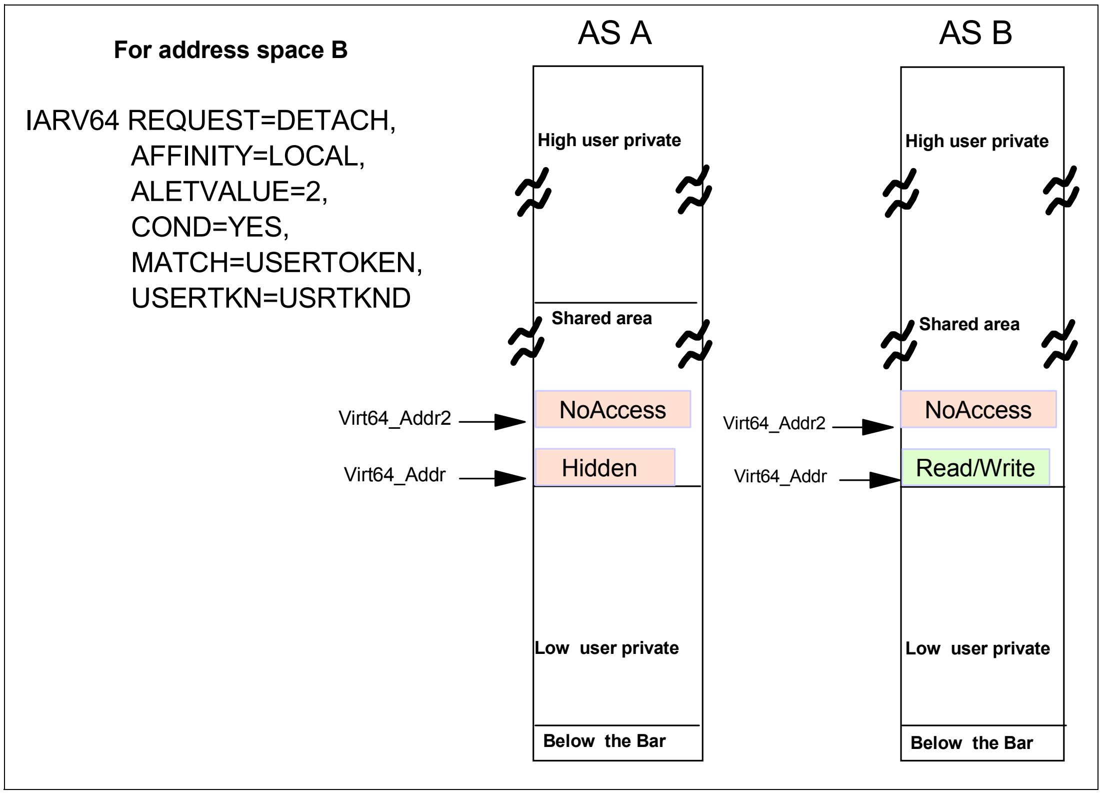
The parameters on the IARV64 macro in Figure 3-10 on page 48 are as follows:

MATCH=SINGLE,MEMOBJSTART= The memory object to be detached can be identified by its origin address.

MATCH=USERTOKEN, USERTKN= A related set of memory objects can be detached by providing the user token specified when they were created.

COND=YES/NO COND=YES helps to make the request conditional to avoid having the program abend because it asked to free a memory object that does not exist.

Figure 3-11 shows an example of deleting a shared memory object from “AS B,” as specified by ALETVALUE=2.



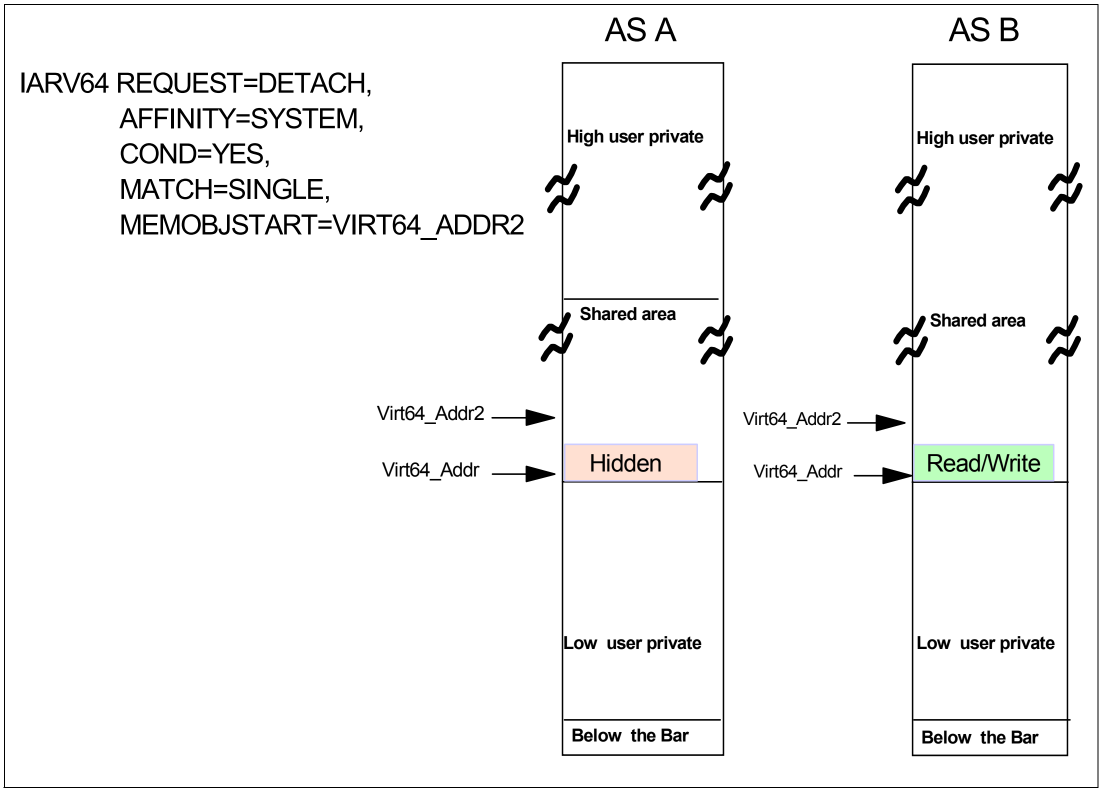
```
IARV64 REQUEST=DETACH,
          AFFINITY=LOCAL,
          ALETVALUE=2,
          COND=YES,
          MATCH=USERTOKEN,
          USERTKN=USRTKND
```

Figure 3-11 Freeing the shared memory object from AS B with AFFINITY=LOCAL

AFFINITY=SYSTEM

When AFFINITY=SYSTEM is specified against a shared memory object, the system interest in the shared memory object is removed; after this happens no new requests for the SHAREMEMOBJ are satisfied.

Figure 3-12 on page 50 shows an example of freeing a shared memory object with AFFINITY=SYSTEM.



```
IARV64 REQUEST=DETACH,
          AFFINITY=SYSTEM,
          COND=YES,
          MATCH=SINGLE,
          MEMOBJSTART=VIRT64_ADDR2
```

Figure 3-12 Freeing a shared memory object with AFFINITY=SYSTEM

The parameters on the IARV64 macro in Figure 3-12 are as follows:

MATCH=SINGLE,MEMOBJSTART= The memory object to be detached can be identified by its origin address.

MATCH=USERTOKEN, USERTKN= A related set of memory objects can be detached by providing the user token specified when they were created.

COND=YES/NO COND=YES helps to make the request conditional to avoid having the program abend because it asked to free a memory object that does not exist.

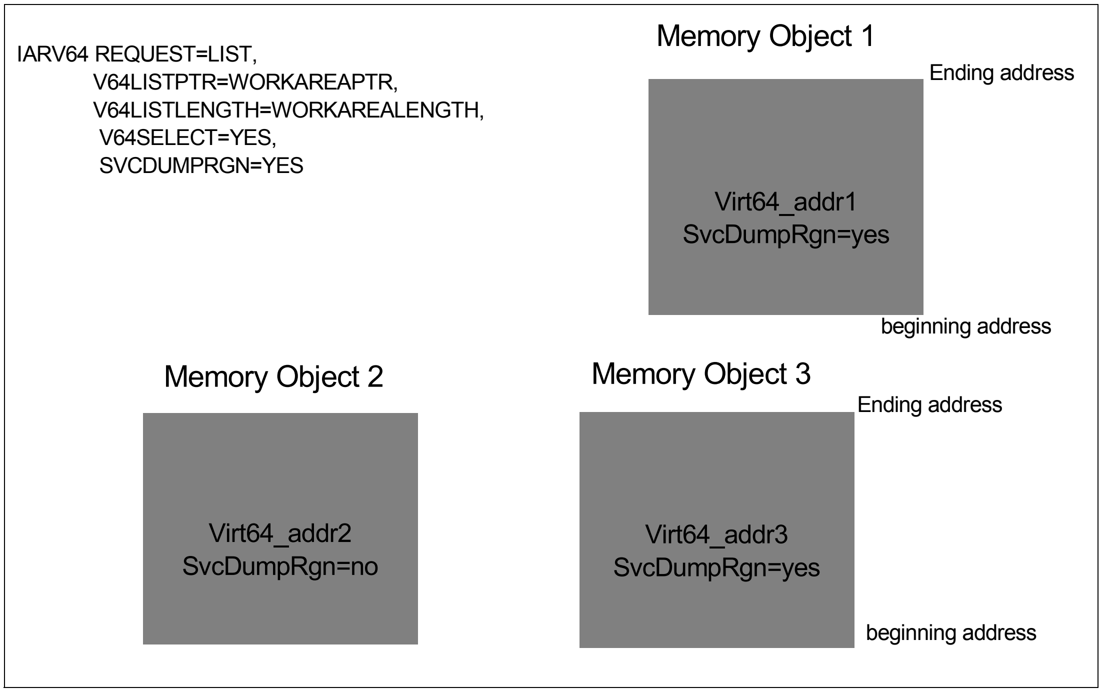
3.2.10 Obtain information about use of virtual storage

IARV64 REQUEST=LIST service is used to obtain the information about the virtual storage used by the program above the bar. This provides:

- Memory objects for the entire address space.
- Memory objects in the entire address space that have been marked as SVCDUMPRGN=YES.
- All shared memory objects that are allocated in the system.

The information is returned in a specific work area which is mapped by IARV64WA.

Figure 3-13 on page 51 shows an example of REQUEST=LIST service.



```
IARV64 REQUEST=LIST,
          V64LISTPTR=WORKAREAPTR,
          V64LISTLENGTH=WORKAREALENGTH,
           V64SELECT=YES,
           SVCDUMPRGN=YES
```

Figure 3-13 Obtain information about memory objects

The parameters on the IARV64 macro in Figure 3-13 are as follows:

V64LISTPTR A required input parameter that specifies the address of the work area which contains the results of the list request. This work area must be in fixed storage addressable from the address space for which the LIST request is made.

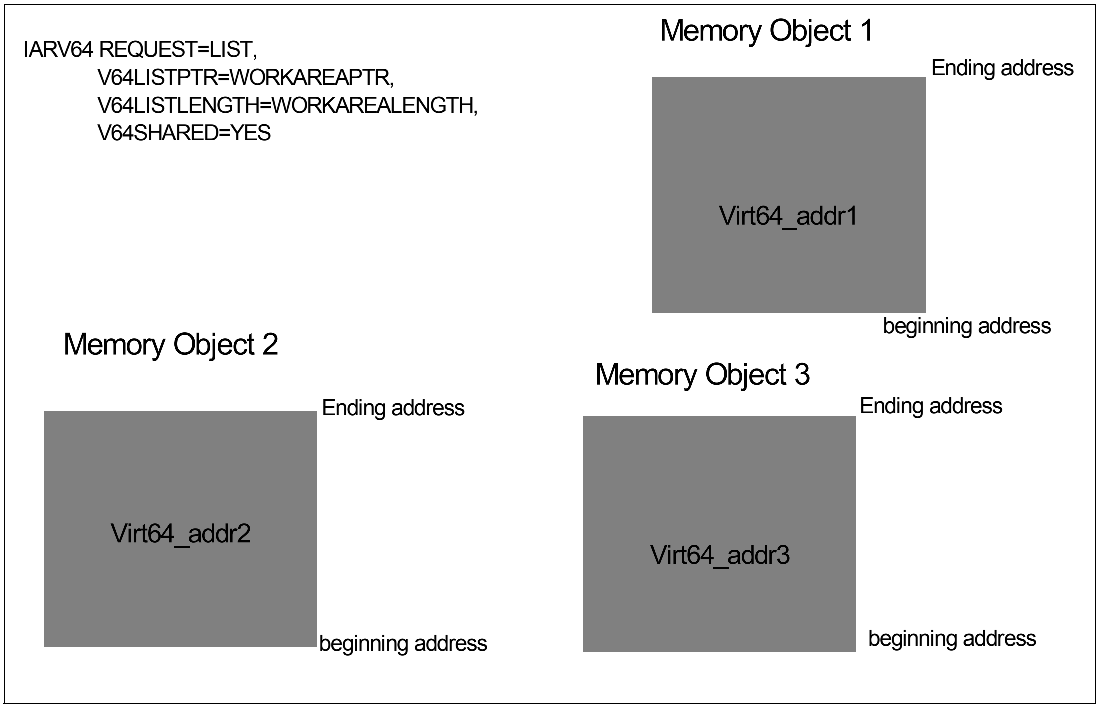
V64LISTLENGTH A required input parameter that specifies the length of the work area which contains the results of the list request.

V64SELECT=YES/NO An optional parameter that specifies whether the list request is for all allocated memory objects or for a subset of the allocated memory objects. The default is V64SELECT=NO.

SVCDUMPRGN=YES/NO An optional parameter that specifies whether the memory object should be included within the set of memory object descriptions returned by the LIST request. The default is SVCDUMPRGN=YES.

As you can see in this example, Virt64_addr2 has been created with SVCDUMPRGN=NO, so details about this object are not returned since REQUEST=LIST has specified SVCDUMPRGN=YES.

Figure 3-14 on page 52 shows another example with parameter V64SHARED.



```
IARV64 REQUEST=LIST,
         V64LISTPTR=WORKAREAPTR,
         V64LISTLENGTH=WORKAREALENGTH,
         V64SHARED=YES
```

Figure 3-14 Obtain information about memory object

The parameter on the IARV64 macro in Figure 3-5 not previously defined is as follows:

V64SHARED=YES/NO An optional input parameter that specifies whether the list of memory objects is for address space or a list of shared memory objects defined by the system via GETSHARED. The default is V64SHARED=NO.

Managing physical resources

To help the system manage main memory that backs high virtual pages of a memory object, a program can alert the system of its intended use of some of those pages.

- The program can notify the system that the data in certain pages will not be used for some time (measured in seconds) and that they are good candidates for paging out of real storage.
  - Use **IARV64 REQUEST=PAGEOUT**
- The program can notify the system that the data in certain pages will be referenced soon and that it would be good to page them into real storage if they are not already backed by real storage.
  - Use **IARV64 REQUEST=PAGEIN**
- The program can notify the system that it no longer needs the data in certain pages and that the system can free them.
  - Use **IARV64 REQUEST=DISCARDDATA**

64-bit shared virtual storage and termination

When an address space requests access to a shared memory object via a SHAREMEMOBJ request, the local address space affinity that is established is associated with the TCB whose address is stored in ASCBXTCB for the address space that is to be given access to the shared memory object.

When the TCB whose address is in ASCBXTCB terminates, the system implicitly removes any local affinities associated with that TCB.

System affinity needs to be explicitly removed via a IARV64 DETACH AFFINITY=SYSTEM request.

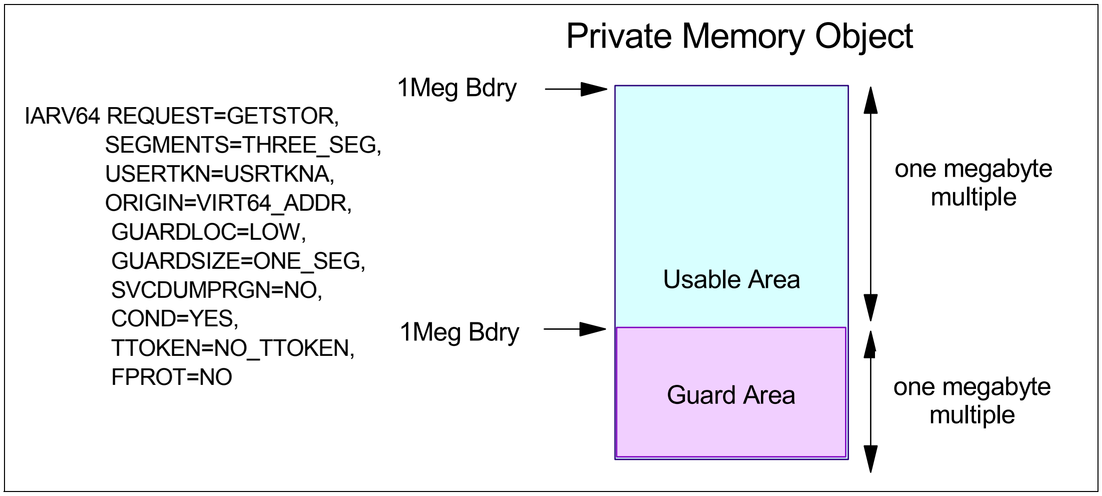
In order to avoid leaving behind orphaned objects, it is recommended that you establish RESMGRs for End of Task (EOT) and End of Memory (EOM) to ensure that any local affinities and the system affinity for a shared memory object are removed.

3.3 Multiple guard area support

z/OS V1R5 has a new enhancement called multiple guard area support.

The GUARDLOC keyword is an optional keyword on the GETSTOR request that specifies whether the guard location is at the low virtual end of the memory object or the high virtual end.

Figure 3-15 shows an example of creating a guard location at the low virtual end.



```
IARV64 REQUEST=GETSTOR,
       SEGMENTS=THREE_SEG,
       USERTKN=USRTKNA,
       ORIGIN=VIRT64_ADDR,
        GUARDLOC=LOW,
        GUARDSIZE=ONE_SEG,
        SVCDUMPRGN=NO,
        COND=YES,
        TTOKEN=NO_TTOKEN,
        FPROT=NO
```

Figure 3-15 Creating a guard area

3.3.1 Changing the amount of usable memory

Use IARV64 REQUEST=CHANGEGUARD to increase or decrease the amount of usable space in a memory object by adjusting the size of the guard area.

The boundary between the usable memory in a memory object and the guard page is moved higher or lower, depending upon the type of request and whether the guard area is high or low in the memory object.

The amount converted is in 1 MB multiples and the whole memory object can be guarded.

A TOGUARD request converts the specified amount of usable storage to guard area. The data in the converted area is released. This operation reduces the amount of virtual storage that contributes towards the MEMLIMIT for the address space.

A FROMGUARD request converts the specified amount of guard area to usable area. The converted area appears as pages of zeroes. This operation increases the amount of virtual storage that contributes towards the MEMLIMIT for the address space.

One reason for asking for a guard area is to reserve the area for future use. A second reason for using a guard area is so that the program requesting the object can protect itself from accidentally referencing storage beyond the end of a memory object and overlay data in another adjacent object.

Figure 3-16 shows an example.

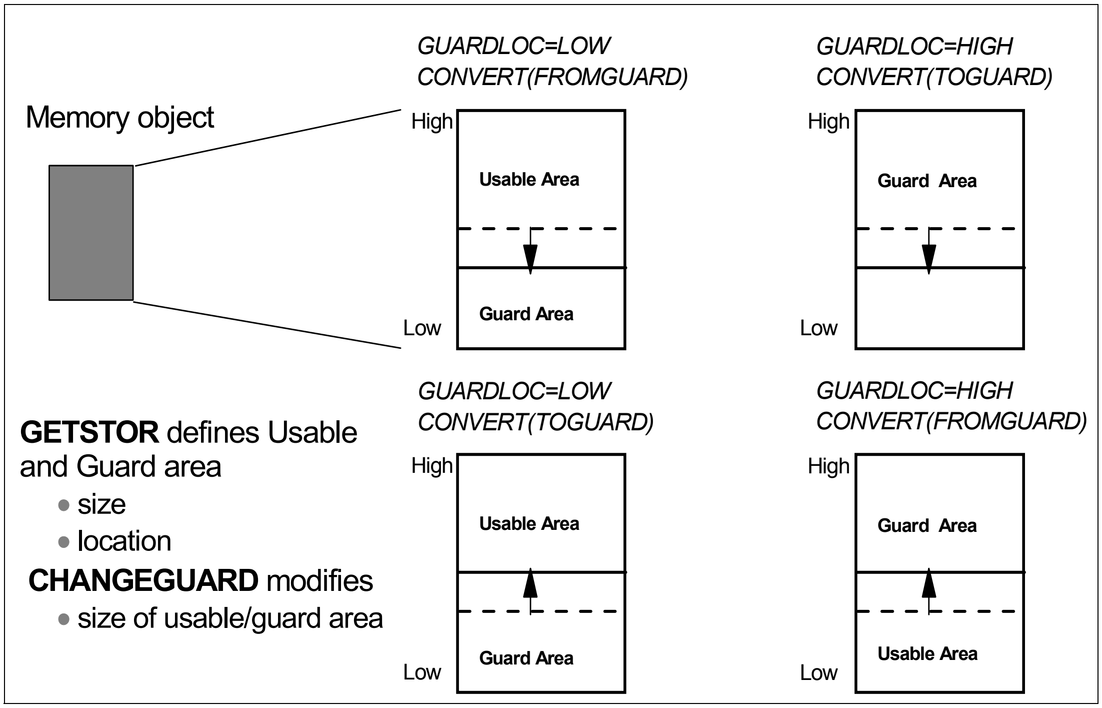
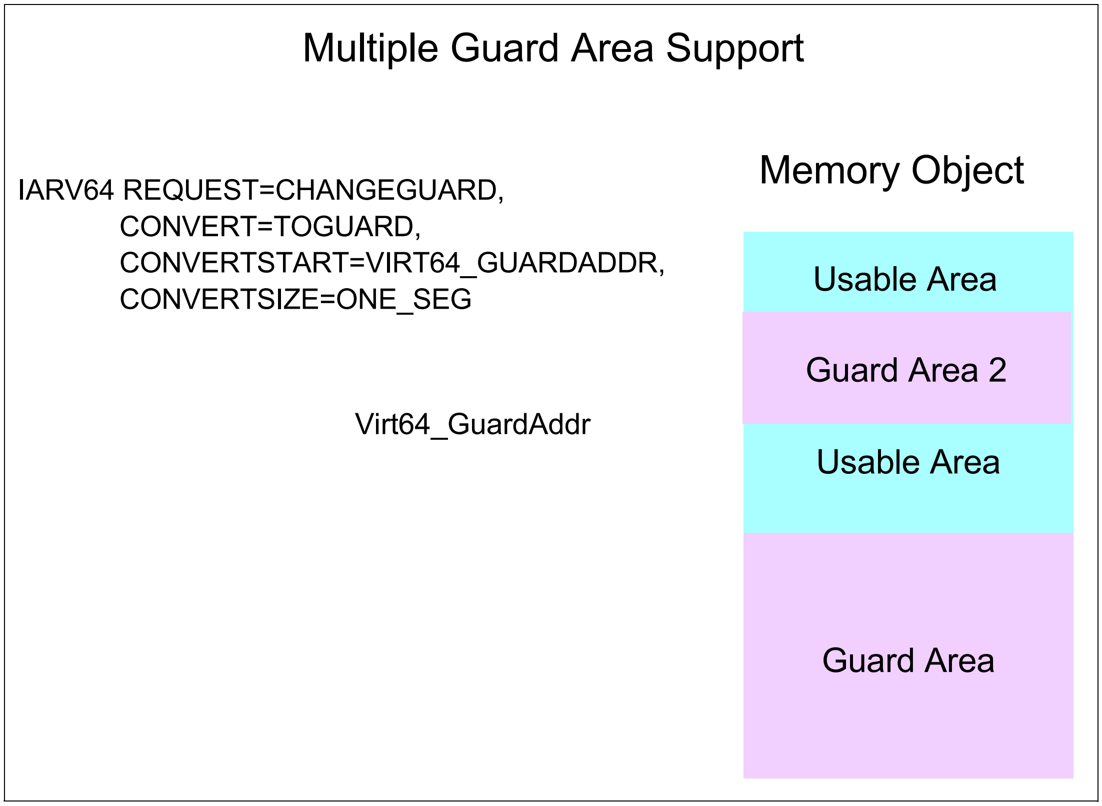


Figure 3-16 Changing the usable memory

3.3.2 Multiple guard area support

When CONVERTSTART is used with a TOGUARD request, the guard area is created from the usable area starting with the address specified and continuing for the number of segments specified by CONVERTSIZE. The convert start address must be on a segment boundary. Figure 3-17 on page 55 shows an example of multiple guard areas in the memory object.



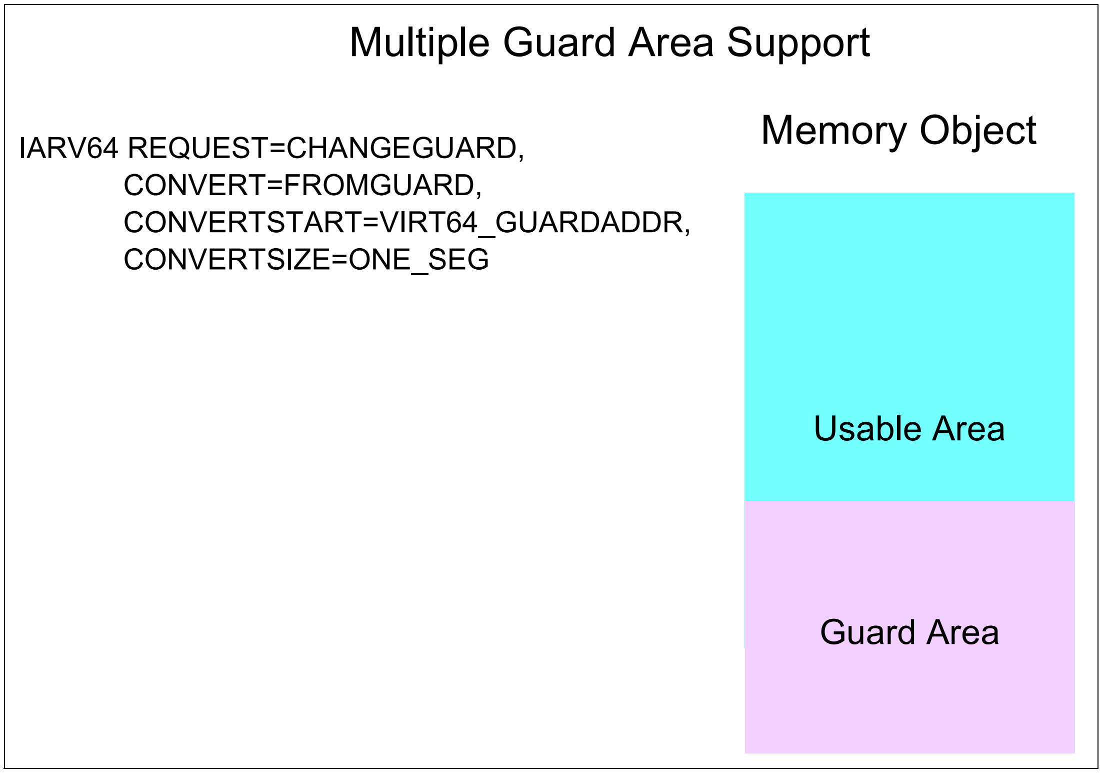
Multiple Guard Area Support

```
IARV64 REQUEST=CHANGEGUARD,
          CONVERT=TOGUARD,
          CONVERTSTART=VIRT64_GUARDADDR,
          CONVERTSIZE=ONE_SEG
```

Figure 3-17 Multiple guard area support 1

Similarly, when CONVERTSTART is used with a FROMGUARD request, the usable area is created from the guard area starting with the address specified and continuing for the number of segments specified by CONVERTSIZE.

The CONVERTSTART address must be on a segment boundary. Figure 3-18 on page 56 shows an example of converting the guard area between two usable areas to the usable area.



Multiple Guard Area Support

```
IARV64 REQUEST=CHANGEGUARD,
          CONVERT=FROMGUARD,
          CONVERTSTART=VIRT64_GUARDADDR,
          CONVERTSIZE=ONE_SEG
```

Figure 3-18 Multiple guard area support 2

3.3.3 Dumping virtual storage above 2GB

The SVC dump service has been enhanced for 64-bit support (SDUMPX macro).

Binary dumps taken to SYSMDUMP data sets were enhanced for 64-bit support.

The dumps taken to SYSABEND and SYSUDUMP data sets have not been enhanced for 64-bit support.

Note: To limit the size of an SVC dump, use SDATA=RGN with caution when dumping address spaces with memory objects that were created with the default parameter SVCDUMPRGN=YES. Data in high virtual is user data containing no control information, and is likely of low diagnostic value.

In lieu of SDATA=RGN in address spaces with high virtual, consider specifying the specific virtual address range that you want to dump.

3.3.4 64-bit virtual support restriction

When 64-bit virtual storage support was introduced in z/OS 1.2, a request to get high virtual in an address space that has currently or previously obtained a subspace, or vice versa, would cause the request to ABEND.

In z/OS V1R5 an address space is allowed to own high virtual and also have subspaces. However, the following considerations are necessary:

- IARV64 services cannot be issued in subspace mode and an attempt to do so results in a DC2 ABEND.
- Accessing 64-bIt virtual storage in subspace mode results in an 0C4 ABEND.

3.4 IPCS enhancements

The IPCS component of z/OS V1R5, HBB7708 has been changed to make its processing of multi-volume dump data sets more efficient and less prone to variations in the time needed to process the same transaction.

3.4.1 IPCS support upgrade for multi-volume dumps

Over the last several years, many dump data sets, both system dumps and SADMPs, have occupied multiple volumes. The reasons for multi-volume dump data sets are:

- The dump data set needs more than 64K tracks.

  When the dump data set requires more than 64K tracks, you cannot store the dump data set in the same volume with DSORG=PS. That is because DSORG=PS APIs use a Track Track record (TTR), with two binary bytes being used, to refer to relative track on a volume.

  Data sets with DSORG=PS-E address this problem. They substitute a 3-byte block logical transfer (BLT) value, which allows more than 64K tracks on a volume to be used. They also support striping and compression technologies.

- Sufficient space is not available in a single volume to allocate the dump data set.

  Though the dump data set occupies less than 64K tracks, it goes multi-volume because no sufficient space is available on the volume to allocate the data set.

- A multi-volume SADUMP data set requires DSORG=PS.

IPCS can process the multi-volume dump data sets without any problem. However, you may notice erratic performance of transactions for multi-volume dump data sets.

Multi-volume SADMPs must be placed in DSORG=PS data sets. For other multi-volume dump data sets, we encourage you to use DSORG=PS-E. However, it may not be always practicable to implement this standard. So, you may have many multi-volume dump data sets to be processed by IPCS.

Prior to z/OS V1R5, IPCS uses a single DCB to access a dump during the periods that you make random accesses to its records.

When a multi-volume dump is being processed and IPCS needs to read a record from a volume other than the one being accessed, it closes the DCB and reopens it to access the other volume. This exposes a complex transaction to Sysplex-wide serialization since VTOCs always need to be accessed and catalogs may also need to be accessed. As a result, a given transaction may complete very quickly once and very slowly another time. But, that's not what we all want from interactive applications!

The enhancement in z/OS V1R5 allows IPCS to employ one DCB for each volume of a multi-volume dump. The DCBs are opened on demand on the first occasion in a session that IPCS needs to access a given volume. This improves the IPCS transaction response time while processing the multi-volume dumps.

Note: Whether or not you have z/OS V1R5 installed, you should move towards the use of DSORG=PS-E data sets to hold your dumps. Their advantages are as follows:

- They can hold up to 16M records on a single logical volume.
- They can exploit more than 64K tracks on a physical volume.
- Striping can be used to accomplish a number of desirable goals.
- DSORG=PS-E data sets can be compressed when they are recorded.

Based on some informal measurements that we have made, these large data sets consume 30% to 60% less DASD space when compressed. We have also run a few complex transactions on a compressed dump and on a conventional copy of that dump and we could not notice any difference in response time.

3.5 Program management enhancements

This section provides an introduction to program management services and discusses various enhancements done to the binder in z/OS V1R5.

3.5.1 Program management services

z/OS provides program management services that let you create, load, modify, list, read, and copy executable programs. With the program management binder, you can create executable modules in either of two formats and store them (depending on the format) in PDS or PDSE libraries, or in z/OS UNIX files. The two types of executable modules are load modules and program objects; they may collectively be referred to as “program modules.” Of these two formats, program objects are the newer. Program objects remove many of the restrictions of the load module format and support new functionality. You can use the z/OS loader to load saved program modules into virtual memory for execution. You can also use the program management binder to build and execute a program in virtual storage in a single step (with some restrictions).

z/OS continues to support the older linkage editor and batch loader programs. However, the program management binder is a functional replacement for these older programs and has many additional enhancements. Because subsequent releases of z/OS might not support these components, it is strongly recommended that you use the binder exclusively. In addition, the program management binder is a functional replacement for the Language Environment prelinker, although z/OS continues to support the use of the prelinker as a separate intermediate step between compilation and binding for the relevant language translators.

Figure 3-19 on page 59 shows how the program management components work together and how each one is used to prepare an executable program.

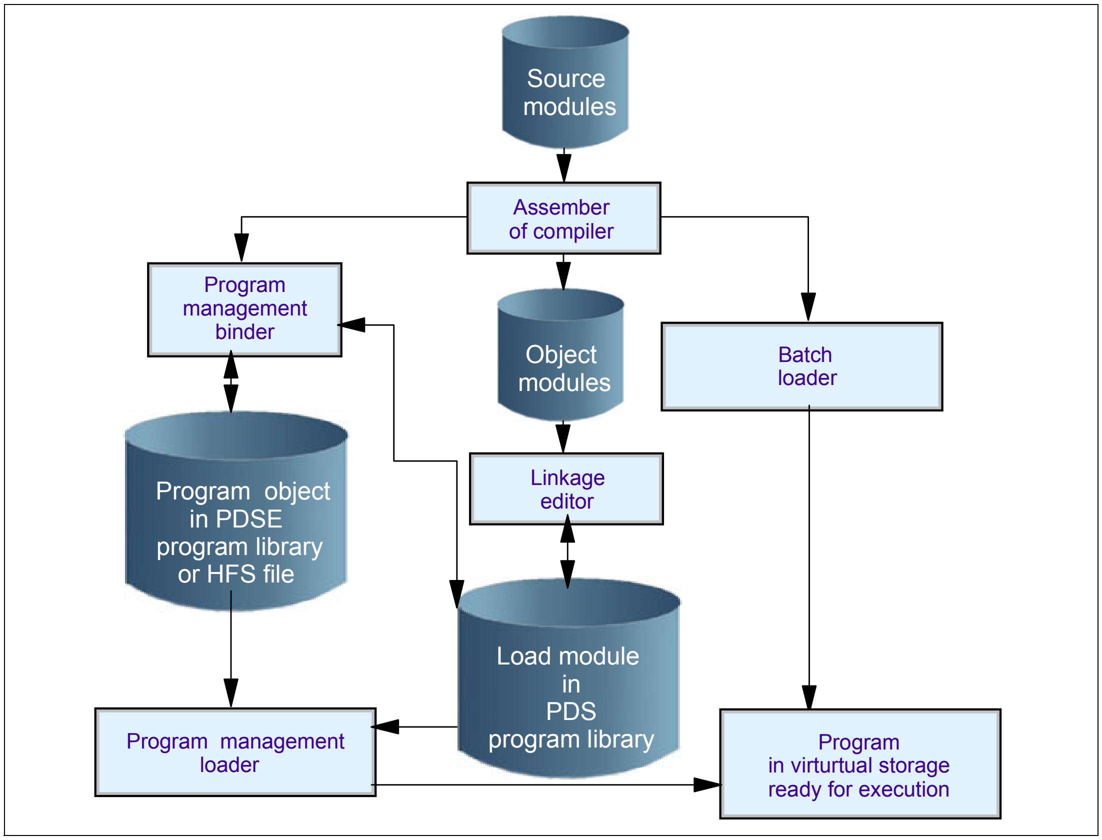

Source modules
Assember of compiler
Program management binder
Object modules
Batch loader
Linkage editor
Program object in PDSE program library or HFS file
Load module in PDS program library
Program management loader
Program in virturtual storage ready for execution

Figure 3-19 Using program management component

3.5.2 Program management binder

The binder converts the output of language translators and compilers into an executable program unit that can either be read directly into virtual storage for execution or stored in a program library.

Binding program modules

You can use the binder to:

- Convert object or load modules, or program objects, into a program object, and store the program object in a partitioned data set extended (PDSE) program library or in a z/OS UNIX file.
- Convert object or load modules, or program objects, into a load module, and store the load module in a partitioned data set (PDS) program library. This is equivalent to what the linkage editor can do with object and load modules.
- Convert object or load modules, or program objects, into an executable program in virtual storage and execute the program. This is equivalent to what the batch loader can do with object and load modules.

The binder processes object modules, load modules and program objects, *link-editing* or *binding* multiple modules into a single load module or program object. Control statements specify how to combine the input into one or more load modules or program objects with contiguous virtual storage addresses. Each object module can be processed separately by the binder, so that only the modules that have been modified need to be recompiled or reassembled. The binder can create programs to be loaded into either 24-bit address or

31-bit address storage (for example, RMODE=24 or RMODE=ANY). Beginning with z/OS V1R3, the binder can create programs which execute in 64-bit addressing mode (including support for 8-byte address constants). The binder can also create overlay load modules or program objects. Programs can be stored in program libraries and later brought into virtual storage by the program management loader.

The binder can also combine basic linking and loading services into a single job step. It can read object modules, load modules and program objects from program libraries into virtual storage, relocate the address constants, and pass control directly to the program upon completion. When invoked in this way, the binder does not store any of its output in program libraries after preparing it for execution. Like the batch loader, you can use the binder for high-performance loading of modules that do not need to be stored in a program library.

3.5.3 Enhancements to the binder compared to linkage editor

The binder also provides enhancements compared to the linkage editor. It provides advantages in the following areas:

- Support for single and multi-segment program objects
- Support for a new object module format Generalized Object File Format (GOFF)
- Easing or elimination of many linkage editor restrictions
- Application programming interface for binding programs
- Increased usability

Program objects

Depending on the library type specified by SYSLMOD, the binder creates either program objects or load modules. Program objects include and extend the functions of load modules. They are stored in partitioned data set extended (PDSE) program libraries or z/OS UNIX files instead of partitioned data set program libraries, and have fewer restrictions than load modules. Program objects remove many of the limitations and restrictions inherent in the old load module format.

Note: For details, refer to *MVS Program Management: User's Guide and Reference*, SA22-7643, in the chapter "Program objects: Features and processing characteristics."

New object module support

The binder supports a modified extended object module, produced by the COBOL, C, and C++ compilers, and a new object module format introduced in a previous release, called Generalized Object File Format (GOFF). The extended object module format (XOBJ) allows C, C++, and COBOL programmers to use long external names. The GOFF format (currently produced by the High Level Assembler and the IBM C and C++ compilers) supports long names, multipart modules, and Associated Data (ADATA).

Fewer linkage editor restrictions

The binder and program objects ease or eliminate many restrictions of the linkage editor and load modules. The linkage editor limited aliases to 64 and external names to 32767. With the binder, the number of aliases and external names for programs stored in a PDSE or z/OS UNIX file is limited only by the space available to store them.

Application programming interface

The binder also provides the ability for programs to invoke the binder and request services individually. Binder services can be invoked directly, allowing your programs to access, update, and print the contents of load modules and program objects.

Usability improvements

The binder provides other usability improvements over the linkage editor and batch loader. Messages and diagnostics have been enhanced, producing diagnostic output that is more detailed and easier to understand than the output of the linkage editor. Binder listings are also improved, printing out more complete information about the run that produced a module, including enhancements to the module map and cross reference table and a summary of the data sets used.

3.5.4 Adding an ESPIE exit routine to binder

Binder currently uses only an Extended Specify Task Abnormal Exit (ESTAE) exit to recover from errors such as program checks or logic errors. But, many recently written high level languages are called in the Language Environment (LE). LE establishes an Extended Specify Program Interruption Exit (ESPIE) exit unless specifically suppressed. The ESPIE exit gets control before binder's ESTAE exit during error recovery. This results in binder interface problems often surfacing as LE dumps rather than error codes to the caller. This makes binder logic errors and program checks harder to debug. This is because the calling program's ESPIE exits are unlikely to know how to recover from binder program checks, and are likely to cover up binder logic errors in a way which is difficult to debug.

Now the binder creates an ESPIE exit routine in addition to the present ESTAE exit, which gets control before a caller's ESPIE exit, allowing for easier debugging of binder program checks and logic errors.

New binder ESPIE exit overview

As part of the solution for better error recovery, the existing logic for recovering from interface validation program checks has been moved from the ESTAE exit to the new ESPIE exit. Now control goes to ESPIE first, which processes a program check or passes control to the ESTAE in the case of a binder logic error. The new feature has the following effects:

- The ESPIE exit routine will now return proper error codes when invalid interface information is used.
- Binder's ESPIE will receive control before a caller's ESPIE, so an IEWDUMP rather than the caller's dump routine can be output for better error diagnosis.
- A new binder option, TRAP can be used to control whether or not an ESPIE and/or ESTAE exit is created.

Binder ESPIE usage and invocation

The new binder ESPIE exit can be controlled through the new option, TRAP, which is specified as TRAP=OFF|<u>ON</u>|ABEND and is used as follows:

TRAP=<u>ON</u> Binder will establish both an ESTAE and an ESPIE exit. This traps all ABENDs and program checks that occur while the binder is in control. A key aspect is that parameter validation done by the binder API will return the documented results even if some program in the binder calling sequence has a program check exit.

TRAP=OFF Binder will not establish an ESTAE or ESPIE. This allows callers of the binder to trap all ABENDs and program checks.

TRAP = ABEND The binder establishes an ESTAE exit but not an ESPIE exit. This traps all abends, but program checks are caught by the binder only if no program in the binder calling sequence has an ESPIE exit.

The TRAP option can be specified only through the following mechanisms:

- The PARM=string specified when the binder is invoked from JCL
- The first parameter in the parameter list passed when calling the binder from another program (IEWBLINK,IEWBLOAD,IEWBLODI, IEWBLDGO)
- The IEWBIND API FUNC=STARTD OPTIONS= or PARMS=parameters

3.5.5 Binder API enhancements

Existing debug data generated by z/OS C/C++ compilers is not readily extensible to the 64-bit world. Therefore, the binder now supports the Common Debug Architecture (CDA) by saving the original source of sections across rebinds, and tracking the compile unit-section associations. This data will be kept in the new format of program object.

Tracking of compile units

A compile unit (CU) is roughly equivalent to an object module and may contain several CSECTs. All CSECTs in a single CU will be assigned the same CU number in the binder section list.

The GETN function is enhanced to let you obtain the compile unit numbers for sections specified by NTYPE=S.

NTYPE={SECTION | CLASS} specifies the type of names to be returned and counted. SECTION causes the names of all sections in the workmod, including special sections, to be returned. In addition, the compile unit (CU) numbers are provided for buffer version 6 or higher. CLASS causes the names of all classes in the workmod containing data to be returned. The value for NTYPE can be abbreviated as S or C. SECTION is the default.

Get compiler units information

The GETC function is a new binder API request that returns data mapped to a new compile unit list buffer. The COMPILEUNITLIST parameter determines which data is returned.

The syntax of the GETC call is shown in Figure 3-20.

IEWBIND FUNC=GETC
,**VERSION**=*version*
[,**RETCODE**=*retcode*]
[,**RSNCODE**=*rsncode*]
,**WORKMOD**=*workmod*
[,**COMPILEUNITLIST**=*compileunitlist*]
,**AREA**=*buffer*
,**CURSOR**=*cursor*
,**COUNT**=*count*

Figure 3-20 Syntax of GETC function

FUNC=GETC Requests that data from items in a workmod be returned to a specified location.

VERSION Specifies the version of the parameter list to be used (6 or higher).

RETCODE RX-type address or register (2-12) that specifies the location of a fullword integer that is to receive the return code returned by the binder.

RSNCODE RX-type address or register (2-12) that specifies the location of a 4-byte field that is to receive the reason code returned by the binder. Reason codes are documented as a sequence of 8 hexadecimal digits.

WORKMOD RX-type address or register (2-12) that specifies the location of an 8-byte area that contains the workmod token for this request.

COMPILEUNITLIST Determines which data is returned. If COMPILEUNITLIST is specified, one record for each compile unit in a list of compile units will be returned. If COMPILEUNITLIST is omitted, one record of each of all compile units will be returned. The header record, the first compile unit record, is built when the cursor is zero. The header record contains the ddname, pathname, or data set name when INTENT=ACCESS is specified in the CREATEW API call.

AREA RX-type address or register (2-12) that specifies the location of a CUI buffer to receive the data. The binder returns data until either this buffer is filled or the specified items have been completely moved.

CURSOR RX-type address or register (2-12) that specifies the location of a fullword integer that contains the position within the item(s) where the binder should begin processing. Specifying a zero for the argument causes the binder to begin processing at the start of the item. The cursor value is specified in bytes for items in the TEXT class, in records for all other classes. The value is relative to the start of the item. The cursor value is modified before returning to the caller.

COUNT RX-type address or register (2-12) that specifies the location of a fullword that is to receive the number of bytes of TEXT or the number of entries returned by the binder.

Get data (GETD) function

The GETD function is enhanced to return library path information which is mapped to a text buffer. Also, RELOC= is a new optional parameter used with GETD that lets you specify the location containing the compile unit list.

RELOC Specifies the location of a compile unit list. You can only use this parameter with VERSION=6 or higher. You will need to know the load segment for the data you are requesting. You can map text classes into load segments using GETN. Reloc is a single 8–byte address. The relocation address will relocate the adcons in the returned text buffer as though the program segments had been loaded at the designated address. If you do not use the RELOC parameter, it should set to zero.

Note: Refer to *MVS Program Management: Advanced Facilities*, SA22-7644 for details.

3.5.6 Fast data access

Fast data allows vendor- and user-written applications to obtain module data from program objects, residing in a PDSE or USS file, more efficiently. This service may be invoked from any language that supports the required data types. The amount of data returned by fast data may differ slightly from binder API. Binder and fast data use different algorithms to determine how much data to return in the caller's buffer area. The mapping of the data returned is the same as that returned by the binder API.

Fast data access now has an additional call interface, the request code call interface. The request code call interface provides the functions of the GETCompilunit, GETData, GETEsd and GETNames binder APIs. The new function is accessed via user-generated parameter list. The IEWBFDA macro has not been enhanced to provide the new function.

To use this interface, the caller must load the fast data access service module, IEWBFDAT. The entry point address must then be saved. If your compiler does not support a LOAD function, you need to call an assembler subroutine to issue the LOAD macro and return the entry point address. When your program is complete, use the DELETE macro to remove IEWBFDAT from your execution environment.

The fast data access request code call interface operates in a manner similar to the binder API in that a series of calls is required to extract data; at a minimum one to start a session, one to get data, and one to end the session. The caller provides a parameter list for each call that specifies the service being requested. General purpose register 1 must contain the address of that parameter list. The high-order bit in the last address of the list must be set to one. This bit signifies the end of the list of addresses.

For each call a function code, interface level, and MTOKEN must be provided. The MTOKEN must be initialized to zero by the caller before the first call. Fast data access returns the MTOKEN with a value to be used in subsequent calls. The returned value should not be modified between calls. The function code, interface level, and MTOKEN are common to all calls. Upon return from fast data access, you can examine the return and reason codes. Fullword return and reason codes are returned in registers 15 and 0 respectively.

Start service request

A sequence of calls must be started with one of the start requests services as shown in Figure 3-21. Each service is shown with the parameter list passed to it.

```
SB - Input data set indicated by DCB and BLDL DEPTR
 DC A(FUNCTION_CODE & INTERFACE_LEVEL)
 DC A(DCBPTR)
SJ - Input data set indicated by DDNAME/MEMBER or Pathname
 DC A(FUNCTION_CODE & INTERFACE_LEVEL)
 DC A(MTOKEN)
 DC A(DDNAME)   |  A(PATHNAME)
 DC A(MEMBER
SQ - Input data set indicated by token returned by CSVQUERY macro
 DC A(FUNCTION_CODE & INTERFACE_LEVEL)
 DC A(EPTOKEN)
SS - Input data set indicated by DCB and DEPTR. Caller must be in supervisor state of
     key 0.
 DC A(FUNCTION_CODE & INTERFACE_LEVEL)
 DC A(MTOKEN)
 DC A(DCBPTR)
 DC A(DEPTR)
```

Figure 3-21 Start service request

Request data

Start requests are followed by one or more occurrences of the services shown in Figure 3-22. The services are shown with the parameter list passed to it.

```
GC - Request compile unit data
    DC A(FUNCTION_CODE & INTERFACE_LEVEL)
    DC A(MTOKEN)
    DC A(CULIST)
    DC A(AREA)
    DC A(CURSOR)
    DC A(COUNT)
GD - Request module data by class
    DC A(FUNCTION_CODE & INTERFACE_LEVEL)
    DC A(MTOKEN)
    DC A(CLASS)
    DC A(SECTION)
    DC A(AREA)
    DC A(CURSOR)
    DC A(COUNT)
    DC A(RELOC)
GE - Request External Symbol Dictionary (ESD) information
    DC A(FUNCTION_CODE & INTERFACE_LEVEL)
    DC A(MTOKEN)
    DC A(CLASS)
    DC A(SECTION)
    DC A(AREA)
    DC A(CURSOR)
    DC A(COUNT
GN - Request section or class names from a program object
    DC A(FUNCTION_CODE & INTERFACE_LEVEL)
    DC A(MTOKEN)
    DC A(NTYPE)
    DC A(AREA)
    DC A(CURSOR)
    DC A(COUNT)
```

Figure 3-22 Request data

Finish requesting data

Finally, the connection created by any of the start requests is terminated by the service shown in Figure 3-23. The service is shown with the parameters passed to it.

```
EN
   DC A(FUNCTION_CODE & INTERFACE_LEVEL)
   DC A(MTOKEN)
```

Figure 3-23 Finish requesting data

The following are brief descriptions of parameters passed to all the services described previously:

DCBPTR A 4-byte pointer variable containing the address of an open DCB which represents a PDSE program object library.

DEPTR A 4-byte pointer variable containing the address of a directory entry which represents the program object library member.

DDNAME A 2-byte length field followed by up to 8 characters representing the DDNAME associated with the data set to be accessed.

MEMBER A 2-byte length field followed by up to 1024 characters representing the member name or alias in the PDSE to be accessed.

PATHNAME A 2-byte length field followed by up to 1024 characters representing the PATH associated with the file to be accessed.

CULIST Compile unit list as returned by GN function request. This parameter is used to request specific compile unit information. The compile unit list is a structure with a variable number of fullword entries preceded by a fullword count.

AREA A standard buffer which will receive the data.

CURSOR A 4-byte field indicating the position, relative to record, where the API should begin processing.

COUNT A 4-byte field which will receive the number of record entries returned by fastdata.

CLASS A 2-byte length field followed by up to 16 characters containing the name of the class of data to be returned.

SECTION A 2-byte length field followed by up to 32767 characters containing the name of the section for which data is to be returned. This is an optional parameter. If it is not specified, class data will be returned for the entire program object.

RELOC Specifies an 8-byte address to be used to relocate adcons in the returned text buffer. This is an optional parameter.

NTYPE A 1-byte character field which indicates the type of name to be returned to the caller. An NTYPE of “C” requests that class names in the program object be returned to the caller. An NTYPE of “S” requests that section names in the program object be returned to the caller. Type value is not case sensitive.

Note: For details refer to the Fast Data Access chapter in *MVS Program Management: Advanced Facilities*, SA22-7644.

3.6 64-bit compiler support

Many IBM compilers, now or soon, will be producing “RMODE 64” code and will support loading Writable Static Area (WSA) above the bar. Now the binder has been enhanced to accept object modules with RMODE 64 contents. In addition, the binder provides support for loading data portions (WSA) of an application above the bar.

Note: The High level Assembler already allows generation of RMODE 64 code.

RMODE 64 toleration

The External Symbol Directory (ESD) records specifying RMODE 64 are accepted, as follows:

- From Generalized Object File Format (GOFF) or traditional object format
- Not from prelinker External Object Module (XOBJ) format

If a module is saved to a PDS (load module) format, RMODE fields are changed to RMODE ANY. Thus there can still be at most two initial load segments, RMODE 24 and RMODE ANY.

If a module is saved to a PDSE, RMODE 64 information will be saved but hidden from loader. RMODE 64 contents in a Program Object (PO) force new PO compat level of z/OSV1R5.

3.6.1 Writable static area (WSA) above the bar

WSA is a read-write data area used by C-type reentrant programs. Now the compilers can generate a new class called C_WSA64 marked as RMODE 64. When this program is executed in a system with the appropriate loader support, C_WSA64 will be loaded above the bar. However, a single program object may not contain both the classes C_WSA and C_WSA64. This enhancement provides virtual storage constraint relief. Figure 3-24 shows a z/OS V1R5 PO format program object during execution.

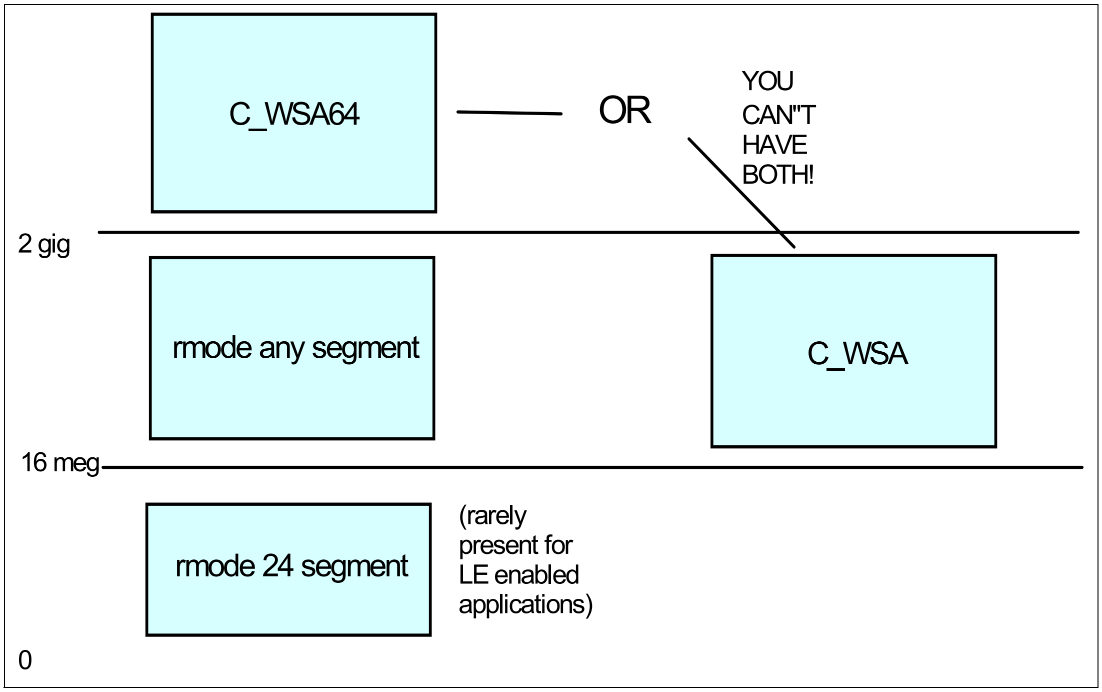


Figure 3-24 z/OS V1R5 format program object execution

When performing autocall processing against c370lib or archive object module libraries, binder will attempt to find a member which matches the AMODE of the caller. Binder allows modules with mixed AMODE 64 and non-AMODE 64 code. However, the reference and definition must match. Otherwise, an error message IEW2469E will be issued.

3.6.2 Program object formats

There are four program object formats. OS/390 DFSMS Version 1.1 introduced program object format 1 (PO1 format).

1. A PO1 format program object can be executed when using any supported release of OS/390 or z/OS. PO1 is the only format (other than the old load module format) which supports overlay structure within programs. Specifying COMPAT(PM1) or OVLY will cause a module to be saved in PO1 format.
2. Program object format 2 (PO2 format) was introduced in OS/390 DFSMS Version 1.3. A PO2 format program object can be executed on any currently supported release of OS/390 or z/OS. Specifying COMPAT(PM2) will cause a module to be saved in PO2 format.
3. Program object format 3 (PO3 format) was introduced in OS/390 DFSMS Version 1.4. All currently supported releases of OS/390 or z/OS also support PO3 format. Specifying

COMPAT(PM3) or identifying a currently supported release older than z/OS Version 1.3, such as COMPAT(OSV2R10), will cause a module to be saved in that format.

4. Program object format 4 (PO4 format) was introduced in z/OS Version 1.3. Only z/OS Version 1.3 and later support PO4 format. Specifying COMPAT(PM4) or identifying ZOSV1R3 or ZOSV1R4 will cause a module to be saved in that format.

z/OS V1R5 support

A variant of PO4 format is introduced in z/OS V1R5. It cannot be rebound or inspected (by Fast Data, the binder API, or AMBLIST) on earlier releases, but it can be loaded and executed on other systems supporting PO4 format. Specifying COMPAT(ZOSV1R5) causes a module to be saved in that format.

Each program object format introduced support for features not previously available and, except for overlay structure, each format supports all features provided by earlier formats. By default, the binder will choose the earliest format supporting all of the features being used.

> **Note:** The binder also continues to support the old load module format. Note the difference in terminology. A load module is stored in a standard partitioned data set in a format compatible with older operating systems. A program object is stored in a PDSE (for example, DSNTYPE=LIBRARY) in one of the formats listed above. The choice between load module and program object for binder output is based solely on the type of data set the program is being stored into.

3.6.3 Program object directory

Since program object internal contents have been enhanced, there needs to be some way to identify the new program object. Two fields in the program object directory (mapped by data area IEWPMAR) determine the program object level:

PMAR_LVL This is the directory level and the level used by the loader.

PMARL_PO_SUBLVL This is the level of the binder data. It is applicable to rebinding and to utility programs that call the binder to process or extract data from program objects. PMARL_PO_SUBLVL will always be zero if PMAR_LVL is less than 4.

The values of these fields are determined by the binder COMPAT option. The default is the minimum program object format that will support the requested function. This default is the same as specifying COMPAT=MIN.

Binder COMPAT option

The COMPAT options are shown in Figure 3-25.

```
COMPAT =
   {CURRENT | MIN | LKED | PM1 | PM2
   | {PM3 | OSV2R8 | OSV2R9 | OSV2R10 | ZOSV1R1 | ZOSV1R2}
   | {PM4 | ZOSV1R3 | ZOSV1R4}
   | ZOSV1R5}
```

Figure 3-25 COMPAT binder level option

CURRENT Specifies that the output is to be defined for the current level of the binder. CURRENT is the same as ZOSV1R5.

ZOSV1R5 COMPAT=ZOSV1R5 is the minimum level that can be specified if RMODE 64 has been specified by a compiler for deferred load data segments.

PM4 COMPAT=PM4 | ZOSV1R3 | ZOSV1R4 is the minimum level that can be specified if any of the following features are used.

Input modules contain 8-byte adcons, as follows:

- Any ESD record is AMODE 64
- Input contains symbol names longer than 1024, unless EDIT=NO
- A value of 64 is specified on the AMODE option or control statement

PM3 COMPAT=PM3 | OSV2R8 | OSV2R9 | OSV2R10 | ZOSV1R1 | ZOSV1R2 is the minimum level that can be specified if any of the following features are used:

- Binding modules compiled using the XPLINK attribute
- DYNAM=DLL
- XOBJ format input to the binder without going through the Language Environment prelinker, or rebinding modules containing input from such sources
- Hidden aliases (from ALIASES control statement)
- Support for deferred classes or initialized text in merge classes in GOFF format input modules or data buffers passed via the binder API

If COMPAT=PM3 and OVLY are both specified, COMPAT=PM3 is changed to PM1. PM3 supports all PM2 and PM1 features.

PM2 COMPAT=PM2 is the minimum level that can be specified if any of the following are used:

- User-defined classes passed in GOFF format input as well as certain other information supported only in GOFF format
- Names (from input modules or created by control statements which cause renaming) that are longer than 8 bytes
- Use of RMODE=SPLIT

If OVLY is specified, COMPAT=PM2 is changed to PM1. PM2 supports all PM1 features.

PM1 This is the minimum level which supports binder program objects.

OVLY is supported, and will force PM1 to be used.

MIN This is the default, and indicates that the binder should select the minimum PM level that supports the features actually in use for the current bind.

LKED Specifies that certain binder processing options are to work in a manner compatible with the linkage editor.

Note: A program object cannot be REBOUND using a binder of a level lower than the compat level. Utility programs which call the binder will also fail unless the binder level and the compat level match. A program cannot be LOADED or EXECUTED at an operating system level lower than the one which supports the PMAR level (given by the PMAR_LVL field). PMAR level 4 was supported starting in z/OS V1R3.

Level reporting

The level is reported in the binder listing (or can be found by using the AMBLIST service aid). For example:

```
PROGRAM TYPE     PROGRAM OBJECT(FORMAT 4 OS COMPAT LEVEL z/OS V1R5)
```

Message changes

The following messages have been changed:

- **IEW2469E** `THE ATTRIBUTES OF A REFERENCE TO` *`symbol name`* `DO NOT MATCH THE ATTRIBUTES OF THE TARGET SYMBOL. REASON 3`

  "Reason 3" indicates either the reference or the target is in AMODE 64 and the AMODEs do not match.

- **IEW2491E** `BOTH CLASSES C_WSA AND C_WSA64 ARE PRESENT IN THE MODULE.`
- **IEW2618I** `RMODE 64 ESD ATTRIBUTES HAVE BEEN CHANGED TO RMODE ANY.`
- Messages diagnosing program object format and contents mismatches have been enhanced to incorporate the new sublevel field. For example:

  IEW2606S `MODULE INCORPORATES z/OSV1R5 PROGRAM OBJECT FEATURES AND CANNOT BE SAVED IN A z/OSV1R3 COMPATIBLE PROGRAM OBJECT FORMAT.`

3.7 JCL enhancements

JCL has been enhanced to add JCL OUTPUT keywords for Infoprint Server E-mail support. Also, the PRMODE keyword was added to the PRINTDEV JCL statement. See "JCL OUTPUT keywords for Infoprint Server e-mail support" on page 182.

3.7.1 PRMODE keyword in PRINTDEV JCL

The PRMODE keyword is added to the PRINTDEV JCL statement, which is used by PSF to determine what type of optional processing should be performed. This keyword allows the user to set a default value for a printer in the PRINTDEV statement, rather than requiring all JCL to contain the PRMODE keyword on the OUTPUT statement.

Note: This function has been rolled down to z/OS V1R2, V1R3, and V1R4 via APAR OA02478.

PRMODE indicates the default processing mode PSF uses to print data sets containing both single-byte and double-byte fonts. The following options are used on the PRMODE keyword:

PRMODE=SOSI1 Specifies that each shift-out, shift-in code is converted to a blank and a Set Coded Font Local text control.

PRMODE=SOSI2 Specifies that each shift-out, shift-in code is converted to a Set Coded Font Local text control.

PRMODE=SOSI3 Specifies that the shift-in code is converted to a Set Coded Font Local text control and two blanks. A shift-out code is converted to a Set Coded Font Local text control.

PRMODE=SOSI4 Specifies that each shift-out, shift-in code is to be skipped and not counted when calculating offsets for the print data set. SOSI4 is used when double-byte character set (DBCS) text is converted from ASCII to EBCDIC. When SOSI4 is specified, the page definition offsets are correct after conversion; therefore, the user does not need to account for SOSI characters when computing FIELD offsets. The data conversion that PSF makes for SOSI4 is the same as for SOSI2.

Note: You must be running z/OS V1R2 or higher with APAR OA02478 to use PRMODE in the PRINTDEV statement. PSF only uses this parameter if you are not using the Printer Inventory or Exit 7 and the PRMODE keyword is not specified on the OUTPUT JCL statement. If this parameter is not specified in the PRINTDEV statement, the Printer Inventory, or Exit 7, and the PRMODE keyword is not specified on the OUTPUT JCL statement, PSF defaults to either line data or MO:DCA™-P, depending on the type of data stream.

3.8 SMF record type 6 enhancements

The SMF Type 6 record changed in two ways in z/OS V1R5. The mapping macro for the Type 6 record was restructured and separated into two macros. Additionally, new fields were added to the file transfer section for IP PrintWay extended mode. These changes were implemented to maintain compatibility with previous levels of the SMF Type 6 record and its mapping. The segments of the Type 6 record that are written only by IP PrintWay or PSF were moved to a new mapping macro, AOPSMF6. The SMF Type 6 mapping macros are:

- **IFASMFR** - maps segments written in all Type 6 records
  - Header
  - First extension
  - Common section
  - Enhanced sysout section
- **AOPSMF6** - maps segments written by IP PrintWay and PSF
  - Second extension (APA section)
  - Multi-bins section
  - Multi-bins counter section
  - File transfer section

The IFASMFR macro internally invokes the AOPSMF6 macro. Thus, there is no change to the invocation from the user accounting program, if any, the way it was done in previous releases.

3.8.1 IP PrintWay support

The flag bytes indicate whether IP PrintWay basic mode or IP PrintWay extended mode wrote the SMF Type 6, as follows:

- IP PrintWay basic mode
  - SMF6SBS = 9
  - SMF6INDC = 1
- IP PrintWay extended mode
  - SMF6SBS = 9
  - SMF6INDC = 7

The content of the file transfer section differs between the SMF Type 6 record written by IP PrintWay extended mode and IP PrintWay basic mode. New fields have been added that will only be present for IP PrintWay extended mode.

SMF fields

The number of bytes transmitted to the printer now appears in two fields, SMF6BYTE, and the new field, SMF6BYTD. For files smaller than 2 gigabytes in size, these two fields will contain the same value. For files larger than 2 gigabytes, SMF6BYTE contains x'7FFFFFFF' and SMF6BYTD contains the actual number of bytes transferred, stored as a 64-bit integer.

A new level indicator, SMF6FTL has been added to the file transfer section. This bit indicates which format of the file transfer section is being written. If SMF6FTL=1, this indicates that the new fields described here are present.

The new SMF6URI field contains the Uniform Resource Indicator (URI) for the target device. The length of this variable-length field is contained in SMF6URIL.

The fields that contained the dotted decimal IP address for IP PrintWay basic mode (SMF6IP1, SMF6IP2, SMF6IP3, SMF6IP4) contain binary zeros when IP PrintWay extended mode writes the SMF Type 6 record.

The changes are illustrated in Figure 3-26.

```
New fields for IP PrintWay extended mode:
SMF6BYTD
   Total bytes transmitted to printer
   64-bit integer; supports larger file sizes
   SMF6BYTE will also contain byte count (up to 2 gigabytes)
SMF6FTL
   Level indicator for the file transfer section
      1 = IP PrintWay extended mode
      0 = IP PrintWay basic mode
SMF6URI
   Indicates what protocol was used
      Address of the target printer
SMF6URIL
   Length of the SMF6URI field
SMF6IP1, SMF6IP2, SMF6IP3, SMF6IP4 contain binary zeros for IP PrintWay extended mode.
```

Figure 3-26 New fields in SMF record type 6

Table 3-1 shows the format of the SMF6URI field for each protocol supported by IP PrintWay extended mode.

Table 3-1 SMF6URI field format in SMF record type 6

Protocol	Format	Example
Lpr	lpr://hostname/queue	lpr://lexmark.xyz.com/TEXT
direct sockets	direct_sockets://hostname:port	direct_sockets://laserjet.xyz.com:9100
e-mail	mailto:	mailto:
IPP	Any URI format for IPP or HTTP as protocol type	ipp://printer1.xyz.com/myqueuename http://printer1.xyz.com:631/ipp

> **Note:** The protocol type is indicated at the beginning of the string. For the LPR protocol, the SMF6URI field contains the hostname followed by the queue name. The queue name is found in two places. It is contained in SMF6PRTQ as before, and can also be parsed as the third subparameter of the SMF6URI field.

For the direct sockets protocol, the SMF6URI field contains the hostname followed by the port number.

E-mail protocol

For the e-mail protocol, only the protocol is identified. SMF6URI will always contain the value “mailto:”. The actual e-mail addresses of the recipients are not captured in the SMF Type 6 record.

IPP protocol

For IPP, any URI or HTTP format can be used to identify the target printer. The SMF6URI field will show IPP or HTTP as the protocol, depending on how the URL was specified in the printer definition.

3.8.2 Migration considerations

Accounting programs may require modifications to use the new fields in the SMF Type 6 record, and should be recompiled to pick up the new mapping macros.

If you are using the IP PrintWay SMF exit, ANFUXSMF, the exit must be recompiled with the new mapping macros. Some functions previously performed in ANFUXSMF apply only to IP PrintWay basic mode. You should examine your exits to see if they are still required for IP PrintWay extended mode. If the processing is applicable to both basic mode and extended mode, both versions of IP PrintWay can call the same SMF exit. For IP PrintWay basic mode, it is possible to use different versions of an exit for each FSS by using a STEPLIB to the exit. For IP PrintWay extended mode, only one version of ANFUXSMF can be called. The exit must either be in the system linklist or pointed to in the STEPLIB environment variable when Infoprint Server daemons are started.

3.9 3590 model H tape drive support

This enhancement provides support in MVS Allocation for the latest offering in the 3590 family, the newly-introduced 3590 Model H drive. The software supports the new 384-track recording format of the 3590 Model H drive. This support will allow Model H drives to coexist with Model B and Model E drives in the same library.

The Dynamic Allocation Text Unit, DALINCHG has been changed to allow the new recording technology. The following two HEX values are added:

53 High Performance Cartridge Tape requested; 384-track recording technique requested.

54 Extended High Performance Cartridge Tape requested; 384-track recording technique requested.

Note: For details, refer to *MVS Programming: Authorized Assembler Services Guide*, SA22-7608.

3.10 File sequence number >9999 support

Currently the maximum value of the file sequence number (also referred to as data-set-sequence-number) of a data set on a tape volume set is 9999. This support will increase the maximum to 65535 for IBM Standard Label tapes (SL, SUL and LTM), unlabeled (NL) tapes, and when using Bypass Label Processing (BLP). This will allow stacking of files on tape volumes to fully utilize large capacity tape cartridges. It is not applicable to ISO/ANSI (AL) tapes.

3.10.1 JCL support

JCL has not been changed to include this support. The sequence number specified in the JCL LABEL parameter is still limited to a maximum of 9999. So, this support is only available by one of the following methods:

- Using the RDJFCB macro to obtain the JFCB, updating the file sequence number in the JFCB and using the OPEN, TYPE=J macro if the data set isn't cataloged.
- If the data set is cataloged and a file sequence number is not specified in the LABEL parameter, then the OPEN macro can be used. The file sequence number will be provided from the catalog.

3.10.2 DYNALLOC support

The DALDSSEQ text unit has been changed to allow an increase of the maximum value from 9999 to 65535. DALDSSEQ specifies the relative position of a data set on a tape volume (data set sequence number). It is mutually exclusive with the SYSOUT (DALSYSOU) key. When you code DALDSSEQ, # must be one, LEN must be two, and PARM contains the sequence number. The maximum PARM value is FFFF (65535).

3.11 Unicode enhancements

The Unicode standard is the universal character encoding standard used for representation of text for computer processing. It is fully compatible with the second edition of International Standard ISO/IEC 10646-1:2000, and contains all the same characters and encoding points as ISO/IEC 10646. The Unicode standard also provides additional information about the characters and their use. Any implementation that is conformant to Unicode is also conformant to ISO/IEC 10646.

Unicode provides a consistent way of encoding multilingual plain text and brings order to a chaotic state of affairs that has made it difficult to exchange text files internationally. Computer users who deal with multilingual text—business people, linguists, researchers, scientists, and others—will find that the Unicode Standard greatly simplifies their work. Mathematicians and technicians, who regularly use mathematical symbols and other technical characters, will also find the Unicode Standard valuable.

The design of Unicode is based on the simplicity and consistency of ASCII, but goes far beyond ASCII's limited ability to encode only the Latin alphabet. The Unicode Standard provides the capacity to encode all of the characters used for the written languages of the world. It uses a default 16-bit encoding that provides code points for more than 65,000 characters. To keep character coding simple and efficient, the Unicode Standard assigns each character a unique numeric value and name.

While 65,000 characters are sufficient for encoding most of the many thousands of characters used in major languages of the world, the Unicode standard and ISO 10646 provide an extension mechanism called UTF-16 that allows for encoding as many as a million more characters, without use of complex modes or escape codes. This is sufficient for all known character encoding requirements, including full coverage of all historic scripts of the world.

The following sections cover the enhancements made to z/OS Unicode.

3.11.1 Unicode and DB2 V8

Unicode processing is integral to DB2 V8. It uses z/Architecture unicode hardware support. DB2 V8 cannot function without certain unicode conversion tables. If the installation does not customize for unicode conversion table (UNI=xx parmlib member), z/OS loads a pre-built image of the required conversion table at DB2 initialization. If the installation has customized for unicode conversion tables, but did not include all the tables required for DB2, DB2 will complain when the tables are needed.

3.11.2 Unicode collation service

Unicode string collation is a culturally correct comparison of two Unicode strings. In order to do so, a collation key is generated for each input string, and the two keys are then binary compared. The Unicode collation algorithm is described in detail in the Unicode consortium's technical report #10. For the detailed report, refer to URL:

```
http://www.unicode.org/unicode/reports/tr10/
```

The collation service can be called through stub routine CUNLOCOL for AMODE (31), or CUN4LCOL for AMODE (64). To create a Unicode image with collation, the COLLATE control statement must be present in the image generator (CUNMIUTL).

Collation levels

Collation works under two basic schemes: binary comparison between two Unicode strings, and generation of a sort key vector. To make a binary comparison or generate a sort key vector, it is necessary to specify a collation level (CUNBOPRM_Coll_Level BIN(8)). There are four collation levels (Ln) supported, which are applied in a inheritance way, as follows:

- L1 = alphabetic ordering
- L2 = L1 + diacritic ordering
- L3 = L2 + case ordering
- L4 = L3 + tie-breaking

The collation levels are taken as a reference to build the collation elements array, which is a collection of "weights" for each character in a determinate level. In this array, all the collation rules are applied in order to generate a binary comparison or a sort key vector. The input for this array comes from the allkeys.txt Unicode file. The allkeys.txt Unicode file can be found at:

```
http://www.unicode.org/unicode/reports/tr10/allkeys.txt
```

An example is shown in Figure 3-27.

```
...
0061  ; [.0A15.0020.0002.0061] # LATIN SMALL LETTER A
...
0041  ; [.0A15.0020.0008.0041] # LATIN CAPITAL LETTER A
006F  ; [.0B4B.0020.0002.006F] # LATIN SMALL LETTER O ...
```

Figure 3-27 Binary comparison example

In this example, The character "a" (latin small letter) has the code point "0061" and contains weights for L1 = 0A15, L2 = 0020, L3 = 0002, and L4 = 0061.

If we compare 0061 (small letter "a") and 0041(capital letter "A") we will get a difference up to L3 (case), where 0002 < 0008.

Collation sample - binary comparison

This is the most common use for the collation service. Two Unicode strings are input by the caller to be compared (collated) in a culturally correct manner. Prior to collation, the caller can optionally set a desired collation level, alternate weighting, and other options in the collation parameter area, to specify a particular comparison type. Once the collation service is called, it will return a compare result and a return and reason code. For two given Unicode input strings A and B, the comparison result shows how one string is related to the other in the following way:

- “-1” means Str1 < Str 2
- “0” means Str 1 = Str 2
- “1” means Str1 > Str 2

L1 comparison - alphabetical order

For example:

- Str 1 = “Michael Jordan”
- Str 2 = “Michael jackson”

Result: The difference is between the letter “o” and “a” so, the comparison result is “1” (Str 1 > Str 2).

L2 comparison - diacritical order

For example:

- Str 2 = “MiguEl Martinez”
- Str 1 = “Miguel Martínez”

Result: At the L1 there is no difference but, there is at the second level between “i” and “í”. In this case, the result is “-1” (means Srt 1 < Str 2).

L3 comparison - case order

For example:

- Str 1 = “Miguel Martínez”
- Str 2 = “MiguEl Martínez”

Result: At the L1 and L2 there are no differences. The difference it's between small letter “e” and capital letter “E” so, the result is “-1” (Str 1 < Str 2).

Note: If we compare Str1 and Str 2 “up to Level 2” we will get 0 as result, which means that there are no differences at Level 2.

L4 comparison - tie-breaking

This level is used to apply some specific collation rules for some code points called *variables*. The collation elements that are marked with an asterisk in a Unicode allkeys.txt file are known as *variable collation elements*.

For example:

- 0020 ; [*0209.0020.0002.0020] # SPACE

Applying “Shifted rule,” its weights will be changed to:

- 0000 0000 0000 0209

Note: Whatever “weight” value contains 0000 is considered ignorable. In a practical case, that weight is not compared nor added to the sort key vector (it depends on the case).

3.11.3 Collation sample - sort key vector generation

The sort key vector is the opposite action to the binary comparison which is the default option. The sort key vector will be generated if CUNBOPRM_SKey_Opt BIT(1) is ON. This behavior gives to the user the possibility to save and/or compare the sort keys by its own algorithms.

Collation sort key vector format is:

```
wwww0000xxxx0000yy0000zzzz
```

Where:

wwww Represents level one (two bytes)

xxxx Represents level two (two bytes)

yy Represents level three (one byte)

zzzz Represents level four (two bytes)

0000 Represents the collation level separator (two bytes).

Figure 3-28 shows an example.

```
Consider Str 1 = ab and Collation Level 4

weights from the allkeys.txt file

0061  ; [.0A15.0020.0002.0061] # LATIN SMALL LETTER a
0062  ; [.0A29.0020.0002.0062] # LATIN SMALL LETTER b

Sort Key format:

0A150A2900000020002000000202000000610062
```

Figure 3-28 Example of sort key format

Collation rules

Collation works with some rules/options to make binary comparisons and sort key vectors. The rules are set up in the collation parameter list field CUNBOPRM_Mask BIT(16) as follow:

- Variable option
- Compare order
- Sort key generation
- Normalization type

Figure 3-29 on page 78 shows the description of CUNBOPRM_Mask. This parameter is 2 bytes in length.

```
2 CUNBOPRM_Mask  BIT(16),           ! Collation Mask
3 CUNBOPRM_Variable_Opt BIT(3),     ! Where :
                                    !   0 - Shifted
                                    !   1 - Blanked
                                    !  10 - Non Blanked
                                    !  11 - Shift-Trimmed
                                    ! 100 - No Variable Behaviour
                                    !       (NAVARIABLECE)
3 CUNBOPRM_Cmp_Order BIT(1),        ! Where :
                                    !   0 - Forward
                                    !   1 - Backward (French)
3 CUNBOPRM_SKey_Opt BIT(1),         ! Where:
                                    !  0 - Not Get Sork Key
                                    !  1 - Get Sork Key
3 CUNBOPRM_Norm_Type BIT(3),        ! Normalization Form
                                    !000 - No Apply Normalization
                                    !001 - Apply NFD
                                    !010 - Apply NFC
                                    !011 - Apply NFKD
                                    !100 - Apply NFKC
3 * Char(1),                        !Reserved
```

Figure 3-29 Collation mask sub-field description

Variable collation elements

These rules are applied to the variable collation elements. They are declared as CUNBOPRM_Variable_Opt BIT(3) and contain one the following values:

- 0 - Shifted (default)

  Variable collation elements are set to be ignorable at levels one through three. In addition, the L4 weight is appended whose value depends on the type. For example:

  Before 0020 ; [*0209.0020.0002.0020] # SPACE

  After [.0000.0000.0000.0209]

- 1 - Blanked

  Variable collation elements are reset so that their weights at levels one through three are zero. For example:

  Before 0020; [*0209.0020.0002.0020] # SPACE

  After [.0000.0000.0000.0020]

- 2 - Non Blanked

  Variable collation elements are not reset to ignorable, and get the weights explicitly mentioned in the file. For example:

  Before/After 0020 ; [*0209.0020.0002.0020] # SPACE

- 3 - Shift-Trimmed

  The same as Shifted, except that all trailing FFFFs are trimmed from the sort key. This option is designed to emulate POSIX behavior.

- 4 - No variable behavior

Compare order

In some languages (notably French), accents are sorted from the back of the string to the front of the string and work with Level 2 (diacritic). CUNBOPRM_Cmp_Order BIT(1) contains one the following values:

0 - Forward (Default)

1 - Backward (French)

Sort key generation

Collation has two options: make a binary comparison between two Unicode strings or generate a sort key vector. CUNBOPRM_SKey_Opt BIT(1) contains one the following values:

0 - Do not Get Sort Key (Default)

1 - Get Sort Key

Normalization type

This option specify the normalization type to the Unicode string inputs (whether or not you choose to make a binary comparison or sort key vector). CUNBOPRM_Norm_Type BIT(3) contains one the following values:

- **0** - Do not Apply Normalization (default)
- **1** - Apply NFD (canonical decomposition)
- **2** - Apply NFC (canonical composition)
- **3** - Apply NFKD (compatibility decomposition)
- **4** - Apply NFKC (compatibility composition)

Work buffer length considerations

The work buffer length has the same consideration for both 31 bit and 64 bit. There are two main considerations related with the collation level specified:

- **Case 1** - CUNBOPRM_Coll_Level = 1, 2 or 3.

  For this case, you must have to consider at least twice the value of the source length (CUNBOPRM_SrcX_Buf_Len * 2)

  Where X could be Src1 or Src2.

- **Case 2** - CUNBOPRM_Coll_Level = 4.

  For this level, you must provide at least three times the value of the source (SrcX_Buf_Len * 3)

  Where X could be Src1 or Src2.

Target buffer consideration

The target buffer length has the same considerations for 31 bit and 64 bit. This section is a reference for setting the size of the CUNBOPRM_TargX_Buf_Len parameter (where X could be Src1 or Src2).

- **Binary comparison** - The target buffer is used to normalize the Unicode strings, depending on the type of normalization services that are required. The size of the target buffer will depend on the normalization forms. As a general parameter (not a rule) for decompositions (NFD, NFKD) we can provide two times the SrcX buffer length; and for compositions, we might specify the same size as the SrcX buffer length.

  If you do not require to normalize during a binary comparison you cannot provide target buffers.

- **Sort Key Vector** - In order to generate the sort key vector, we can follow these rules based in the collation level:

  For L1, TargX_Buff_Len >= CUNBOPRM_SrcX_Buf_Len * 2

  For L2, TargX_Buff_Len >= CUNBOPRM_SrcX_Buf_Len * 4 + 2

  For L3, TargX_Buff_Len >= CUNBOPRM_SrcX_Buf_Len * 5 + 2

  For L3, TargX_Buff_Len >= CUNBOPRM_SrcX_Buf_Len * 6 + 2

  Where:

  X makes reference to a Src1, Wrk1,Targ1 or Src2, Wrk2, Targ2. The buffers works in parallel (1 with 1 and 2 with 2, as needed).

New reason codes

The following new reason codes are added for return code 8.

RETURN CODE 8

- `CUN_RS_INV_COLL_LEVEL FIXED(31) CONSTANT(10)`

  An unsupported Collation Level was specified.

  Action: Use a valid collation Level in IDF CUNBOIDF

  Module: cunmocol/cun4mcol

- `CUN_RS_NO_SERV_AVAILABLE FIXED(31) CONSTANT(11)`

  An unavailable service was called in the active image.

  Action: Use SET command to load an image with the service available.

  Module: cunlocol/cun4lcol

- `CUN_RS_WRK_EXHAUSTED FIXED(31) CONSTANT(12)`

  The work buffer was exhausted before all the code points (source buffers) were represented in collation elements - (weights) (work buffers).

  Action: Recall the service with new parameter value in the work buffer.

  Module: cunmogce/cun4mgce

- `CUN_RS_TARG_EXHAUSTED FIXED(31) CONSTANT(13)`

  The target buffer was exhausted before all collation elements (work buffers) were represented in a sort key (target buffers).

  Action: Recall the service with new parameter value in the target buffer.

  Module: cunmogsk/cun4mgsk

New messages

The following new messages are added.

- **`CUN1032W`** `DUPLICATE COLLATE STATEMENT`

  Explanation: The COLLATE statement is specified exactly as a previous one, and therefore, is ignored.

  System action: Processing continues.

  Operator response: None.

  System programmer response: Verify that this is acceptable. If not, change the input control statements and resubmit the job.

Source: Image generator for z/OS support for Unicode

z/OS Unicode collation services settings

There are some prerequisites to use Unicode collation service, specifically:

- Build a Unicode Image with Collation Services enabled.
- Satisfy all migration requirements from previous releases.

For more information, see the following document:

- IBM publication *Support for Unicode: Using Conversion Services*, SA22-7649
- Unicode consortium, Collation Technical Report, available at:

  `http://www.unicode.org/reports/tr10/`

3.11.4 z/OS Unicode collation services invocation

z/OS support for Unicode provides the collation service to make a culturally correct binary comparison between two Unicode strings. It can also generate a sort key, which can later be used by the caller to do binary comparisons between strings.

The collation service is called using a stub routine named CUNLOCAL for AMODE (31), or CUN4LCOL for AMODE (64). It is called by any user and the external output goes to SYSPRINT log.

Establish a Unicode collation image

You can use the JCL shown in Figure 3-30 to set a Unicode Image. (Notice the COLLATE statement is in the SYSIN section.)

```
//UNIUTL JOB NOTIFY=&SYSUID,
// MSGCLASS=X,MSGLEVEL=(1,1),TIME=60,CLASS=A,
//CUNMIUTL EXEC PGM=CUNMIUTL
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* SYSIMG must be a FB 80 dataset *****************
//SYSIMG DD DSN=UNI.IMAGES(CUNIMG01),DISP=SHR
//TABIN DD DISP=SHR,DSN=SYS1.SCUNTBL
//SYSIN DD *
  /*******************************/
  /* example of input statements */
  /*******************************/
  /* Normalization Control Statement will be able */
  /* to work with D, C, KD and KC Normalization */
  /* forms                                      */
NORMALIZE;   /* Normalization include D,C,KD,KC */
             /* Collation Statement            */
             /* Tables support L1 to L4 level   */
COLLATE;     /* comparison                     */
CONVERSION 850, /* ASCII */
1047, /* EBCDIC */
RE; /* TECHNIQUE-SEARCH-ORDER *//*
*/
```

Figure 3-30 JCL to build a Unicode image

Collation tables

The collation tables are:

CUNOBACE	CE Main Table
CUNOEXIN	Expansions Index Table
CUNOEXDA	Expansion Elements Table
CUNOTIDX	Contraction Elements Index Table
CUNOCODA	Contraction Elements Data Table
CUNOMIDX	Main Index Table
CUNOFCD	Fast Normalization Table
CUNOTHLA	Rearrangement Values (Thai and Lao) Table
CUNOFKD	Fast Canonical Decomposition
CUNOFCO	Fast Composition

Collation limitations

Following are the collation limitations.

- **Surrogates tables**

  Two tables are reserved for dealing with surrogates. However, we will not deal with surrogates at this point since the Unicode Normalization services do not support surrogates. These tables will be left for definition in the future when surrogates can be normalized; right now surrogates will be ignored.

- **Tailoring tables**

  When tailoring is implemented in this development, more tables will have to be defined. These will hold the language rules needed to build the customized (tailored) CE Main Table and it's derived expansion/contraction/surrogate table counterparts. There will be one Rules Table for each tailoring defined.

Loading the Unicode image

Use the SET UNI command specifying the parmlib CUNUNIxx member which points to the Unicode Image.

Use the Display UNI command to verify that collation is enabled (look for the following lines).

This line shows you the service availability in the system:

```
SERVICE: CHARACTER      NORMALIZATION  CASE  COLLATION
```

This line shows you that Collation Tables were added to the image and are ready to be used:

```
COLLATE: ENABLED
```

z/OS Unicode services operation environment

Figure 3-31 on page 83 shows all the inter-component relations between all modules of collation service.

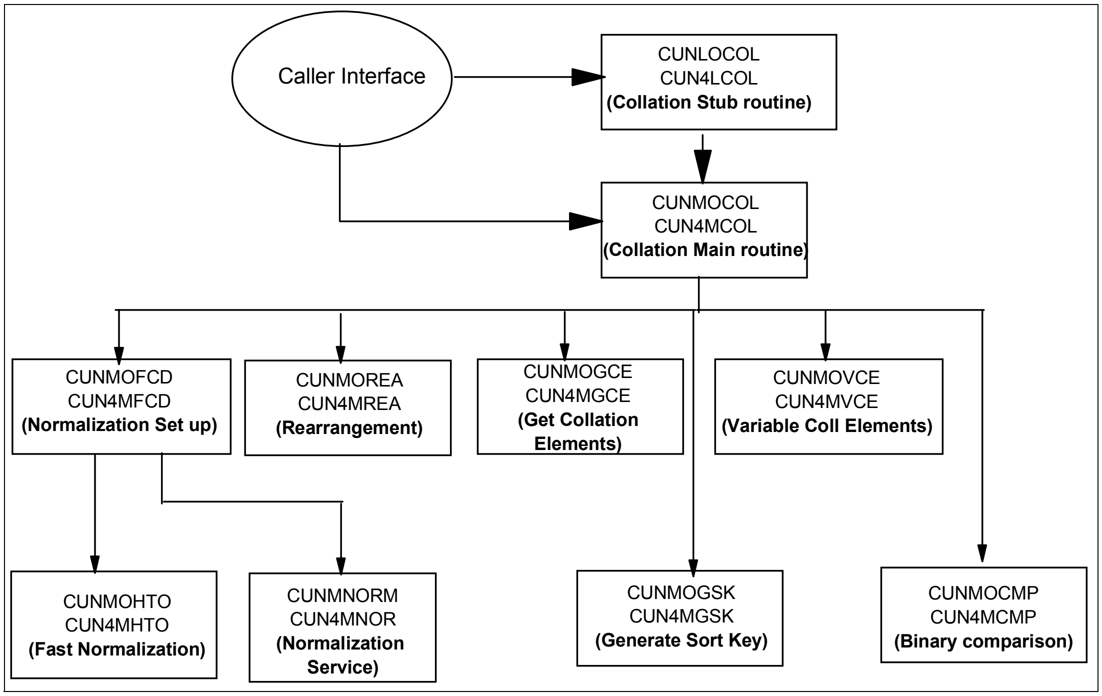


Figure 3-31 Unicode collation flow

You must provide the collation level required for the collate action. Valid values are:

- CUNBOIDF_PRIMARY
- CUNBOIDF_SECONDARY
- CUNBOIDF_TERTIARY
- CUNBOIDF_QUATERNARY

Collation interfaces

You can use the following interfaces with the Unicode.

- As other Unicode services, collation has 2 different interfaces (for C callers and ASM callers). The ASM interface is provided in 31-bit AMODE and 64-bit AMODE.
- Callers have to include this interface where prototypes and Unicode structures like parameter list are defined.
- **CUNBOIDF** for 31-bit and **CUN4BOID** for 64-bit in ASM language.
- **CUNHC** for 31-bit in C language.

> **Note:** For details, refer to the Collation chapter in *Support for Unicode: Using Conversion Services,* SA22-7649.

3.11.5 Unicode 64-bit support

z/OS V1R5 provides 64-bit APIs for z/OS Unicode services for exploitation by AMODE=64 applications.

- – HLASM APIs are provided on z/OS V1R2 via APAR OW56073.
- – C/C++ APIs are provided only on z/OS V1R6.

z/OS Unicode conversion service

The other major part of z/OS support for Unicode is the conversion services. They consist of a variety of conversion types and permit character conversion as well as case conversion for which the appropriate conversion tables were provided by the infrastructure.

Character conversion

z/OS support for Unicode provides direct conversion between character streams that are encoded with Coded Character Set Identifiers (CCSID). For character conversion, the conversion services are called using a stub routine named CUNLCNV for AMODE (31), or CUN4LCNV for AMODE (64). z/OS support for Unicode must be called in primary mode. The corresponding interface file for AMODE(64) is CUN4BCID.

Figure 3-32 shows the call syntax in HLASM for the calling stub routine CUN4LCNV.

```
GETMAIN .........        Obtain storage for parameter
 *                            area in primary address space
     LR R4,R1                 Save parameter area address
     USING CUN4BCPR,R4        Make parameter area addressable
     XC CUN4BCPR,CUN4BCPR     Init PARAMETER AREA to BINARY 0
     LA R15,CUN4BCPR_VER      Get Version
     ST R15,CUN4BCPR_VERSION  Version Store to get parameter area
     LA R15,CUN4BCPR_LEN      Initialize length
     ST R15,CUN4BCPR_LENGTH   Move to parameter area
     MVC CUN4BCPR_TECHNIQUE,=CL8' '  Take default technique
     MVC CUN4BCPR_SRC_CCSID,=FL4'1047'  From CCSID
     MVC CUN4BCPR_TARG_CCSID,=FL4'13488'  To CCSID
 *
 *      Supply source buffer pointer, length and ALET.
 *      Supply target buffer pointer, length and ALET.
 *      Supply work buffer pointer, length and ALET. (Not
 *        required for a conversion from 1047 to 13488).
 *      Supply DDA buffer pointer, length and ALET.
 *      Note: A DDA is always required. The required DDA length
 *        is defined by constant CUN4BCPR_DDA_REQ.
 *      Set flags
 *
     CALL CUN4LCNV,((R4))     Call stub routine with CUN4BCPR
 *                            address as argument.
     CUN4BCID DSCET=YES       Provide Mappings (CUN4BCPR, return
 *                            and reason codes, constants for
 *                            version and length).
```

Figure 3-32 Invoke CUN4LCNV

Case conversion

z/OS support for Unicode provides case conversions that allow conversion of Unicode characters to their upper case equivalent or their lower case equivalent. For more details about the case mappings, refer to the tables provided by the Unicode Consortium.

For case conversion, the conversion services are called using a stub routine named CUNLASE for AMODE (31), or CUN4LASE for AMODE (64). The corresponding interface file for AMODE(64) is CUN4BAID.

Figure 3-33 on page 85 shows the call syntax in HLASM for the calling stub routine CUN4LASE.

```
GETMAIN .........        Obtain storage for parameter
 *                            area in primary address space
     LR R4,R1                 Save parameter area address
     USING CUN4BAPR,R4        Make parameter area addressable
     XC CUN4BAPR,CUN4BAPR     Init PARAMETER AREA to BINARY 0
     LA R15,CUN4BAPR_VER      Get Version
     ST R15,CUN4BAPR_VERSION  Version Store to get parameter area
     LA R15,CUN4BAPR_LEN      Initialize length
     ST R15,CUN4BAPR_LENGTH   Move to parameter area
     LA R0,CUN4BAPR_TO_UPPER  Get conversion type
     ST R0,CUN4BAPR_CONV_TYPE Store to parameter area
 *
 *      Supply source buffer pointer, length and ALET.
 *      Supply target buffer pointer, length and ALET.
 *      Supply DDA buffer pointer, length and ALET.
 *      Note: A DDA is always required. The required DDA length
 *        is defined by constant CUN4BAPR_DDA_REQ.
 *      Set flags
 *
     CALL CUN4LASE,((R4))     Call stub routine with CUN4BAPR
 *                            address as argument.
     CUN4BAID DSCET=YES       Provide Mappings (CUN4BAPR, return
 *                            and reason codes, constants for
 *                            version and length).
```

Figure 3-33 Invoke CUN4LASE

Normalization

z/OS support for Unicode provides support that allows the normalization (decomposition or composition) of Unicode characters to one of the normalization forms. For a detailed explanation of normalization, including specific information about the normalization forms, refer to Technical Report #15 provided by the Unicode Consortium, available at:

`http://www.unicode.org/unicode/reports/tr15/`

The normalization service is called using a stub routine named CUNLNORM for AMODE (31), or CUN4LNOR for AMODE (64). The corresponding interface file for AMODE(64) is CUN4BNID.

Figure 3-34 on page 86 shows the call syntax in HLASM for the calling stub routine CUN4LNOR.

```
GETMAIN .........        Obtain storage for parameter
 *                            area in primary address space
     LR R4,R1                 Save parameter area address
     USING CUN4BNPR,R4        Make parameter area addressable
     XC CUN4BNPR,CUN4BNPR     Init PARAMETER AREA to BINARY 0
     LA R15,CUN4BNPR_VER      Get Version
     ST R15,CUN4BNPR_VERSION  Version Store to get parameter area
     LA R15,CUN4BNPR_LEN      Initialize length
     ST R15,CUN4BNPR_LENGTH   Move to parameter area
     LA R0,CUN4BNPR_D         Get normalization type
     ST R0,CUN4BNPR_NORM_TYPE Store to parameter area
 *
 *      Supply source buffer pointer, length and ALET.
 *      Supply target buffer pointer, length and ALET.
 *      Supply work buffer pointer, length and ALET.
 *      Supply DDA buffer pointer, length and ALET.
 *      Note: A DDA is always required. The required DDA length
 *        is defined by constant CUN4BNPR_DDA_REQ.
 *      Set flags
 *
     CALL CUN4LNOR,((R4))     Call stub routine with CUN4BNPR
 *                            address as argument.
     CUN4BNID DSCET=YES       Provide Mappings (CUN4BNPR, return


 *                            and reason codes, constants for
 *                            version and length).
```

Figure 3-34   Invoke CUN4LNOR

Collation

z/OS support for Unicode provides the Collation Service to make a culturally correct binary comparison between two Unicode strings. It can also generate a sort key, which can later be used by the caller to do binary comparisons between strings. For a detailed explanation of the Unicode collation process, refer to the Unicode Consortium's Technical Report #10 at:

```
http://www.unicode.org/unicode/reports/tr10/
```

The collation service is called using a stub routine named CUNLOCOL for AMODE (31), or CUN4LCOL for AMODE (64). The corresponding interface file for AMODE(64) is CUN4BOID.

You can find a sample for calling the stub routine CUN4LCOL in the samples library (SYS1.SAMPLIB) as member CUN4SOSA for 64 bit.

> **Note:** Callers invoking 64 bit interfaces should provide large enough Dynamic Data Area (DDA) size for conversion. DDA size has to be at least 8K in a 64-bit interface. This support has been provided via APAR OA03519.

z/OS Unicode services operation environment

The 64-bit calls to z/OS Unicode services is using an independent @PROCESS definition for each service (which means new entry points for each addressing mode).

The new entry points are loaded in the CUNMUNI load module at IPL time. At load time we detect the addressing mode and load just the necessary support for each environment. (MVS/ESA™ = 31 Bit and MVS/ESAME = 31 and 64 Bit).

Current callers won't be affected by these changes because 31-bit interfaces and modules will maintain current behavior.

Figure 3-35 illustrates the implementation to handle different object modules from the same source code.

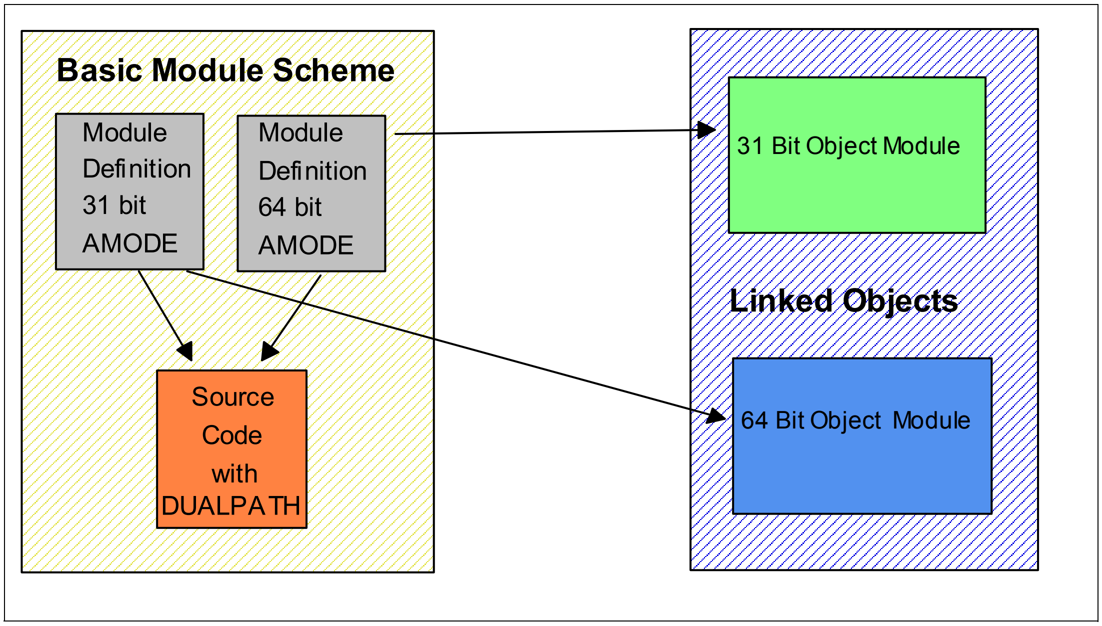

Basic Module Scheme
Module Definition 31 bit AMODE
Module Definition 64 bit AMODE
Source Code with DUALPATH
31 Bit Object Module
Linked Objects
64 Bit Object Module

Figure 3-35 Handling object modules

Figure 3-36 illustrates the 64-bit implementation.

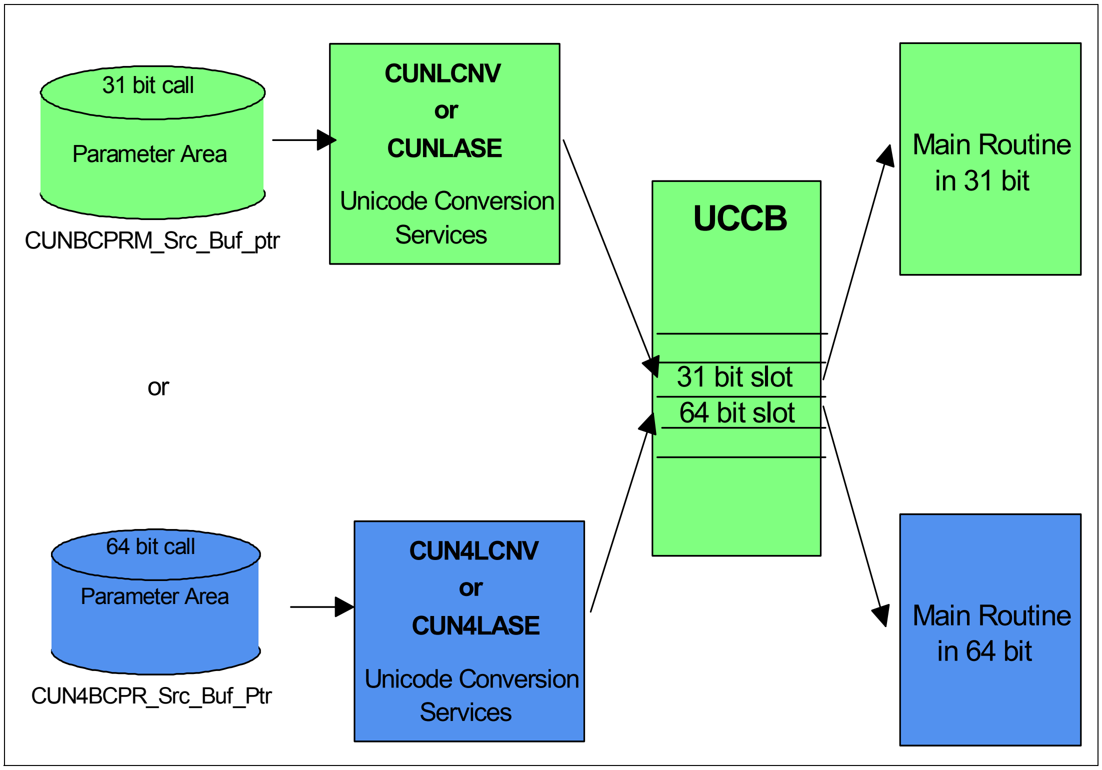


Figure 3-36 64-bit scheme

3.11.6 Providing iconv()-like behavior

A new functionality has been added to identify, detect, and react to malformed and inconvertible characters, in the same fashion as the C/C++ iconv() function.

- Inconvertible characters are characters that are valid on the source codepage but have no equivalence on the target codepage. These characters can't be converted.
- Malformed (or invalid) characters are characters whose structure isn't valid on the source codepage. These characters can't be converted.

Note: This function has been implemented in the base code of z/OS V1R5 and has been rolled down to z/OS V1R2 via APAR OW56074.

Current behavior

Current conversion behavior is to take the input character, look for its conversion on the conversion table, and place the converted character on the output buffer when a match is found. With the current algorithm, it does not tell the difference between an incomplete multiple byte character and a character not found on the conversion table. Both are flagged as inconvertible characters. DB2 needs to be able to differentiate between the two types.

Enhancement

A new flag CUNPCPRM_Mal_Action is defined in the CUNBCIDF structure (part of CUNBCPRM_Flag1) as shown in Table 3-2.

Table 3-2 CUNPCPRM_Mal_Action

Bit Position	Name
xxx1 xxxx	CUNBCPRM_Mal_Action

CUNBCPRM_Mal_Action specifies the action to take when a source character is malformed on the source CCSID:

- **0**: indicates that the substitution character is to be put in the target buffer and the conversion is to continue when malformed characters are found.
- **1**: indicates that the conversion is to terminate with Return Code=4, Reason Code=12, when malformed characters are found. This is the same behavior one gets when using the LE's iconv() function. This is the default.

Note: CUNPCPRM_Mal_Action only works if CUNPCPRM_Sub_Action is enabled (set to 1).

A second new flag CUNPCPRM_Mal_Found is defined in the CUNBCIDF structure (part of CUNBCPRM_Flag2). This is shown in Table 3-3.

Table 3-3 CUNBCPRM_Mal_Found

Bit position	Name
x1xx xxxx	CUNBCPRM_Mal_Found

CUNBCPRM_Mal_Found indicates to the caller whether the conversion service has encountered a malformed character in the source buffer:

- **0**: indicates that the conversion service did not find a malformed character in the source buffer. This is the default.

- **1**: indicates that the conversion service did find a malformed character in the source buffer (or the service was called with the bit set to 1).

Note: This bit needs to be reset by the caller.

User action

When a malformed character is found, change or remove the malformed source character from the source buffer and perform the conversion again.

During character conversion, the source buffer may also contain byte-strings that don't represent a character in the source code page. These characters are called malformed characters and cannot be converted to valid target code points. Specifically, if the CUNBCPRM_Flag1 parameter bit CUNBCPRM_Sub_Action says “substitute” and CUNBCPRM_Mal_Action says “terminate,” then when a malformed character is encountered, conversion will terminate with RC=4 and RSN=12. But if CUNBCPRM_Mal_Action says “substitute,” the malformed character will be substituted. The only exception will be when converting from UTF8 code pages. When CUNBCPRM_Flag1 parameter bit CUNBCPRM_Sub_Action is set to “terminate,” the CUNBCPRM_Mal_Action parameter bit is ignored for all characters that have no equivalence in the To-CCSID and will be treated as malformed characters, thus resulting in RC=4 and RSN=12,

If CUNBCPRM_Sub_Action says “substitute” and CUNBCPRM_Mal_Action says “substitute,” normal substitution behavior will occur.

RC=4 and RSN=12

Return code 4 and reason code 12 found during the conversion mean that a character, found in the source buffer, is not a valid source character and cannot be converted to a To-CCSID character and the CUNBCPRM_Mal_Action flag specifies “terminate with error”. To resolve this, check whether the input string is correct and whether the correct conversion tables are used. An incomplete character may be causing a range check to fail.

3.12 2 GB FICON channel support

The FICON switch port card and the FICON channel attached to it negotiate the speed at which I/O proceeds through the channel. For example, a 1GB port card attached to a 2GB channel will negotiate an I/O speed of 1GB through the channel. This is shown in Figure 3-37.

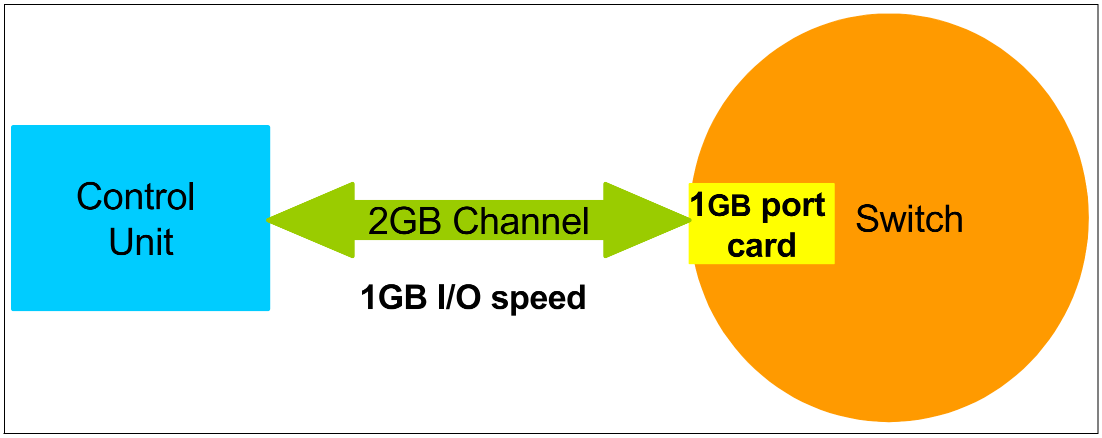


Figure 3-37 FICON channel speed negotiation

This negotiated I/O speed is stored in the channel path measurement facility's channel path measurement characteristics. Previously, IOS only refreshed a channel's channel path measurement characteristics when the channel was configured online using command CF CHP(xx),ONLINE.

If the switch port card was replaced, causing a negotiation to a different I/O speed without first configuring the channel offline and, then, online after the replacement, the channel path measurement characteristics were not refreshed.

With the 2GB FICON channel support, when the port card is replaced without first configuring the channel offline and, then, online after the replacement, IOS channel path recovery will refresh the channel path measurement characteristics.This is illustrated in Figure 3-38.

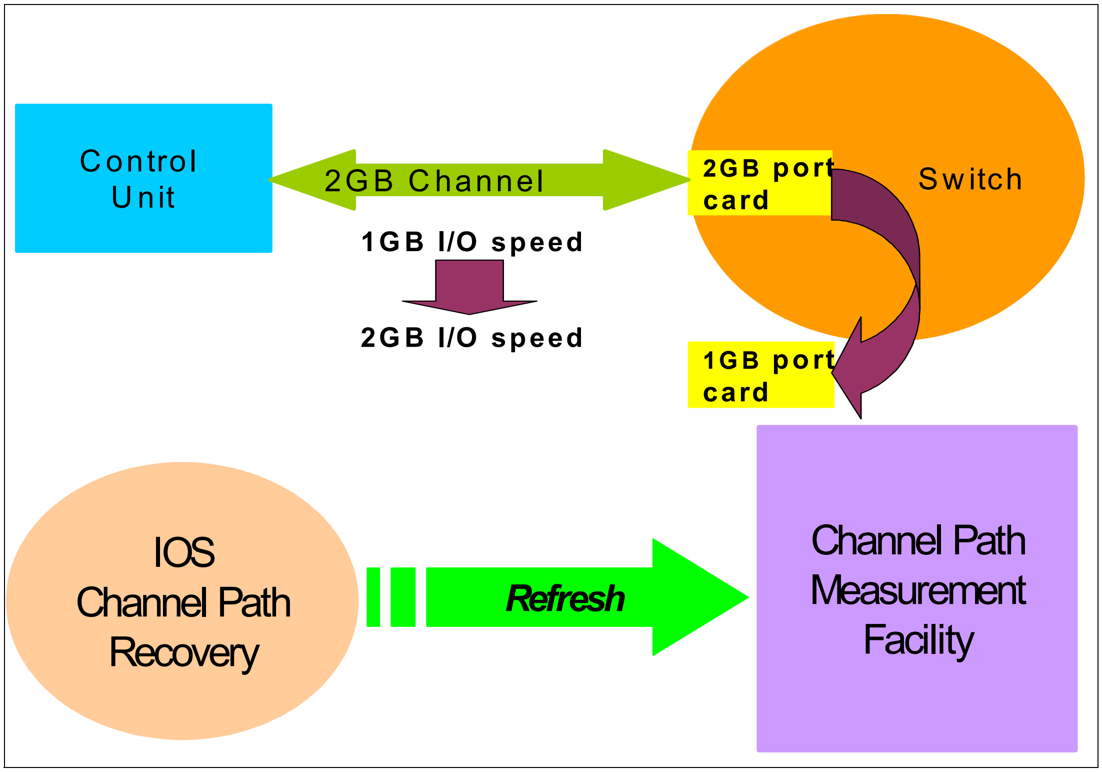


Figure 3-38 2 GB FICON support

3.13 IARV64 system trace support

System trace capability has been provided for the high virtual storage services (IARV64). A trace entry will be created in the system trace table upon completion of the IARV64 request. Trace entries will be created for the following IARV64 requests:

```
GETSTOR, DETACH, PAGEFIX, PAGEUNFIX, PAGEOUT, PAGEIN, DISCARDDATA, CHANGEGUARD,
GETSHARED, SHAREMEMOBJ, CHANGEACCESS
```

The trace entry will be a system service entry (SSRV) with a new SSRV entry identifier of x'14B' to identify it as an IARV64 trace entry. An additional 1-byte field within the trace entry unique fields will identify the IARV64 service for which the entry was created.

The detailed trace entry is shown in Appendix B, “IARV64 system trace entry” on page 493.

3.14 System Logger enhancement

The purpose of the System Logger enhancement is to provide a better way for the System Logger address space to manage contention and any request made against a logstream.

If the offload processing cannot complete there is an impact on the activity of the System Logger address space which could lead to a sysplex outage.

This is the reason why, as a first step, APAR OW51854 was created and shipped with OS/390 V2R10 (FMID HBB7703). As this was not enough, the Logger task monitor was implemented.

3.14.1 Migration and installation

The new functions shipped with V1R5 have no impact on migration, with the exception that there are some subtleties in the following messages which are described later:

- IXG271I
- IXG272E
- IXG275I

The new functions shipped with z/OS V1R5 do not affect coexistence, as follows:

- A user can run with mixed level sysplex.
- No extra hardware support is required.
- The z/OS V1R5 release requires the same software products to be installed that the previous release of this product required.

3.14.2 Logger task monitor

Logger now monitors its allocations and HSM recall service tasks for delays through the Logger task monitor and provides a mechanism (via WTORS) to interrupt these delay requests. Logger handles the interruption as an error condition for the current request. However, removing the delayed request allows other log stream resource requests then to be processed.

The Logger task monitor monitors two important tasks, allocation and HSM recall; and provides the new services IXGOFLDS, IXGDELAB, and IXGDELLS.

These more practical new services are somewhat equivalent to the macros IXGOFFLD, IXGDELET, and IXGINVNT services described in *z/OS MVS Assm Services Reference IAR-XCT*, SA22-7607.

Additionally, the new messages IXG275i, IXG272e, IXG314i, IXG266i, and IXG271i are issued to ease the processing of a delayed logstream resource request. IXG271i and IXG272e are essentials.

IXG271i message

```
IXG271I LOGGER DATA SET REQUEST IN taskname SERVICE TASK DELAYED DURING THE PAST seconds
SECONDS FOR LOGSTREAM logstream staging DSN=dsname, DIAG=diag
```

IXG272e message

```
IXG272E LOGGER taskname TASK DELAYED, REPLY "MONITOR", "IGNORE", "FAIL", "EXIT".
```

IXG271i is issued to inform the user that an allocation task or migrated data set task is delayed. Then the WTOR IXG272e is issued to have the operator make a decision, which could be either MONITOR, IGNORE, FAIL, or EXIT, as follows:

MONITOR Continues to process the delayed request.

IGNORE Stops the delayed request.

FAIL Interrupts the delayed request that is being processed.

EXIT Terminates the Logger Service Task Monitoring.

The recommended procedure for responding to IXG272e is described in the section titled “Offload and Service Task Monitoring” in *z/OS MVS Setting Up a Sysplex*, SA22-7625.

With all these new features, a delayed request does not stop another logstream resource request from running, the System Logger address space and the logstream usage will have much more availability on one system, and the availability of the System Logger address space within the sysplex will increase.

3.14.3 Logger data set deletion

The problem that existed is that during the offload, the System Logger both allocates a new data set to offload the logstream and deletes one or several offload data sets according to the RETPD value set in the LOGR policy. See *z/OS MVS Setting Up a Sysplex*, SA22-7625 for a discussion of RETPD value.

Offload data sets

The problem is, if the System Logger cannot actually delete offload data sets because they are allocated or migrated, it has an impact on performance because the services used to delete the offload data sets and to offload the data sets are the same. Thus, the offload processing can hang.

To solve this, this support now limits the amount of deletion activity and uses the Logger task monitor for internal monitoring. The internal monitoring has the Logger task monitor monitoring the logstream service task that deletes data sets during the offload activity. Also, now the Logger task monitor quiesces the data set deletion activity so that other log stream requests can be processed.

> **Note:** The data set deletion activity is quiesced if after 50 seconds the deletion of the logstream has not completed and if there is another logstream request queued.

The internal monitoring issues the IXG266i message to inform the user that the data set deletion activity is quiesced and this minimizes the bottlenecks and prevents sysplex outages.

IXG266i message

```
IXG266I LOGGER DATA SET DELETION ACTIVITY FOR LOGSTREAM logstream WAS QUIESCED BY
taskname TASK.
```

3.14.4 The utility function procedures

In the past there was no easy way to initiate a logstream offload besides using the IXGOFFLD macro via a program API. To solve this, utility procedure IXGOFLDS was developed.

Two additional utility procedures were developed: IXGDELAB, and IXGDELLS. They are shipped in SYS1.SAMPLIB and must be copied to SYS1.PROCLIB.

These procedures should only be used when it is necessary to take an installation action on the log stream. Refer to the documentation about the subsystem or application that makes use of this log stream to understand any interaction or expectations before running any of these procedures.

The description of their use is as follows:

IXGOFLDS Initiates an offload for all valid log blocks in primary storage to DASD. When the procedure completes successfully, all the log blocks in the log stream will be off-loaded to DASD and message IXG273 will be issued to the console. When the procedure fails, message IXG274I will be issued to the console stating which function failed and listing the return and reason code.

IXGDELAB Deletes all the log blocks marked logically deleted. When the procedure completes successfully, all the log blocks in the log stream will be logically deleted and message IXG273I will be issued to the console. When the procedure fails, message IXG274I will be issued to the console stating which function failed and listing the return and reason code.

IXGDELLS Attempts to delete a defined log stream from the LOGR couple data set. When the procedure completes successfully, the log stream is deleted from the logger inventory and message IXG273I is issued to the console. When the procedure fails, message IXG274I is issued to the console stating which function failed and listing the return and reason code.

Messages IXG273I and IXG274I are issued in response to the execution of the procedures as follows:

IXG273i message `IXG273I function COMPLETED SUCCESSFULLY`

IXG274i message `IXG274I subfunction FAILED FOR function, RETCODE=retcode, RSNCODE=rsncode`

> **Note:** To execute theses procedures, the user must have UPDATE access on the logstream name resource in the RACF class LOGSTRM. Users should have a look at the RACF security server documentation to activate the LOGSTRM class, to define the profile, and to set the access level to profiles defined in this class.

3.14.5 Understanding share options (3,3)

Though the use of share options (3,3) was recommended in previous releases of z/OS when defining the logstream and staging data sets, messages IXG267I and IXG268I are now issued to inform the user of a potential problem before using the data sets. They are as follows:

IXG267i message

```
IXG267I DataSetType DATASET DataSetName ALLOCATED WITH INCORRECT VSAM SHAREOPTIONS, USE
IS ACCEPTED BUT NOT RECOMMENDED.
```

IXG268i message

```
IXG268I DataSetType DATASET DataSetName COULD NOT BE OPENED FOR JOB JobName DUE TO
ShrOpt LossOfData
```

3.14.6 LE enhancements in z/OS V1R5

The enhancements introduced in z/OS V1R5 Language Environment (LE) and z/OS V1R5 C/C++ are as follows:

- New and enhanced C/C++ functions

  - LE enhancements to IEEE math library, discussed in the next section
  - Enhanced ASCII support, see “Enhanced ASCII support” on page 95
  - itoa() family of functions, see “itoa() family of functions” on page 96
- Multilevel security support, see “MLS support” on page 97
- RAS improvements, see “RAS enhancements” on page 98
- Performance enhancements, see “Performance enhancements” on page 98
- XPLINK, see “XPLINK” on page 99
- Debugging enhancements, see “Debugging enhancements” on page 99
- National language support (NLS), see “National language support (NLS)” on page 101

3.14.7 LE enhancements to IEEE math library

In this release of z/OS V1R5 some enhancements were made to the Language Environment (LE) 31-bit IEEE math library.

The LE math library contains 140 callable services that perform mathematical operations on hexadecimal floating-point numbers. There currently are a number of C library functions that perform the equivalent operation on IEEE floating-point numbers. However, there were a few missing functions. Therefore, IEEE floating-point math functions were added to the library. This makes the IEEE math library functions equivalent to the hexadecimal math functions.

C/C++ applications that intend to use the IEEE math library should:

- Be compiled with the FLOAT(IEEE) C/C++ compiler option
- Include the <math.h> header for best performance

Table 3-4 New math functions

acoshf()	__cotan()	expm1f()	log1pl()	remquol()
acoshl()	__cotanf()	expm1l()	log2()	tgamma()
asinhf()	__cotanl()	fdim()	log2f()	tgammaf()
asinhl()	erfcf()	fdimf()	log2l()	tgammal()
atanhf()	erfcl()	fdiml()	lround()	trunc()
atanhl()	erff()	hypotf()	lroundf()	truncf()
cbrtf()	erfl()	hypotl()	remainderf()	truncl()
cbrtl()	exp2()	lgammaf()	remainderl()	
copysignf()	exp2f()	lgammal()	remquo()	
copysignl()	exp2l()	log1pf()	remquof()	

These new math functions are part of the C run-time library and can be invoked by C/C++ applications that are compiled with the FLOAT(IEEE) compiler option. Check the *C/C++ Run-Time Library Reference*, SA22-7821 for more information about the IEEE floating point math functions.

3.14.8 Enhanced ASCII support

Enhanced ASCII was introduced in z/OS V1R2 and provides C/C++ run-time library support for applications compiled in the ASCII codeset, enabling UNIX applications which were built to run on ASCII-based systems to be more easily ported to z/OS.

Enhanced ASCII usage requires ASCII and XPLINK compiler options. The ASCII option instructs the compiler to convert strings and literals to the ISO8859-1 codeset. Language Environment C/C++ headers, and appropriate feature test macros, are required to force mapping of the supported Enhanced ASCII functions to the proper library stub routines which handle any ASCII to EBCDIC translation needed in order for the function to perform as desired on the z/OS operating system.

The initial Enhanced ASCII support did not include all of the C/C++ library functions. Many functions still required the application to perform their own ASCII ↔ EBCDIC conversions for character data on input and/or output from the function. In this release of z/OS V1R5 thirty-one additional functions were added for enhanced ASCII to support the functionality that was needed.

> **Note:** Using Enhanced ASCII extensions, applications can now exploit the implicit support for ASCII strings as input/output to thirty-one additional C/C++ library functions. This means ported applications can now remove z/OS-specific code that was needed to handle ASCII ↔ EBCDIC conversions.

There is an APAR available for those of you who want to use this enhancement on z/OS releases prior to z/OS V1R5. APAR PQ63405 produced the following PTF numbers:

- z/OS 1.2 UQ73076
- z/OS 1.3 UQ73077
- z/OS 1.4 UQ73078

Installing the PTFs associated with APAR PQ63405 or migrating to z/OS V1R5 will not affect existing Enhanced ASCII applications that are currently using any of these thirty-one functions (and doing their own conversion). Even a re-compile of the application will not be affected. The new Enhanced ASCII support must be explicitly asked for at compile time with a new feature test macro. Therefore, the new feature test macro _ENHANCED_ASCII_EXT must be defined before inclusion of the first header file. It also must be set to 0x41020010 or higher:

```
#define _ENHANCED_ASCII_EXT 0x41020010
 -D _ENHANCED_ASCII_EXT=0x41020010
 DEFINE(_ENHANCED_ASCII_EXT=0x41020010)
```

The new feature test macro is needed because the EBCDIC versions of these functions might already be used in an existing ASCII application. We cannot just map to the ASCII interfaces when a re-compile occurs. The application must also change. The *C/C++ Run-Time Library Reference*, SA22-7821 contains a more extensive description of the new feature test macro, and why it is needed.

Important: Failure to set the new _ENHANCED_ASCII_EXT feature test macro to an appropriate value or failure use the required header will not cause problems until the application is executed. The problems range from the C library function failing because it was given ASCII data when it expected EBCDIC to the application failing because the C library returned EBCDIC data when the application expected ASCII. This situation can occur because the Enhanced ASCII Extensions support is for already existing functions. Application developers can verify the proper interface is used by looking at the symbol map generated by the compiler. The @@Axxxx stub name will be there if the Enhanced ASCII interfaces are properly requested.

3.14.9 itoa() family of functions

The itoa() family of functions are added to aid in porting applications to z/OS. These functions are common to other platforms, although not part of a standard. The itoa() family of functions provide integer to alpha conversion. In other words, an integer value is converted to a string representation.

Note: Previously, applications being ported to z/OS which make use of the itoa() family of functions on other platforms were required to make code changes.

The itoa() support consists of six new C functions. They provide support for (unsigned) integer to alpha conversions. The users of these functions will be applications being ported to z/OS or any other application that would like a more simple interface for doing these types of conversions. The main benefit for using the itoa() support is that it provides a way for converting (unsigned) integer values to various string representations without using sprintf().

Feature test macro _OPEN_SYS_ITOA_EXT must be defined before inclusion of the first header in the application source code to expose the itoa() family of functions external C/C++ Run-Time Library interfaces.

In addition, the target execution environment, defined using the TARGET compiler option, must be set to a minimum value of 0x41050000 to expose itoa() family of functions externals. This value is the default when using the z/OS C/C++ compiler and z/OS Language Environment that go with the z/OS V1R5 operating system.

Note: Enhanced ASCII support is provided.

The six new C functions represent the following itoa() prototypes:

```
char *itoa(int n, char *buffer, int radix);
char *utoa(unsigned int n, char *buffer, int radix);
char *ltoa(long n, char *buffer, int radix);
char *ultoa(unsigned long n, char *buffer, int radix);
char *lltoa(long long n, char *buffer, int radix);
char *ulltoa(unsigned long long n, char *buffer, int radix);
```

Figure 3-39 on page 97 shows a sample C program that uses the ultoa() function. The ultoa() function coverts the `unsigned long` into a character string. The string is placed in the buffer passed, which must be large enough to hold the output.

```
#define _OPEN_SYS_ITOA_EXT
#include <stdlib.h>
#include <stdio.h>
int main() {
  char buffer[30]; /*result string*/
  char *ch;
  unsigned long n = 56734UL;
  ch = ultoa(n, buffer, HEX);
  if (ch != NULL) printf("buffer(hex)=%s\n",ch);
  else perror("");
}
```

Figure 3-39 Sample C program

Finally the buffer output is presented in hexadecimal format, as shown in Figure 3-40.

```
buffer(hex)=dd9e
```

Figure 3-40 Output of the C sample

3.14.10 MLS support

Multi-level security (MLS), as a requirement of the US government, was added in this release of z/OS V1R5. MLS is an additional level of security, classifying users and resources and imposing mandatory security controls.

MLS allows the classification of data and users based on a system of hierarchical security levels, combined with a system of non-hierarchical security categories.

In this release of z/OS V1R5 the Language Environment provides support for MLS through the following:

__poe() Port Of Entry information used in determining various levels of permission checking.

__writedown() Query or change the setting of the write-down privilege of an ACEE (access control environment element)

__lchattr() Change the Attributes of a file or directory when they point to a symbolic or external link.

S_ISSECLABEL Check the availability of the SECLABEL field in a given structure (where s = seclabel field).

A number of headers were changed also for this work. In all cases, the structures are equivalent to their USS/Kernel analogs:

- <sys/__getipc.h>
- <sys/__stat.h>
- <sys/socket.h>
- <sys/modes.h>

For more information about MLS see “Support for multilevel security” on page 128 and “Multilevel security” on page 320.

3.14.11 RAS enhancements

“Reliability, availability, serviceability,” also known as “RAS,” is part of the zSeries strategy for autonomic computing. One new enhancement in this field is in the area of heap pools seviceability. In this release of z/OS V1R5 the Language Environment (LE) provides service support using tools and tracing capability to diagnose enabled heap pools.

The Heappools support within LE is a method of obtaining dynamic storage in a more efficient manner, especially with many similar size storage requests and/or when the application is multi-threaded, than normal LE storage management routines.

It is now possible to create a trace of heappool **get** and **free** requests which can be formatted from IPCS. The trace is specific to each pool and with a maximum of 1024 trace entries per pool. It is useful to know that after 1024 trace entries the trace will wrap within each pool. Being provided with additional debug capability is an great advantage, although it will slow down performance when it is used.

A new sub-option on the HEAPCHK run-time option is provided. It now looks like this:

```
HEAPCHK(ON|OFF,freq,delay,call depth, pool depth)
```

Where **pool depth** indicates the maximum number of call levels (traceback) which will be stored in the trace table for each trace entry. If zero (0) is specified the trace will be turned off, which is the same as specifying OFF. We recommend setting the pool depth to 10 when tracing is turned on. As shown in Figure 3-41, the HEAPCHK setting enables the heappools trace with up to 10 traceback entries per trace entry.

```
HEAPCHK(ON,0,0,0,10)
```

Figure 3-41 HEAPCHK setting

3.14.12 Performance enhancements

The LE run-time option HEAPPOOLS was introduced in OS/390 V2R4 and provides C/C++ applications a faster and more efficient way to allocate memory without rewriting or recompiling the applications. This is accomplished by pre-allocating pools of storage elements (cells) of specific sizes. This support allowed for 6 different cell sizes, up to 2K, to be specified.

The HEAPPOOLS support was hooked into malloc() and free() calls (C++ new and delete use malloc() and free()). By using the pre-allocated pools of storage (inside of normal LE heap segments) increased performance was attained for these functions. Locking mechanisms are also simplified, which allows even greater performance improvements for multi-threaded applications.

The initial HEAPPOOLS support allowed for only 6 different cell sizes with a maximum size of only 2K. In this release of z/OS V1R5, support for 12 different cell sizes with a maximum size of 64K has been provided. This improves the performance of multi-threading applications with storage requests greater than 2KB and that use the HEAPPOOLS run-time option.

Attention: Changes to existing CEEDOPT and CEECOPT usermods will be required to support the new sub-options of the HEAPPOOLS run-time option. See the sample CEEDOPT and CEECOPT provided with z/OS V1R5.

3.14.13 XPLINK

Prior to this release of z/OS V1R5, almost all LE functions that were written in PL/X or Assembler ran on the non-XPLINK stack. XPLINK C and non-XPLINK applications were both using the same non-XPLINK copy of these routines in the common execution library (CEL). So when an XPLINK C/C++ program called one of these non-XPLINK functions, a slow stack-swap operation occurred before and after the call.

XPLINK versions were initially provided for only a few high-performance routines. None of the mutex routines and only a few of the storage management routines (not including malloc() and free()) were supplied in both XPLINK and non-XPLINK versions. Changes were made to the XPLINK high-performance LE functions. It now provides high-performance support for the following functions in an XPLINK environment:

- malloc()
- free()
- pthread_mutex_lock()
- pthread_mutex_unlock()
- pthread_rwlock_rdlock()
- pthread_rwlock_wrlock()
- pthread_rwlock_unlock()

Let’s look at malloc() for example. The combination of HEAPPOLS(OFF) and XPLINK is the major path that is affected. HEAPPOOLS(ON) and XPLINK versions of malloc() and free() were already high performance.

These new XPLINK versions are called from the XPLINK version of the C run-time, which is used by XPLINK C and C++ applications. It is faster for the XPLINK C run-time to stay on the XPLINK stack and call these new XPLINK functions, rather than switch to the non-XPLINK stack and back when calling the original non-XPLINK versions.

Using XPLINK Hi-Perf Functions, you can do the same functions as before (but faster). The new XPLINK functions should save the stack swapping required when using the old functions. This can be considered a major advantage. The only disadvantage is that they take additional space in LE load modules.

3.14.14 Debugging enhancements

There are new interfaces that will be of interest to debugger writers introduced in z/OS V1R5 LE enhancements as follows:

- Loading the debugger event handler from the HFS.
- Using a C CWI to activate and deactivate execution hooks.

Debugger event handler

Loading the event handler from the HFS is only available for the debugger event handler. The profiler event handler cannot be loaded from the HFS. The debugger event handler is loadable by Language Environment with one of the following:

- The name CEEEVDBG
- The __CEE_DEBUG_FILENAME31 environment variable

  If __CEE_DEBUG_FILENAME31 exists, LE uses the value specified by that variable and loads it from the HFS. If this variable does not exist, LE uses CEEEVDBG as the name of the debugger event handler and loads it from a MVS data set. This is what happened before the changes in z/OS V1R5. The attempt to load the Debug Event Handler is performed from either the HFS or MVS, not both.

Restriction: Loading the debug event handler from HFS is not supported when running under CICS.

This name, __CEE_DEBUG_FILENAME31, combined with the HFS “path” specified in the LIBPATH environment variable, provide the fully qualified pathname for the debugger event handler.

Examples

In the examples, “`useful`” is the program that is being run and the debugger event handler is `/debug/testvdbg`.

```
From TSO:
useful  ENVAR('LIBPATH=/debug','_CEE_DEBUG_FILENAME31=testvdbg') TEST/

From UNIX Shell:
export LIBPATH=$LIBPATH:/debug
export _CEE_DEBUG_FILENAME31=testvdbg
export _CEE_RUNOPTS=TEST
useful
```

Figure 3-42 Examples to specify the name of the debugger event handler to be loaded from the HFS

C CWI _setHookEvents()

The C CWI _setHookEvents() can be used by both debuggers and profilers. A compiler option (usually TEST) causes the compiler to generate EX instructions that point to one of several fields in Language Environment's Common Anchor Area (CAA). These fields contain NOPR instructions when the hooks are not active and BAL instructions when the hooks are active. When the hooks are active, the BAL instruction will result in the debugger or profiler event handler being called for event 133.

The __setHookEvents() CWI sets the execute hook events state for all threads owned by the target enclave and referenced using asfTargetThreadRef as specified by the eventsMask parameter. Callback functions let you provide address space free access to storage in the target process.

The __setHookEvents() API sets the execute hook events state for all threads owned by the target enclave referenced via asfTargetThreadRef as specified by the eventsMask parameter, shown in Figure 3-43 on page 100. Callback functions allow the caller to provide address space free access to storage in the target process.

```
#include <__ledbug.h>
int __setHookEvents( int eventsMask,
                     const asfCalbackFunctions *asfCallbacks,
                     const asfTargetRef *asfTargetThreadRef,
                     const threadSpec *reservedForFutureUse);
```

Figure 3-43 C CWI __setHookEvents() syntax

For a complete discussion of this new support, see *z/OS Language Environment Vendor Interfaces*, SA22-7568.

3.14.15 National language support (NLS)

National Language Support (NLS) is the set of variables used by the system to set the correct language settings. NLS provides commands and Standard C Library subroutines for a single worldwide system base. An internationalized system has no built-in assumptions or dependencies on language-specific or cultural-specific conventions such as:

- Code sets
- Character classifications
- Character comparison rules
- Character collation order
- Numeric and monetary formatting
- Date and time formatting
- Message-text language

All information belonging to cultural conventions and language is obtained at process run time. The following capabilities are provided by NLS to maintain a system running in an international environment:

- Separation of messages from programs
- Conversion between code sets

In English, the word *locale* means a place where something happens or has happened. This is a key term in globalization. A locale is defined by these language and cultural conventions. An internationalized system processes information correctly for different locations. For example, in the United States, the date format, 9/6/2002, is interpreted to mean the sixth day of the ninth month of the year 2002. The United Kingdom interprets the same date format to mean the ninth day of the sixth month of the year 2002. The formatting of numeric and monetary data is also country-specific, for example, the U.S. dollar and the U.K. pound.

All locale information must be accessible to programs at run time so that data is processed and displayed correctly for your cultural conventions and language. This process is called *localization*. Localization consists of developing a database containing locale-specific rules for formatting data and an interface to obtain the rules.

In z/OS V1R5 support has been added to allow conversions with code pages IBM-4933 and IBM-13124.

An XPLINK and a base version of each new EBCDIC locale are shipped. For the HFS Name of the XPLINK version, the suffix '`.xplink`' will be added to the name of the locale. ASCII locales are only XPLINK. Source files for locales in the HFS are located in directory /usr/lib/nls/localedef. Object files for locales in the HFS are located in the /usr/lib/nls/locale directory.

The new locales are:

- Simplified Chinese in Hong Kong S.A.R. (EBCDIC)
- Simplified Chinese in Singapore (EBCDIC)
- Ukrainian in Ukraine (EBCDIC)
- Hindi in India (ASCII)
- Tamil in India (ASCII)
- Telugu in India (ASCII)
- Kazakh in Kazakhstan (ASCII)

Note: The new locales are part of the *G11N White Paper Currency II*. G11N is an abbreviation of globalization and the term is widely used in software development. One of its goals is to be involved in making the software flexible to handle locale-specific data, for example, to handle text in local languages, and handle date, time, numeric, and currency data according to local formatting convention.

3.15 DFSORT

DFSORT, an optional feature of z/OS, is the IBM high performance sort, merge, copy, analysis, and reporting tool. DFSORT, in conjunction with DFSMS and RACF, forms the software base for an IBM system managed infrastructure. This version of DFSORT has been enhanced significantly in z/OS V1R5, specifically in the areas of performance, usability, and serviceability.

An excellent document available on the Web describes what’s new in DFSORT for z/OS V1R5. It can be found at:

```
http://www.storage.ibm.com/software/sort/mvs/release_14/pdf/sortnew.pdf
```

3.15.1 Performance improvements

The performance benefits provided by the enhancements in z/OS V1R5 DFSORT are automatic. No changes are required on the customer’s behalf to take advantage of the enhancements. However, memory object sorting does require 64-bit architecture.

3.15.2 Memory objects

Prior to this release of DFSORT, sorting a large number of records required mass amounts of intermediate disk storage for work data sets. With this release, it is now possible for DFSORT to take advantage of memory objects to reduce this requirement.

DFSORT can dynamically determine, based on the size of the file being sorted and current central storage usage, the maximum size of a memory object that is used for memory object sorting.

Use of memory object sorting reduces I/O processing, sort elapsed time, the total number of EXCPs necessary to complete the sort, and overall channel utilization. Additionally, less disk space is required for sort work data sets during large sorting operations.

MOSIZE option

A new installation default and run-time option has been created to govern how memory object sorting will make use of central storage. The option is, MOSIZE. It is used to specify the maximum size of a memory object to be used for memory object sorting in 64-bit real architecture. The MOSIZE option is coded as follows:

```
MOSIZE=(MAX,n,p%)
```

Where:

MAX Directs DFSORT to dynamically determine the maximum size for a memory object to be used for memory object sorting, based on the size of the file to be sorted and current central storage usage activity.

n Directs DFSORT to dynamically determine the maximum size for a memory object to be used for memory object sorting up to a maximum value of “n” megabytes. The value specified for “n” must be between 0 and 2147483646. The actual value used for a memory object will not exceed the value specified, but may be less than “n” based on the size of the file to be sorted and current central storage usage activity. If MOSIZE=0 is specified, memory object sorting will not be used.

p% Directs DFSORT to dynamically determine the maximum size for a memory object to be used for memory object sorting, subject to an upper limit of p% of available central storage. The value specified for “p” must be between 0 and 100. If MOSZIE=0% is coded, memory object sorting will not be used.

ICEIEXIT exit

The ICEIEXIT user exit has been enhanced to provide support for the MOSIZE installation option to enable additional control over resources allocated for memory objects.

ICEIEXIT is a user-written and -installed DFSORT user exit that can be used to provide more control over selected installation and run-time options. When it is activated, ICEIEXIT receives control during sort initialization after it scans and validates the combination of installation and run-time options. These selected validated options are then passed to ICEIEXIT for processing.

Examples of how an installation might use the ICEIEXIT user exit to modify installation and run-time options at execution time include:

- Increasing or decreasing storage limits for sort jobs with different performance requirements.
- Setting different maximum storage limits for production and test sort jobs.
- Allocation of more storage for jobs with a large number of sort data sets.

New messages

The following new messages have been added to support memory object sorting:

```
ICE133I Options MOSIZE=r
```

Where r is MAX or decimal value.

```
ICE199I Memory Object Storage Used=nMBytes
```

Where n is the number of megabytes DFSORT used for a memory object during this sort.

3.15.3 Usability improvements

ICEMAC installation option defaults have been updated and new installation and run-time options have been provided to better control processing in situations where no records are written to sort output data sets.

ICEMAC installation defaults

ICEMAC default options have been changed to correct out of date defaults and to select defaults which are more in line with customers’ usual option designations. This may, for most customers, circumvent the need to build and execute an SMP/E usermod to override standard IBM installation defaults.

COBEXIT The default has been changed from COB1 to COB2. This parameter specifies the runtime library for COBOL E15 and E35 routines. COB1 is actually an obsolete parameter, but is still available for compatibility reasons.

DSA The default has been changed from 32MB to 64MB. This parameter specifies the maximum amount of storage available to DFSORT for dynamic storage adjustment during a blockset sort application when SIZE=MAX or MAINSIZE=MAX is specified.

TMAXLIM The default has been changed from 4MB to 6MB. This parameter specifies the maximum amount of total storage available to DFSORT for a blockset sort application when SIZE=MAX or MAINSIZE=MAX is specified.

ZDPRINT The default has been changed from NO to YES. This parameter specifies whether or not positive zoned-decimal fields resulting from summation should be converted to printable numbers.

3.15.4 NULLOUT installation and runtime option

The NULLOUT option allows the installation to specify what action DFSORT will take when no records are written to the SORTOUT data set. The installation can choose if an informational message or an error message is to be issued, and which return code is issued (0,4,or 16). This allows better control of jobstream actions through the use of conditional execution JCL and/or message processing based automation when potentially unexpected conditions arise.

The following table shows the possible return codes and message combinations, and the resulting impact on DFSORT processing.

Return Code issued	**Message issued**	**Processing**
Return Code = 0	ICE173I	Continues
Return Code = 4	ICE173I	Continues
Return Code = 16	ICE206A	Terminates

3.15.5 NULLOFL installation option and OUTFIL runtime option

The NULLOFL option allows the user to specify what DFSORT will do when no records are written to the OUTFIL data set.The installation can choose if an informational message or an error message is to be issued, and which return code will be issued (0, 4, or 16). This allows better control of jobstream actions through the use of conditional execution JCL and/or message processing based automation when potentially unexpected conditions arise.

The following table shows the possible return codes and message combinations, and the resulting impact on DFSORT processing.

Return Code issued	**Message issued**	**Processing**
Return Code = 0	ICE174I	Continues
Return Code = 4	ICE174I	Continues
Return Code = 16	ICE209A	Terminates

3.15.6 New messages

The new messages produced in support of the NULLOUT and NULLOFL options are:

```
ICE173I NO RECORDS FOR THE SORTOUT DATA SET - RC=N
```

where n = 0 or 4

```
ICE174I NO DATA RECORDS FOR AN OUTFIL DATA SET - RC=N
```

where n = 0 or 4

```
ICE206A NO RECORDS FOR THE SORTOUT DATA SET - RC=16
ICE209A NO DATA RECORDS FOR AN OUTFIL DATA SET - RC=16
```

3.15.7 Serviceability improvements

DFSORT messages have been updated to contain the APAR levels of DFSORT modules used during execution. This enhancement will assist the user in gathering diagnostic information that may be required by IBM service, and will eliminate the need to run SMP/E reports or create dumps to determine module service levels.

3.15.8 Migration considerations

Customers using previous versions of DFSORT that were dependent on the standard IBM installation defaults for COBEXIT, DSA, TMAXLIM, and ZDPRINT need to review the new defaults. If necessary, they may need to override the installation defaults using an SMP/E usermod or by specifying overriding run-time options.

Users may also want to consider the automation of job processing based on the customized return code and messages produced when a “no records written” condition occurs during DFSORT execution.

4

JES3 V1R5 enhancements

This chapter describes the following functional changes made to JES3 in z/OS V1R5:

- Support for Workload Manager (WLM) balanced initiators
- Truncated blanks restore support
- A new ENF58 signal type when a print checkpoint is taken
- JES3 Loop and Wait Monitor
- Greater than 8 media types
- Miscellaneous enhancements

4.1 Support for Workload Manager (WLM) balanced initiators

The following problems existed prior to JES3 V1R5:

- If initiators are removed from constrained systems and jobs on the select queue are eligible to execute only on these systems, these jobs would have to wait in the job select queue until WLM decides to start more initiators. This could cause unnecessary delays in existing job processing. With information supplied by JES3 about system affinity jobs for those systems, WLM does not remove initiators from the systems and hence the delays would be avoided for such jobs.

  If a job is eligible to run on more than one system and any one of the eligible systems is unconstrained, this job is not counted for constrained purposes.
- Prior to JES3 z/OS 1.5, WLM would only balance initiators when starting new ones. A system could be overcommitted, but because initiators have already been started there, jobs continue to be scheduled to that system. Another system with available resources could run those jobs quicker but WLM did not pay attention to the imbalance on the load on the system with the available initiator.

4.1.1 z/OS V1R4 support

Beginning with WLM in z/OS V1R4 and higher, changes were made to rebalance the distribution of initiators between the sysplex members. This capability is now supported by JES3 V1R5. While in earlier releases a balancing of initiators between high and low loaded systems was only done when new initiators were started, this is now done when initiators are already available.

The number of initiators on highly utilized systems is reduced, while new ones are started on less utilized systems. This enhancement can improve sysplex performance with better use of the processing capability of each system.

WLM attempts to distribute the initiators across all members in the sysplex to balance the utilization of the systems while taking care that jobs with affinities to specific systems are not hurt by WLM decisions. Initiators are stopped on systems that are utilized over 95% when another system in the sysplex offers the required capacity for such an initiator.

WLM also increases the number of initiators more aggressively when a system has low utilization and jobs are waiting for execution.

Initiator management

WLM with its new support tries to migrate initiators to significantly lower used systems by more aggressively reducing them on constrained systems and starting new ones on low usage systems. This check for potential rebalancing is made every 10 seconds.

Consider an example where there are four jobs in the batch queue and two of those jobs can only run on SY1, one on SY2, and one on either of them. That is what JES3 reports to WLM so that WLM can make an informed decision about whether to move the initiators from the constrained system (SY1) where three jobs could run to a less constrained system (SY2) where two could run.

Note: If the reduction of initiators on constrained systems is very aggressive, jobs with affinity to that system could be treated worse than before.

4.1.2 Determining constrained systems

Assume JES3 is doing sampling on SY1, as shown in Figure 4-1. First, JES3 finds out via the WLM service, IWMBQRY, what systems are constrained. JES3 provides a count of jobs that have the following:

- An affinity to the current system
- Eligibility to run on that system
- No affinity to any unconstrained systems

Job affinities

Jobs with an affinity to only constrained systems (SY1 and SY2, as shown in Figure 4-1) are also counted when reporting to WLM. If a job must run on SY1, it is included in the (constrained) count because it has no other place to run. With information about affinity for jobs for those systems, WLM would not remove initiators from the systems and hence the delays would be avoided for such jobs.

Jobs eligible for multiple systems

If a job is eligible to run on more than one system and any one of the eligible systems is unconstrained, we do not count this job for constrained purposes. If the job could run on an unconstrained system (SY3 in our example), the job is not counted.

If a job can run anywhere except SY3, the job is counted. If SY1 takes away its initiators, the job can only run on SY2, and SY2 isn't doing any better than SY1.

If a job can run anywhere except SY1, the job is not counted because SY1 can't do anything about it. Furthermore, this doesn't preclude SY2 from reducing its initiators. SY3 is the better choice and it has room for more initiators.

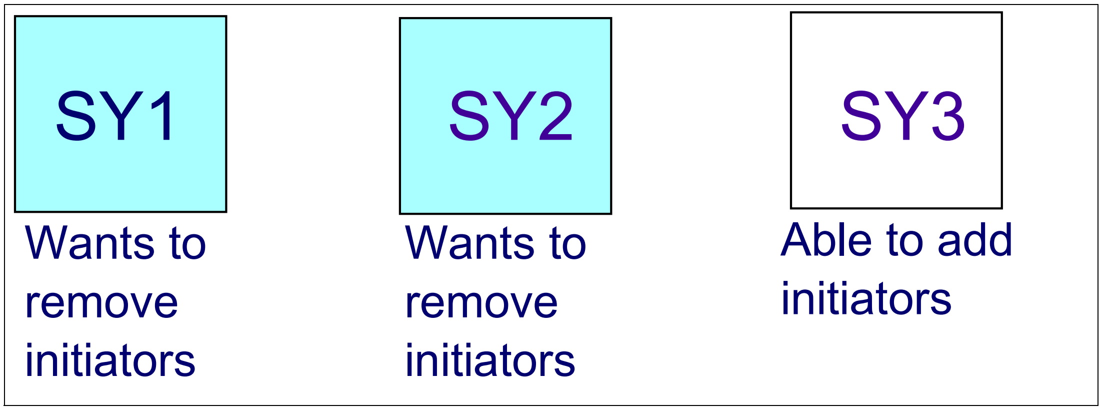


Figure 4-1 Three JES3 systems in a sysplex

4.2 ENF checkpoint signal

In z/OS V1R5, a new signal qualifier is added for applications that use the client print interface. A signal with this new qualifier is issued whenever a print application makes a checkpoint request or, if printing on a JES3-managed printer, when JES3 takes an internal checkpoint.

As with all signals of this type, the signals are issued only for data sets that were created with a client token.

4.3 ENF58 signal

When the ENF58 signal is issued, it includes the approximate line, page, and copy count of the file, except that on JES3 managed printers the page count is not provided.

The purpose of this signal is to provide a rough progress monitor so that an application can see how far a print file has progressed.

4.4 JES3 and blank truncation

Blank truncation suppression support is provided to solve the problem of JES3 stripping out trailing x'40's after assuming that they are blanks; but they are not always blanks and they are lost to printing functional subsystems that depend on the trailing x'40's to print the record.

The only workaround for this problem is to define a special SYSOUT class with TRUNC=NO, but this is not a good long term solution because it wastes spool space.

4.4.1 JES3 V1R5 and blank truncation

With JES3 V1R5, the x'40' characters are still truncated, but the original record length is preserved in each spool record. In the Functional Subsystem Interface the original length is passed back so that PSF and other applications can reconstruct the record back to the way it looked before truncation.

JES3 support has changed the DATCC to remember the original record length.

Application support

For an application to use this new support, the IATXDATX macro is used to address user data following a DATCC with or without extension. It also provides the data length and logical record length of the current record. The logical record length returned is zero if DATCC extension (DATCCX) does not exist.

Use the LRECL=YES on the IATXDATX macro to specify whether you want IATXDATX to return the original logical record length of the record, prior to truncation of trailing blanks. If you specify LRECL=YES, the original logical record length is returned in register 15.

Figure 4-2 on page 111 shows the new DATCCX that contains the original record count before blank truncation. The DATCC is the first 4 bytes and contains the following information:

Byte 1 This byte contains a set of flags to determine the kind of data record; page data (DATCPDS), the carriage control character is in machine code (DATMAC), or the carriage control character is in ASA code (DATASA).

The DATOPTCD flag specifies that the MRF to be opened contains table reference characters.

The DATSPLTB flag indicate that this record is split between this buffer and the next buffer.

The DATCON flag indicates that this record is a continuation from the previous buffer.

The DATSPAN flag indicates that this record spans this buffer, the previous buffer, and the next buffer.

The DATDATX flag indicates the presence of DATCC extension (JES3 V1R5)

Byte 2 Reserved by IBM for future use.

Byte 3-4 These two bytes contain the length of the spool record plus four bytes which is the length of the DATCC.

The DATCCX is 2 bytes and contains the original logical record length of a record (JES3 V1R5). If blank truncation is used, this is the original record length before truncation.

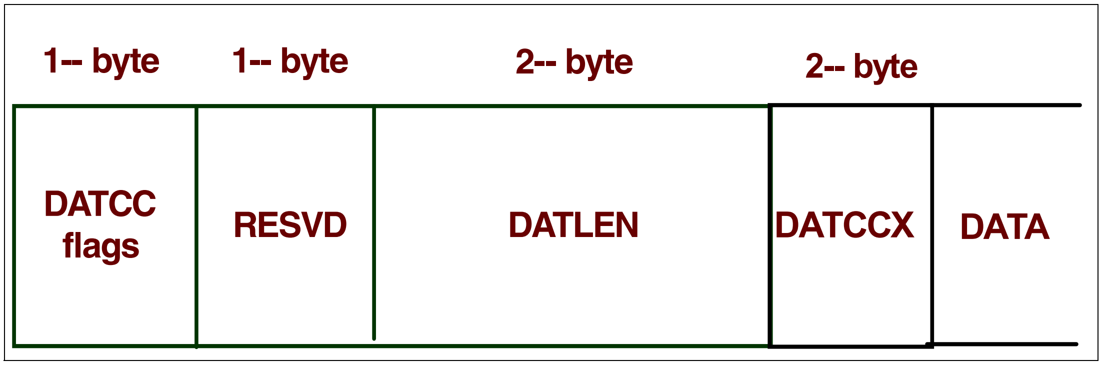


Figure 4-2 DATCC and new DATCCX with JES3 V1R5

Migration considerations

In order to keep the original record length it was necessary to extend the DATCC prefix of every spool record. But data records that are created before z/OS V1R5 are not extended. Therefore, user code that depends on the length has to be changed.

The new macro IATXDATX handles records with and without an extension. It locates the start of the data within the record, provides the current length, and provides the original length. Code changes, if applicable, must be made prior to introducing z/OS V1R5 on the global or any local.

Coexistence considerations

APAR OW57535 is required on OS/390 V2R10, z/OS V1R1, z/OS V1R2, z/OS V1R3, and z/OS V1R4 for coexistence, migration, and fallback because the extension of the DATCC causes a spool incompatibility. This same APAR is needed on any processor where a fallback from z/OS V1R5 is made to one of these releases, as it:

- Allows jobs to be present across the migration
- Allows jobs to be dumped and restored across the migration using the DJ DSP

  The Dump Job record header was similarly extended and OW57535 is needed to cope with the presence of an extension at a down level release.

APAR OW57535 includes a smaller version of IATXDATX that:

- Locates start of data whether or not a DATCCX is present
- Determines the current (truncated) record length
- Does not determine the original record length

User code changes to use IATXDATX must be in place on all processors before bringing z/OS V1R5 into the complex.

4.5 Loop and Wait Monitor

In this release, JES3 provides a Loop and Wait monitor to detect and report on loops and other exception conditions. The health monitor can prevent unnecessary outages.

This monitor checks for operator settings left in an undesirable state for a long period of time (for example, holding the queue or a priority and forgetting to release it later). A mistake like this has been known to cause a self-inflicted outage in the past.

If a dump is taken, the monitor can perform the same kind of analysis on a dump as it does on a live system, thereby performing automated first pass diagnosis. This can find some of the obvious problems, allowing time to be better spent on the diagnosis that is really needed.

Most importantly, the monitor can prevent outages. It detects loops while they are occurring. It also gives the operator the option to abend the looping task. In many cases the abend is recoverable and provides first failure data capture without causing an outage.

The JES3 Loop and Wait Monitor makes sure the two main JES3 tasks, the Nuc and Aux tasks, are not suspended or in a never-ending loop. When that happens, no other JES3 functions (called FCTs) can be dispatched besides the currently active one. Because the condition is difficult to detect by an operator, JES3 will periodically examine the status of those two functions and inform the operator, via WTOR IAT6410, about the unusual condition.

The operator can take action based on that information by either failing the currently active FCT or by taking the whole JES3 global down.

4.5.1 Loop and Wait Monitor commands

This monitor is a function of JES3 that monitors the activity of JES3 and detects any functions that are dispatched continuously for too long. The Loop and Wait Monitor is activated automatically on the global. Two parameters can be modified via an operator command:

- The interval at which JES3 checks what the active FCT is.
- The threshold for how long an FCT can be continuously active before the operator is notified.

There are two commands with this support, as follows:

- An operator command can be issued to detect certain unusual conditions on request.
- IPCS command can be used to detect unusual conditions after the fact when a dump has been taken and is being diagnosed.

JES3 operator command

The operator command allows the operator to check the “health” of the JES3 address space. A summary report is issued to the console listing any unusual conditions found when issuing the following command:

```
F JES3,CHK
```

Parameters to control the monitor

The **F JES3** command has two parameters to control the monitor activity as follows:

F JES3,INT=n This command allows changing of the Monitoring interval for the Wait and Loop monitor. n specifies the interval in seconds. The default value is 10 seconds. The maximum value is 59940 (999 minutes). The interval specification controls how frequently JES3 checks what the currently dispatched FCT is.

F JES3,THRSH=n This command allows changing of the Monitoring threshold for the Wait and Loop Monitor. n specifies the threshold in seconds. The default value is 30 seconds. The maximum value is 300 seconds.

> **Note:** The threshold determines when messages IAT6397, IAT6398, IAT6415, and IAT6410 are issued for an FCT that has been continuously active for the specified number of seconds.

Issuing the command

Figure 4-3 shows the results obtained at the console when the following command is issued on a busy system:

```
F JES3,CHK
```

```
IAT6397   FCT ISDRVR   (NODEVICE) HAS BEEN DISPATCHED CONTINUOUSLY FOR:
IAT6398     00000 HOURS  00 MINUTES  38 SECONDS
IAT6415 at 056B814C in LOADMOD=IATUX28 , EPNAME=IATUX28 ,EPADDR=056B8120,LEN=000080
*15 IAT6410 Reply 'JES3    ' to fail the FCT or 'nnn' to ask later
15,20
IEE600I REPLY TO 15 IS;20
IAT6397   FCT ISDRVR   (NODEVICE) HAS BEEN DISPATCHED CONTINUOUSLY FOR:
IAT6398     00000 HOURS  01 MINUTES  22 SECONDS
IAT6415 at 056B814C in LOADMOD=IATUX28 , EPNAME=IATUX28 ,EPADDR=056B8120,LEN=000080
*16 IAT6410 Reply 'JES3    ' to fail the FCT or 'nnn' to ask later
```

Figure 4-3 Output messages from F JES3,CHK command

- Figure 4-4 on page 114 shows the result if the operator issues a `FAIL DSP`**`, 16,JES3,`** for the DSP.

```
SY1  16,jes3
SY1  IEE600I REPLY TO 16 IS;JES3
SY1  IAT3713  FAILURE LOGOUT
IAT3713  *************************************************************
IAT3713  *************************************************************
IAT3713  DATE = 2003169 TIME = 1636249   JES3 z1.5.0
IAT3713  JES3 FAILURE NUMBER = 0001   FAILED  DM134
IAT3729  FAILURE EXPLANATION:
IAT3832  OPERATOR FAILED THE FCT VIA ACTIVE FCT MONITOR.
IAT3713  ACTIVE FCT = ISDRVR     DEVICE = NONE      FCT FAIL NO = 0001
IAT3713  MODULE = NOT JES3       MOD BASE = 00000000  DISP = 000000
IAT3713  APAR NUMBER =           PTF NUMBER =
IAT3713  CALLING SEQUENCE (HIGHEST LEVEL MODULE LISTED LAST)
IAT3713  MODULE = IATISJB        MOD BASE = 05A6F710  DISP = 000C10
IAT3713  APAR NUMBER =           PTF NUMBER = 1.5.0
IAT3713  MODULE = IATISLG        MOD BASE = 05A6CE48  DISP = 000420
IAT3713  APAR NUMBER =           PTF NUMBER = 1.5.0
IAT3713  MODULE = IATISDV        MOD BASE = 05A6A138  DISP = 000F04
IAT3713  APAR NUMBER =           PTF NUMBER = 1.5.0
IAT3713  PSW AT TIME OF FAILURE  071C0000 856B814C  ILC  2
IAT3713  THE FAILING INSTRUCTION IS   8190
IAT3713  REGISTERS AT TIME OF FAILURE
IAT3713  REGS  0- 3   00000000   0565532C   00000001   7F241964
IAT3713  REGS  4- 7   05A69B80   7F3BC00C   7F241929   05A69B80
IAT3713  REGS  8-11   7F3BC360   05A70710   056B8120   05877050
IAT3713  REGS 12-15   05607000   05A67F18   85659F8E   056B8120
IAT3713  *************************************************************
IAT3713  *************************************************************
SY1  IAT3702 ISDRVR             FAILED  DM134 - JES3 FAILURE NO. 0001
```

Figure 4-4 Operator fails DSP messages

- On a fairly idle system, the messages shown in Figure 4-5 were received.

```
F JES3,CHK
********************************************************
*  General system status
*
********************************************************
Task Information
  TCB 007E9E88 Task has been set nondispatchible

DLOG Information
  DLOG got a busy return from WTL at some point in time

Workload Manager Information
  The WLM FCT is in sleep mode
********************************************************
*
*  Spool Related Exceptions
*
********************************************************
Extent Table Information
  02F4B5D0 SPOOL1   This extent contains a dynamic STT
IAT6413 COMMAND CHK PROCESSING ENDED
```

Figure 4-5 F JES3,CHK command

- Diagnosing a dump, a new IPCS command formats the same report provided by the **F JES3,CHK** operator command for the dump, to assist in first level problem diagnosis.

  This monitor can be run on a dump using a new IPCS verb exit. The same exception conditions that the operator command checks on a live system are checked in a dump. This can provide first level problem diagnosis automatically, so that valuable diagnosis time is saved by not having to manually check some of the obvious things first.

  When diagnosing a dump, issue the following command:

  ```
  VERBX JES3 'OPTION=CHK'
  ```

Note: In addition, for JES3 dump formatters in general, formatting data for various areas that are not on the JES3 global now use the SVT where possible instead of being dependent on the JES3 TVT. This reduces the number of occurrences of the message "JES3 TVT NOT ACCESSIBLE - FORMATTING TERMINATED" when formatting information in a dump.

4.6 Greater than 8 tape media types

JES3 z/OS V1R5 has new support to provide the BCP support for up to 255 media types. The combination of z/OS V1R5 JES3 and BCP are required on the following systems:

- JES3 global
- Any processor on which a job needing the support might execute
- Any processor on which a job needing the support might go through converter/interpreter in a CIFSS address space

4.6.1 DFSMS support

When the system-managed tape library was originally designed, it supported up to 8 media types and 15 recording technologies. Four of the eight media types (MEDIA1/CST, MEDIA2/ECCST, MEDIA3/HPCT, and MEDIA4/EHPCT) and five of the 15 recording technologies (18-TRACK, 36-TRACK, 128-TRACK, 256-TRACK, and 384-TRACK) are used today.

The tape library now supports 255 media types and recording technologies to allow you to easily use additional media types and recording technologies as they become available. As a result of this change, the LIBRARYENTRY in the VOLCAT has increased in size.

4.7 Miscellaneous enhancements

A number of small changes have been made to JES3 V1R5, based on user requirements.

4.7.1 Message IAT3040 enhancement

Message IAT3040 has been modified to indicate what system is the global. The source of the message is the Complex Status Record (CSR). If the global is the current system coming up, the name is bracketed by plus signs. If the global is deemed to be somewhere else, its name is embedded within asterisks. The IAT3040 message now looks like the following:

```
SY1  IAT3040 STATUS OF JES3 PROCESSORS IN JESXCF GROUP nodex
SY1  IAT3040 SY1     +  +, SY2     (  ), ...
```

+ + Indicates the active system is the global
*** *** Indicates what other system is the global

4.7.2 Job segment scheduler (JSS)

You can use the NOREQ parameter on the JES3 **start** command, as follows:

```
S JES3,PARM=NOREQ
```

This parameter instructs JES3 initialization to simulate the ***S JSS** command as soon as JES3 comes up. This way you can avoid delays waiting for the operator to issue the command assuming you do not use your automation to issue the command.

4.7.3 WTO logging

There was a subtle change made to the way WTOs are logged. A job can issue a WTO using the JOBID keyword and specify someone else's jobid, provided they are authorized. In prior releases, the message might have been logged in the target job whose jobid was used. In JES3 z/OS V1R5, the message is logged in the issuing job's JESMSGLG data set using the issuing job's ASID.

4.7.4 Dumping JES3

When the ***DUMP** command is issued in this release, it does not terminate JES3 after taking the dump.

4.7.5 DSI processing

JES3 DSI processing is changed as part of the fix for FIN APAR OW48297. The ENQ now prevents two globals from coming up at the same time.

The exclusive ENQ that protects this is now:

```
SYSZIAT/GLOBAL.ckptvolser.ckptdsname
```

ckptvolser The Volume Serial from where the checkpoint data set was allocated

ckptdsname The checkpoint data set name

The ENQ prevents another global from coming up in the complex:

- During DSI
- At any other time

It allows a failed DSI to revert to the old global.

4.7.6 New parameter on SETPARAM statement

The new parameter in the SETPARAM initialization statement is SDEPZERO. It indicates whether jobs that require a tape mount, but are in a CLASS which has SDEPTH=0 defined, should wait on the MDS allocate queue (which is the default), or be sent to the MDS error queue (meaning it failed allocation).

The SDEPZERO parameter is used as follows:

```
SDEPZERO= WAIT | ERROR
```

MDS Inquiry command

To display the SDEPZERO option in addition to the information that was previously displayed use the following command as the SDEPZERO specification.

```
*I S
```

5

JES2 V1R5 enhancements

This chapter describes the enhancements and changes that have been incorporated into JES2 V1R5. The following topics are discussed:

- Multilevel security support
- ENF 58 enhancements
- 1 Byte console ID elimination
- CONDEF statement
- JES2 command character in SSI (54)
- New $DSERV macro
- Message additions and enhancements
- JES2 patching facility enhancements
- Monitor $TRACE IPCS support
- Multisystem JES2 dump enhancements
- Control block changes - $HASXB
- JES2 z/OS V1R5 installation and migration
- Changed installation exits

5.1 Multilevel security support

JES2 V1R5 can now limit job selection based on security labels, and you can now specify a subset of members to which SECLABELs apply. A RACF **SETROPTS** command option (security label by system) now controls whether SECLABELs are active on all systems or only those you specify. JES2 maps the systems for which a SECLABEL is active against an affinity mask associated with each batch job. JES2 then uses that affinity mask to determine where a job can be selected for conversion and execution. This new SECLABEL affinity can increase security within your system, but it can also prevent job selection.

5.1.1 SECLABEL by system

The function that limits where a SECLABEL is active is called *SECLABEL by system*. The security administrator can define a security label to be active only on certain members of a sysplex. Using security labels on a per-system basis allows the installation to separate work based on security classification while still sharing the RACF database. The security administrator activates the use of system-specific security labels, as follows:

```
SETROPTS SECLBYSYSTEM
SETROPTS RACLIST(SECLABEL) REFRESH
```

When SECLBYSYSTEM is active, JES2 insures that no job is run on a member that does not have an appropriate security label active. If no system is available on which a job's security label is active, the job remains in the conversion phase.

Note: NOSECLBYSYSTEM is the RACF default.

Defining system-specific SECLABELs

To define system-specific security labels, the security administrator specifies on which systems a security label is to be active by adding a member list to the SECLABEL resource class profile. The member names are system SMF IDs (for example SY4, SY7) containing one to four characters. For example, to define the security label named BATCHIMP as being active only on the systems with SMF system IDs SYSA and SYSB, the security administrator could define BATCHIMP with a command like:

```
RDEFINE SECLABEL BATCHIMP....ADDMEM(SY4,SY7)
```

If there are no entries in the member list for a SECLABEL profile, then the SECLABEL is active on all systems. The values specified in the member list are the SMF IDs of the systems where the SECLABEL is active. If RACF checks on a system where a SECLABEL is not active, the check fails in the same manner it would if the SECLABEL was not defined.

5.1.2 JES2 affinity masks

JES2 support for system-specific security labels involves maintaining an affinity mask for each job that has not executed and that has a list of systems where the SECLABEL assigned to the job is active.

When you activate security label by system (SECLABEL by system), you can limit the MAS member(s) on which a job can run. Therefore, a job might not be able to run if there is no system that satisfies the requirements for:

- System affinity (SYSAFF=)
- Scheduling environment affinity (SCHENV_AFF=)
- SECLABEL affinity (SECLABEL_AFF=) - (new affinity mask with JES2 V1R5)

New affinity mask

The new affinity mask available in JES3 V1R5 is defined as discussed in "Defining system-specific SECLABELs" on page 118. JES2 uses the new mask to determine the following:

- Where a job can convert (due to RACF checks at conversion)
- Where the job can execute

To build the mask, JES2 has to track the SMF IDs for all members of a MAS. JES2 also listens to a new RACF ENF that indicates when the SECLABEL class is:

- Activated (RACLISTed)
- De-activated (NORACLISTed)
- Updated (RACLIST REFRESH)

Affinity mask for job selection

For a job to be selected, JES2 must find an active system in all three affinity lists. The intersection of the three sets of possible affinity masks, shown in Figure 5-1, defines the select group of systems on which the job can run. If the intersection is empty, the job cannot run. In the example that shows all three affinities, only members in the intersections of the affinities can run or convert jobs.

In this case, the job can convert on members SY4 and SY7. The job can only run on member SY7.

Note: Also, for example, if a job has no SCHENV affinity defined, then this affinity is not considered in the decision.

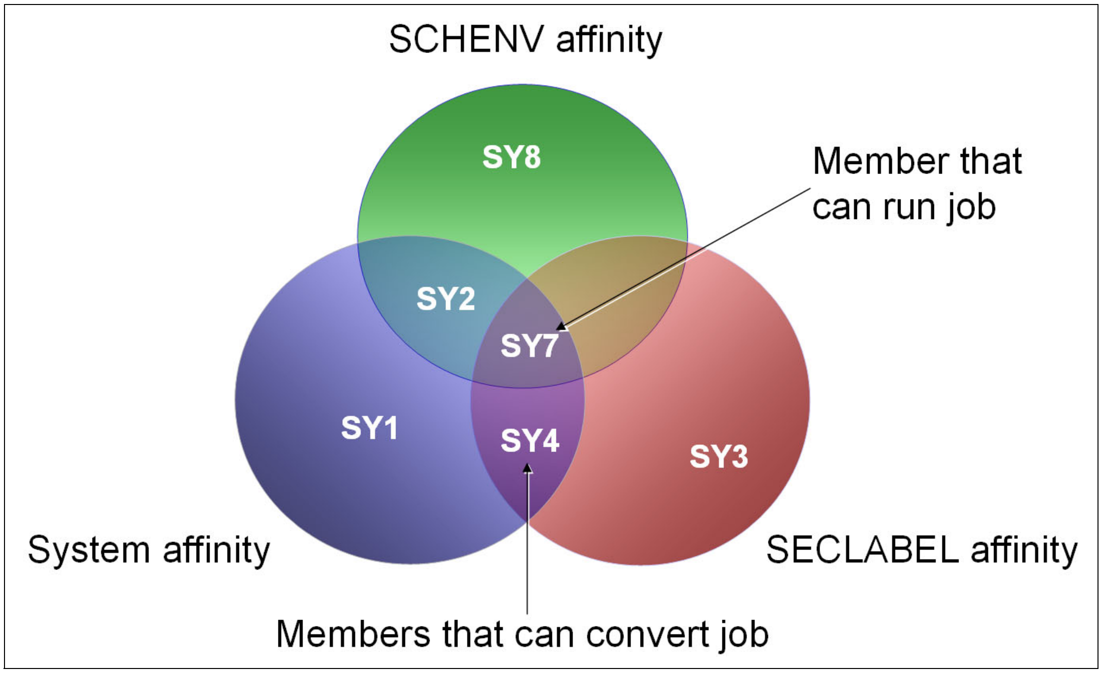


Figure 5-1 Affinity example for scheduling a job for conversion and execution

5.1.3 Commands for SECLABEL affinity for jobs

The following commands have been modified to display information about SECLABELs:

- A new keyword is added to the **$DJ** command, SECLABEL_AFF=member, that displays the JES2 MAS members on which the SECLABEL for this job is available. This is only shown if the RACF SECLBYSYSTEM option is also active. SECLABEL is added in the display of the following commands:

  ```
  $DJ100,LONG or $DJ100,SECLABEL_AFF=member
  ```

- The **$DJ**, with the option **DELAY** command is updated with SECLABEL delay as follows:

  The command displays the delay reason why a pre-execution job will not enter execution and the new reason is:

  ```
  SECLABEL - The security label (SECLABEL) assigned to the job is not defined
  as active on any member that is active.
  ```

- The **$DMEMBER** command is updated with the MVS SMF ID, as follows:

  SMFID[=smfid] - Specifies that the SMF ID of the MVS image of the member is to be displayed. You can also specify wildcards on this parameter. The SMF ID will not be displayed if it is equal to the member name (NAME=) or if the member is inactive.

5.1.4 Other changes for SECLABEL affinity for jobs

The SECLABEL by system affinity for jobs has other changes for the following functions:

- Extended status SSI returns new mask

  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 JES2 queue now displays the SECLABEL for the job or SYSOUT.

- Now used in **$SJ** and WLM sampling

  The SECLABEL affinity mask also affects what is reported in WLM sampling data and system selection for **$SJ** commands.

- Considerations in JES2 Exit 14 (job queue work select exit)

  Installations using exit 14 (replace JES2 job selection) have to examine the exit logic to determine if modifications are needed for the new affinity mask.

5.1.5 Device SECLABELS for SDSF

One of the objectives of the new support is to be able to “hide” objects that users cannot access at their current SECLABEL. For displays of jobs and SYSOUT (JOEs) this is easy since the corresponding data areas have SECLABEL fields. However, for devices and NJE nodes, the SECLABELs are not readily available. For SDSF to extract the SECLABELs every time it displays the page would greatly impact SDSF performance. Instead, JES2 will extract the SECLABELs for SDSF’s use, as follows:

- When JES2 is initialized
- When a device is defined via **$ADD** commands
- When a device is started
- When an NJE full path is done

Due to the volumes of SECLABELs that may need to be extracted, a request to look at devices just after JES2 starts (particularly a hot start) may not display some devices if the SECLABELs have not been extracted yet. Retrying the request at a later time (a few minutes later) should give JES2 a chance to complete the extracts. JES2 does not get notified when the SECLABELs associated with a device is updated. If the SECLABEL associated with a device is updated, restart the device to extract the SECLABELs for SDSF.

Note: JES2 security checks are not affected by the SECLABELs extracted for a device.

5.1.6 SECLABELS for JES2 address space

The JES2 address space is a server address space that performs work for users running with different security labels. In particular, during conversion processing, JES2 anchors ACEEs with SECLABELs matching the job being converted to the converter subtask. Because of changes in the way RACF does security checking for address spaces that create task level ACEEs, if multi-level security options are active, then the user ID associated with the JES2 started task should have a default security label of SYSMULTI. Failure to do so results in jobs failing to convert with the following messages:

```
ICH408I USER(userid) GROUP(group) NAME(programmername)
LOGON/JOB INITIATION - USER SECLABEL NOT COMPATIBLE WITH SERVER
$HASP313 jobname JES2 is unable to create the security environment
SECURITY PRODUCT RETURN CODE = 00000038 REASON CODE = 00000014
```

5.2 ENF58 enhancements

JES2 generates an ENF58 record whenever a checkpoint is taken by a device processing a SYSOUT data set. Periodic checkpointing monitors the processing and progress of printing to ease print restart in the event of a printing failure.

This new type of ENF58 record allows applications that are monitoring the processing of SYSOUT (data sets with a client token) to not only track that a data set is printing but also how much of the data set has printed. The frequency of ENF58 record writing is dependent on how often JES2 takes a checkpoint. For 3800 devices, the CKPTLINE= and CKPTPAGE= parameters on the PRTnnnn initialization statement influence the rate at which JES2 takes checkpoints (and thus the frequency of ENF58 records). For FSS devices, the CKPTSEC= or CKPTPAGE= parameter can be used to control checkpoints.

The ENF58 signal provides the current counts for:

- Pages
- Records
- Copies completed - Count starts at 0

5.3 1-Byte console ID elimination

In JES2, message processing was updated in previous releases to eliminate all uses of 1 byte console IDs. However, the use of the MGCR macro was overlooked. In this release all MGCRs were updated to MGCRE and 4-byte console IDs.

The IBM-supplied external writer was still using 1 byte console IDs for message routing. It also did not support the use of CARTs to connect messages to the command that issued it. Both these problems were fixed in this release; now “4-byte” console IDs are used for messages where appropriate.

5.4 CONDEF statement

A close parenthesis, ")" is no longer a valid CONCHAR= (JES2 commands identifier) or RDRCHAR= (JES2 commands from local or remote card reader identifier) specification on the CONDEF initialization statement.

CONCHAR specifies the character that will be used to identify JES2 commands from local consoles. If a command from a local console begins with the character specified for CONCHAR, JES2 assumes that the command is a JES2 command and attempts to process it.

> **Note:** You must use the default CONCHAR= character ($) on JES2 commands imbedded in the JES2 initialization data set regardless of what you specify for CONCHAR=.

5.4.1 Migration actions

If you are using a ")" as your CONCHAR= and in addition RDRCHAR= symbol specification, be certain to change it to any other valid symbol. The remaining valid characters are:

```
( % : . &+ _ # < ! - > @ " / ? = ¬ *
```

5.5 New $DEVSERV macro

A number of JES2 macros that were added in OS/390 V1R4 require a DSERV address when used from the subtask environment. A DSERV is a representation of a checkpoint version which is needed to access checkpoint data outside the main task. To simplify the code needed to obtain a DSERV, a new macro $DSERV and service was created to obtain a DSERV and return it when done.

Use $DSERV to request that JES2 invoke checkpoint version processing to either obtain (GET) or free (FREE) a DSERV, the parameter list used by authorized programs to request job information service from the JES2 checkpoint data space.

This macro can be used with other JES2 services:

- $DOGJQE - Deliver or get JQE (job queue element)
- $DOGCAT - Deliver or get CAT (class attributes table)
- $DOGWSCQ - Deliver or get WLM (work load manager) service class

> **Note:** To isolate your exit routines that need to obtain a checkpoint version from any future changes IBM might implement to checkpoint version processing, we recommend you use the new $DSERV service.

5.6 Message additions and enhancements

The following messages are new:

- $HASP892 - QUEUE_ERROR(n) with reason codes added
- $HASP9107- {No} JES2 ERROR COUNTS SINCE
- $HASP9133 - JES2 ERROR HISTORY
- $HASP9163 - FAST SPOOL GARBAGE COLLECTION (SPOOLDEFGCRATE=FAST)

The following messages have been changed:

- $HASP003 - during $scan processing, an error was detected
  - new reason code (100) added
  - message says that at least one data set is required in proclib concatenation
- $HASP607 - issued when JES2 cannot terminate cleanly ($PJES2)
  - Message contains reasons why JES2 cannot terminate cleanly
  - Problem is message sometimes DOMed too soon
  - Message is now retained and updated as conditions are cleared

5.7 JES2 patching facility enhancements

The JES2 patching facility applies temporary patches to the in-storage copy of JES2 modules.

The following enhancements were made:

- Access to JES2 data space (by data space name)
- Deferred patch to CKPT data until CKPT is read

5.8 Health monitor enhancements

IPCS support is for dump analysis only and does not affect a running system. You can dynamically update or replace the JES2 IPCS parmlib member and IPCS panels and modules.

The monitor data areas:

- Perform formatting as tables not just control block formatting
- Give the same information as monitor commands

5.9 $TRACE IPCS support

IPCS has been enhanced to format the JES2 monitor data areas. The areas are formatted both as raw control blocks and as tables similar to what is displayed in monitor commands.

IPCS has been enhanced to support formatting instorage $TRACE tables. This is useful if you want to run a JES2 trace without actually logging it to a SYSOUT data set. Some traces have individual fields formatted ($SAVE/$RETURN) but others are just the trace headers with hex dumps.

5.10 Multisystem JES2 dump enhancements

JES2 already has support to dump other members when an error is detected. The problem is that the current support only allows the local member or all members to be dumped. However, there are cases where you may want to dump the local member and one other member. In the past, we did not use the multi-system dump capability in these cases because of the impact dumping all members can have on a sysplex.

To address this, JES2 recovery code was updated to support having a list of systems to be dumped. This support is used when errors are detected with data from another system. As service finds more cases where another member would be useful, the recovery processing for those cases will be updated. This is not something that installations need to do anything about. It is just a new function that should aide in the debugging of certain problems.

5.11 Control block changes - $HASXB

In z/OS V1R5 JES2, JES2 frees the $HASXB at job termination time, not at end-of-memory time as in previous releases.

If your processing is dependent upon user fields (HXBUSER1) being available at end-of-memory time, you should consider using an alternate field.

Note: Examine your exits for references to HXBUSER1 and make updates as necessary if your exit depends on $HASXB user fields being accessible at end-of-memory time.

5.12 JES2 z/OS V1R5 installation and migration

The new release of JES2 still supports running with an R4 mode checkpoint. This is in addition to the z2 mode checkpoint. However, it is recommended that installations migrate to z2 mode when they no longer require running in a MAS with a OS/390 V2R10 system. In a future release, when OS/390 V2R10 is no longer supported in a MAS, support for R4 mode will be discontinued.

z/OS V1R5 can coexist in a MAS with JES2 versions OS/390 V2R10, z/OS V1R2, and z/OS V1R4. However, APAR OW55708 *must* be installed on any down-level members.

Note: Review *z/OS JES2 Migration,* GA22-7538 for any exit changes that may be needed.

5.13 Changed installation exits

JES2 installation exits 14, 34, 36, and 37 were modified with JES2 V1R5. Make sure these changes are considered if any of these exits are to be used.

5.13.1 Exit 34 (Subsystem interface data set unallocation)

On entry to this exit, at 16 bytes (decimal) into the data area pointed to by register 1 (R1) is the address of the PDDB (peripheral data definition block) of the data set being unallocated. Previously, this address could be zero only if the data set type was an internal reader or an unknown data set type. It is now also set to zero if a PSO unallocation was performed after the JOB-step TCB ended. In previous releases, the pointer in this case was residual and pointed to storage that could have been re-used for another PSO request.

Note: Examine your Exit 34 code to ensure that the potential zero address for PSO unallocation can be processed correctly.

5.13.2 Exits 14, 36, and 37

In support of multi-level security support through RACF, JES2 can now limit job selection based on “security label by system” (SECLABELs). JES2 maps the systems for which SECLABELs are active against an affinity mask associated with each batch job. If you use Exit 14 to replace normal JES2 job selection or Exit 36 and Exit 37 to provide pre- and post-security authorization calls, you need to consider the following processing interactions.

Exit 14: (job queue work select - $QGET)

You can use Exit 14 (Job queue work select - $QGET) to replace the normal JES2 job selection for job execution or conversion and use security label by system to effect system affinity. If the RACF SETROPT option for SECLABEL by system is active, then JES2 considers SECLABEL settings when selecting a job. A new field, JQASCLAF, contains an affinity mask of JES2 members where the SECLABEL is available. SECLABEL affinity applies to selection of job for conversion and execution only.

Exit 36 and Exit 37: (pre-security and post-security authorization call)

With these routines, JES2 can now pass a new RACF request type to the exit. JES2 can request a “branch entry extract” to extract information from SECLABEL profiles (WAVREQST field set to WAVRXTRB). In addition, JES2 now also uses the RACF extract (non-branch entry) to extract SECLABELs from various other profiles (WAVREQST field set to WAVRXTRT). Previously, JES2 defined this call but had never used it. New function codes ($SEASCLA - SECLABEL affinity extract and $SEASCLE - DCT SECLABEL extract) are now defined for these requests.

> **Note:** Examine your Exit 14, Exit 36, and Exit 37 code to ensure that any exit routine processing that needs to access the new security-related fields does so and the exits do not compromise JES2 by not including code needed to replace normal JES2 processing.

5.14 Workload manager policy and default processing

MVS uses the default workload manager (WLM) policy when MVS is started without a WLM couple data set or if there is a problem or error when accessing the current policy in the WLM couple data set.

The default WLM policy assigns a null service class (SRVCLASS= on a `$DJ` command) to all jobs.

> **Note:** Previous releases of JES2 did not support running jobs in WLM-managed initiators if the service class of the job was null.

The null SVRCLASS restriction has been removed, and now allows jobs with blank service classes to be executed under a WLM-managed initiator.

> **Note:** This may impact local exits or system management applications.

6

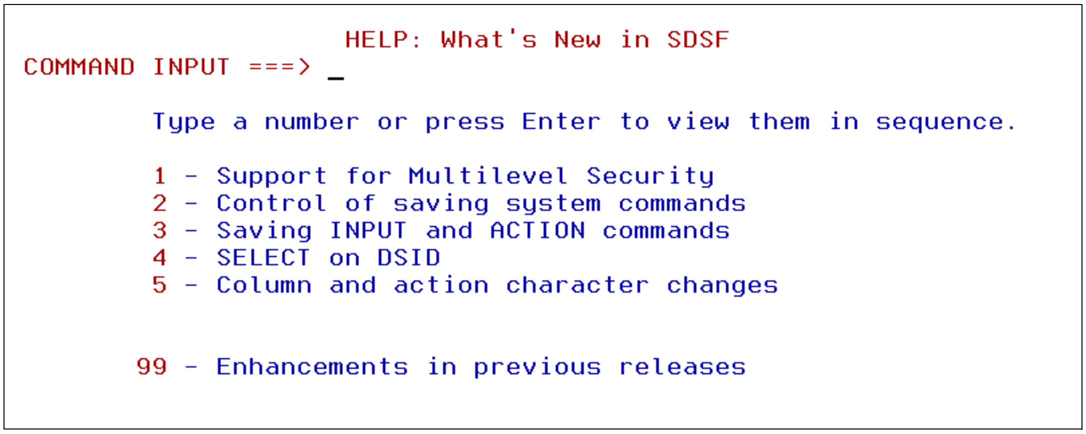
SDSF enhancements

This chapter describes the enhancements and changes that have been incorporated into z/OS V1R5 SDSF. The following topics are discussed:

- What’s new in SDSF
- Support for multilevel security
- Control of saving system commands
- Saving INPUT and ACTION commands
- Simplification of command authorization
- Server enhancement
- Column and action character enhancements
- Controlling the format of CPU information
- Improved folding of mixed case text
- Removal of SNAP dump
- Protection of RMF services
- Select with DSID on JDS

6.1 What is new in SDSF

For a quick reference of what is new in SDSF, go to the SDSF primary option menu, type in HELP or PF1, and select option 1 from the HELP menu that is displayed. The menu shown in Figure 6-1 is displayed.



```
                     HELP: What's New in SDSF
COMMAND INPUT ===> _

        Type a number or press Enter to view them in sequence.

        1 - Support for Multilevel Security
        2 - Control of saving system commands
        3 - Saving INPUT and ACTION commands
        4 - SELECT on DSID
        5 - Column and action character changes


       99 - Enhancements in previous releases
```

Figure 6-1 SDSF help panel - Option 1

6.2 Support for multilevel security

SDSF adds a number of enhancements to support the multilevel security (MLS) function in z/OS V1R5. Multilevel security allows the classification of data and users based on a system of hierarchical security levels combined with non-hierarchical security categories.

SDSF support changes for this release include the following:

- When MLS is in effect, SDSF filters the rows displayed on tabular panels based on rules for SECLABEL dominance. SDSF also checks for SECLABEL dominance when browsing a data set. See “SECLABEL dominance” on page 129.
- SDSF adds the user's SECLABEL to the response for the **WHO** command.
- SDSF adds a SECLABEL column to the DA, INIT, PUN, and RDR panels (for the job). SDSF also adds a DSecLabel column to the NO, PR, PUN, RDR, LI, and SO panels (for the device).
- SDSF’s support for multilevel security requires the SDSF SAF (security authorization facility) support to be used.
- SDSF does not explicitly prevent the use of ISFPARMS with multilevel security; if SAF is not active, authorization falls back to ISFPARMS.
- SDSF’s sysplex support for the device panels and browse function, which requires WebSphere MQ, cannot be used in the multilevel security environment.
- Installations already using the sysplex support should disable it by removing the server group definitions from ISFPARMS.

Note: Multilevel security is not available with SDSF’s internal security module, ISFPARMS.

6.2.1 Dependencies

The new columns in SDSF require z/OS V1R5 JES2.

For multilevel security, the installation must ensure that the appropriate SAF classes are active. Refer to z/OS Planning for Multilevel Security.

In a sysplex environment with some systems at levels below z/OS V1R5 SDSF, SDSF cannot check the user's dominance in relation to the lower level systems on the device displays, and so will assume that the user does not have dominance.

6.2.2 SECLABEL dominance

With multilevel security support, SDSF restricts the rows displayed on a panel using SECLABEL dominance by invoking the RACROUTE REQUEST=DIRAUTH service. SDSF caches the results of the DIRAUTH calls so that dominance checks are reused.

SDSF compares the SECLABEL of the object to the SECLABEL of the user. For example, to control a printer, the user would need:

- Access to the Printer panel (PR command)
- SECLABEL dominance over the specific printer
- Authority to type an action character or modify a field

Wherever possible, SDSF obtains the SECLABEL for an object (job, device, and so forth) from the system. When SDSF cannot obtain a SECLABEL for an object, SDSF assigns a SECLABEL to it.

Note: No SECLABELs are associated with the SYSLOG panels.

6.2.3 SECLABEL source for SDSF panels

For APPC transactions running under ASCHINIT and z/OS UNIX processes running under BPXAS, the SECLABEL of the transaction is not available; SDSF uses the SECLABEL of the initiator.

Table 6-1 SDSF panels that display SECLABELS (1)

Panel	Object	SELABEL Source
DA	Address Space	JQE (RMF) / SJB (non-RMF) *
I, ST	Job	JQE
O, H	Output	JOE
JDS	Data set	PDDB
OD	Output descriptor	PDDB

* For jobs not running under the local JES2, SDSF assigns SYSHIGH.

For browse, the dominance check is part of the verify call that SDSF makes to ensure the user is authorized to the data set.

Table 6-2 SDSF panels that display SECLABELS (2)

Panel	Object	SECLABEL Source
INIT	Initiator	SJB if active* / SYSNONE
PR, PUN, RDR, LINE, SO	Device	DCT

Panel	Object	SECLABEL Source
NO	Node	NIT
PS	Processes	OdmvSecLabel / SYSHIGH
Browse	Data set	Rtoken

* This is the SECLABEL for the job that is active on the initiator.

Table 6-3 SDSF panels that display SECLABELS (3)

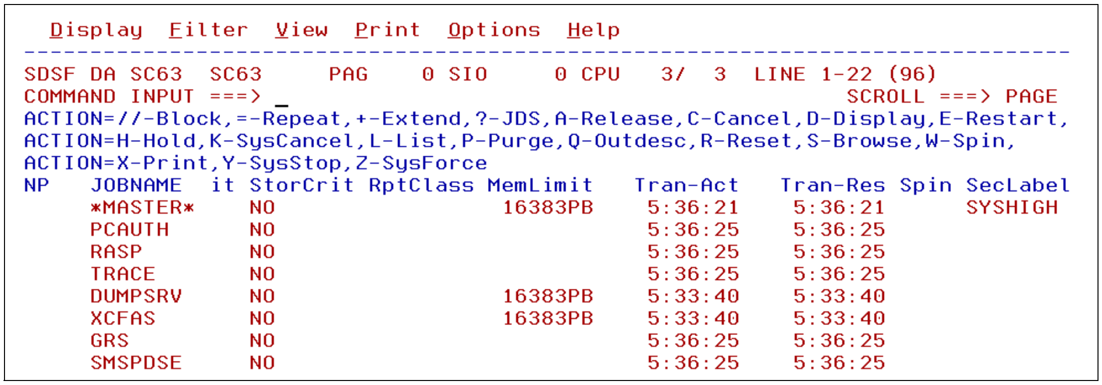
Panel	Object	Source - assigned by SDSF
MAS	System	SYSNONE
JC	Job class	SYSNONE
SP	Spool volume	SYSNONE
SE	Sched env	SYSNONE
RES	WLM resource	SYSNONE
ENC	Enclave	SYSHIGH
SR	Reply message	SYSHIGH

6.2.4 Panel and command changes

The security label is passed to the table build exit for the DA, INIT, PS, and RDR panels.

SecLabel column

The SecLabel column is added to these panels as shown in Figure 6-2.



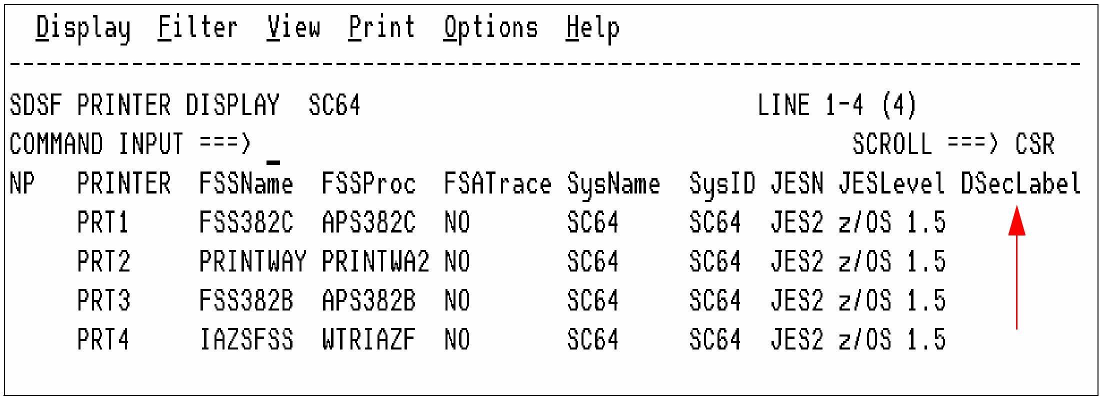
```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF DA SC63  SC63     PAG    0 SIO     0 CPU   3/   3  LINE 1-22 (96)
COMMAND INPUT ===> _                                          SCROLL ===> PAGE
ACTION=//-Block,=-Repeat,+-Extend,?-JDS,A-Release,C-Cancel,D-Display,E-Restart,
ACTION=H-Hold,K-SysCancel,L-List,P-Purge,Q-Outdesc,R-Reset,S-Browse,W-Spin,
ACTION=X-Print,Y-SysStop,Z-SysForce
NP   JOBNAME  it StorCrit RptClass MemLimit   Tran-Act   Tran-Res Spin SecLabel
     *MASTER*    NO                 16383PB    5:36:21    5:36:21      SYSHIGH
     PCAUTH      NO                            5:36:25    5:36:25
     RASP        NO                            5:36:25    5:36:25
     TRACE       NO                            5:36:25    5:36:25
     DUMPSRV     NO                 16383PB    5:33:40    5:33:40
     XCFAS       NO                 16383PB    5:33:40    5:33:40
     GRS         NO                            5:36:25    5:36:25
     SMSPDSE     NO                            5:36:25    5:36:25
```

Figure 6-2 Display activity (DA) panel with SecLabel column

DSecLabel column

The device security label is passed to the table build exit for the PR, PUN, RDR, LI, SO, and NO panels.

The DSecLabel column is added to these panels as shown in Figure 6-3 on page 131.



```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF PRINTER DISPLAY  SC64                                LINE 1-4 (4)
COMMAND INPUT ===> _                                             SCROLL ===> CSR
NP   PRINTER  FSSName  FSSProc  FSATrace SysName  SysID JESN JESLevel DSecLabel
     PRT1     FSS382C  APS382C  NO       SC64     SC64  JES2 z/OS 1.5
     PRT2     PRINTWAY PRINTWA2 NO       SC64     SC64  JES2 z/OS 1.5
     PRT3     FSS382B  APS382B  NO       SC64     SC64  JES2 z/OS 1.5
     PRT4     IAZSFSS  WTRIAZF  NO       SC64     SC64  JES2 z/OS 1.5
```

Figure 6-3 Printer panel with DSecLabel column

Who command

When the SECLABEL class is active, SDSF adds the user's security label to the response to the **WHO** command.

6.2.5 Recommendations

- If you want to use SECLABELS to provide security for the SDSF displays, consider the following options:
  - For certain panels where every user needs access, protect that panel with a SECLABEL of SYSHIGH.

    For example, the MAS panel is protected by the profile ISFCMD.ODSP.MAS.jes2. You can add a SECLABEL to this profile by using the RALTER or RDEFINE commands.
  - Also, protect the System Requests panel with a SECLABEL of SYSHIGH.
- ULOG (user log) panel

  Control activation of the EMCS (extended MVS console services) console using a SECLABEL on the associated OPERCMDS profile (MVS.MCSOPER.consolename)
- SR (system requests) panel

  If you don't protect access to the panel with SYSHIGH, users might access the panel but see no rows (the rows are protected with SYSHIGH).

  If you use the SDSF server to process ISFPARMS, assign a SECLABEL of SYSLOW to the server, or whatever security label is appropriate to access PARMLIB.

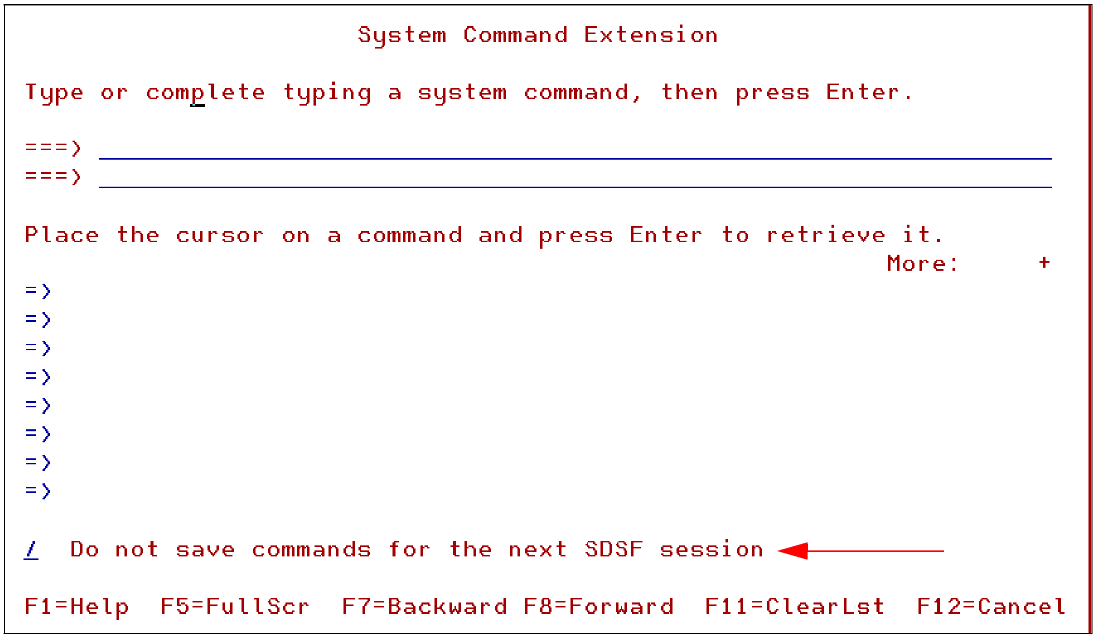
6.3 Control of saving system commands

SDSF adds an option to the System Command Extension pop-up, shown in Figure 6-4 on page 132, that controls the saving of system commands. If saving is turned off, all saved system commands are removed from the profile when the user exits SDSF. During the current session, however, they continue to be saved and displayed in the selection list on the pop-up.

The number of saved system commands entered through SDSF has been increased to twenty. The saving of system commands is supported as follows:

- If saving is turned off, all saved system commands are removed from the profile when the user exits SDSF.

- During the current session, however, they continue to be saved and displayed in the selection list on the pop-up.



```
                        System Command Extension

 Type or complete typing a system command, then press Enter.

 ===> ________________________________________________________________
 ===> ________________________________________________________________

 Place the cursor on a command and press Enter to retrieve it.
                                                          More:     +
 =>
 =>
 =>
 =>
 =>
 =>
 =>
 =>

 /  Do not save commands for the next SDSF session

 F1=Help  F5=FullScr  F7=Backward F8=Forward  F11=ClearLst  F12=Cancel
```

Figure 6-4 System Command Extension panel

Note: To access the system command extension panel, enter a “/” on the SDSF command input line.

6.4 Saving INPUT and ACTION commands

SDSF now saves the values for the **INPUT** and **ACTION** commands in the ISPF profile when SDSF is running as an ISPF dialog.

INPUT This command controls whether SYSIN data sets are displayed when you browse a job.

ACTION This command controls which WTORs are displayed at the bottom of the SYSLOG panel.

SDSF also adds the INPUT keyword to ISFPARMS, to allow an initial setting for INPUT to be made for a group of users.

When it is running under ISPF, SDSF uses the values from the previous session for INPUT and ACTION.

Note: You can query the current value with the **INPUT ?** or **ACTION ?** SDSF commands.

6.4.1 Migration actions

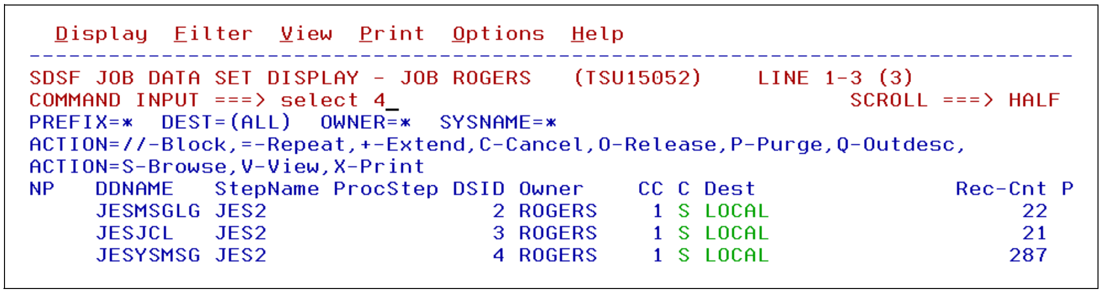
Specify a value of ON for the INPUT keyword in ISFPARMS if you want the initial value for the INPUT command to be *ON* for that group of users.

6.5 SELECT with DSID on JDS

The Job Data Set panel allows the user to display information about SYSOUT data sets for a selected job, started task, and TSO user. On the JDS (job data set) display panel, the **SELECT** command now accepts DSID as a parameter as shown in Figure 6-5.

SELECT is a fast-path filter that temporarily overrides any existing filters.

Users can quickly filter job data sets by DSID.



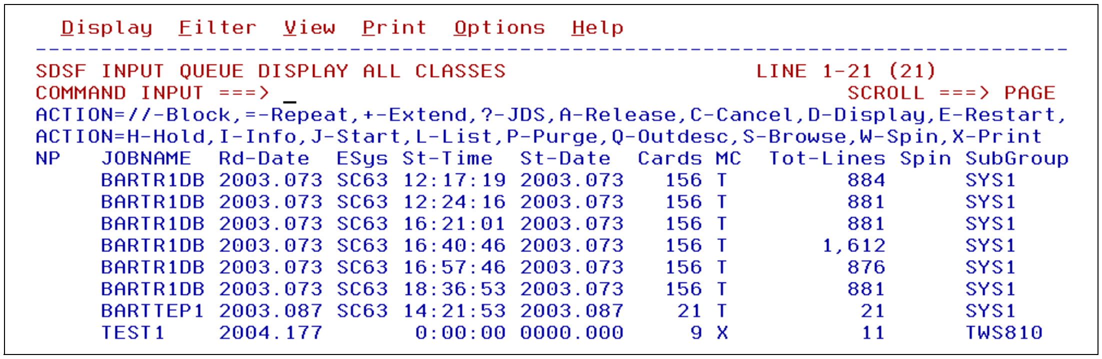
```
  Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF JOB DATA SET DISPLAY - JOB ROGERS   (TSU15052)    LINE 1-3 (3)
 COMMAND INPUT ===> select 4_                                  SCROLL ===> HALF
 PREFIX=*  DEST=(ALL)  OWNER=*  SYSNAME=*
 ACTION=//-Block,=-Repeat,+-Extend,C-Cancel,O-Release,P-Purge,Q-Outdesc,
 ACTION=S-Browse,V-View,X-Print
 NP   DDNAME   StepName ProcStep DSID Owner    CC C Dest              Rec-Cnt P
      JESMSGLG JES2                   2 ROGERS    1 S LOCAL                22
      JESJCL   JES2                   3 ROGERS    1 S LOCAL                21
      JESYSMSG JES2                   4 ROGERS    1 S LOCAL               287
```

Figure 6-5   SELECT=DSID shown on the JDS panel

6.6 Column and action character enhancements

SDSF adds a column for submitter group, SubGroup column, to the H, I, O, and ST panels as shown in Figure 6-6.

It is added to the end of the alternate field lists because access is delayed, but it can be arranged to a different location by the system programmer or the user. The subgroup is the submitter group in the user's RACF profile.



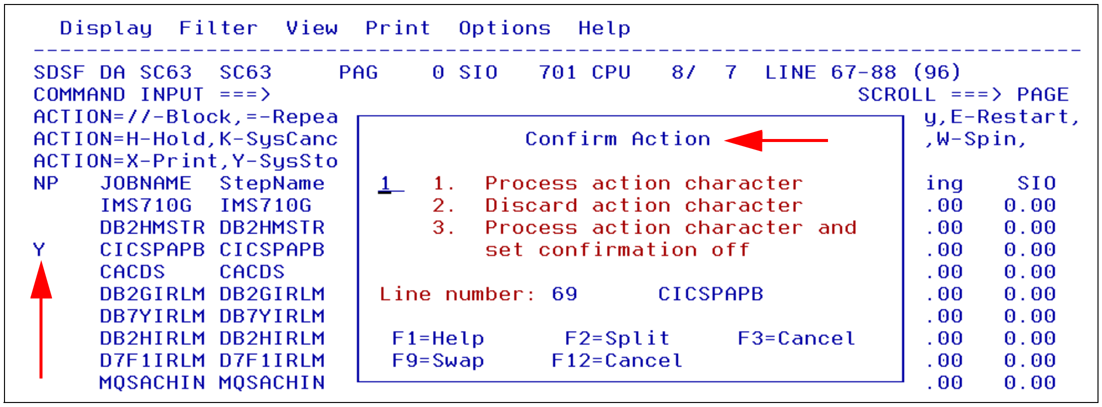
```
  Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF INPUT QUEUE DISPLAY ALL CLASSES                  LINE 1-21 (21)
 COMMAND INPUT ===> _                                          SCROLL ===> PAGE
 ACTION=//-Block,=-Repeat,+-Extend,?-JDS,A-Release,C-Cancel,D-Display,E-Restart,
 ACTION=H-Hold,I-Info,J-Start,L-List,P-Purge,Q-Outdesc,S-Browse,W-Spin,X-Print
 NP   JOBNAME  Rd-Date  ESys St-Time  St-Date  Cards MC  Tot-Lines Spin SubGroup
      BARTR1DB 2003.073 SC63 12:17:19 2003.073   156 T         884      SYS1
      BARTR1DB 2003.073 SC63 12:24:16 2003.073   156 T         881      SYS1
      BARTR1DB 2003.073 SC63 16:21:01 2003.073   156 T         881      SYS1
      BARTR1DB 2003.073 SC63 16:40:46 2003.073   156 T       1,612      SYS1
      BARTR1DB 2003.073 SC63 16:57:46 2003.073   156 T         876      SYS1
      BARTR1DB 2003.073 SC63 18:36:53 2003.073   156 T         881      SYS1
      BARTTEP1 2003.087 SC63 14:21:53 2003.087    21 T          21      SYS1
      TEST1    2004.177        0:00:00 0000.000    9 X          11      TWS810
```

Figure 6-6   Input queue display with new SubGroup field

Character enhancement on the DA panel

SDSF adds an action character, Y, to the DA panel that generates an MVS **P** (**STOP**) command as shown in Figure 6-7 on page 134.

This is useful for stopping started tasks. This action requires confirmation when confirmation is active. Confirmation can be controlled with the **SET CONFIRM** command.



```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF DA SC63  SC63     PAG    0 SIO    701 CPU    8/  7  LINE 67-88 (96)
COMMAND INPUT ===>                                          SCROLL ===> PAGE
ACTION=//-Block,=-Repea                                        y,E-Restart,
ACTION=H-Hold,K-SysCanc         Confirm Action                 ,W-Spin,
ACTION=X-Print,Y-SysSto
NP    JOBNAME  StepName   1_  1.  Process action character     ing    SIO
      IMS710G  IMS710G         2.  Discard action character    .00   0.00
      DB2HMSTR DB2HMSTR        3.  Process action character and .00   0.00
Y     CICSPAPB CICSPAPB            set confirmation off          .00   0.00
      CACDS    CACDS                                            .00   0.00
      DB2GIRLM DB2GIRLM   Line number: 69      CICSPAPB          .00   0.00
      DB7YIRLM DB7YIRLM                                         .00   0.00
      DB2HIRLM DB2HIRLM    F1=Help      F2=Split      F3=Cancel .00   0.00
      D7F1IRLM D7F1IRLM    F9=Swap     F12=Cancel               .00   0.00
      MQSACHIN MQSACHIN                                         .00   0.00
```

Figure 6-7   New Y action character on the DA panel

ENC panel enhancement

The reset action character on the ENC panel (which includes the quiesce function) now requires confirmation when confirmation is active.

6.7 Simplification of command authorization

SDSF enhances the AUTH keyword in ISFPARMS to make it possible to authorize a group of users to:

- All authorized SDSF commands
- All *operator* commands
  Operator commands are those in group 2 of SDSF's sample ISFPARMS.
- All *end user* commands
  End user commands are those in group 3 of SDSF's sample ISFPARMS.

With this enhancement, system programmers can define their groups so they no longer need to update ISFPARMS when new authorized commands are added to SDSF.

The AUTH keyword now accepts these values:

ALL The group is authorized to all SDSF commands.

ALLOPER The group is authorized to all operator commands.

ALLUSER The group is authorized to all end user commands.

6.7.1 Migration actions

Review the AUTH keyword for each group in ISFPARMS and determine if the new keywords (ALL, ALLOPER, and ALLUSER) can be substituted for existing lists of commands. If so, update AUTH with the appropriate keyword.

6.8 Server enhancement

SDSF adds a keyword to the COMM statement in ISFPARMS that causes SDSF to use predefined Web Sphere MQ queues, rather than creating new queues during initialization with Web Sphere MQ DEFINE commands. SDSF uses the queues to provide sysplex support for browse and device displays. If you choose to have SDSF use the predefined queues, the

SDSF server does not need administrative authority to the Web Sphere MQ DEFINE command. However, you then have to define the queues manually.

6.8.1 Migration actions

Add QDEFINE(NO) to the COMM keyword in ISFPARMS if you don’t want the SDSF server to issue DEFINE commands to define queues.

Define the model queue and the client request queue with DEFINE commands. This is optional, but required if you specify QDEFINE(NO).

6.8.2 Migration actions

If you have customized field lists in ISFPARMS, add the new column to the field lists for the H, I, O, and ST panels.

Authorize users to the Y action character with ISFPARMS or SAF.

Installations with their own field lists in ISFPARMS may want to remove PGN and DOMAIN from the list for the DA panel, and PGN from the list for the ENC panel. These columns are never displayed since they are not shown when the system is running in goal mode and, with z/OS V1R5, the system is always in goal mode.

6.9 Controlling the format of CPU information

SDSF adds an initialization option to control the format of CPU busy information on the title line of the DA panel.

Installations can choose to display both the MVS and LPAR views (the default, when in LPAR mode) or only the MVS view.

The option is specified with a new group-level keyword in ISFPARMS, CPUFMT.

6.9.1 Migration actions

Specify a value of SHORT for the CPUFMT keyword in ISFPARMS if you want the DA title line to show only the MVS view of CPU.

Note: CPUFMT=LONG is the default.

6.10 Improved folding of mixed case text

SDSF implements a change to the CTITLE parameter in ISFPARMS.

Rather than controlling only the folding of column titles, it now also controls the folding of text on the primary option menu, print pop-ups, and other mixed-case text displayed by SDSF.

This is useful when SDSF is used with codepages that do not include lowercase characters

6.10.1 Migration actions

Review the CTITLE keyword in ISFPARMS and specify the appropriate value - ASIS or UPPER.

Note: CTITLE=ASIS is the default.

6.11 Removal of SNAP dump

Beginning with this release, SDSF uses only SDUMP when an abend occurs. The SNAP dump is no longer taken. As a result, the SDSFDUMP card is no longer used.

6.12 Protection of RMF services

RMF adds support for protecting the services that SDSF uses to gather data for the DA panel.

RMF protects its services with the SAF profile ERBSDS.MON2DATA or a generic-style profile, ERBSDS.*, both in the FACILITY class.

- If either profile exists, SDSF users must have READ access to it to access the data.
- If neither profile exists, then SDSF users can access the data.

If either of the new profiles exists, users who do not have access to it will see an error message on the DA panel.

Note: This change applies only when the DA panel is using RMF as the source of its values.

6.12.1 Migration actions

Protect use of the RMF services using the new resources in the FACILITY class. Authorize SDSF users to the resources, so that they may see RMF data on the DA panel.

7

Infoprint Server

This chapter describes the following functional enhancements in the z/OS Infoprint Server in z/OS Version 1 Release 5:

- Infoprint Server overview
- Summary of the new enhancements
- Changes to the ISPF primary panel
- Changes to the configuration file, AOPD.CONF
- Changes to the security environment

7.1 Infoprint Server overview

Infoprint Server is an optional feature of z/OS that uses z/OS UNIX System Services. This feature is the basis for a total print serving solution for the z/OS environment. It lets you consolidate your print workload from many servers onto a central z/OS print server.

Infoprint Server delivers improved efficiency and lower overall printing cost with the flexibility for high-volume, high-speed printing from anywhere in the network. With Infoprint Server, you can reduce the overall cost of printing while improving manageability, data retrievability, and usability.

As shown in Figure 7-1, Infoprint Server provides the following capabilities for:

- Businesses that print statements, such as banking statements, invoices, and bills of materials, to print both on LAN-attached printers and on higher-volume, host-attached printers.
- Workstation users that need to print documents, such as memos, e-mail, Web documents, and manuals on LAN-attached Printer Control Language (PCL) and PostScript printers.
- Business applications, such as payroll, accounting, and inventory-control applications, that need to print on the same system where the data resides.
- z/OS batch applications and z/OS UNIX System Services applications that need to print on host-attached printers.

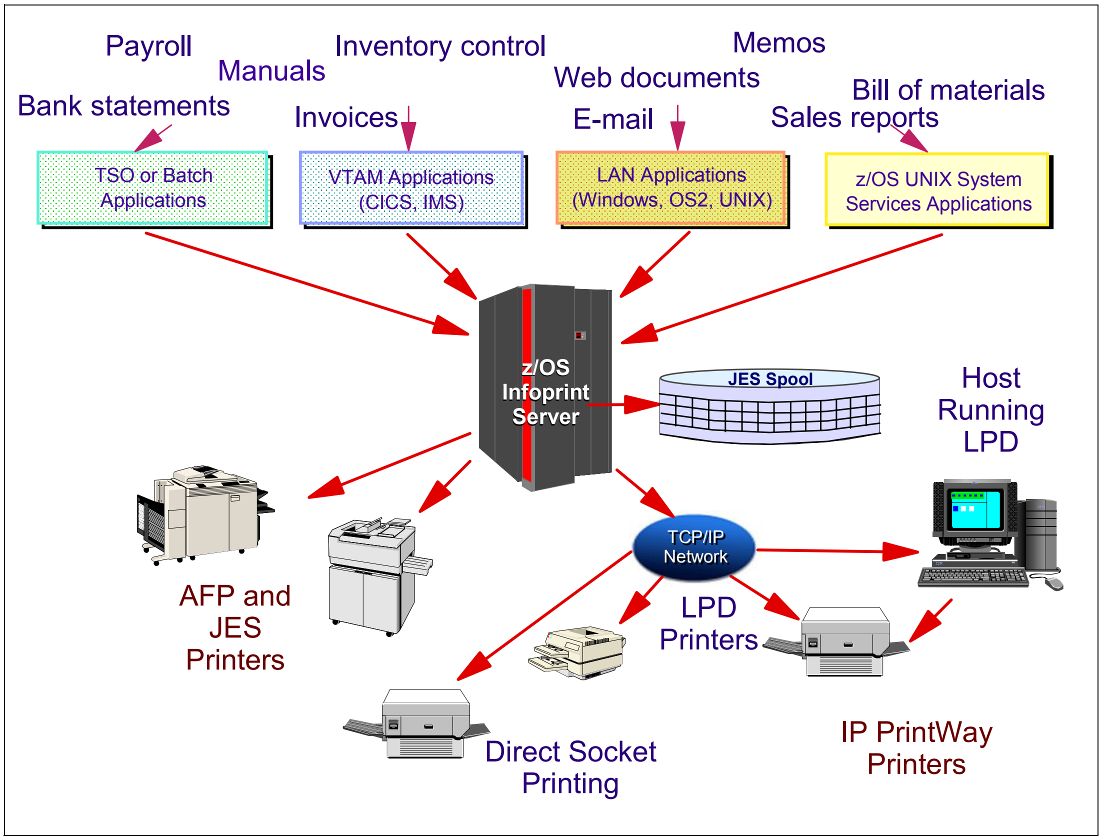


Figure 7-1 z/OS Infoprint Server overview

7.2 Infoprint Server enhancements with z/OS V1R5

The following enhancements have been introduced with z/OS V1R5:

- Miscellaneous enhancements and modifications:
  - Infoprint Server ISPF Printer Inventory Manager panel - See “Infoprint Server primary ISPF panel” on page 139.
  - Sample configuration file AOPD.CONF enhancements - See“Sample configuration file” on page 140.
  - Security enhancements - See “Security with PRINTSRV class” on page 141.
- Infoprint Central:

  Infoprint Central lets help desk operators and other authorized users or job submitters work with print jobs, printers, and NetSpool logical units (LUs); display printer definitions; and check system status. Infoprint Central is a Web-based print management system. See “Infoprint Central” on page 143.
- Security enhancements:
  - Read-only access to the Printer Inventory - See “Protect resources” on page 141.
  - Control access to Infoprint Central functions.
  - New profiles in RACF PRINTSRV class - See“Security with PRINTSRV class” on page 141.
- IP PrintWay extended mode:

  Another significant enhancement to Infoprint Server in z/OS V1R5 is the re-architecting of the offering to use the Sysout Application Programming Interface (SAPI) to access jobs and job information from the JES Spool rather than the JES Functional Subsystem Interface (FSI) connected to one or more Functional Subsystem Applications (FSAs).

  IP PrintWay extended mode uses the SAPI interface to access jobs and job information on the spool. This change gives Infoprint Server the ability to be more flexible in the way it accesses and processes print jobs, and improves scalability and reliability for large distributed print environments. See “IP Printway extended mode” on page 150.
- Common Message Log - See “Common message log” on page 177.
- New daemons - See “New daemons with z/OS V1R5” on page 140.
- IPP Server now supports Java™ 1.4.
- IP PrintWay - miscellaneous enhancements:
  - SMF Type 6 enhancements
  - IP PrintWay support for large files > 2 GB
  - IPv6 support
- NetSpool enhancements.
- E-mail enhancements - See“E-mail printers” on page 179.

7.2.1 Infoprint Server primary ISPF panel

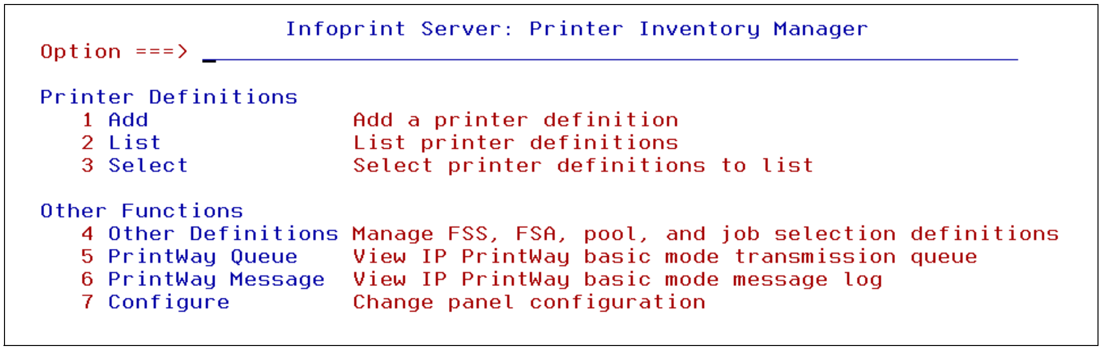
On the Infoprint Server: Printer Inventory Manager panel, there are several changes in z/OS V1R5, as indicated in Figure 7-2 on page 140.

New options changes

The changes to this panel for z/OS V1R5 are as follows:

Option 4 Job selection definitions (rules) for IP PrintWay extended mode are new.

Options 5 and 6 These options now say they only work for basic mode.



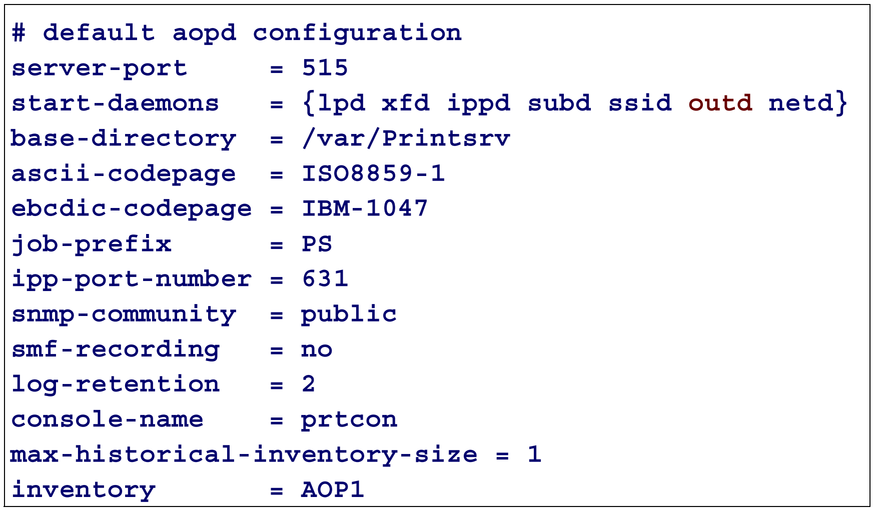
```
                    Infoprint Server: Printer Inventory Manager
Option ===> _____________________________________________________________

Printer Definitions
   1 Add                 Add a printer definition
   2 List                List printer definitions
   3 Select              Select printer definitions to list

Other Functions
   4 Other Definitions Manage FSS, FSA, pool, and job selection definitions
   5 PrintWay Queue      View IP PrintWay basic mode transmission queue
   6 PrintWay Message    View IP PrintWay basic mode message log
   7 Configure           Change panel configuration
```

Figure 7-2 Infoprint Server primary option panel

7.2.2 Sample configuration file

The Infoprint Server configuration file, shown in Figure 7-3, contains attributes that customize Infoprint Server. The Infoprint Server configuration file, aopd.conf, lets you customize the Printer Inventory Manager and other components of Infoprint Server.

Note: This file is optional. If the configuration file does not exist or if an attribute in the configuration file is omitted, default values are used.



```
# default aopd configuration
server-port      = 515
start-daemons    = {lpd xfd ippd subd ssid outd netd}
base-directory   = /var/Printsrv
ascii-codepage   = ISO8859-1
ebcdic-codepage = IBM-1047
job-prefix       = PS
ipp-port-number = 631
snmp-community   = public
smf-recording    = no
log-retention    = 2
console-name     = prtcon
max-historical-inventory-size = 1
inventory        = AOP1
```

Figure 7-3 Infoprint Server configuration file AOPD.CONF

New daemons with z/OS V1R5

The new functions added in this release are started by specifying new daemons in the configuration file.

netd Starts the NetSpool daemon, aopnetd

outd Starts the IP PrintWay extended mode daemons, aopoutd and aopwsmd

When you start IP PrintWay again, any IP PrintWay printers that were stopped return to the started state, and any printers that were redirected to other printers are no longer redirected.

ssid Starts the Infoprint Central daemon, aopssid

Important: If you add a value to the start-daemons = while Infoprint Server is running, restart Infoprint Server to start the new daemons. You do not need to first stop Infoprint Server daemons that are running. Also, restart the z/OS HTTP Server if you run Infoprint Central to pick up the change.

See “Infoprint Server daemons and tasks status” on page 147 to check the status of the daemons defined to Infoprint Server using Infoprint Central.

7.3 Security with PRINTSRV class

In z/OS V1R5, the AOP.ADMINISTRATOR profile in the PRINTSRV class replaces the AOPADMIN profile in the FACILITY class.

Note: After you define the AOP.ADMINISTRATOR profile, Infoprint Server no longer checks the AOPADMIN profile.

Consider giving those users and groups that currently have READ (or higher) access to the AOPADMIN profile UPDATE access to the AOP.ADMINISTRATOR profile.

- READ access to the AOPADMIN profile is equivalent to UPDATE access to the AOP.ADMINISTRATOR profile.
- After all systems that share the RACF database have been migrated to z/OS V1R5, you can delete the AOPADMIN profile from the FACILITY class.

7.3.1 PRINTSRV class protection

The following Infoprint Server resources can be protected by the PRINTSRV class:

- The Printer Inventory - See “Controlling access to Printer Inventory” on page 142.
- Infoprint Central security - See “Infoprint Central security” on page 146.
- The common message log.
- Operator commands and daemons.
- Infoprint Server daemons.

If you define any profiles in the PRINTSRV class, you must activate the PRINTSRV class in RACF. In addition, to improve performance, you should copy profiles in the PRINTSRV class into virtual storage and refresh the PRINTSRV class after you change any profiles or permit new users to the profiles to make the changes effective, as follows:

```
SETROPTS CLASSACT(PRINTSRV)
SETROPTS RACLIST(PRINTSRV)
```

Protect resources

The following profiles can protect Infoprint Server resources:

```
RDEFINE PRINTSRV (AOP.ADMINISTRATOR) UACC(READ)
SETROPTS RACLIST(PRINTSRV) REFRESH
```

The UACC is used as follows:

READ READ access lets users view the Printer Inventory using ISPF panels, Infoprint Central, or the Printer Inventory Definition Utility (PIDU), as shown in

Figure 7-4. Users do not need READ access simply to list names of printer definitions with the **lpstat** command or with Infoprint Port Monitor.

UPDATE UPDATE access lets users update the Printer Inventory using ISPF panels or PIDU, and lets users view the Printer Inventory using Infoprint Central, as shown in Figure 7-4.

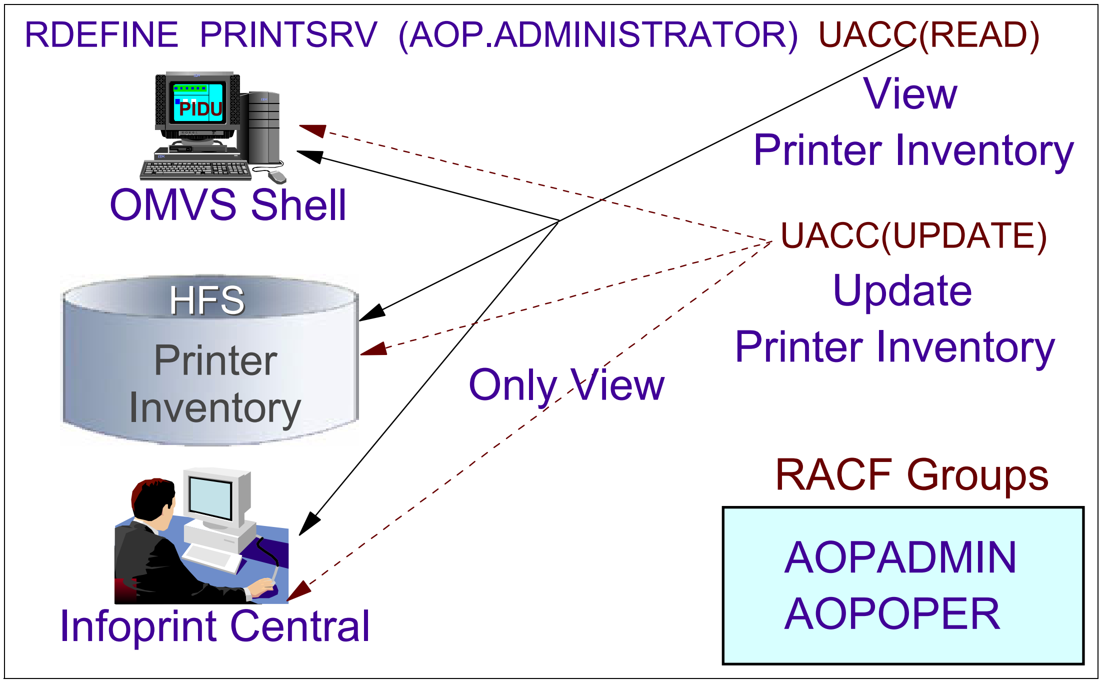


Figure 7-4 Giving users access to protected resources

You may have defined groups to RACF for Infoprint Server administrators and operators using the Infoprint Server suggested names, shown in Figure 7-4, as:

AOPADMIN The AOPADMIN group is for Infoprint Server administrators, who can view and update the Printer Inventory and view all messages in the common message log.

AOPOPER The AOPOPER group is for Infoprint Server operators, who can start and stop Infoprint Server daemons.

You can choose any names for these groups.

7.3.2 Controlling access to Printer Inventory

Define the resource profile AOP.ADMINISTRATOR with universal access of NONE, as follows:

```
RDEFINE PRINTSRV (AOP.ADMINISTRATOR) UACC(NONE)
```

Access for users

Give users who need to view the Printer Inventory and the common message log, using either Infoprint Server ISPF panels or Infoprint Central, READ access to the AOP.ADMINISTRATOR profile, as follows:

```
PERMIT AOP.ADMINISTRATOR CLASS(PRINTSRV) ACCESS(READ) ID(userid or group)
```

Access for administrators

Give administrators and other authorized personnel UPDATE access to allow updating of the Printer Inventory and viewing of Infoprint Central, as follows:

```
PERMIT AOP.ADMINISTRATOR CLASS(PRINTSRV) ACCESS(UPDATE) ID(AOPADMIN)
SETROPTS RACLIST(PRINTSRV) REFRESH
```

7.4 Infoprint Central

Infoprint Central is a Web-based print management system primarily for help desk operators. However, other authorized users or job submitters can also use it. Infoprint Central works with IP PrintWay extended mode. With Infoprint Central, you can:

- Work with print jobs
- Work with printers
- Work with NetSpool logical units (LUs)
- Display printer definitions
- Check system status

7.4.1 Starting Infoprint Central

To use Infoprint Central, add the *ssid* value to any existing values in start-daemons= in the configuration file, shown in Figure 7-3 on page 140. All daemons specified are started when you use the **aopstart** command or AOPSTART JCL procedure to start the Infoprint Server. This starts the **aopssid** daemon.

Infoprint Central requires the z/OS HTTP Server and a Web browser. No applications other than a Web browser need to be installed on a user's workstation.

Customizing an HTTP server

For improved performance, and because you must customize the HTTP Server for Infoprint Central, start a separate HTTP Server to be used exclusively by Infoprint Central.

Because the HTTP Server can display Infoprint Central Web pages only for Infoprint Server running on the same z/OS system as the HTTP Server, start an HTTP Server on each z/OS system where Infoprint Server is running.

> **Tip:** To encrypt and decrypt information that passes between the user's browser and the HTTP Server, you can customize the HTTP Server to use the Secure Sockets Layer (SSL) protocol.

z/OS HTTP Server configuration file

The z/OS HTTP Server configuration file, httpd.conf, contains directives that customize the HTTP Server. You must add directives so that the HTTP Server can display Infoprint Central Web pages. In addition, you can add directives to protect access to Infoprint Central Web pages and to tune the HTTP Server.

Add the Pass directives, in the order shown, before the ServerInit and Service directives and before the generic Pass directive, Pass /*. File /usr/lpp/Printsrv/samples/httpd.conf.updates contains the required directives, as follows:

```
Pass /Infoprint/Scripts/*     /usr/lpp/Printsrv/InfoprintCentral/Scripts/*
Pass /Infoprint/Images/*      /usr/lpp/Printsrv/InfoprintCentral/Images/*
Pass /Infoprint/help/En_US/*  /usr/lpp/Printsrv/InfoprintCentral/help/En_US/*
Pass /Infoprint/En_US/*.html  /usr/lpp/Printsrv/InfoprintCentral/html/En_US/*.html
```

Web browsers

The following browsers are supported:

- Microsoft® Internet Explorer 5.5 (or a higher level)
- Netscape Navigator 7.0 (or a higher level)
- IBM Home Page Reader 4.0 (or a higher level) for users with disabilities

Protecting Infoprint Central Web pages

The z/OS HTTP Server lets you use any of several methods to protect Web pages. You must use a protection method that provides Infoprint Central with a unique z/OS user ID and password that has been authenticated by RACF or a similar security system.

7.4.2 Accessing Infoprint Central from a browser

To access Infoprint Central from a workstation, as shown in Figure 7-5, enter the following URL:

```
http://wtsc65oe.itso.ibm.com/Infoprint/En_US/IPS.html
```

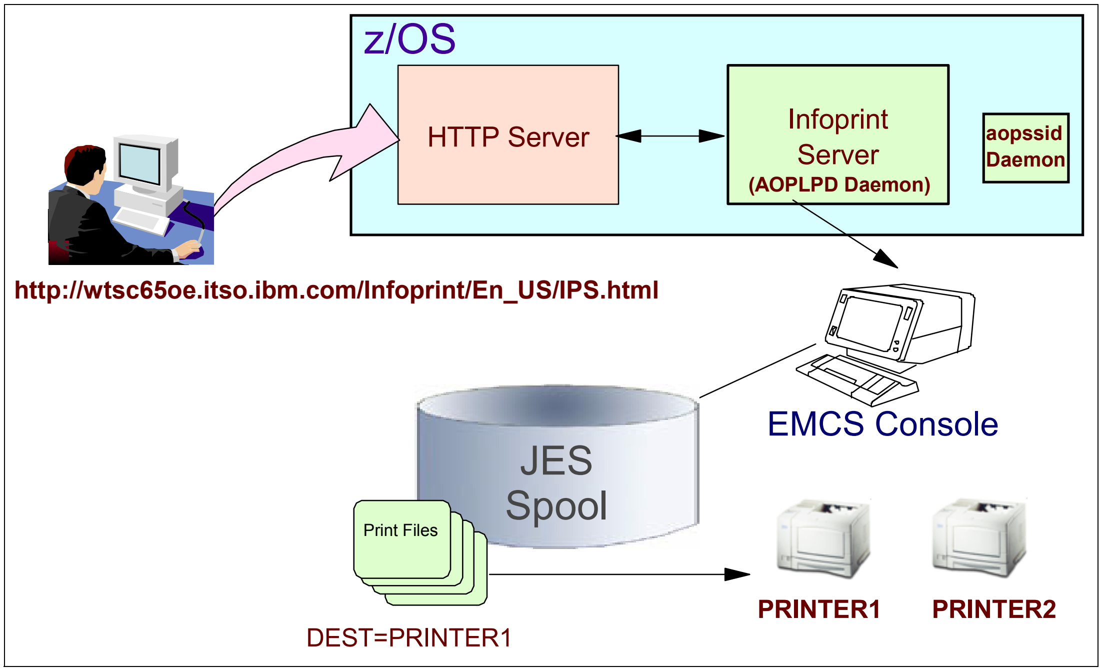


Figure 7-5 Infoprint Central administrator communicating with Infoprint Server

EMCS console

The EMCS console, shown in Figure 7-5, is used to issue JES and MVS commands and receive responses to the commands issued by Infoprint Server on behalf of the administrator or operator using the browser. The console name is defined in the aopd.conf file and a default name of AOP1 is used if no name is specified.

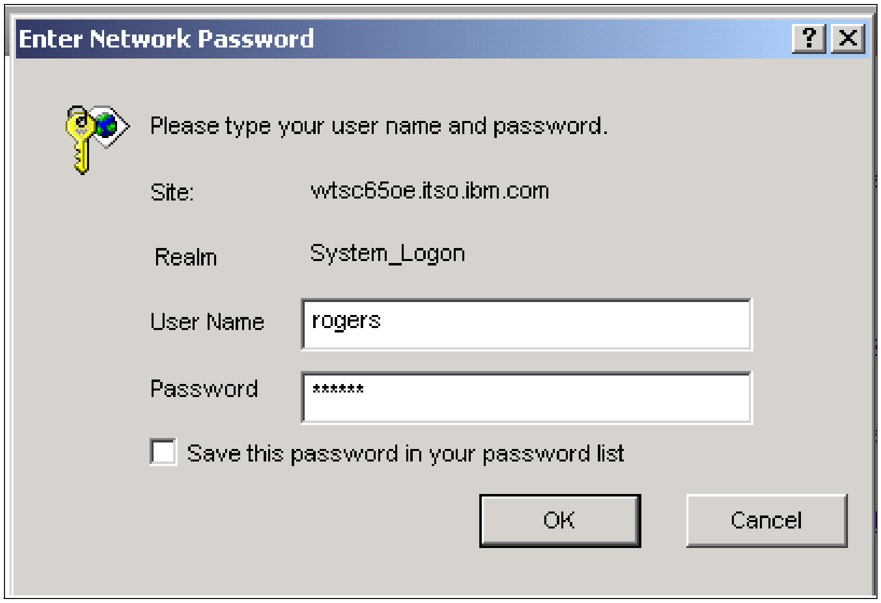
User authentication

Two requirements for access to Infoprint Central are, as shown in Figure 7-6:

- RACF user ID
- Password

Unauthenticated users are not allowed access, but a user who needs access must be a z/OS UNIX user, while a TSO segment is not required. A minimum level of READ access to the Printer Inventory is required as follows:

```
PERMIT AOP.ADMINISTRATOR CLASS(PRINTSRV) ACC(READ) ID(userid or groupid)
```



Enter Network Password

Please type your user name and password.

Site: wtsc65oe.itso.ibm.com

Realm System_Logon

User Name rogers

Password ******

□ Save this password in your password list

Figure 7-6 Authentication of user ID and password

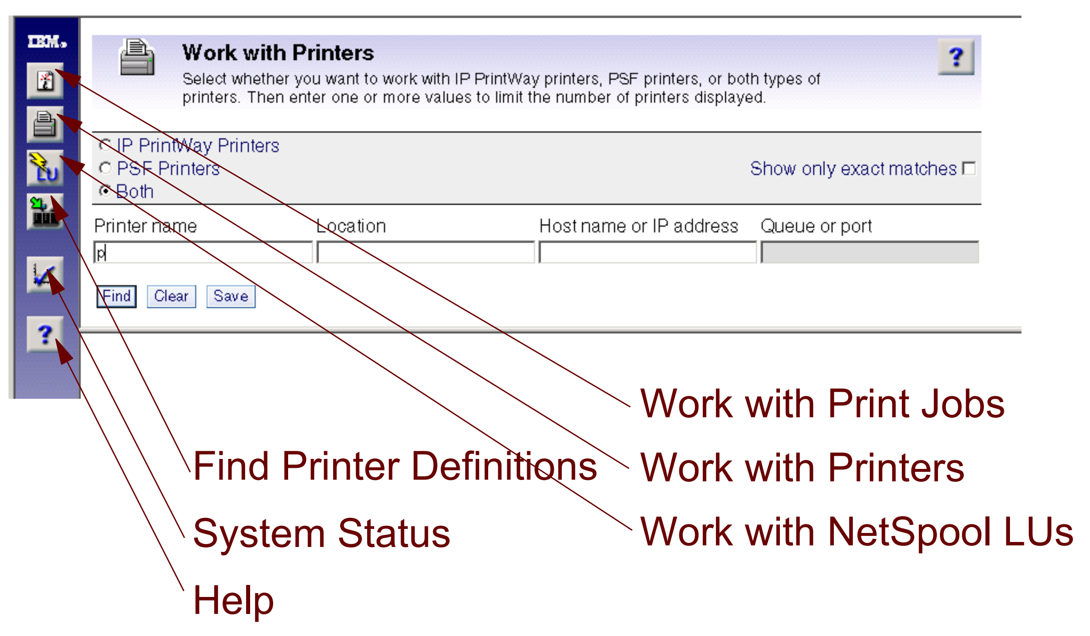
7.4.3 Infoprint Central initial panel

The first panel to be displayed following authentication is shown in Figure 7-7 on page 146.

Work with Printers is the first panel displayed when you access Infoprint Central from a Web browser. You can find and work with printers that are defined in the Printer Inventory, which includes:

- IBM AFP printers that PSF controls (called PSF printers)
- TCP/IP-attached printers to which IP PrintWay extended mode sends print jobs (called IP PrintWay printers)

> **Note:** You can find and work with IP PrintWay printers only when you run IP PrintWay extended mode.



Work with Printers

Select whether you want to work with IP PrintWay printers, PSF printers, or both types of printers. Then enter one or more values to limit the number of printers displayed.

○ IP PrintWay Printers
○ PSF Printers
● Both

Show only exact matches □

Printer name	Location	Host name or IP address	Queue or port
p			

Figure 7-7 Infoprint Central primary panel

7.5 Infoprint Central security

You must define users of Infoprint Central to RACF as z/OS UNIX users. While Infoprint Central is a Web-based print management system primarily for help desk operators, in addition, authorized job submitters can also use it.

Using the PRINTSRV class, you can protect who can use Infoprint Central and which printers a user can control. The types of printers that can be controlled are:

- IP PrintWay extended mode printers
- PSF printers

The PRINTSRV class can be used to specify which jobs a user can see and what actions the user can perform on jobs and on printers. Administrators, operators, or general users can be authorized to:

- Start and stop printers
- Change and set job selection rules
- Change and set NetSpool LUs
- Display status of Infoprint Server daemons

7.5.1 User access to Infoprint Central

To make sure proper access is given to general users, operators, and administrators when accessing Infoprint Central, consider the following areas of access control to the functions:

- Setting up security for Infoprint Server daemons. See “Infoprint Server daemons and tasks status” on page 147.
- Setting up security for printers.
- Setting up security for IP PrintWay job selection rules.

- Setting up security for print jobs.
- Setting up security for printer and printer pool definitions.
- Setting up security for NetSpool logical units (LUs).

7.5.2 Infoprint Server daemons and tasks status

Using Infoprint Central, you can see the status of all Infoprint Server daemons and tasks. This can help you determine the cause of a printing problem. Infoprint Central lets users display the status of Infoprint Server daemons to see whether they are started. You can define profile AOP.DAEMON to restrict who can display daemons. If you do not define profile AOP.DAEMON, any Infoprint Central user can display the status of daemons. Infoprint Central does not let users start and stop daemons.

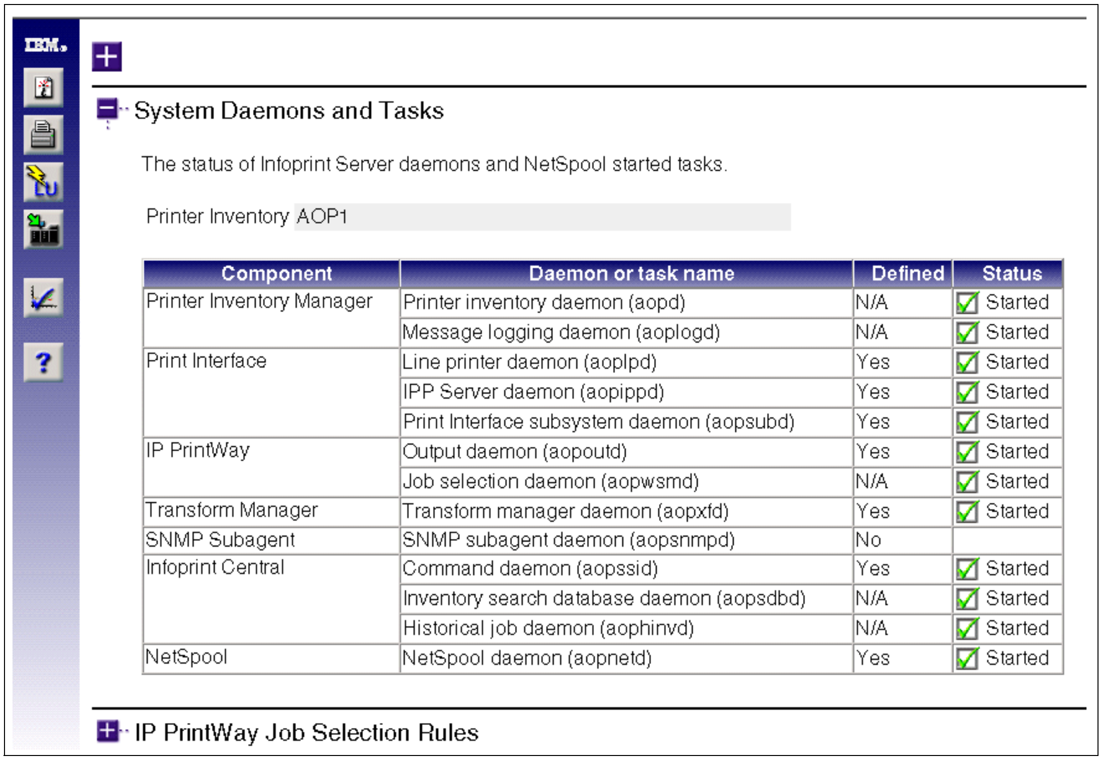
AOP.DAEMON profile

Who can display daemons is determined by the following profiles defined in the PRINTSRV class:

```
RDEFINE PRINTSRV (AOP.DAEMON) UACC(NONE)
```

To give users access to the AOP.DAEMON resource profile, issue the following commands:

```
PERMIT AOP.DAEMON CLASS(PRINTSRV) ACCESS(READ) ID(userid or groupid)
SETROPTS RACLIST(PRINTSRV) REFRESH
```



System Daemons and Tasks

The status of Infoprint Server daemons and NetSpool started tasks.

Printer Inventory AOP1

Component	Daemon or task name	Defined	Status
Printer Inventory Manager	Printer inventory daemon (aopd)	N/A	✓ Started
	Message logging daemon (aoplogd)	N/A	✓ Started
Print Interface	Line printer daemon (aoplpd)	Yes	✓ Started
	IPP Server daemon (aopippd)	Yes	✓ Started
	Print Interface subsystem daemon (aopsubd)	Yes	✓ Started
IP PrintWay	Output daemon (aopoutd)	Yes	✓ Started
	Job selection daemon (aopwsmd)	N/A	✓ Started
Transform Manager	Transform manager daemon (aopxfd)	Yes	✓ Started
SNMP Subagent	SNMP subagent daemon (aopsnmpd)	No	
Infoprint Central	Command daemon (aopssid)	Yes	✓ Started
	Inventory search database daemon (aopsdbd)	N/A	✓ Started
	Historical job daemon (aophinvd)	N/A	✓ Started
NetSpool	NetSpool daemon (aopnetd)	Yes	✓ Started

IP PrintWay Job Selection Rules

Figure 7-8 Displaying Infoprint Server daemons and tasks status

You can also start and stop IP PrintWay job selection rules to change which print jobs IP PrintWay processes.

7.6 IP PrintWay enhancements

The IP PrintWay component of Infoprint Server transmits output data sets from the JES spool to remote printers, print servers, and to e-mail destinations. You can run either of two modes, as follows:

- **IP PrintWay basic mode**: IP PrintWay basic mode, the original mode of operation, uses the z/OS Functional Subsystem Interface (FSI) to obtain output data sets from the JES spool, as shown in Figure 7-13 on page 153.
- **IP PrintWay extended mode**: IP PrintWay extended mode, new in z/OS V1R5, uses the z/OS Sysout Application Programming Interface (SAPI) to obtain output data sets from the JES spool. It provides better performance, improved usability, and more function than IP PrintWay basic mode.

Note: IP PrintWay basic mode and IP PrintWay extended mode use the same printer definitions in the Printer Inventory. The two modes can also be operational at the same time.

7.6.1 IP PrintWay protocols

The IP PrintWay printing protocols, illustrated in Figure 7-9 on page 149, are defined in an IP PrintWay printer definition panel for each printer. Figure 7-9 shows the protocols using the IP PrintWay basic mode. These protocols are shown in an IP PrintWay extended mode in Figure 7-10 on page 149.

You can select one of the following transmission protocols that IP PrintWay can use to transmit output data sets from the JES spool to the target system:

- **LPR**: The LPR protocol is a TCP/IP protocol defined by RFC 1179. An LPD that adheres to RFC 1179 must be running in the remote printer or system.
- **Direct sockets:** The direct sockets printing protocol is a TCP/IP protocol in which data is transmitted directly to a designated port. The remote printer or print server must support direct sockets printing.
- **IPP:** Internet Printing Protocol (IPP) is a standard TCP/IP protocol for printing over the Internet. An IPP server must be running in the remote printer or system.
- **VTAM®:** Virtual Telecommunications Access Method (VTAM) is a protocol that only IP PrintWay basic mode supports. This support allows printing to any printer that is defined to VTAM as LU type 0, LU type 1, or LU type 3. Supported output data streams are SNA character string (SCS) and Data Stream Compatible/Data Stream Extended (DSC/DSE). The Coax Printer Support feature of Infoprint Server Transforms is required to print on VTAM-controlled printers. IP PrintWay extended mode does not support the VTAM protocol.
- **E-mail:** IP PrintWay can use the z/OS UNIX sendmail function to send print output to one or more e-mail addresses. IP PrintWay uses an e-mail protocol to send the output, which can be in any data format, as an e-mail attachment.

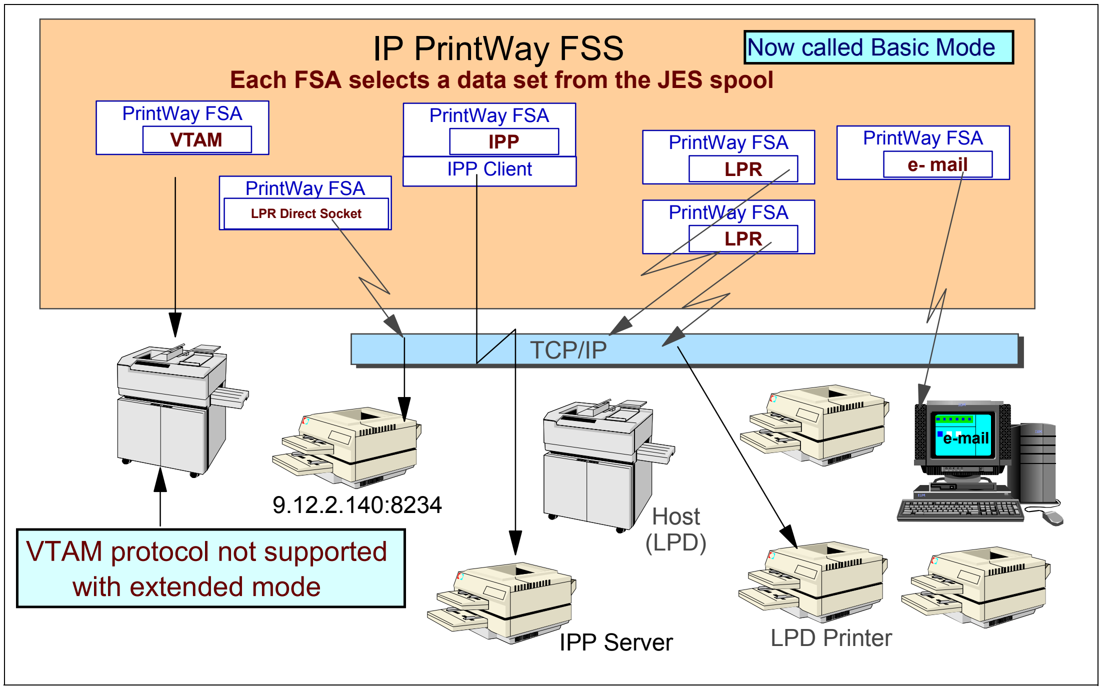


Figure 7-9   IP PrintWay printing protocols

Note: You may have multiple FSS address spaces with multiple FSAs driving printers.

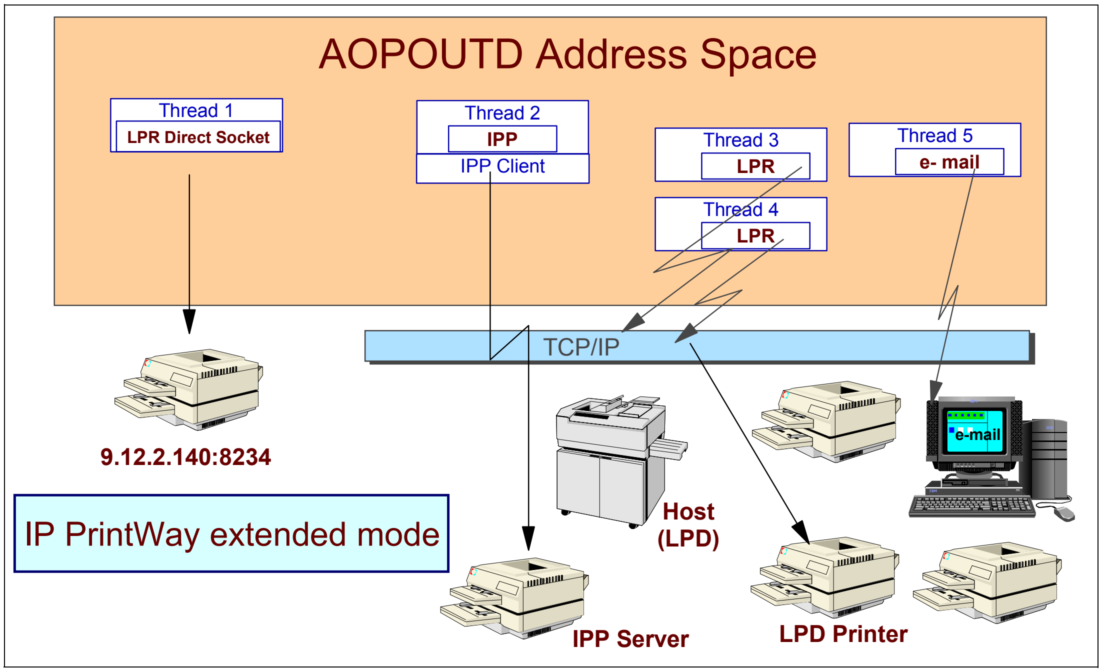


Figure 7-10   IP PrintWay protocols with extended mode

7.7 IP Printway extended mode

With z/OS V1R5, IP PrintWay can operate in a new mode called IP PrintWay extended mode. IP PrintWay extended mode uses the z/OS Sysout Application Programming Interface (SAPI) to obtain output data sets from the JES spool. This implementation results in better performance and improved usability. In addition, IP PrintWay extended mode provides new functions that help you manage printers and print jobs.

IP PrintWay extended mode and IP PrintWay basic mode use the same printer definitions in the Printer Inventory.

If you run IP PrintWay extended mode, use Infoprint Central because it lets you work with IP PrintWay extended mode printers. It also lets you work with output data sets that IP PrintWay extended mode is currently processing and displays the status of output data sets, including whether IP PrintWay has retained them.

7.7.1 IP PrintWay extended mode enhancements

Using IP PrintWay extended mode, you can take advantage of a number of functional enhancements and differences. With IP PrintWay extended mode you can:

- Manage all IP PrintWay printers and print jobs from the Web using Infoprint Central.
- Redirect print jobs from one printer to another.
- Delete and hold jobs that IP PrintWay extended mode is currently processing.
- Retain any number of print jobs, up to the limits of the size of the JES spool.
- Call data stream transforms and other filters directly without the resubmit for filtering option.
- Process print jobs larger than 2 gigabytes.
- Process and retain more data sets on the JES spool without running out of address space. This reduces the possibility of ending abnormally with an F02 abend code.
- Print output data sets in priority order.
- Print to printers that have IPv6 addresses.
- The host name (instead of the colon-hexadecimal address) must be used in the DEST=IP: JCL parameter, in Infoprint Server job attributes, and in printer definitions.
- Write the printer address for all protocol types in the SMF type 6 record.

Advantages over basic mode

In addition, IP PrintWay extended mode will provide the following advantages over basic mode:

- Better performance
- Increased reliability
- Improved usability
- New printer management functions
- New print job management functions
- All messages written to common message log
- Automatic detection of data format and verification that job will print on selected printer

7.7.2 Using IP PrintWay extended mode

Specifying start-daemons= in the configuration file, shown in Figure 7-3 on page 140, starts the daemons when you use the `aopstart` command or AOPSTART JCL procedure to start Infoprint Server. To run IP PrintWay extended mode, add the outd value to any existing values in this attribute to start the **aopoutd** and **aopwsmd** daemons, shown in Figure 7-11. Enclose all values in braces.

aopwsmd This daemon selects output from the JES spool using the SAPI SSI function call and the defined and active job selection rules, as defined in “Defining job selection rules” on page 155.

aopoutd The aopwsmd daemon starts a task in the aopoutd address space to select the output data set selected and process it to a printer.

Benefits of using SAPI SSI

When using IP PrintWay extended mode printers, the SAPI SSI call is used to select output data sets from the JES spool. The benefits of using SAPI versus the basic mode FSA job selection are the following:

- Better control of work selection criteria.
- Improved performance.
- No need to define FSA printers to JES.
- Currently, selected data sets may not be modified using SDSF because IP PrintWay has acquired and owns the selected data set.
- IP PrintWay basic mode allows a limited number of FSAs within an FSS.
- Eliminate duplicate work currently done in Print Interface and Printway.

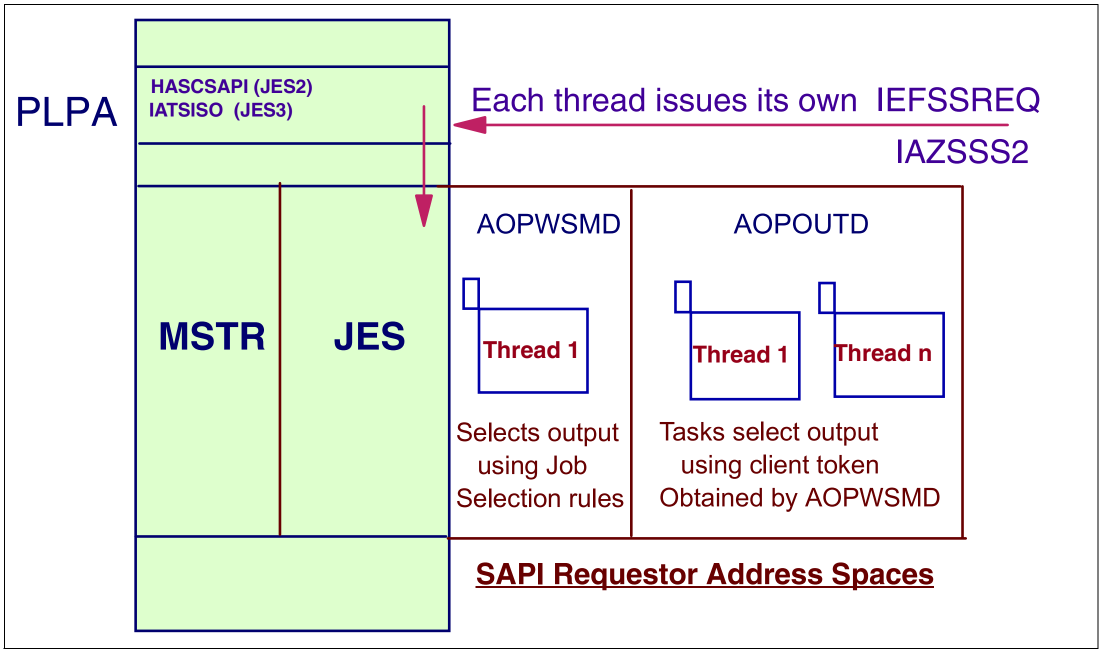

PLPA
HASCSAPI (JES2)
IATSISO (JES3)
Each thread issues its own IEFSSREQ
IAZSSS2
MSTR
JES
AOPWSMD
AOPOUTD
Thread 1
Thread 1
Thread n
Selects output using Job Selection rules
Tasks select output using client token Obtained by AOPWSMD
SAPI Requestor Address Spaces

Figure 7-11 IP PrintWay extended mode daemons using SAPI interface

SAPI SSI call

The SYSOUT Application Program Interface (SSI function code 79) allows JES to function as a server for the Infoprint Server daemons that need to access SYSOUT data sets residing on the JES spool. Use of the SAPI SSI call (IEFSSREQ), shown in Figure 7-12 on page 152, allows the data sets to be selected independently from the normal JES-provided functions (such as print or network). SAPI supports multiple, concurrent requests from the applications' address spaces. Each issuer of the IEFSSREQ macro is referred to as an *application thread* as shown in Figure 7-11 on page 151.

IAZSSS2 mapping macro

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

PUT/GET request

PUT/GET request processing, shown in Figure 7-12, 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. See “Defining job selection rules” on page 155.

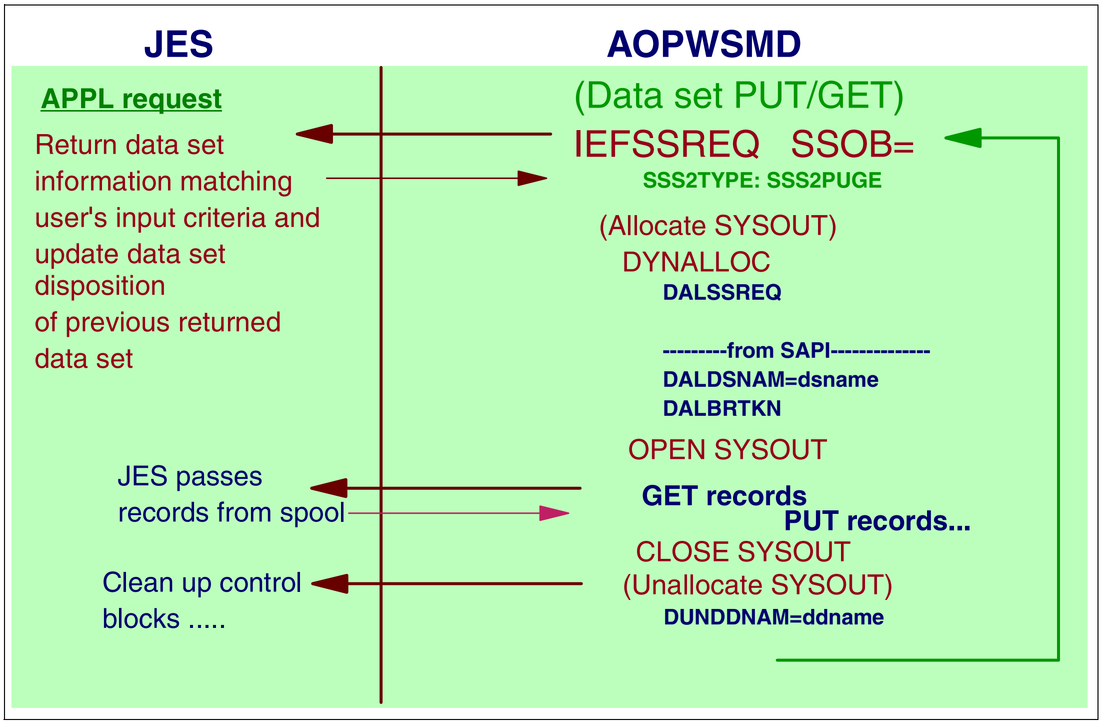


Figure 7-12 IP PrintWay extended mode data set selection

7.8 IP PrintWay basic mode

IP PrintWay basic mode is the name used for the original IP PrintWay mode of operation, to distinguish the original mode from the new IP PrintWay extended mode. You can continue to run IP PrintWay basic mode in z/OS V1R5. However, no enhancements have been made to IP PrintWay basic mode.

Attention: IBM will make enhancements in future releases only to IP PrintWay extended mode.

7.8.1 Using IP PrintWay basic mode

Both the FSA and JES use the FSIREQ GETDS parameter list to pass information to one another (see Figure 7-13). The FSA must initialize certain fields of the FSIREQ GETDS parameter list for each issuance of the GETDS request.

The GETDS request goes to JES to select a SYSOUT data set to process. The JES-supplied GETDS routine in the IP PrintWay FSS address space receives control when the FSA issues the FSIREQ GETDS macro. This routine communicates with the JES address space to process GETDS requests. The basic function of GETDS processing is to attempt to satisfy the GETDS request immediately by selecting a JES output data set and then despooling that data set to the FSA. JES uses its own data set selection criteria to select the appropriate data set.

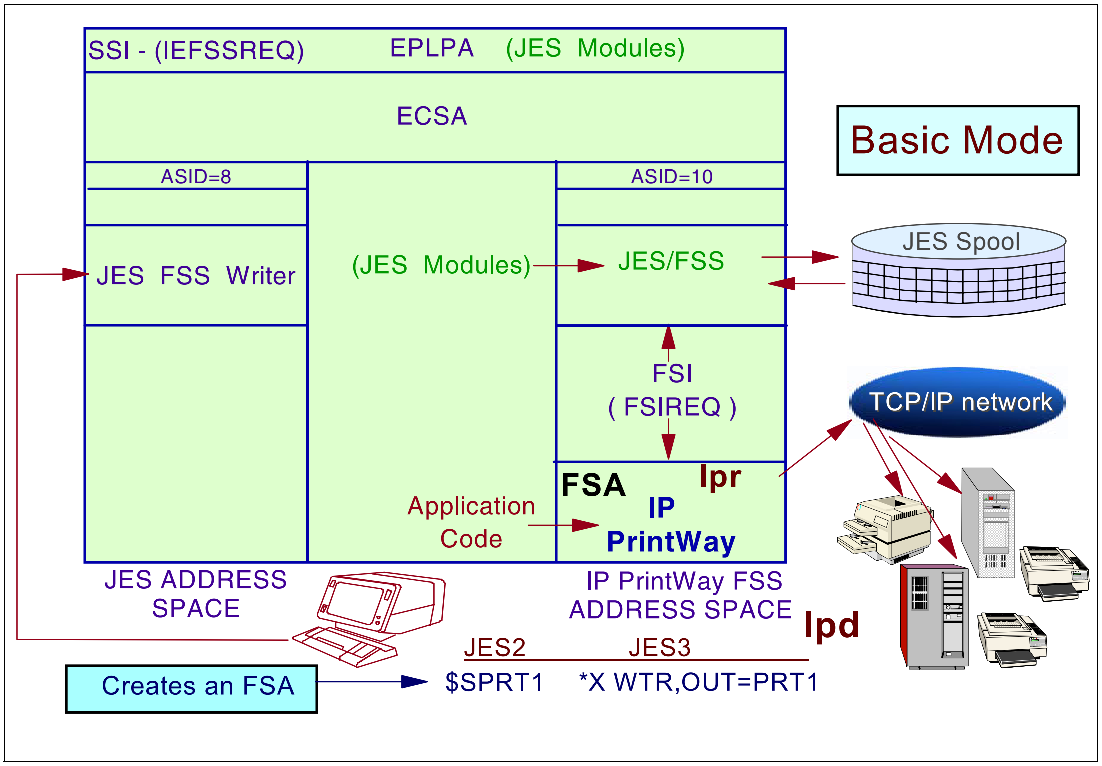


Figure 7-13 IP PrintWay basic mode using an FSS address space

Multiple FSAs

A functional subsystem application (FSA) is a collection of programs residing in the FSS address space that control one device. There can be multiple FSAs per FSS. It is recommended that each of the FSAs for the FSS be a separate task, as shown in Figure 7-14 on page 154. The FSA can be thought of as a logical subset of the FSS and is the lowest level of connection with JES. There is a JES function in the JES address space that starts and communicates with the FSA. Also, you can define multiple IP PrintWay FSS address spaces.

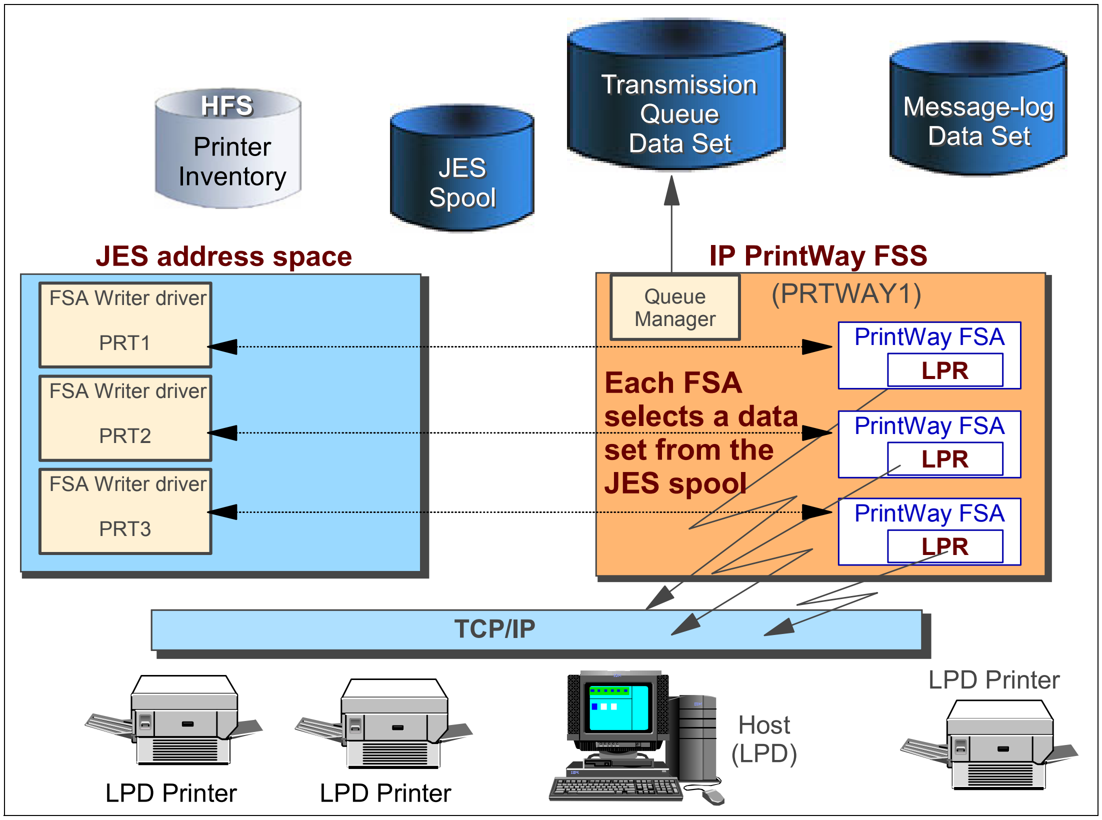


Figure 7-14 IP PrintWay basic mode data set selection

7.9 Working with output data sets

When you run IP PrintWay extended mode, operators can use Infoprint Central to work with output data sets that IP PrintWay is processing or has retained on the JES spool, as shown in Figure 7-15 on page 155. In addition, Infoprint Central lets you work with data sets that IP PrintWay has not yet selected for processing.

When you run IP PrintWay basic mode, the IP PrintWay transmission queue data set contains information about output data sets that IP PrintWay is processing or has retained on the JES spool, as shown in Figure 7-15 on page 155. Operators must use Infoprint Server ISPF panels to manage these data sets.

You can use JES and SDSF or (E)JES commands to do these actions on output data sets, depending on whether you run IP PrintWay extended mode or basic mode.

Note: There are differences between the way IP PrintWay extended mode and IP PrintWay basic mode work with output data sets.

7.9.1 IP PrintWay extended mode

If IP PrintWay extended mode is not currently processing a data set, you can delete, hold, and release it. You can also change its attributes, such as its priority, DEST, CLASS, and FORMS values. You can do these actions on data sets that IP PrintWay has:

- Not yet processed

- Waited to retry
- Retained on the JES spool after completion

You cannot do these actions on data sets that IP PrintWay extended mode is processing. Infoprint Central, however, lets you work with data sets that IP PrintWay extended mode is processing.

IP PrintWay basic mode

If IP PrintWay basic mode has not selected a data set for processing, you can delete, hold, and release it. You can also change its attributes, such as its DEST, CLASS, and FORMS values.

However, you cannot delete, hold, or release data sets that IP PrintWay is waiting to retry, or data sets that IP PrintWay has retained on the JES spool. The Infoprint Server ISPF panels, however, let you work with data sets that IP PrintWay basic mode is retrying or has retained on the JES spool by using the IP PrintWay transmission queue.

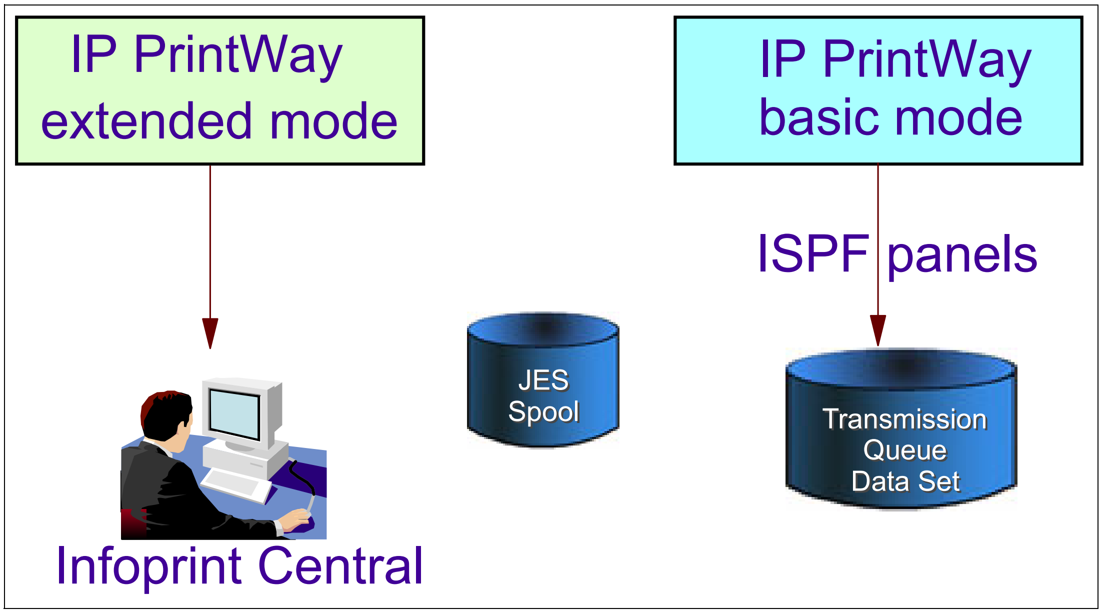


Figure 7-15 Working with output data sets

7.10 Defining job selection rules

When you use IP PrintWay extended mode, an administrator defines the job selection rules in the Printer Inventory to specify which print jobs the extended mode writers can select. You can use either the Infoprint Server ISPF panels or the Printer Inventory Definition Utility (PIDU) program to create, modify, and delete job selection rules in the Printer Inventory.

Note: To select a data set using the job selection rule, the attributes of the print job must match all of the values in a rule.

For Forms, Dest, and WRITER, you can specify the exact name or include one or more asterisks in any position in the name to represent zero or more unknown characters or one or more question marks in any position in the name to represent one unknown character.

7.10.1 Job Selection Rule panel

IP PrintWay extended mode uses the job selection rules to determine which output data sets to select from the JES spool for printing. Consider the following:

- You must create at least one job selection rule for IP PrintWay to select print jobs.
- You can create as many job selection rules as you need.
- The attributes of the print job must match all of the values in a rule to be selected.
- All attributes are optional. If you do not specify any attributes, IP PrintWay extended mode selects all output data sets.

From the Infoprint Server primary panel, shown in Figure 7-2 on page 140, select option 4 to see the FSA, FSS, Pool, and Job Selection Rule Management panel. From this panel, select option 10 to create a job selection rule.

The Job Selection Rule panel, shown in Figure 7-16 on page 157, is defined by filling in the desired fields and options from among the following:

Rule name The name of a job selection rule. This field is required.

Operator security profile The name of the RACF resource profile in the PRINTSRV class that controls who can work with this job selection rule using z/OS Infoprint Central for the Web. This field applies only to IP PrintWay extended mode. See “Operator security profile” on page 157.

DEST IP PrintWay selects only print jobs with this destination name.

CLASS IP PrintWay selects only print jobs in one of these JES output classes.

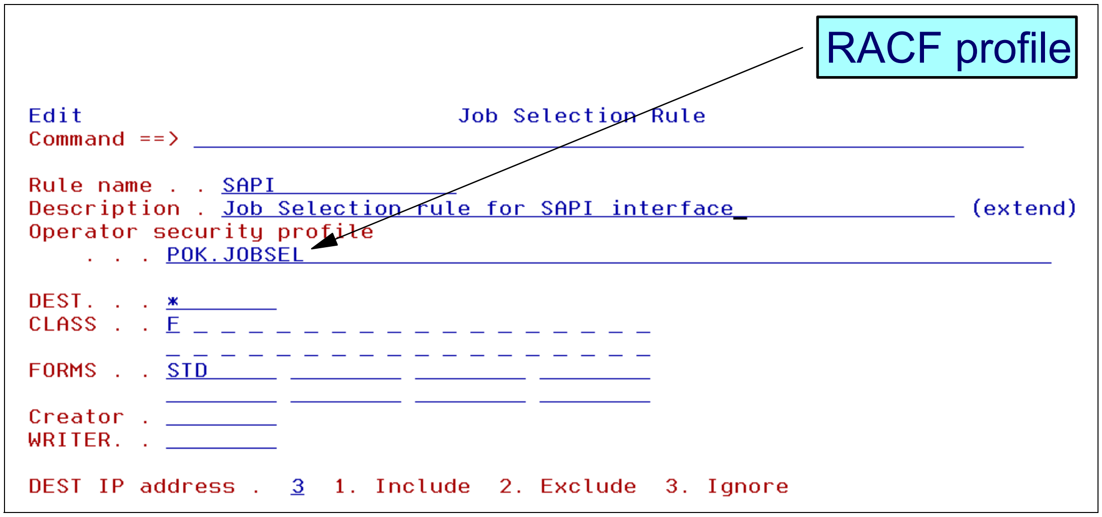
FORMS IP PrintWay selects only print jobs with one of these forms names.

Creator The user ID of the person who submitted the print job. The ID applies only to RACF user IDs, such as TSO user IDs. IP PrintWay selects only print jobs that this user ID submitted.

WRITER An external writer is an IBM- or installation-written program. IP PrintWay selects only print jobs with this writer name.

DEST IP Address Indicates whether IP PrintWay selects print jobs that specify an IP address in the DEST=IP parameter on the OUTPUT JCL statement, as follows:

- 1 = Include: Select only print jobs that specify an IP address.
- 2 = Exclude: Do not select any print jobs that specify an IP address.
- 3 = Ignore: Select any print jobs, regardless of whether they specify an IP address (default).



```
Edit                              Job Selection Rule
Command ==> ____________________________________________________________

Rule name . . SAPI_____________
Description . Job Selection rule for SAPI interface_________________ (extend)
Operator security profile
    . . . POK.JOBSEL_______________________________________________________

DEST. . . *_______
CLASS . . F _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
          _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
FORMS . . STD_____ ________ ________ ________
          ________ ________ ________ ________
Creator . ________
WRITER. . ________

DEST IP address .  3  1. Include  2. Exclude  3. Ignore
```

Figure 7-16 Job Selection Rule panel to create selection rules for extended mode

Note: You must create at least one job selection rule for IP PrintWay to select print jobs.

7.10.2 Operator security profile

To provide protection for who can display, start, and stop job selection rules using Infoprint Central, the security administrator can define a RACF resource profile in the PRINTSRV class, as shown in Figure 7-16 with POK.JOBSEL. See Figure 7-17 on page 158.

Note: This resource profile can be any combination of letters, numbers, and special characters except for commas, semicolons, parentheses, and blanks. If the value contains special characters, enclose it in single or double quotation marks. Lowercase letters are converted to uppercase. Do not start names with AOP.

You must specify the name of that RACF resource profile in the job selection rule. If you do not specify a RACF resource profile in the job selection rule, any Infoprint Central user authorized to read the Printer Inventory can display, start, and stop the job selection rule. You can specify the following RACF commands to allow no one access:

```
RDEFINE PRINTSRV POK.JOBSEL UACC(NONE)
RDEFINE PRINTSRV (AOP.ADMINISTRATOR) UACC(NONE)
```

To display, start, or stop the job selection rules, the following access levels are required:

Display `PERMIT AOP.ADMINISTRATOR CLASS(PRINTSRV) ACCESS(READ) ID` (userid or group)

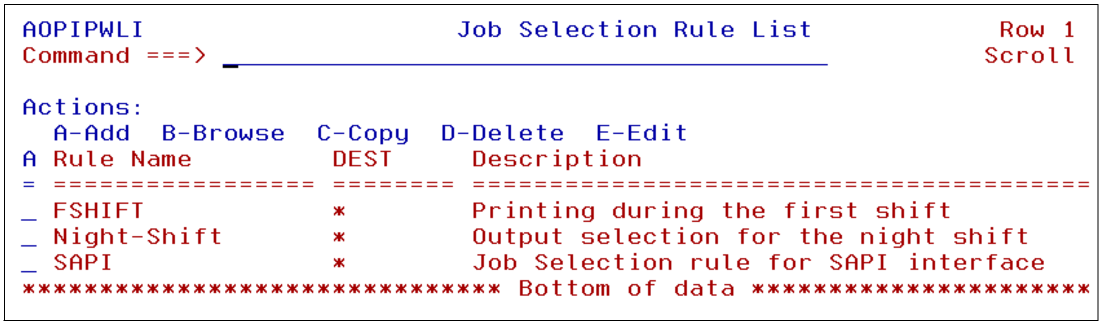
Start `PERMIT POK.JOBSEL CLASS(PRINTSRV) ACC(CONTROL) ID` (userid or group)

Stop `PERMIT POK.JOBSEL CLASS(PRINTSRV) ACC(CONTROL) ID` (userid or group)

Defined job selection rules

To display the currently defined job selection rules, use either the ISPF panels or Infoprint Central, as follows:

ISPF panels From the Infoprint Server primary panel, shown in Figure 7-2 on page 140, select option 4 to see the FSA, FSS, Pool, and Job Selection Rule Management panel. From this panel, select option 11 to list the defined rules, as shown in Figure 7-17 on page 158.

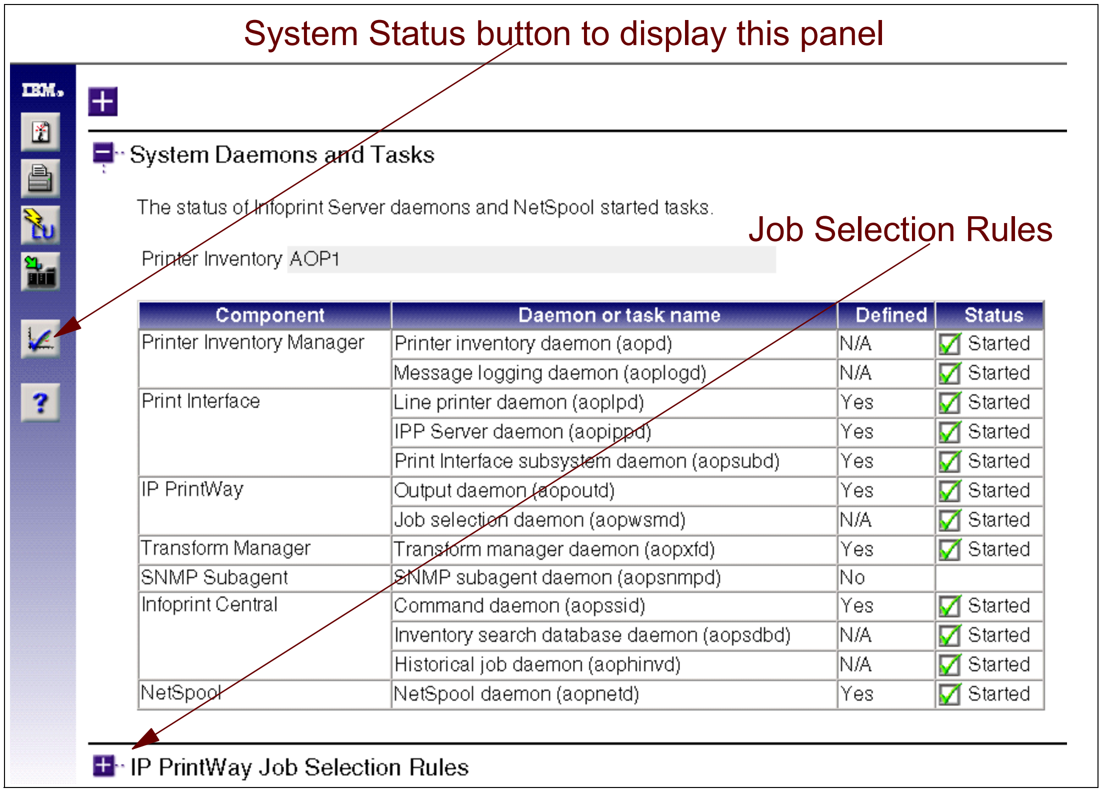


```
AOPIPWLI                    Job Selection Rule List                   Row 1
Command ===> _______________________________________                  Scroll

Actions:
  A-Add  B-Browse  C-Copy  D-Delete  E-Edit
A Rule Name          DEST     Description
= ================== ======== ========================================
_ FSHIFT              *       Printing during the first shift
_ Night-Shift         *       Output selection for the night shift
_ SAPI                *       Job Selection rule for SAPI interface
******************************* Bottom of data **********************
```

Figure 7-17 Listing the job selection rules with the ISPF panels

7.10.3 Starting and stopping job selection rules

Use Infoprint Central to select the system status icon on the Infoprint Central panel. This displays the system status of the daemons and tasks, shown in Figure 7-18. Click the Job Selection Rules button to display them, as shown in Figure 7-19 on page 159.



System Status button to display this panel

Job Selection Rules

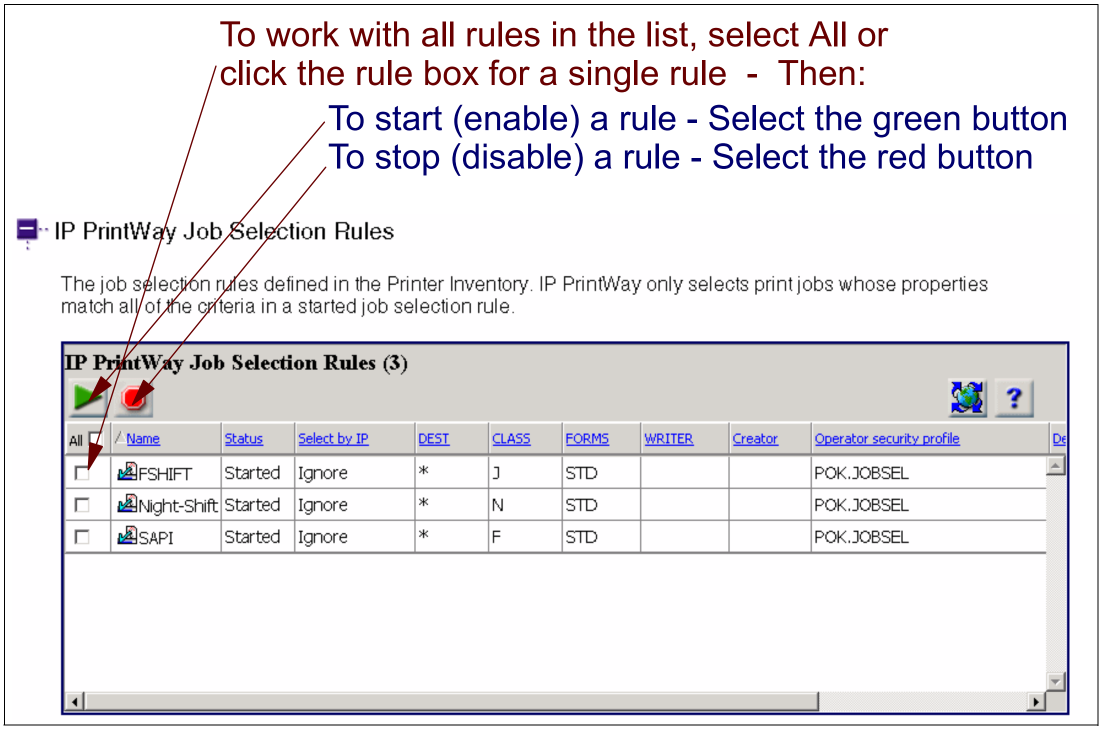
The status of Infoprint Server daemons and NetSpool started tasks.

Printer Inventory AOP1

Component	Daemon or task name	Defined	Status
Printer Inventory Manager	Printer inventory daemon (aopd)	N/A	✓ Started
	Message logging daemon (aoplogd)	N/A	✓ Started
Print Interface	Line printer daemon (aoplpd)	Yes	✓ Started
	IPP Server daemon (aopippd)	Yes	✓ Started
	Print Interface subsystem daemon (aopsubd)	Yes	✓ Started
IP PrintWay	Output daemon (aopoutd)	Yes	✓ Started
	Job selection daemon (aopwsmd)	N/A	✓ Started
Transform Manager	Transform manager daemon (aopxfd)	Yes	✓ Started
SNMP Subagent	SNMP subagent daemon (aopsnmpd)	No	
Infoprint Central	Command daemon (aopssid)	Yes	✓ Started
	Inventory search database daemon (aopsdbd)	N/A	✓ Started
	Historical job daemon (aophinvd)	N/A	✓ Started
NetSpool	NetSpool daemon (aopnetd)	Yes	✓ Started

Figure 7-18 System Status panel for access to Job Selection Rules

The operator can use Infoprint Central to start and stop the job selection rules, as shown in Figure 7-19, or the administrator can automate the starting and stopping of job selection rules by using the `cron` daemon.



To work with all rules in the list, select All or click the rule box for a single rule - Then:

To start (enable) a rule - Select the green button
To stop (disable) a rule - Select the red button

IP PrintWay Job Selection Rules

The job selection rules defined in the Printer Inventory. IP PrintWay only selects print jobs whose properties match all of the criteria in a started job selection rule.

IP PrintWay Job Selection Rules (3)

All □	Name	Status	Select by IP	DEST	CLASS	FORMS	WRITER	Creator	Operator security profile	De
□	FSHIFT	Started	Ignore	*	J	STD			POK.JOBSEL	
□	Night-Shift	Started	Ignore	*	N	STD			POK.JOBSEL	
□	SAPI	Started	Ignore	*	F	STD			POK.JOBSEL	

Figure 7-19 Starting and stopping job selection rules with Infoprint Central

Attention: If you want to run IP PrintWay extended mode and IP PrintWay basic mode at the same time, make sure that IP PrintWay extended mode does not select the same print jobs that IP PrintWay basic mode does. For example, both IP PrintWay basic mode and IP PrintWay extended mode should not select print jobs in the same class.

7.11 Working with printers

Using Infoprint Central, you can display the properties of any printer definition in the Printer Inventory. You can use a variety of search criteria to find printer definitions, including the printer definition name and the printer's location. This can help you find the name of a printer in your area. You can find and work with the following types of printers that are defined in the Printer Inventory:

- IBM AFP printers controlled by PSF (called PSF printers)

  You can start, stop, space, interrupt, pause (JES2 only), and ping PSF printers. You can also change forms and other job-selection criteria for PSF printers.

- TCP/IP-attached printers to which IP PrintWay extended mode sends print jobs (called IP PrintWay printers).

  You can start, stop, redirect, restore, and ping IP PrintWay printers. You can also see messages in the common message log for printers. See “Common message log” on page 177.

Note: You can find and work with IP PrintWay printers only when you run IP PrintWay extended mode. See “Managing printers with Infoprint Central” on page 162.

7.11.1 Security for access to printers

When using Infoprint Central, you can protect IP PrintWay and PSF printers with profiles in the following RACF classes:

PRINTSRV class This new RACF class in z/OS V1R5 is used by Infoprint Server to create profiles to restrict who can work with IP PrintWay and PSF printers. The profiles you can define in the PRINTSRV class can apply to both IP PrintWay and PSF printers.

You can define profiles in the following ways:

- A separate profile to protect each printer
- One profile to protect a group of printers
- One profile for all printers

OPERCMDS class Profiles in the OPERCMDS class restrict who can work with PSF printers using operator commands.

Note: If you define profiles in both classes, users must have access to both profiles to perform actions on PSF printers.

With Infoprint Central, users can work with printers of the following types:

IP PrintWay printers Define profiles in the PRINTSRV class to protect printers. Otherwise, any Infoprint Central user can work with any IP PrintWay printer.

Specify the profile name in the printer's printer definition.

PSF printers Define profiles in the PRINTSRV class to protect printers, or define profiles in the OPERCMDS class to protect printer actions. Otherwise, any Infoprint Central user can work with any PSF printer.

Specify the profile name in the printer's FSA definition.

7.11.2 Using Infoprint Central with printers

You can use Infoprint Central to start, stop, redirect, restore, and ping IP PrintWay printers. Messages related to these printers can be seen in the common message log for printers. Users require the proper access level to be able to work with printers with Infoprint Central.

Protecting printer access with the PRINTSRV class

You can define profiles in the PRINTSRV class to restrict who can work with printers. You create your own profile names in the PRINTSRV class. The following examples of profiles in the PRINTSRV class can apply to both IP PrintWay and PSF printers.

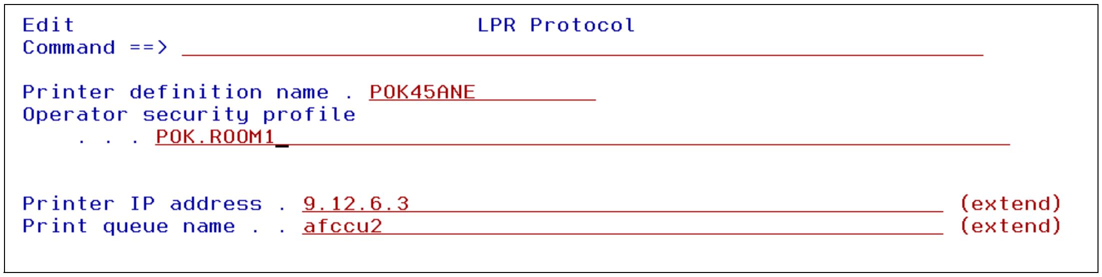
There are several types of profiles that can be defined in the PRINTSRV class where you create your own resource profile name for the printers in the Printer Inventory, as follows:

- Define a separate profile to protect each printer, as follows:

```
RDEFINE PRINTSRV (POK.PRINTER1) UACC(READ)
```

- Define one profile to protect a group of printers or all printers. An example of these profiles is shown in Figure 7-20 on page 161.

```
RDEFINE PRINTSRV (POK.ROOM1) UACC(READ)
```



```
Edit                                  LPR Protocol
Command ==> ____________________________________________________________

Printer definition name . POK45ANE_________
Operator security profile
    . . . POK.ROOM1_____________________________________________________


Printer IP address . 9.12.6.3________________________________________ (extend)
Print queue name . . afccu2__________________________________________ (extend)
```

Figure 7-20 Protocol panel for defining an operator security profile

- Define a profile for all users to use all printers:

  ```
  RDEFINE PRINTSRV (POK.PRINTALL) UACC(READ)
  ```

The profiles can be defined with UACC(NONE). An example follows:

```
RDEFINE PRINTSRV (POK.ROOM1) UACC(NONE)
PERMIT POK.ROOM1 CLASS(PRINTSRV) ID(user ID or group ID) ACC(READ)
```

Printer access levels

Many of the printer options for controlling them require different access levels. They may require either READ, UPDATE, or CONTROL access levels; the appropriate access is as follows:

READ	Find and display printers Ping printers View log - IP PrintWay printers only View properties
UPDATE	Change forms - PSF printers only Change job selection - PSF printers only
CONTROL	Interrupt - PSF printers only Pause - PSF printers only Redirect - IP PrintWay printers only Repeat - PSF printers only Restore - IP PrintWay printers only Space printers- PSF printers only Start printers Stop printer and delete the current print job Stop printer after the current print job completes

Important: You must specify the name of the profile, for example POK.ROOM1, that applies to each printer in the Printer Inventory.

7.11.3 Adding security profiles to existing printer definitions

The **pidu** list command lists the names of all objects in a specified object class or only objects that meet certain criteria. You can use the list command in combination with the modify command to list all or selected objects in an object class and then modify one or more attributes. For example, the following command lists all IP PrintWay FSS definitions:

```
pidu -c 'list printway-fss ; '
```

The programming language awk lets you work with information stored in files. Since awk is a file-processing language that is well suited to data manipulation and retrieval of information from text files, it can be used with the **PIDU** list command for files that you specify on the command line to provide the input data for awk to manipulate. One at a time, and in order, awk compares each input record with the pattern of every rule in the program. When a pattern matches, awk performs the action part of the rule on that input record.

IP PrintWay printer definitions

The **PIDU** command can be used to specify the name of the same RACF profile in all IP PrintWay printer definitions that do not already contain a profile name. This example of the **PIDU** list command lists the names of all IP PrintWay printer definitions with no value in the operator-security-profile attribute. These names are piped to the awk program, which writes modify commands to modify the printer definitions to file /tmp/defs. An example of the **pidu** list command follows:

```
pidu -qc "list printer where printer-type=ip-printway and
operator-security-profile=null;" | awk'{print "modify printer
" operator-security-profile = \"POK.ROOM1\";"}' > /tmp/defs
```

Inspect the /tmp/defs file to make sure the modify commands are acceptable. Then, enter the following **pidu** command to update the Printer Inventory:

```
  pidu < /tmp/defs
```

> **Note:** See Figure 7-20 on page 161 for the Printer Inventory panel that would get updated after using the **pidu** command.

PSF printer definitions

For PSF printers, the operator security profile is kept in the PSF FSA definitions for TCP/IP-attached printers. For all of these defined printers that do not already have a profile, issue the following command:

```
pidu -qc "list fsa where fsa-type=psf-tcpip and
operator-security-profile=null;" |awk '{print "modify fsa " $1
" operator-security-profile=\"POK.PRINTALL\";"}' > /tmp/defs
```

Inspect the /tmp/defs file to make sure the modify commands are acceptable. Then, enter the following **pidu** command to update the Printer Inventory:

```
  pidu < /tmp/defs
```

> **Note:** The **pidu** list command lists the names of all printer definitions with a specified object. Therefore, when you are scanning the Printer Inventory for printers to modify, you can specify any specific object such as output-class=K. If you wanted to change all class J definitions to class F, issue the following command:
>
> ```
> pidu -qc "list printer where output-class = J ; " |
>       awk '{ print "modify printer " $1 " output-class = \"F\";" }' |
>       pidu
> ```

7.12 Managing printers with Infoprint Central

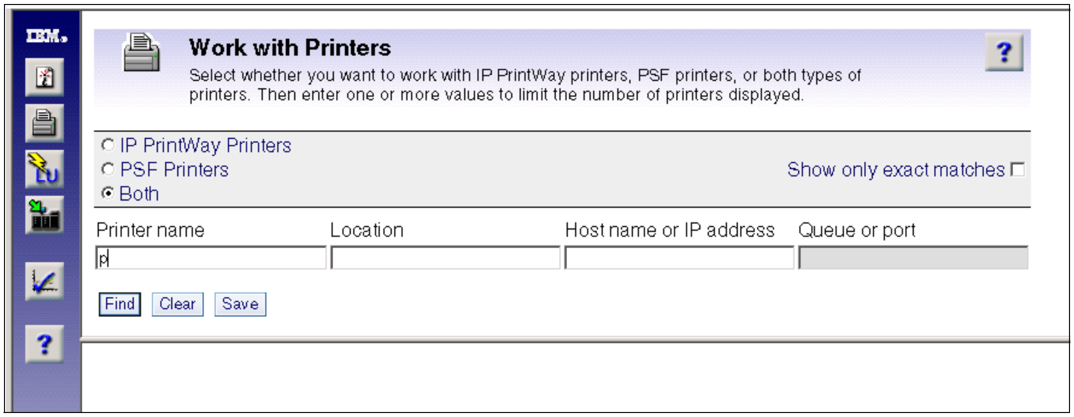
Using Infoprint Central, you can find and work with printers that are defined in the Printer Inventory, including IBM AFP printers controlled by PSF (called PSF printers), and TCP/IP-attached printers to which IP PrintWay can send print jobs (called IP PrintWay printers).

Use the following steps to display the list of printers that match a particular criteria:

1. Select the **Work with Printers** button on the Infoprint Central primary panel, shown in Figure 7-7 on page 146.
2. Enter a value in one or more of the search fields, as shown in Figure 7-21 on page 163.

3. Select the **Find** button.

Printers that match all of the specified values are displayed, as shown in Figure 7-22 on page 164.



Work with Printers

Select whether you want to work with IP PrintWay printers, PSF printers, or both types of printers. Then enter one or more values to limit the number of printers displayed.

○ IP PrintWay Printers
○ PSF Printers
● Both

Show only exact matches □

Printer name: p
Location:
Host name or IP address:
Queue or port:

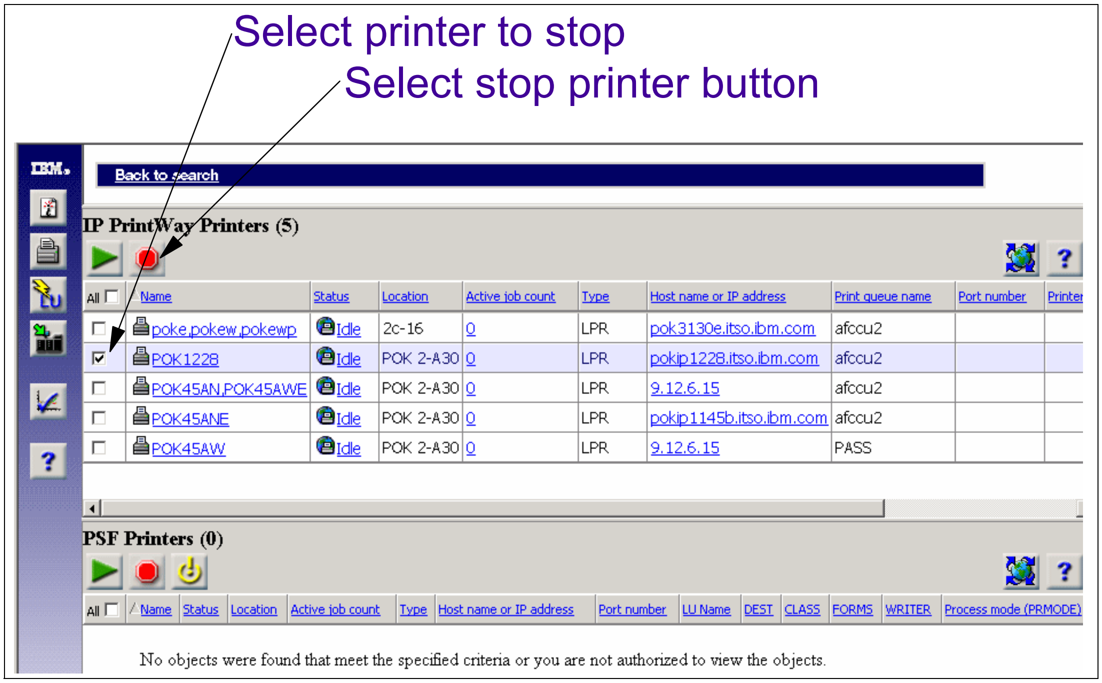
Figure 7-21 Selecting both types of printers that have a name beginning with p

7.12.1 Stopping a printer

The printer list displayed by Infoprint Central, as shown in Figure 7-22, provides two buttons for starting and stopping one or more of the printers on the list. To stop one of the printers, do the following:

- Select the box next to the printer name.
- Click the **Stop** button. This stops the selected printer so that IP PrintWay does not process any more print jobs for the printer. However, data IP PrintWay has already transmitted to a printer and that is already in the printer's buffer continues to print. Infoprint Server and JES continue to accept print requests for stopped printers and add print jobs to the printers' queues.
- To start or stop a printer requires CONTROL access. An example of the RACF profile commands that would allow user ID AOPOPER to stop the printer are the following:

```
RDEFINE PRINTSRV (POK.POK1228) UACC(NONE)
PERMIT POK.POK1228 CLASS(PRINTSRV) ID(AOPOPER) ACC(CONTROL)
```



IP PrintWay Printers (5)

All □	Name	Status	Location	Active job count	Type	Host name or IP address	Print queue name	Port number	Printer
□	poke,pokew,pokewp	Idle	2c-16	0	LPR	pok3130e.itso.ibm.com	afccu2		
☑	POK1228	Idle	POK 2-A30	0	LPR	pokip1228.itso.ibm.com	afccu2		
□	POK45AN,POK45AWE	Idle	POK 2-A30	0	LPR	9.12.6.15	afccu2		
□	POK45ANE	Idle	POK 2-A30	0	LPR	pokip1145b.itso.ibm.com	afccu2		
□	POK45AW	Idle	POK 2-A30	0	LPR	9.12.6.15	PASS		

PSF Printers (0)

All □	Name	Status	Location	Active job count	Type	Host name or IP address	Port number	LU Name	DEST	CLASS	FORMS	WRITER	Process mode (PRMODE)

No objects were found that meet the specified criteria or you are not authorized to view the objects.

Figure 7-22 Operator request to stop a printer using Infoprint Central

Stop printer dialog

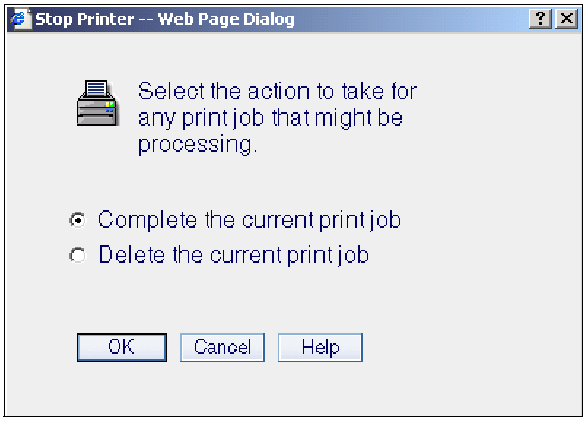
A dialog box is displayed when the Stop printer button is clicked, as shown in Figure 7-23 on page 165. Here you can specify what to do with the data set currently printing.

On the Stop Printer panel, you can select whether to stop the printer immediately or wait until the current print job completes printing. If you stop the printer immediately, the current print job is deleted.

To stop a printer immediately but without deleting the current print job, do the following:

- On the Stop Printer panel, select “Complete the current print job” and click **OK**.
- On the IP PrintWay Printer Information panel, expand the Print Job Queue section.
- On the Infoprint Server Print Jobs panel, hold the print job that is processing.

If a printer is not working, you can redirect all print jobs to an alternate IP PrintWay printer after you stop it. To redirect a printer, select (Redirect). To restart a stopped printer, select (Start). If IP PrintWay is restarted, all stopped printers are automatically restarted.



Stop Printer -- Web Page Dialog

Select the action to take for any print job that might be processing.

● Complete the current print job

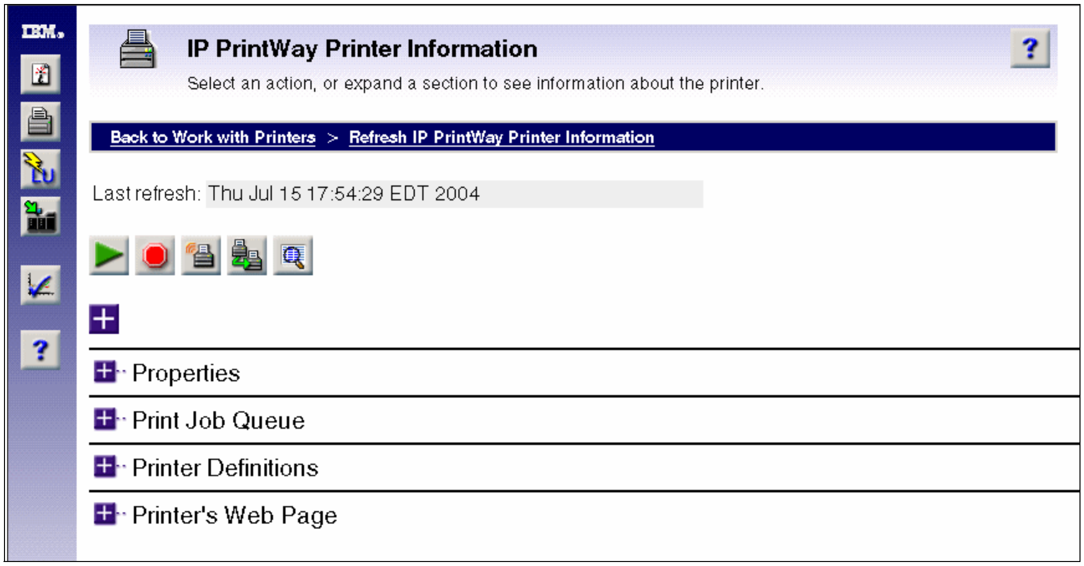
○ Delete the current print job

Figure 7-23 Panel to stop a printer and select the action to take for the data set

Display detailed information for a printer

To display detailed information about one of the printers listed in Figure 7-22, do the following:

- Click the name of the printer, the printer host name, or the IP address field for the printer.
- The IP PrintWay Printer Information panel shown in Figure 7-24 is displayed. By clicking on the plus sign (**+**) beside each item, you can expand any of the listed categories to get details about the printer's properties, print job queue, printer definitions, and its Web page. Figure 7-25 on page 166 shows the same Printer Information screen with the Properties field expanded. To expand all the categories, click the top **+** button.
- The IP PrintWay Printer Information panel also provides a set of action buttons that let you start, stop, ping, redirect, and view the log for the printer. These buttons are identified in Figure 7-25.



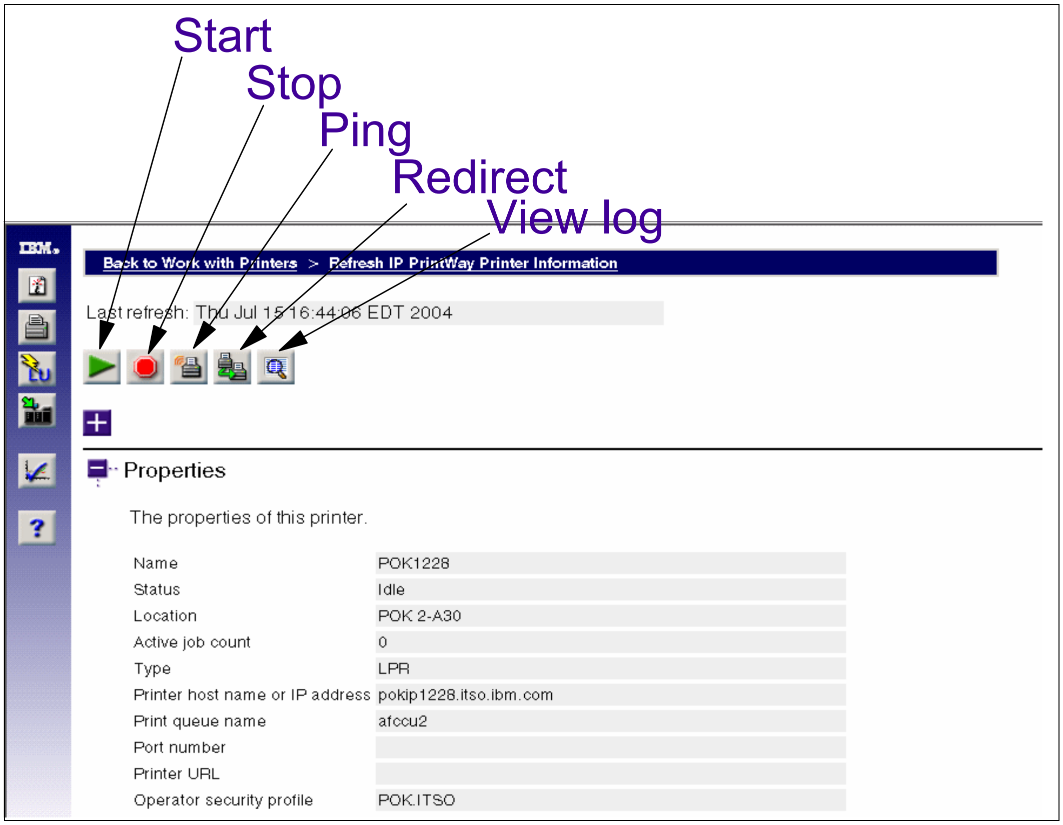
IP PrintWay Printer Information

Select an action, or expand a section to see information about the printer.

IBM
Back to Work with Printers > Refresh IP PrintWay Printer Information

Last refresh: Thu Jul 15 17:54:29 EDT 2004

Figure 7-24 Display of detail printer information panel


Start
Stop
Ping
Redirect
View log
IBM.
Back to Work with Printers > Refresh IP PrintWay Printer Information

Last refresh: Thu Jul 15 16:44:06 EDT 2004

Properties

The properties of this printer.

Name	POK1228
Status	Idle
Location	POK 2-A30
Active job count	0
Type	LPR
Printer host name or IP address	pokip1228.itso.ibm.com
Print queue name	afccu2
Port number	
Printer URL	
Operator security profile	POK.ITSO

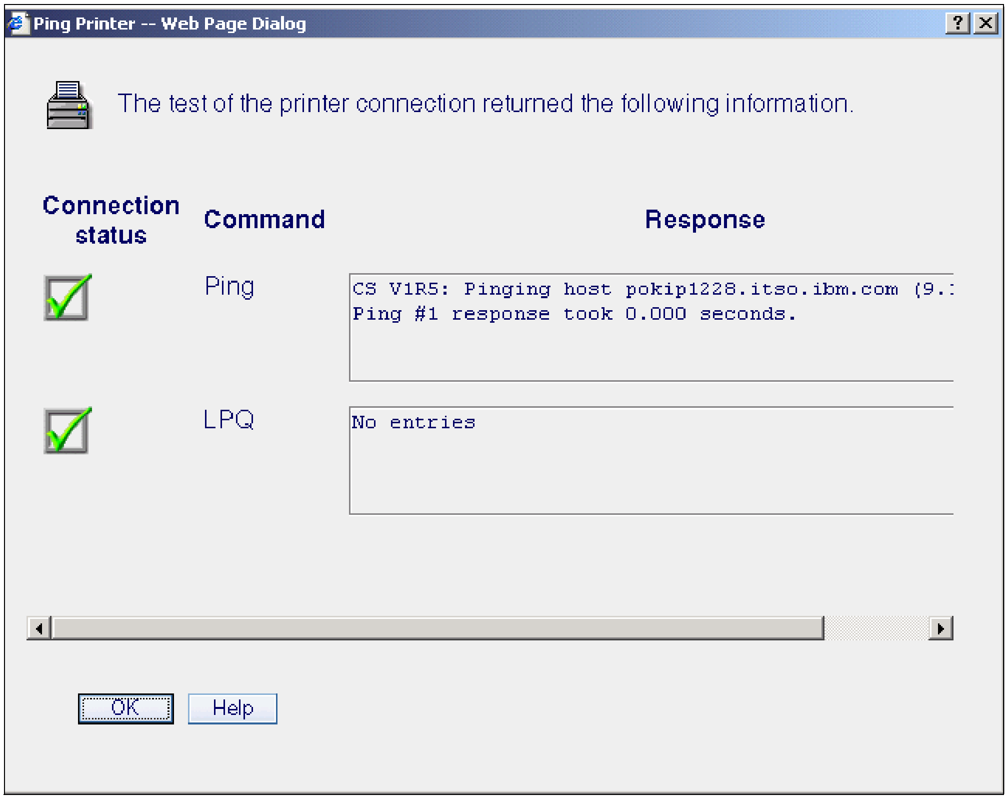
Figure 7-25 Action buttons and properties details for a printer

7.12.2 Ping a printer

When you ping a printer, this tests the TCP/IP network connection to a remote printer or print server. Figure 7-26 on page 167 is the response back from clicking the Ping button in Figure 7-25. The Connection status shows that the printer is active. The possible returned status is either success of failure, as follows:

Success This is indicated by a green check mark in the box, which means the TCP/IP network is working and the z/OS system can communicate with the printer. If you ping a printer instead of a print server, a successful response also means the printer is turned on. However, the printer might be offline.

Failure This is indicated by a red X in the box. This means the printer did not respond in 1 second. This can occur if the network is not working, the printer is not turned on, or the remote host is slow to respond.



The test of the printer connection returned the following information.

Connection status	Command	Response
☑	Ping	CS V1R5: Pinging host pokip1228.itso.ibm.com (9. Ping #1 response took 0.000 seconds.
☑	LPQ	No entries

Figure 7-26 Panel showing the response from a ping of the printer

For LPQ, this provides information about jobs on the LPD's print queue. The information provided depends on the printer's implementation of the LPD. You can see a response from this command only for IP PrintWay printers that use the LPR to LPD protocol to communicate with the printer.

7.12.3 Redirect a printer

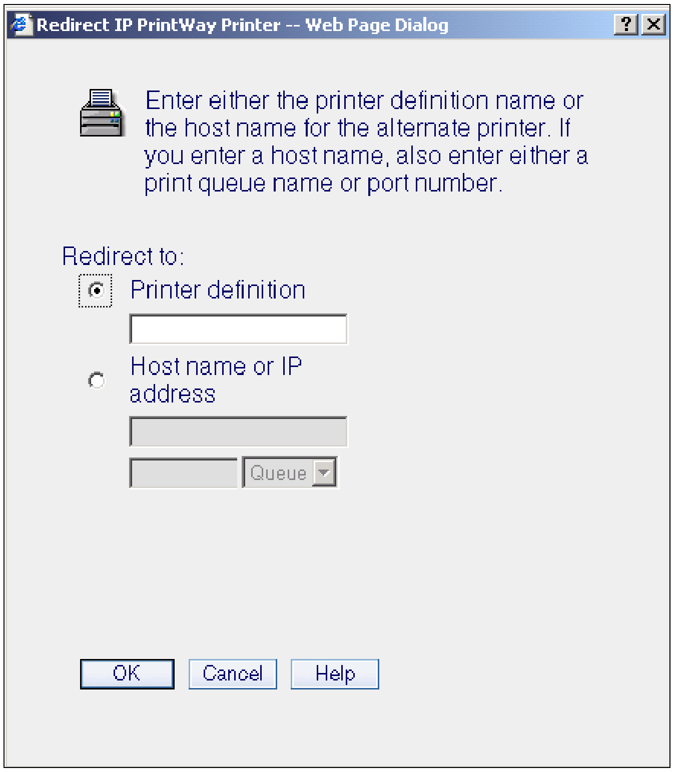
The panel shown in Figure 7-27 on page 168 lets you move all print jobs currently on the queue of an IP PrintWay printer and all print jobs that are subsequently submitted to this printer to an alternate IP PrintWay printer. When you select the alternate printer, consider the following:

- The alternate printer must be an IP PrintWay printer.
- If a printer definition does not exist for the alternate printer in the Printer Inventory, the printer must support either the LPR or direct-sockets printing protocol.
- The alternate printer must not itself be redirected to an alternate printer.
- The alternate printer should be able to print the same types of data streams (for example, PostScript or PCL) as the original printer so that data prints correctly.

Selecting a printer

To select, enter the name of the printer definition for the alternate printer. IP PrintWay redirects print jobs to the printer address specified in this printer definition. However, IP PrintWay continues to use attributes in the printer definition for the original printer to format data.

A second option is to enter the Host name or IP address of an alternate printer. Also, enter either the print queue name or the port number of the alternate printer. Then select either Queue or Port to identify the type of value you entered.



Enter either the printer definition name or the host name for the alternate printer. If you enter a host name, also enter either a print queue name or port number.

Redirect to:

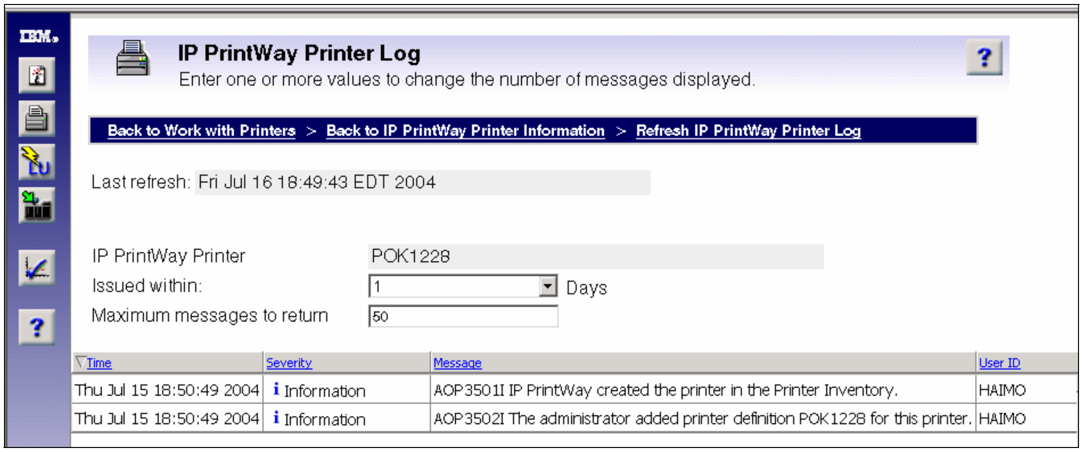
● Printer definition

○ Host name or IP address

Figure 7-27 Panel to redirect a printer

7.12.4 Viewing the printer log

Figure 7-28 shows the printer log, which lists messages from Infoprint Server for the selected printer. Fields before and after the message text contain additional information, such as the time the message was sent.



IP PrintWay Printer Log

Enter one or more values to change the number of messages displayed.

Back to Work with Printers > Back to IP PrintWay Printer Information > Refresh IP PrintWay Printer Log

Last refresh: Fri Jul 16 18:49:43 EDT 2004

IP PrintWay Printer POK1228

Issued within: 1 Days

Maximum messages to return 50

Time	Severity	Message	User ID
Thu Jul 15 18:50:49 2004	i Information	AOP3501I IP PrintWay created the printer in the Printer Inventory.	HAIMO
Thu Jul 15 18:50:49 2004	i Information	AOP3502I The administrator added printer definition POK1228 for this printer.	HAIMO

Figure 7-28 Infoprint View log for the specified printer

Note: To change the time period for which messages are displayed, specify the desired number of days in the "Issued within" field. The "Maximum messages to return" field lets you specify a number of messages to be returned, and can be any value from 1 to 999. Click **Refresh**.

7.13 Working with print jobs

A print job consists of one or more documents submitted in the same job and that prints on the same printer. With Infoprint Central, you can work with all print jobs on the z/OS JES spool.

Using Infoprint Central, you are able to see more information about print jobs that Infoprint Server processes. For example, you can see whether an Infoprint Server print job completed successfully and where it printed even if the print job is no longer on the JES spool.

As shown in Figure 7-29 on page 170, the following print jobs on the JES spool can be displayed:

- **Infoprint Server print jobs**: All print jobs that Infoprint Server is processing or has processed. This includes print jobs that Infoprint Server (either the Print Interface or NetSpool component) creates on the JES spool, and print jobs that are submitted from TSO/E and directed to the IP PrintWay extended mode component of Infoprint Server.

  Also, Infoprint Server processes documents submitted from:

  - Remote workstations through Print Interface
  - VTAM applications (such as CICS and IMS) through NetSpool
  - z/OS jobs to any IP PrintWay-controlled printer
  - z/OS jobs to any printer using the Infoprint Server subsystem.
- **JES print jobs**: All print jobs that are currently on the JES spool, including Infoprint Server print jobs that are currently on the JES spool. JES print jobs also include jobs that are not Infoprint Server print jobs. For example, print jobs submitted from TSO/E and directed to PSF printers are JES jobs, but not Infoprint Server print jobs unless they were submitted to the Infoprint Server subsystem.

Note: If you run IP PrintWay basic mode, you cannot use Infoprint Central to work with jobs submitted to IP PrintWay or IP PrintWay printers. However, you can use Infoprint Central to do other functions.

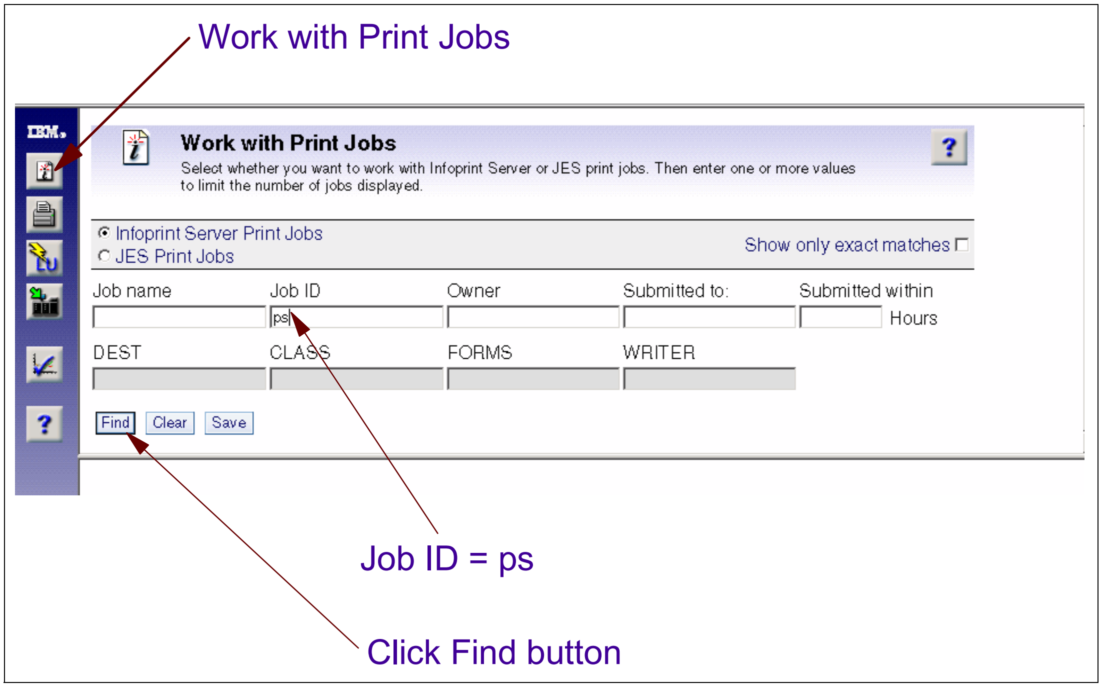

Work with Print Jobs
Work with Print Jobs
Select whether you want to work with Infoprint Server or JES print jobs. Then enter one or more values to limit the number of jobs displayed.
● Infoprint Server Print Jobs
○ JES Print Jobs
Show only exact matches □
Job name
Job ID
ps
Owner
Submitted to:
Submitted within
Hours
DEST
CLASS
FORMS
WRITER
Find
Clear
Save
Job ID = ps
Click Find button

Figure 7-29 Selecting print jobs to display

7.13.1 Displaying print jobs

When print jobs are submitted to z/OS for processing with Infoprint Server, typically that print consists of one output data set. However, a print job can consist of several output data sets submitted together in the same batch job.

A print job is typically one output data set. However, a print job can consist of several output data sets submitted together in the same batch job. In JES2, a print job is called an output group.

Selection criteria for print job search

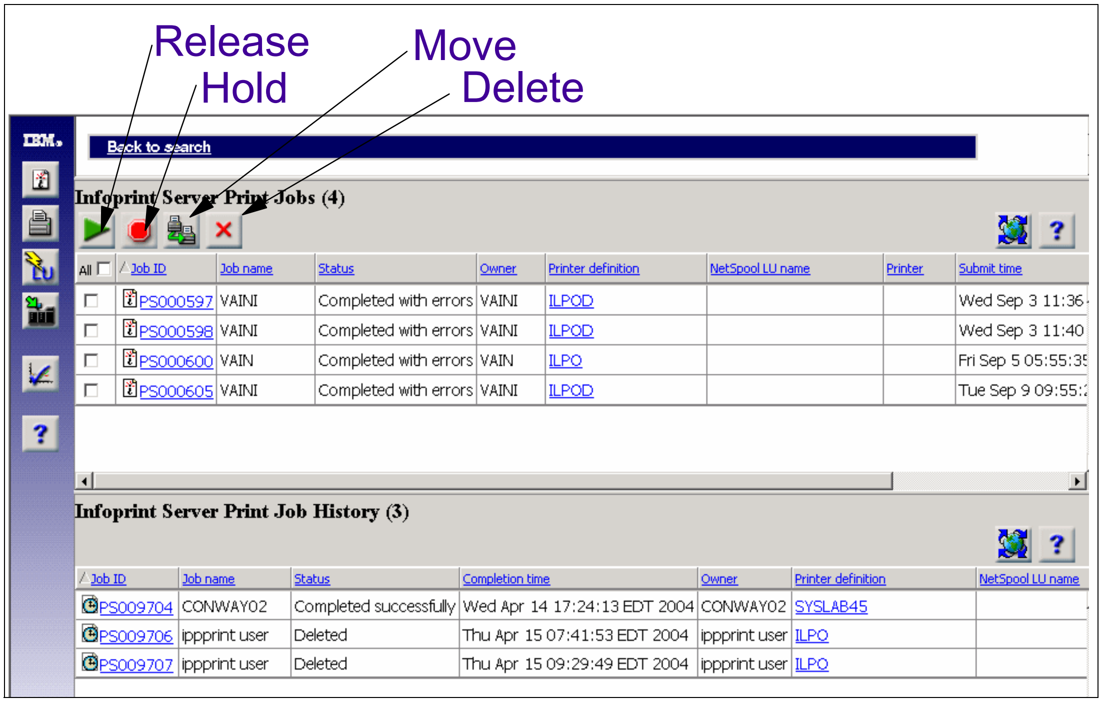
When selecting Infoprint Server print jobs, you can specify the following selection criteria:

Job name	Windows® logon name - TSO job name
Job ID	Job IDs beginning with PS - JES Job IDs
Owner	TSO user ID of the job submitter - Windows logon name
Submitted to:	Name of the printer - NetSpool LU name for the printer

When select JES print jobs, you can specify the following:

Job name	TSO job name
Job ID	JES Job IDs
Owner	TSO user ID of the job submitter
DEST	Printer destination if defined in the Printer Inventory
CLASS	Any JES SYSOUT class
FORMS	Any JES defined forms
WRITER	An external writer that processes the data sets

Figure 7-29 on page 170 shows the Work with Print Jobs panel, where the user is requesting to display all Job IDs beginning with “ps” as a selection criteria. When the user clicks the **Find** button to display the selected jobs, the resulting display is shown in Figure 7-30. Also displayed is a job history of print jobs that have already printed. These are print jobs that Infoprint Server finished processing and that have been deleted from the JES spool. You can see whether historical print jobs completed successfully or not, and you can see all messages in the Infoprint Server common message log for a print job.



Infoprint Server Print Jobs (4)

All □	Job ID	Job name	Status	Owner	Printer definition	NetSpool LU name	Printer	Submit time
□	PS000597	VAINI	Completed with errors	VAINI	ILPOD			Wed Sep 3 11:36
□	PS000598	VAINI	Completed with errors	VAINI	ILPOD			Wed Sep 3 11:40
□	PS000600	VAIN	Completed with errors	VAIN	ILPO			Fri Sep 5 05:55:35
□	PS000605	VAINI	Completed with errors	VAINI	ILPOD			Tue Sep 9 09:55:

Infoprint Server Print Job History (3)

Job ID	Job name	Status	Completion time	Owner	Printer definition	NetSpool LU name
PS009704	CONWAY02	Completed successfully	Wed Apr 14 17:24:13 EDT 2004	CONWAY02	SYSLAB45	
PS009706	ippprint user	Deleted	Thu Apr 15 07:41:53 EDT 2004	ippprint user	ILPO	
PS009707	ippprint user	Deleted	Thu Apr 15 09:29:49 EDT 2004	ippprint user	ILPO	

Figure 7-30 Display of Infoprint Server print jobs

7.13.2 Modifying selected print jobs

To work with a print job, select either the **All** box to specify all selected jobs, or select an individual print job box. Then, click one of the four action buttons, shown in Figure 7-30, to perform the desired action on the selected job.

Release

Release makes the print job available for printing. Select this action to:

- Print a job that has a status of Held.
- Retry the transmission of a print job that has a status of Waiting for retry. The retry occurs immediately instead of at the next scheduled retry time.
- Retry a retained print job that has a status of Completed with errors or Completed successfully.

Hold

Hold prevents the selected print jobs from printing. It also prevents a print job that Infoprint Server has retained on the JES spool from being deleted when the retention time expires. Select this action to:

- Print a print job at a later time.
- Prevent a retained print job from being automatically deleted from the JES spool when the retain time expires.

Move

The **Move** button moves the selected print jobs to an alternate printer of the same type and on the same system. For example, you can move a print job that was submitted to print on an IP PrintWay printer to another IP PrintWay printer. If you select to move more than one print job at a time, you must move all print jobs to the same printer. You might want to move a print job when the printer is busy processing other print jobs.

You can move a print job that is currently processing on an IP PrintWay printer. However, the entire print job, including all copies, prints again from the beginning on the alternate printer.

On the Move Print Job panel, you can specify the name of the printer definition for the alternate printer. If you do not know the printer definition name or if no printer definition exists for the alternate printer, you can specify either the host name or IP address of the alternate IP PrintWay printer; or CLASS, DEST, FORMS, PRMODE, and WRITER values of the alternate PSF printer.

Delete

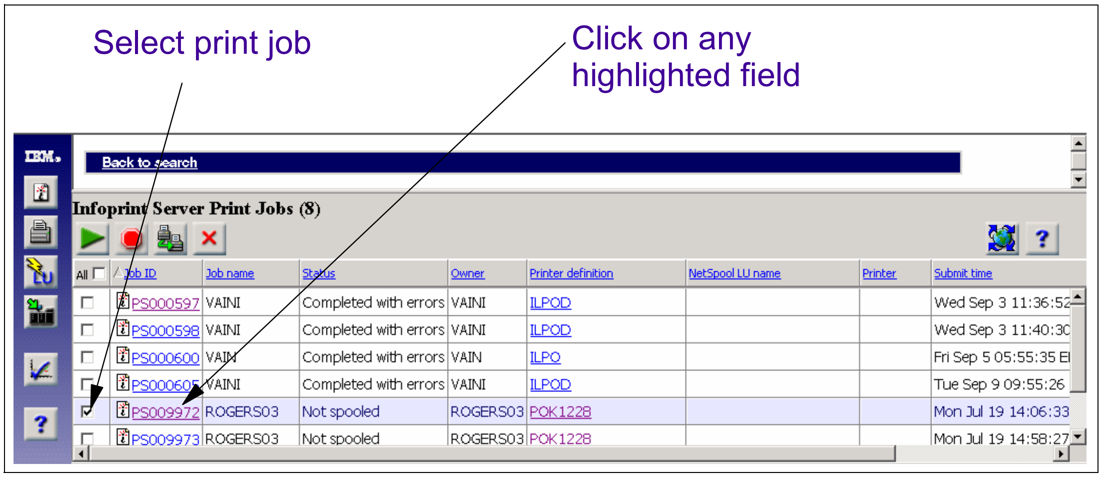
The **Delete** button deletes the selected print jobs from the JES spool. If the print job is currently processing on an IP PrintWay printer, the print job is not retained on the JES spool.

7.13.3 Displaying print job information

To see detailed information about a print job, select a job and click any highlighted values, as illustrated in Figure 7-31. In this example, there are two highlighted fields:

Job ID Displays the Infoprint Server Print Job Information panel, as shown in Figure 7-32 on page 173.

Printer definition Displays the Printer Definition Information panel, as shown in Figure 7-33 on page 173.



Infoprint Server Print Jobs (8)

All □	Job ID	Job name	Status	Owner	Printer definition	NetSpool LU name	Printer	Submit time
□	PS000597	VAINI	Completed with errors	VAINI	ILPOD			Wed Sep 3 11:36:52
□	PS000598	VAINI	Completed with errors	VAINI	ILPOD			Wed Sep 3 11:40:3C
□	PS000600	VAIN	Completed with errors	VAIN	ILPO			Fri Sep 5 05:55:35 EI
□	PS000605	VAINI	Completed with errors	VAINI	ILPOD			Tue Sep 9 09:55:26
☑	PS009972	ROGERS03	Not spooled	ROGERS03	POK1228			Mon Jul 19 14:06:33
□	PS009973	ROGERS03	Not spooled	ROGERS03	POK1228			Mon Jul 19 14:58:27

Figure 7-31 Selecting a job to display print job information

Important: When displaying panels with a **+**, expand these sections by selecting the **+** to the left of each section title.

Print Job Information panel

On the Infoprint Server Print Job Information panel, shown in Figure 7-32, you can see more details about the print job and perform these additional actions: Release, Change priority, Hold, Delete, View properties, and View log.

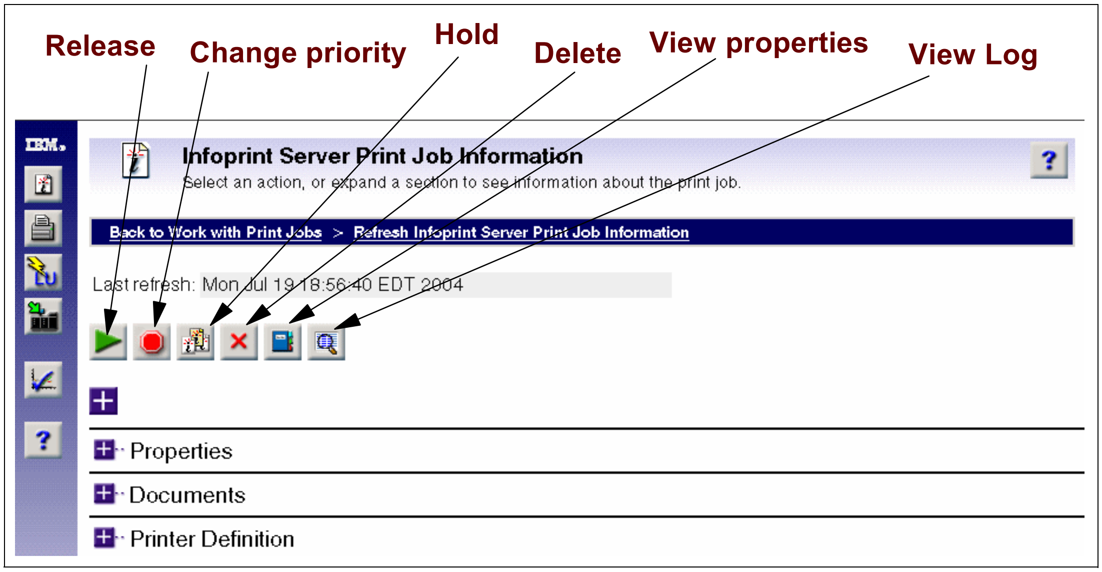

Release
Change priority
Hold
Delete
View properties
View Log
IBM.
Infoprint Server Print Job Information
Select an action, or expand a section to see information about the print job.
Back to Work with Print Jobs > Refresh Infoprint Server Print Job Information
Last refresh: Mon Jul 19 18:56:40 EDT 2004
Properties
Documents
Printer Definition

Figure 7-32 Infoprint Server Print Job Information panel

View properties

To view the selected printer properties, click the **View properties** button as shown in Figure 7-33.

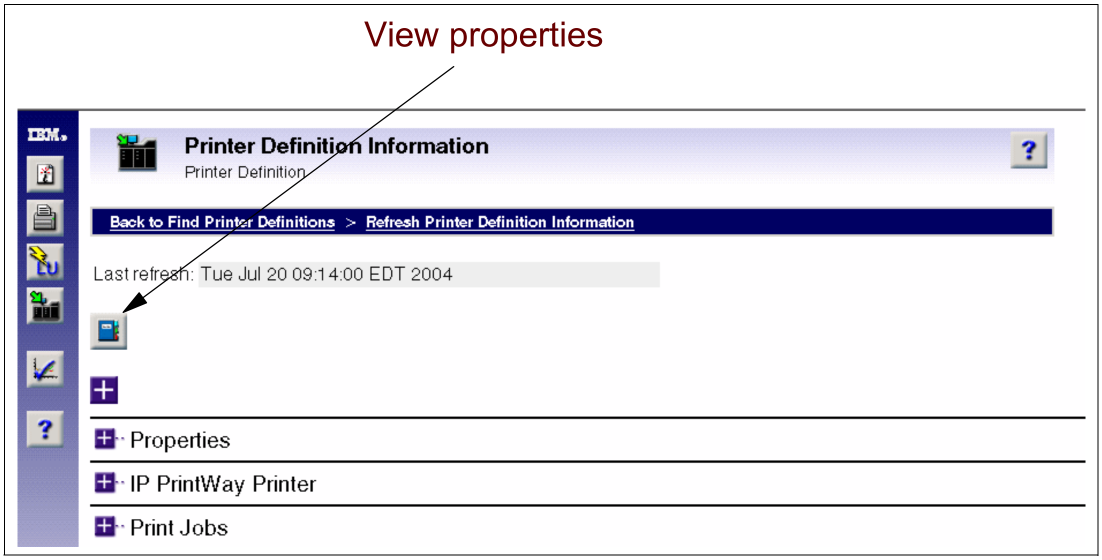
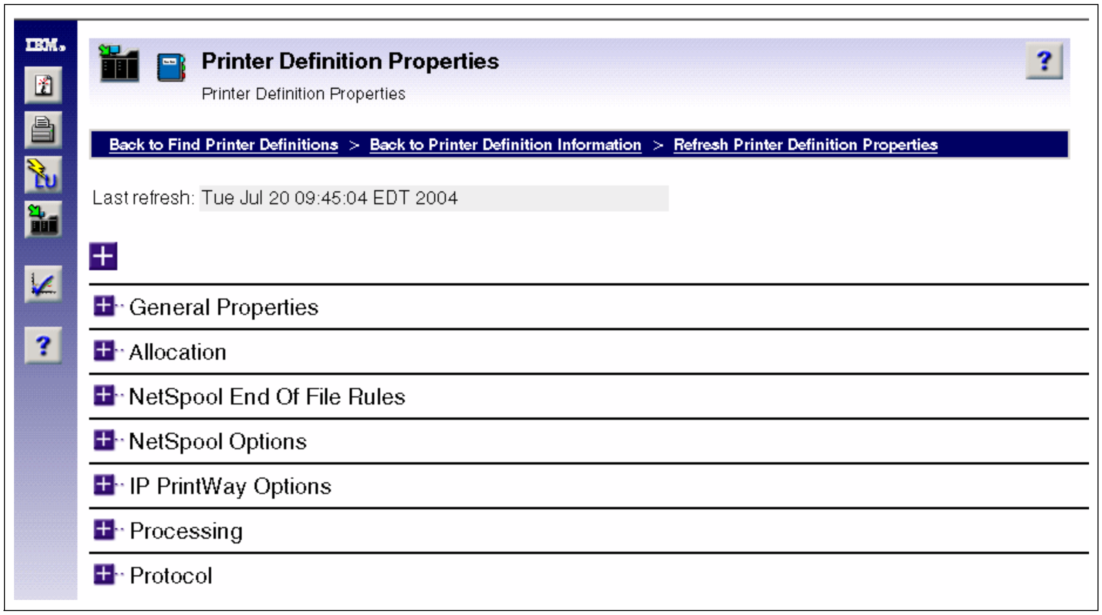

View properties
IBM.
Printer Definition Information
Printer Definition
Back to Find Printer Definitions > Refresh Printer Definition Information
Last refresh: Tue Jul 20 09:14:00 EDT 2004
Properties
IP PrintWay Printer
Print Jobs

Figure 7-33 Printer Definition Information panel

Printer Definition Properties

The panel shown in Figure 7-34 on page 174 displays all the properties of the selected printer that were defined in the Printer Inventory.



Printer Definition Properties

Printer Definition Properties

Back to Find Printer Definitions > Back to Printer Definition Information > Refresh Printer Definition Properties

Last refresh: Tue Jul 20 09:45:04 EDT 2004

- General Properties
- Allocation
- NetSpool End Of File Rules
- NetSpool Options
- IP PrintWay Options
- Processing
- Protocol

Figure 7-34 Printer Definition Properties panel

7.13.4 Other display options for print jobs

After using a search criteria to find print jobs, and specifying a print job to work with, you can use the action buttons shown in Figure 7-32 on page 173 to perform the following functions:

- Release the job if it was held, waiting to be retried, or when the problem with the print job or printer is fixed, so that it prints immediately.
- Change the job’s priority as discussed in “Change a print job’s priority” on page 175.
- Place the job in hold
- Delete a print job
- View the job properties
- View the job’s message log

Active print jobs

These are print jobs that Infoprint Server is writing to the JES spool and print jobs that are on the JES spool that have not yet completed.

Some actions you can take on active print jobs are:

- Check the status of a print job. See the Status column for the jobs displayed in Figure 7-31 on page 172.
- Increase the priority of a print job. This can be done from an Infoprint Server Print Job panel or a JES Print Job panel.
- Release a print job if it was held or is waiting to be retried so that it prints immediately.
- Delete a print job.

Retained print jobs

These are print jobs that have completed processing on IP PrintWay printers but have not been deleted from the JES spool. Infoprint Server retains print jobs on the JES spool for the retention period specified for the printer or print job. After the retention period expires, Infoprint Server automatically deletes the print job.

Some actions you can take on retained print jobs before they are deleted are:

- Release a print job to print when a problem with the print job or printer is fixed.
- Move a print job to another printer and release it to print.
- Hold a print job to prevent the print job from being deleted when its retention period expires.
- Delete the print job if the specified retention period is FOREVER.

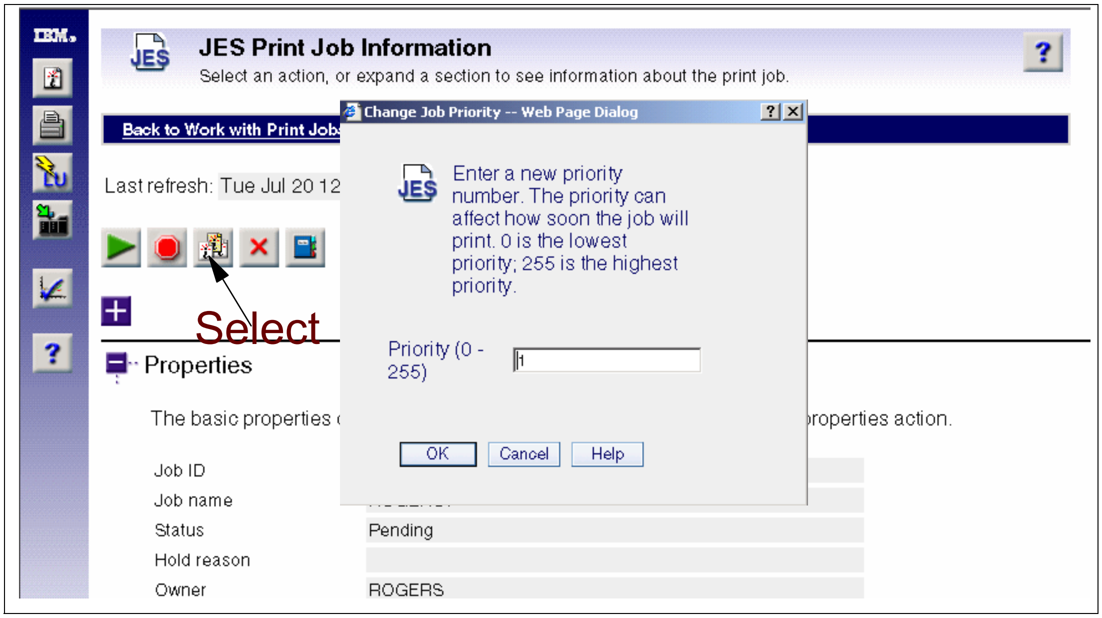
Historical print jobs

These are print jobs that Infoprint Server finished processing and that have been deleted from the JES spool. You can see whether historical print jobs completed successfully or not, and you can see all messages in the Infoprint Server common message log for a print job.

> **Note:** Messages related to the print job are placed into the common message log. See "Common message log" on page 177.

7.13.5 Change a print job's priority

The priority of a print job, either an Infoprint Server print job or a JES print job, can be changed by clicking the **Change priority** button, as shown in Figure 7-35. A pop-up window appears on the same panel to allow you to change the priority of the print data set.



IBM
JES Print Job Information
?
Back to Work with Print Job
Change Job Priority -- Web Page Dialog
Select
Properties
OK
Cancel
Help

Select an action, or expand a section to see information about the print job.

Last refresh: Tue Jul 20 12

Enter a new priority number. The priority can affect how soon the job will print. 0 is the lowest priority; 255 is the highest priority.

Priority (0 - 255) 1

The basic properties (... roperties action.

Job ID

Job name

Status Pending

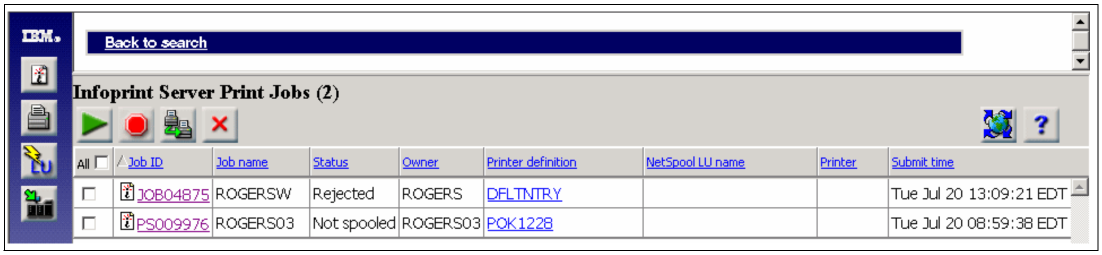
Hold reason

Owner ROGERS

Figure 7-35 Changing a print job's priority

7.13.6 Job submission without a valid destination

A job is submitted, in this example as a JES job to be processed by Infoprint Server for output, and the specified destination printer was not valid. The output data set is assigned the DFLTNTRY printer defined in the Printer Inventory, as shown in Figure 7-36 on page 176.



Infoprint Server Print Jobs (2)

All □	Job ID	Job name	Status	Owner	Printer definition	NetSpool LU name	Printer	Submit time
□	JOB04875	ROGERSW	Rejected	ROGERS	DFLTNTRY			Tue Jul 20 13:09:21 EDT
□	PS009976	ROGERS03	Not spooled	ROGERS03	POK1228			Tue Jul 20 08:59:38 EDT

Figure 7-36 JOB04875 has a Printer definition of DFLTNTRY

Error messages issued

The following messages were issued to the log and the common message log.

```
+AOP3201E The DEST, CLASS, and FORMS JCL keywords do not match a printer definition in
the Printer Inventory. - JOB04875 - ROGERS
+AOP3417I IP PrintWay rejected the print job. The print job remains on the JES spool. -
JOB04875 - ROGERS JOB04875
```

Figure 7-37 Messages for JOB04875 in the OPERLOG and common message log

Explanation for AOP3210E: IP PrintWay extended mode cannot find a printer definition in the Printer Inventory that matches the DEST, CLASS, and FORMS JCL parameters for the print job. The job submitter might have specified incorrect DEST, CLASS, and FORMS values on the OUTPUT JCL statement, or the printer definition in the Printer Inventory might be missing or have incorrect values. The **aoplogu** command provides more information in fields before the message text. IP PrintWay extended mode rejected the print job and did not process it. The print job remains on the JES spool with a status of “rejected” so that IP PrintWay extended mode does not attempt to process the print job again until the operator releases it.

Explanation for AOP3417I: IP PrintWay extended mode cannot process the print job due to an error. The **aoplogu** command provides more information in fields before the message text. The print job remains on the JES spool in the rejected state. Printer messages are shown in Figure 7-38 on page 177.

```
07/20/04 13:09:22 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3204I IP
PrintWay received the document from JES.
07/20/04 13:09:22 (UTC-5) priority:error user:ROGERS job:JOB04875 dsn:ROGERS.ROG
ERSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3201E Th
e DEST, CLASS, and FORMS JCL keywords do not match a printer definition in the P
rinter Inventory.
07/20/04 13:09:22 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3417I IP
PrintWay rejected the print job. The print job remains on the JES spool.
07/20/04 14:08:42 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? program:Infoprint_Central msg:AOP3017I IP PrintWay rejec
ted the request to move the print job to printer lpr://pokip1145b.itso.ibm.com/a
fccu2.
07/20/04 14:09:34 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3204I IP
PrintWay received the document from JES.
07/20/04 14:09:34 (UTC-5) priority:error user:ROGERS job:JOB04875 dsn:ROGERS.ROG
ERSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3201E Th
e DEST, CLASS, and FORMS JCL keywords do not match a printer definition in the P
rinter Inventory.
07/20/04 14:09:34 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3417I IP
PrintWay rejected the print job. The print job remains on the JES spool.
07/20/04 14:12:11 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? program:Infoprint_Central msg:AOP3017I IP PrintWay rejec
ted the request to move the print job to printer lpr://pokip1145b.itso.ibm.com/a
fccu2.
07/20/04 14:14:50 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3204I IP
PrintWay received the document from JES.
07/20/04 14:14:50 (UTC-5) priority:error user:ROGERS job:JOB04875 dsn:ROGERS.ROG
ERSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3201E Th
e DEST, CLASS, and FORMS JCL keywords do not match a printer definition in the P
rinter Inventory.
07/20/04 14:14:50 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3417I IP
PrintWay rejected the print job. The print job remains on the JES spool.
07/20/04 14:15:35 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3204I IP
PrintWay received the document from JES.
07/20/04 14:15:35 (UTC-5) priority:error user:ROGERS job:JOB04875 dsn:ROGERS.ROG
ERSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3201E Th
e DEST, CLASS, and FORMS JCL keywords do not match a printer definition in the P
rinter Inventory.
07/20/04 14:15:35 (UTC-5) priority:info user:ROGERS job:JOB04875 dsn:ROGERS.ROGE
RSW.JOB04875.D000000A.? job_selection_rule:SAPI program:aopwsmd msg:AOP3417I IP
PrintWay rejected the print job. The print job remains on the JES spool.
```

Figure 7-38 Printer messages for JOB4875

7.14 Common message log

In previous releases, Infoprint Server components such as Print Interface, IP PrintWay, and NetSpool wrote messages to different locations. In z/OSV1R5, Infoprint Server components write messages to a common message log. This makes it easier to find messages that might be related to each other. The common message log lets you see messages from most

Infoprint Server components in one place. The log contains messages from all components of Infoprint Server except for IP PrintWay basic mode. It does not contain messages from Infoprint Server Transforms or other transform products. By default, no messages are kept in the common message log.

> **Note:** Only IP PrintWay basic mode and Infoprint Server Transforms do not write messages to the common message log.

IP PrintWay extended mode writes its messages only to the common message log. Other components, such as NetSpool and Print Interface, write their messages to other locations such as the NetSpool message-log data set and the system console log, as well as to the common message log.

Infoprint Central lets authorized users view messages in the common message log for selected print jobs and IP PrintWay extended mode printers. In addition, Infoprint Server administrators can use the **aoplogu** command to select messages in a particular time range and copy them to a file or view them on the terminal. Enter this command from the z/OS UNIX (OMVS) command line in the following ways:

- To specify the begin and end time:

  ```
  aoplogu [-b time] [-e time]
  ```

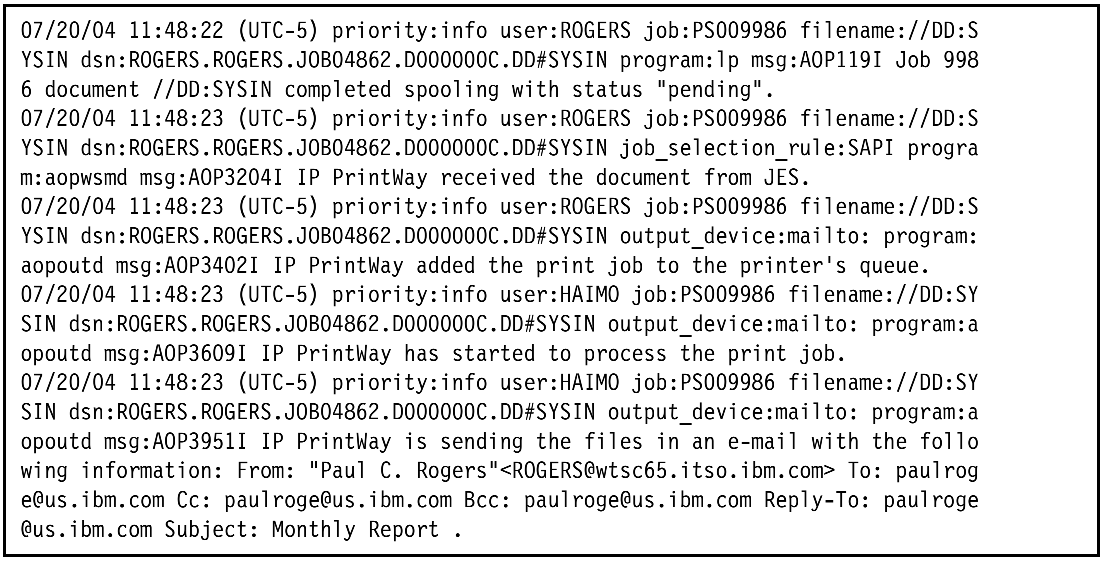
- To specify the most recent messages for the specified time:

  ```
  aoplogu -l time -
  ```

> **Note:** To use the **aoplogu** command, you must be defined to RACF as a z/OS UNIX user and be connected to the RACF AOPADMIN group.

7.14.1 Common message log messages

Figure 7-39 shows some examples of the common message log messages received when you issue the **aoplogu** command.



```
07/20/04 11:48:22 (UTC-5) priority:info user:ROGERS job:PS009986 filename://DD:S
YSIN dsn:ROGERS.ROGERS.JOB04862.D000000C.DD#SYSIN program:lp msg:AOP119I Job 998
6 document //DD:SYSIN completed spooling with status "pending".
07/20/04 11:48:23 (UTC-5) priority:info user:ROGERS job:PS009986 filename://DD:S
YSIN dsn:ROGERS.ROGERS.JOB04862.D000000C.DD#SYSIN job_selection_rule:SAPI progra
m:aopwsmd msg:AOP3204I IP PrintWay received the document from JES.
07/20/04 11:48:23 (UTC-5) priority:info user:ROGERS job:PS009986 filename://DD:S
YSIN dsn:ROGERS.ROGERS.JOB04862.D000000C.DD#SYSIN output_device:mailto: program:
aopoutd msg:AOP3402I IP PrintWay added the print job to the printer's queue.
07/20/04 11:48:23 (UTC-5) priority:info user:HAIMO job:PS009986 filename://DD:SY
SIN dsn:ROGERS.ROGERS.JOB04862.D000000C.DD#SYSIN output_device:mailto: program:a
opoutd msg:AOP3609I IP PrintWay has started to process the print job.
07/20/04 11:48:23 (UTC-5) priority:info user:HAIMO job:PS009986 filename://DD:SY
SIN dsn:ROGERS.ROGERS.JOB04862.D000000C.DD#SYSIN output_device:mailto: program:a
opoutd msg:AOP3951I IP PrintWay is sending the files in an e-mail with the follo
wing information: From: "Paul C. Rogers"<ROGERS@wtsc65.itso.ibm.com> To: paulrog
e@us.ibm.com Cc: paulroge@us.ibm.com Bcc: paulroge@us.ibm.com Reply-To: paulroge
@us.ibm.com Subject: Monthly Report .
```

Figure 7-39 **aoplogu** *command messages*

Note: If you use Infoprint Central to view messages in the common message log, set the TZ and either the LC_TIME or LC_ALL environment variables. These variables affect the date and time displayed in messages.

7.14.2 Historical Inventory

The Historical Inventory contains information about data sets that Infoprint Server has processed but that are no longer on the JES spool because they finished processing or were deleted. Infoprint Central lets authorized users display information about data sets (called print jobs) in the Historical Inventory.

Definition in AOPD.CONF

To customize the common message log and the Historical Inventory, specify these attributes in the aopd.conf configuration file, as shown in Figure 7-3 on page 140. The definitions are as follows:

log-retention The number of days worth of messages to keep in the common message log and information about output data sets that are no longer on the JES spool to keep in the Historical Inventory. You can specify a value from 0 to 59. A value of 0 means that no messages are kept in the common message log, and no information is kept in the Historical Inventory.

```
max-historical-inventory=size=1
```

Note: Because the common message log and Historical Inventory can contain a large amount of data, start with a value of 1 (day). Increase the value by 1 (day) if users request to see more messages or historical information.

7.15 E-mail printers

Infoprint Server e-mail support was added shortly after the z/OS V1R2 release. To customize this support requires the following changes:

- One or more printer definitions in the Printer Inventory.
- When the e-mail protocol is selected in a printer definition, IP PrintWay uses z/OS UNIX sendmail to prepare and send e-mails to the recipients listed in the printer definition. Sendmail is a mail transfer agent provided with z/OS Communications Server that provides enhanced SMTP support. Sendmail version 8.12.1 runs on z/OS 1.5.

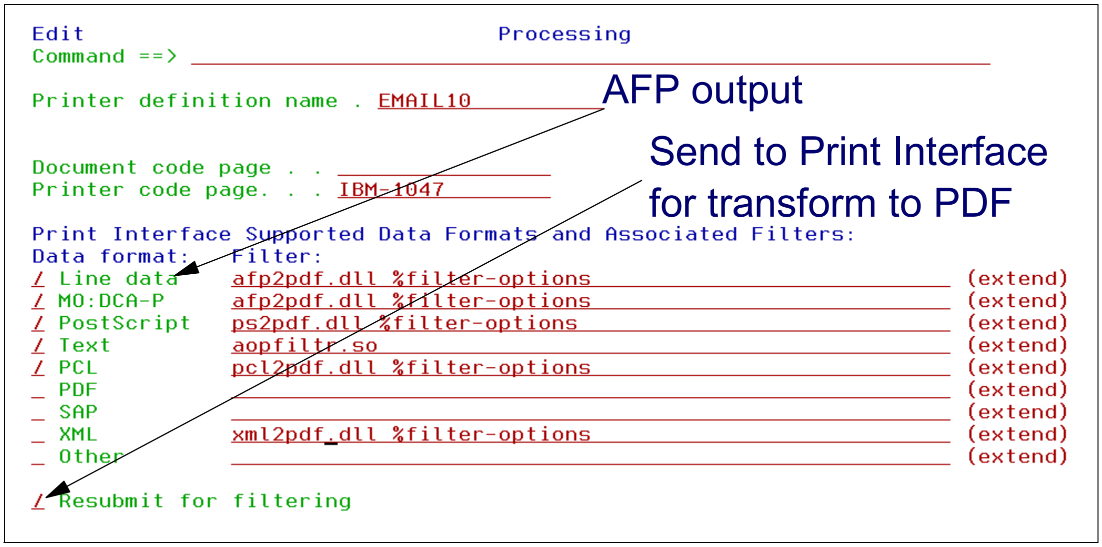
  In the sendmail aliases file, you can create alias names to represent a list of real e-mail addresses. When you create an alias, you should specify the user ID that owns the list in the owner-alias statement. The user ID that owns the list receives notification about bounced e-mails. If you do not specify the user ID that owns the alias name, sendmail sends notification of bounced e-mails to the user ID assigned to the IP PrintWay startup procedure (basic mode) or to the user ID who started the Infoprint Server daemons (extended mode).

7.15.1 Printer Inventory definitions

You define a printer as an e-mail printer just like a normal printer. In the protocol panel for an e-mail printer, you specify the e-mail addresses this printer uses to send the file (preferably a PDF) to the specified users.

Processing panel

On the Processing panel for the e-mail defined printer, make sure that all potential data streams are transformed into a PDF stream.



```
Edit                                Processing
Command ==> ____________________________________________________________

Printer definition name . EMAIL10


Document code page . . ________________
Printer code page. . . IBM-1047________

Print Interface Supported Data Formats and Associated Filters:
Data format:    Filter:
/ Line data     afp2pdf.dll %filter-options_______________________  (extend)
/ MO:DCA-P      afp2pdf.dll %filter-options_______________________  (extend)
/ PostScript    ps2pdf.dll %filter-options________________________  (extend)
/ Text          aopfiltr.so_______________________________________  (extend)
/ PCL           pcl2pdf.dll %filter-options_______________________  (extend)
_ PDF           __________________________________________________  (extend)
_ SAP           __________________________________________________  (extend)
_ XML           xml2pdf_.dll %filter-options______________________  (extend)
_ Other         __________________________________________________  (extend)

/ Resubmit for filtering
```

AFP output

Send to Print Interface for transform to PDF

Figure 7-40 Processing panel for e-mail printer EMAIL10

Other Processing panel options

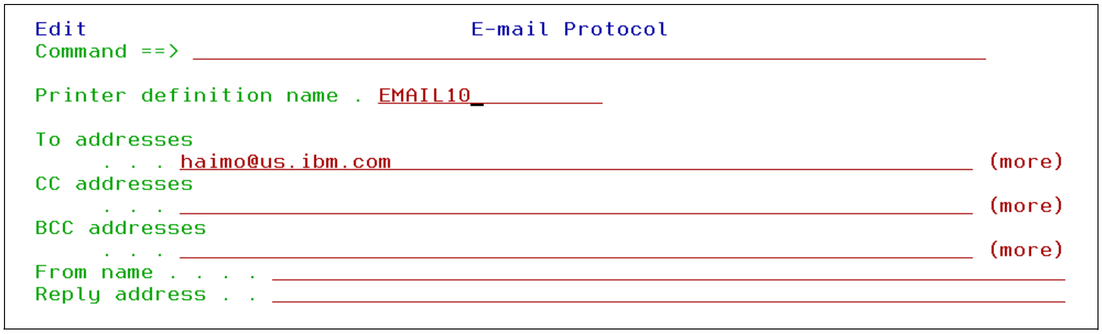
The following panel options are necessary:

- %filter-options causes options that are specified in the filter-options job attribute (specified, for example, on the `lp` command) to be passed to the transform.
- The Resubmit for filtering single-valued attribute indicates whether a filter in the filters attribute is to be used for data sets submitted as batch jobs to IP PrintWay basic mode. When resubmit-for-filtering=yes is specified, IP PrintWay resubmits batch data sets to Print Interface. Print Interface calls the filter (if any) associated with the input data format and then writes the data to a new output data set on the JES spool for subsequent processing by IP PrintWay.

> **Note:** The Resubmit for filtering option is not required with IP PrintWay extended mode defined printers.

Protocol panel

The ISPF panel that you use to fill in the Protocol section of an IP PrintWay printer definition is different for each transmission protocol. You can select one of these transmission protocols when you add a printer definition: LPR, direct sockets, Internet Printing Protocol (IPP), VTAM, or e-mail. The e-mail protocol panel for the EMAIL10 printer is shown in Figure 7-41 on page 181.



```
Edit                              E-mail Protocol
Command ==> ____________________________________________________________

Printer definition name . EMAIL10_________

To addresses
     . . . haimo@us.ibm.com___________________________________________ (more)
CC addresses
     . . . ___________________________________________________________ (more)
BCC addresses
     . . . ___________________________________________________________ (more)
From name . . . . ____________________________________________________________
Reply address . . ____________________________________________________________
```

Figure 7-41 E-mail Protocol panel to define printer EMAIL10

7.15.2 E-mail support before z/OS V1R5

In the following example, a batch job is used to send an AFP™ data set to the e-mail address specified for printer EMAIL10 in the Protocol panel. The following JCL example is used. The output is in SYSOUT class J, which is selected by an FSA writer in the IP PrintWay FSS address space.

```
//ROGERSZ   JOB   (POK,999),MSGCLASS=A,NOTIFY=ROGERS
//PRINT     EXEC PGM=IEBGENER
//SYSPRINT  DD  SYSOUT=A
//SYSUT2    DD  SYSOUT=J,DEST=EMAIL10
//SYSUT1    DD  DISP=SHR,DSN=ROGERS.SAPI.LIST3820
//SYSIN     DD  DUMMY
//
```

Figure 7-42 JCL batch job submitted to send the output data set to an e-mail address

Figure 7-43 on page 182 shows the flow of the output being delivered to the e-mail address, as follows:

1. The batch job is submitted from TSO/E. After executing, the output is now on the JES spool.
2. The IP PrintWay FSS address space selects the AFP output from the JES spool.
3. The Printer Inventory is checked for EMAIL10 printer and the Resubmit for filtering option is detected.
4. The output data set is sent back to the Print Interface for a data stream transform.
5. The Print Interface sends the data set to the Transform Manager for conversion from AFP to PDF.
6. The PDF data set is placed back on the JES spool.

> **Note:** The second sysout data set, which Print Interface allocates on the JES spool, contains the same job name, job ID, and last qualifier of the data set name as the original sysout data set on the JES spool. Therefore, the operator can use these values to find the job submitter's data set on the JES spool.

7. The IP PrintWay FSS address space selects the PDF output from the JES spool.
8. IP PrintWay transmits the PDF via TCP/IP to the e-mail address specified in the Protocol panel.

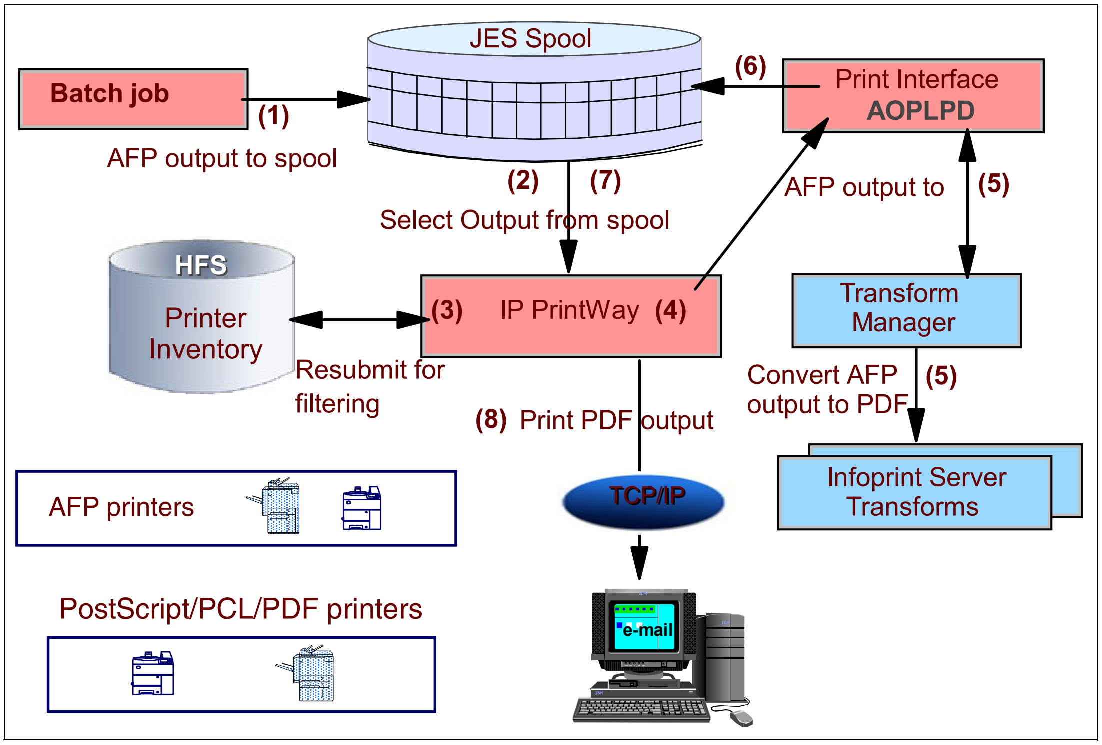


Figure 7-43 Flow of a batch job submission output to an e-mail address

Note: IP PrintWay extended mode transforms data without resubmitting data sets to Print Interface, so the resubmit for filtering option does not apply. IP PrintWay extended mode ignores the Resubmit for filtering field if it is selected.

7.15.3 E-mail enhancements in z/OS V1R5

Infoprint Server lets you send output directly to one or more e-mail addresses instead of printing the output. This support was added on z/OS Version 1 Release 2 of Infoprint Server. In previous releases, you specified the e-mail addresses in a printer definition. In the z/OS V1R5 release, you can now also specify the e-mail addresses in the OUTPUT JCL statement with new parameters. In addition, you can specify these e-mail items in the printer definition in the Printer Inventory, as shown in Figure 7-41 on page 181. The following specifications are possible:

- E-mail addresses of the recipients of the e-mail
- File name of the attachment to the e-mail
- Descriptive name or other identifier of the sender of the e-mail
- Blind copy (bcc) recipients of the e-mail
- Copy (cc) recipients of the e-mail
- E-mail address that recipients of the e-mail can reply to

7.15.4 JCL OUTPUT keywords for Infoprint Server e-mail support

JCL has been enhanced to add JCL OUTPUT keywords for Infoprint Server e-mail support. Also, the PRMODE keyword was added to the PRINTDEV JCL statement.

The new JCL OUTPUT keywords are added to allow end users to specify parameters to be used by Infoprint Server when the user wants to send the output as an e-mail attachment rather than printing it. The keywords added are described in the following section.

MAILTO Use the MAILTO parameter to specify one or more e-mail address of the recipients of an e-mail. The syntax is:

```
MAILTO= {('to address'[,'to address']...)}
        {to-address                      }
```

to address Specifies the e-mail addresses of the recipients. You can code up to 32 addresses. Each address can be 1 to 60 EBCDIC text characters. There is no default for MAILTO.

Example: `//OUTDS2   OUTPUT  MAILTO=('sahoo@cbc.com','bob@abc.com')`

MAILFROM Use the MAILFROM parameter to specify the descriptive name or other identifier of the sender of an e-mail. The syntax is:

```
MAILFROM = {'from address'}
           {from-address  }
```

from address Specifies descriptive name or other identifier of the sender of an e-mail. The from address can be 1 to 60 EBCDIC text characters.

There is no default for MAILFROM. However, Infoprint Server always includes userid@domainname to identify the sender. The user ID is the TSO user ID of the job submitter and domainname is the domain name where Infoprint Server is running.

Example: `//OUTDS2   OUTPUT  MAILFROM='sahoo@abc.com'`

MAILCC Use the MAILCC parameter to specify one or more e-mail addresses of the copy (cc) recipients of an e-mail. A cc means that other recipients of the e-mail can see the cc recipient listed. The syntax is:

```
MAILCC= {('cc address'[,'cc address']...)}
        {cc-address                      }
```

cc address Specifies the e-mail addresses of the copy (cc) recipients of an e-mail. You can code up to 32 cc addresses. Each address can be 1 to 60 EBCDIC text characters. There is no default for MAILCC.

Example: `//OUTDS2   OUTPUT  MAILCC=('rob@abc.com', ‘smith@abc.com')`

MAILBCC Use the MAILBCC parameter to specify one or more e-mail addresses of the blind copy (bcc) recipients of an e-mail. A bcc means that other recipients of the e-mail do not see the bcc recipient listed. The syntax is:

```
MAILBCC= {('bcc address'[,'bcc address']...)}
         {bcc-address                       }
```

bcc address Specifies the e-mail addresses of the blind copy (bcc) recipients of an e-mail. You can code up to 32 bcc-addresses. Each address can be 1 to 60 EBCDIC text characters. There is no default for MAILBCC.

Example: `//OUTDS2   OUTPUT  MAILBCC=('rob@abc.com','smith@abc.com')`

MAILFILE Use the MAILFILE parameter to specify the file name of the attachment to an e-mail. The syntax is:

```
MAILFILE= {'file id'}
          {file-id  }
```

file id Specifies the name of a file attached in an e-mail. The file id can be 1 to 60 EBCDIC text characters. As default, Infoprint Server uses the last qualifier of the data set name as the name of the e-mail attachment.

Example: `//OUTDS2   OUTPUT  MAILFILE='redbook chapter 1'`

REPLYTO Use the REPLYTO parameter to specify the e-mail address to which recipients of the e-mail can respond. The syntax is:

```
REPLYTO = {'reply address'}
          {reply-address  }
```

reply address Specifies the e-mail address to which recipients of the e-mail can respond. The reply address can be 1 to 60 EBCDIC text characters. There is no default for REPLYTO.

Example: `//OUTDS2   OUTPUT  REPLYTO='sahoo@abc.com'`

7.16 NetSpool enhancements

The NetSpool component of Infoprint Server intercepts print data from VTAM applications, such as CICS and IMS; transforms the data streams to EBCDIC line data, PCL, PDF, or other formats that the printer accepts; and creates output data sets on the JES2 or JES3 spool. You can configure NetSpool so that you do not need to change existing VTAM applications. That is, existing VTAM applications can send print requests to NetSpool in the same manner as they currently send print requests to SNA network printers.

VTAM applications, such as CICS or IMS, establish communication sessions with NetSpool printer logical units (LUs) instead of with SNA-network printers. Each NetSpool printer LU must be defined to VTAM as an application logical-unit (LU). NetSpool runs as a VTAM application on the same or different z/OS system. NetSpool can process VTAM print requests sent to different NetSpool printer LUs.

NetSpool can process these types of VTAM data streams:

- SNA character string (SCS) data over an LU type 1 session
- 3270 data over an LU type 3 or LU type 0 session
- A binary data stream over an LU type 0, type 1, or type 3 session

7.16.1 Starting the NetSpool daemon

To start the NetSpool daemon **aopnetd** shown in Figure 7-44 on page 185, with the **`aopstart`** command, add `netd` to the values in the `start-daemons` statement in the aopd.conf configuration file. By default, only the Printer Inventory Manager and the LPD start. Therefore, this statement is required to start the NetSpool daemon. The daemon **aopnetd** controls part of the NetSpool processing. You can only run one NetSpool daemon, but it can control several NetSpool started tasks.

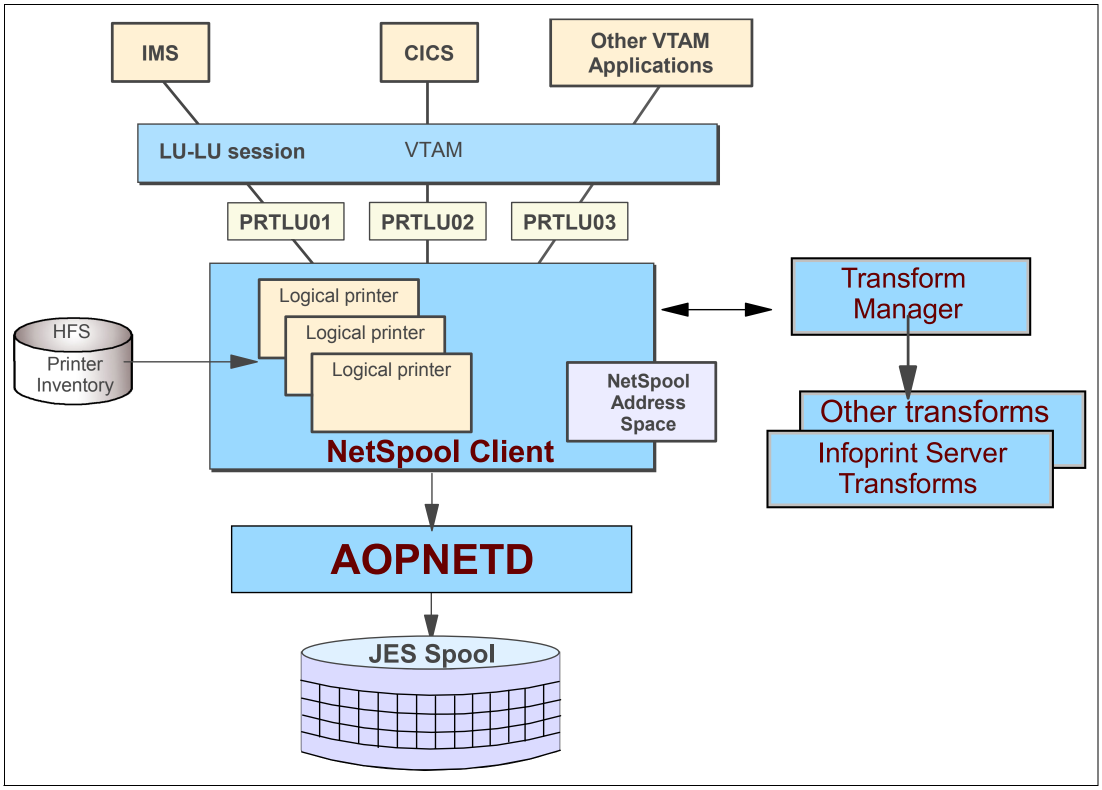


Figure 7-44 NetSpool overview

Starting NetSpool

If you use the NetSpool startup procedure with a member named APIJPJCL in the SYS1.PROCLIB library, enter the following command:

```
START APIJPJCL
```

Note: To stop NetSpool processing, you do not need to stop the NetSpool daemon (aopnetd). When you stop the NetSpool task, NetSpool processing ends. You can restart the NetSpool task without restarting the NetSpool daemon.

NetSpool printer LUs

When you start NetSpool, the NetSpool printer LUs that are assigned to the LU classes specified on the EXEC statement in the NetSpool startup procedure are started. After NetSpool is started, you might want to start an LU that is assigned to a NetSpool LU class that has not been started, or to restart an LU that you stopped. To do this, you can use one of the following methods:

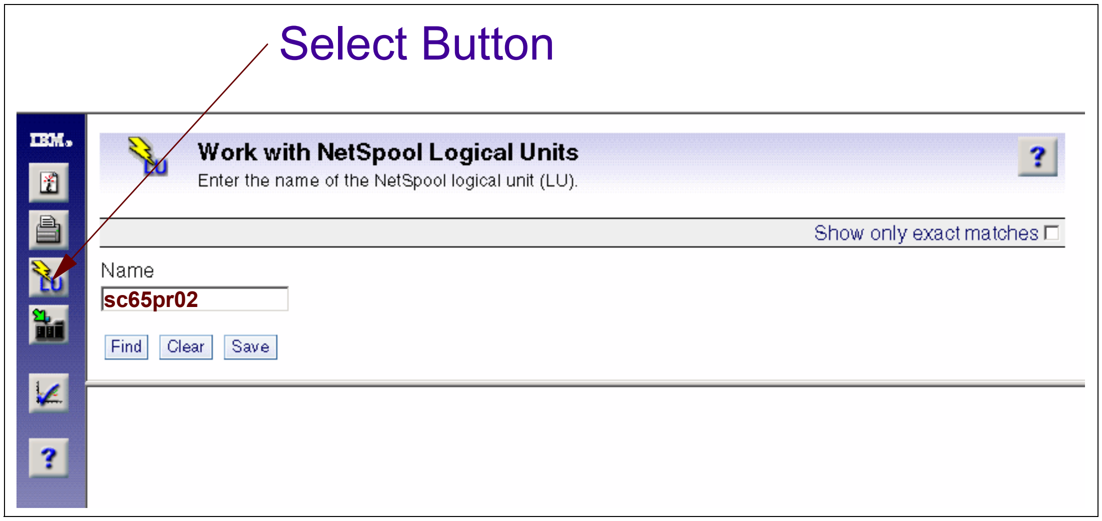
- Infoprint Central panels
- VTAM VARY ACT and LUNAME ADD commands

If you specify a printer LU name in a printer definition after NetSpool is started, NetSpool automatically tries to start the LU if it is assigned to one of the LU classes that NetSpool has already started. This means that you do not need to start the LU with the NetSpool ADD command. However, you must still activate it in VTAM using either Infoprint Central or the VARY ACT command.

7.16.2 Using Infoprint Central

The operator can control NetSpool LUs from Infoprint Central, as shown in Figure 7-45, from the system console, or from extended MCS consoles. For example, the operator can display the status of NetSpool LUs, stop them, and start them.

The operator can use tools such as SDSF and Infoprint Central to find output data sets that NetSpool writes to the JES spool. Infoprint Central can display additional status such as whether the output data sets (called print jobs) completed successfully, were retained due to failed transmission to LAN printers, or were deleted before printing.



Work with NetSpool Logical Units

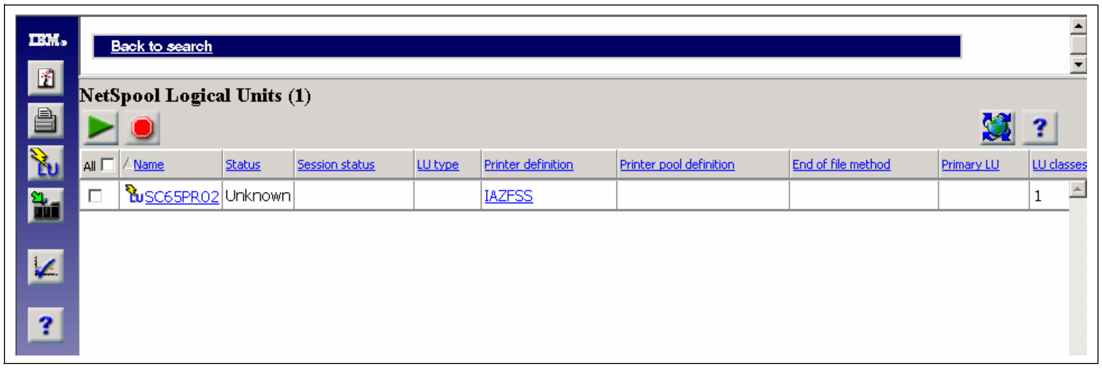
Enter the name of the NetSpool logical unit (LU).

Show only exact matches □

Name: sc65pr02

Figure 7-45 Work with NetSpool LUs with Infoprint Central

After entering the LU name, click the **Find** button to display the LU specified, as shown in Figure 7-46.



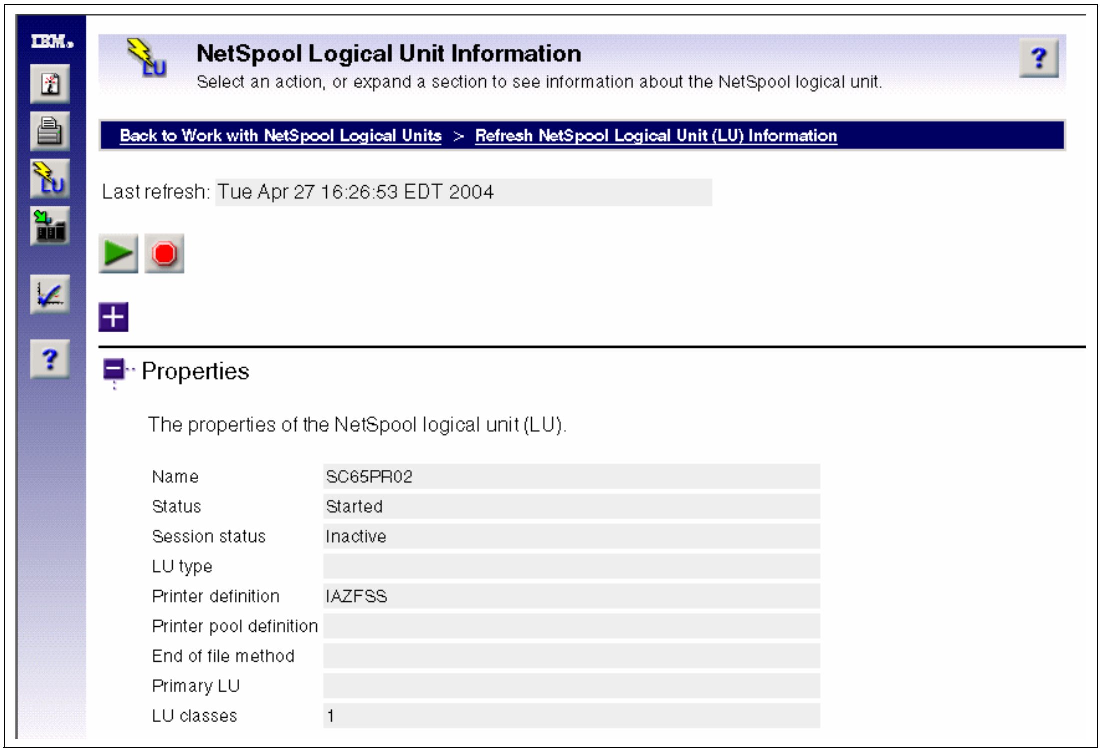
NetSpool Logical Units (1)

All □	Name	Status	Session status	LU type	Printer definition	Printer pool definition	End of file method	Primary LU	LU classes
□	SC65PR02	Unknown			IAZFSS				1

Figure 7-46 NetSpool LU displayed with Infoprint Central

To get the specified definitions for the LU, select the LU name; Figure 7-47 on page 187 is displayed.

Restriction: You cannot use Infoprint Central to start, stop, or display the status of NetSpool LUs if you start more than one NetSpool task. However, you can use Infoprint Central to perform other functions.



NetSpool Logical Unit Information

Select an action, or expand a section to see information about the NetSpool logical unit.

Back to Work with NetSpool Logical Units > Refresh NetSpool Logical Unit (LU) Information

Last refresh: Tue Apr 27 16:26:53 EDT 2004

Properties

The properties of the NetSpool logical unit (LU).

Name	SC65PR02
Status	Started
Session status	Inactive
LU type	
Printer definition	IAZFSS
Printer pool definition	
End of file method	
Primary LU	
LU classes	1

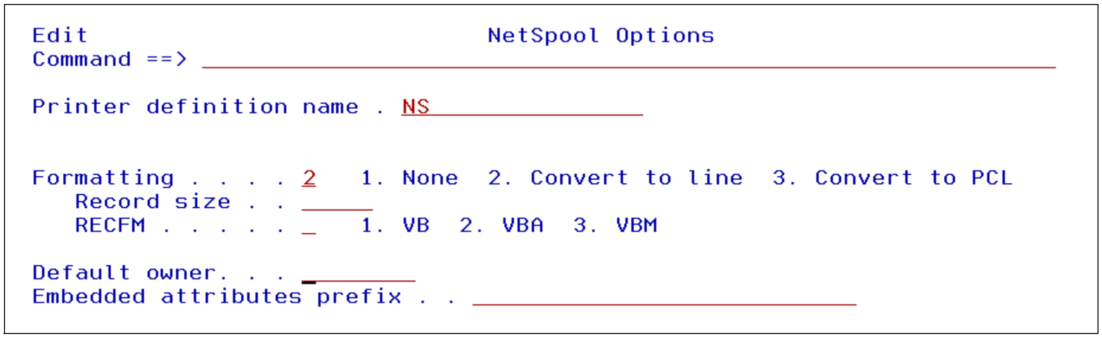
Figure 7-47 NetSpool LU information panel display

7.16.3 NetSpool default owner

A new attribute has been added to the NetSpool-options panel, shown in Figure 7-48: “Default owner” for the associated logical unit. This attribute is used if the print data does not specify an owner. The job owner is used for output data sets created for this NetSpool LU. If this field is blank, the default job owner is the user ID of the NetSpool daemon AOPNETD.

The Infoprint Server job owner helps you find jobs when you are using z/OS Infoprint Central. The JES job owner is always the user ID of the NetSpool task. The job owner in this field is also used as the JES job name if no other owner or job name is specified in the print data.

Note: This Default owner can be printed on jobs’ separator pages.



```
Edit                                NetSpool Options
Command ==> ____________________________________________________________

Printer definition name . NS_________________


Formatting . . . . 2    1. None   2. Convert to line   3. Convert to PCL
   Record size . . _____
   RECFM . . . . . _    1. VB   2. VBA   3. VBM

Default owner. . . ________
Embedded attributes prefix . . ___________________________
```

Figure 7-48 NetSpool Options panel with new field Default owner

7.16.4 NetSpool and MVS commands

To control NetSpool printer LUs, you can use the MVS **MODIFY (F)** command to direct NetSpool commands to NetSpool. You can also use the MVS **STOP** command to stop a NetSpool started task.

To use the NetSpool modify command with the DISPLAY option, issue the following type commands:

- Specific LU by LUNAME:

```
F NETSPOOL, DISPLAY LUNAME=SC65PR02
```

- All LUs that are started:

```
F NETSPOOL, DISPLAY STARTED
```

- All LUs that are selected for processing:

```
F NETSPOOL,DISPLAY SELECTED
API1008I Display of SELECTED LUs.
   API1002I LUPRT002 -- LU IS STARTED SESSION=ACTIVE
            PRINTERNAME=MYPRINTER LUTYPE=1
            PLU=IMS001 EOFRULE=EB.
   API1050I LUPRT000 -- LU IS PENDING CLOSE.
   API1002I LUPRT001 --  LU IS STARTED SESSION=INACTIVE
            PRINTERNAME=YOURPRINTER LUTYPE=0
            PLU=IMS002 EOFRULE=TIMER.
   API1003I LUPRT003 --  LU IS WAITING.
```

Note: When you use the **DISPLAY** command with LUNAME, SELECTED, or STARTED, you see additional information with z/OS V1R5 that can help you diagnose problems with NetSpool LUs, VTAM definitions, and VTAM application programs. The new fields in the messages are PRINTERNAME=, SESSION =, LUTYPE=, POOLNAME=, PLU=, and EOFRULE=

8

z/OS V1R5 ISPF enhancements

This chapter describes the changes made to ISPF for z/OS V1R5. The enhancements are aimed at improving end-user productivity. The set of functions available to dialog writers is also broadened.

This chapter covers the following topics:

- Scrollable fields on ISPF panels
- Miscellaneous changes to the ISPF panel processing
- Catalog name in data set list
- MOVE/COPY alias support
- PDS/PDSE member delete by pattern
- PDS/PDSE member list enhancements
- Additional command tables
- System symbolics in temporary data set names
- Configure min and max scroll amounts
- Configuration table identification
- SuperC command output highlighting
- SCLM enhancements
- Other miscellaneous ISPF changes

8.1 Scrollable fields on ISPF panels

Before this enhancement, the display and input of data in ISPF panels was limited to the panel field size. With z/OS V1R5, ISPF has been enhanced to support scrollable fields. A scrollable field can be used when the size of the field defined on the panel is smaller than the amount of data to be displayed or entered. Defining a field as scrollable provides the ability to the end-user to display and input a variable larger than the display area that the dialog variable occupies.

To support this enhancement:

- A new **)FIELD** panel definition section has been added, providing support for application developers to define panel fields as scrollable.
- A new **EXPAND** command is introduced, and LEFT and RIGHT commands are changed to display the scrollable fields.
- Dialog Tag Language (DTL) includes new tags which support the definition of scrollable fields.

Note: The support for scrollable fields has been rolled down to z/OS 1.2, 1.3, and 1.4 via APR OW57368. However, this is limited to enhancing ISPF panel services to add new keywords and associated logic to define, display, and manipulate scrollable fields. This does not include the enhancements to DTL to support scrollable fields. Documentation for this new support is delivered in SISPSAMP member ISPSCRFL. It can be downloaded and viewed with Adobe Acrobat (PDF).

8.1.1 Defining scrollable fields in ISPF panel services

The)FIELD section of a panel definition specifies what fields, if any, are scrollable fields. Defining a field as scrollable provides the ability to display and input a variable larger than the display area that the dialog variable occupies. The LEFT, RIGHT, and EXPAND primary commands are active when the cursor is positioned within the variable on the display panel. This enables left and right scrolling and expansion of the variable into a pop-up panel. The)FIELD section and its parameters are shown in Figure 8-1.

```
)FIELD
     FIELD(field-name) ........... Identify scrollable panel field
     [LEN(value|field-name)] ...... Length of displayed variable
     [IND(field-name,value)] ...... Left & right scroll indicator
     [LIND(field-name,value)] ..... Left scroll indicator
     [RIND(field-name,value)] ..... Right scroll indicator
     [SIND(field-name,value)] ..... Separator scroll indicator
     [LCOL(field-name)] .......... Left column position indicator
     [RCOL(field-name)] .......... Right column position indicator
     [SCALE(field-name)] ......... Scale indicator
     [SCROLL(value|field-name)] ... Scroll control switch
```

Figure 8-1)FIELD section for scrollable fields

The following are brief descriptions of parameters of)FIELD section:

FIELD(*field-name*) The name of the panel field that is scrollable.

LEN(*value* / *field-name*) Length of the displayed variable.

value: Specify a value between 1 and 32 767.

field-name: The length dialog variable can be used to specify an initial length if it contains a value between 1 and 32 767. After the display, this variable contains the calculated display length.

Calculated display length: The length of the variable is the maximum value of the default display variable length and the specified length.

Default: If the LEN parameter is not specified, the field defaults to the length of the dialog variable, if it exists. For variables referenced in a)MODEL section, the dialog variable length is the maximum of all instances on the current display for that variable.

IND(*field-name,value*)

Left and right scroll indicator dialog variable.

field-name: This must refer to a 2 byte scroll indicator dialog variable that is updated on the panel to indicate whether left and/or right scrolling can be performed.

value: (Default -+) Specify a 2 byte literal (enclosed in quotes) to override the default scroll indicator values. Each byte must be nonblank.

Displays as:

-+ Indicates that you can scroll left and right

- Indicates that you can only scroll left

+ Indicates that you can only scroll right

LIND(*field-name,value*)

Left scroll indicator dialog variable.

field-name: This must refer to a 1 byte left scroll indicator dialog variable that is updated on the panel to indicate whether left scrolling can be performed.

value: (Default "-") Specify a 1 byte nonblank literal (enclosed in quotes) to override the default left indicator value.

Displays as:

value: Indicates that you can scroll left

blank: Indicates you are positioned at the start of the field

RIND(*field-name,value*)

Right scroll indicator dialog variable.

field-name: This must refer to a 1 byte right scroll indicator dialog variable that is updated on the panel to indicate whether right scrolling can be performed.

value: (Default "+") Specify a 1 byte nonblank literal (enclosed in quotes) to override the default right indicator value.

Displays as:

value: Indicates that you can scroll right

blank: Indicates you are positioned at the end of the field

SIND(*field-name,value*)

Separator scroll indicator dialog variable. This field is initialized with the value repeated for the length of the field on the panel. If the field is scrollable to the left, the leftmost byte is the value of the left indicator (default "-"). If the field is right scrollable, the rightmost byte is the value of the right indicator (default "+").

field-name: This must refer to a 3 byte scroll indicator dialog variable that is updated on the panel to indicate whether left and/or right scrolling can be performed.

value: (Default "<->") Specify a 3 byte literal (enclosed in quotes) to override the default separator indicator values. The 3 bytes represent the left scroll indicator, the separator value and the right scroll indicator respectively. Each byte must be nonblank.

LCOL(*field-name*) Left column dialog variable - to display current left position.

field-name: This must refer to a dialog variable that is updated when the field is scrolled to contain the left column value. You can use this to specify an initial left column position for the scrollable field. It must be a numeric value greater than or equal to 1. Values greater than the maximum left column position are set to the maximum left column position.

Note: Fields with the same left column dialog variable will scroll simultaneously and will have the same left column value up to the maximum for each field.

RCOL(*field-name*) Right column dialog variable - to display current right position.

field-name: This must refer to a dialog variable that is updated when the field is scrolled to contain the right column value. It is an output field only. Any pre-existing values are ignored and are replaced with the current right column value.

SCALE(*field-name*) Scale indicator dialog variable. This field is initialized with a scale line reflecting the current columns within the field being displayed. The variable occupies the display length on the panel with the value as follows:

`----+----1----+----2----+----3`...and so forth.

field-name: This must refer to the dialog variable that is placed on the panel in the position at which the scale line is to be initialized.

SCROLL(*value/field-name*) Scroll control field.

value: OFF - Field is not scrollable

ON - Field is scrollable

field-name: specifies a scroll control dialog variable which you can set to a value of OFF to turn scrolling off from the application or from the panel.

Default: If the SCROLL parameter is not specified, the default for the scroll control is ON.

Note: For details of FIELD section description, refer to chapter 7 in *z/OS V1R5.0 ISPF Dialog Developer's Guide and Reference*, SC34-4821.

Primary commands for scrollable fields

You can move to the left or right or expand the scrollable fields using function keys or primary commands. The following commands can be used when the cursor is placed within a scrollable field:

LEFT Scroll left the specified scroll amount.

RIGHT Scroll right the specified scroll amount.

EXPAND Display the variable in a dynamic area in a pop-up expand window. If the scrollable field is defined for input, then you are able to update the variable in the expand window.

The expand panel displays the variable in a scrollable dynamic area. Standard up and down scrolling is supported. You can display the variable in character and hexadecimal using the following primary command.

HEX ON/OFF - Turn hexadecimal display on and off. When set, the setting is remembered for subsequent expand processing.

If a scroll field is found on the current panel, then the scroll amount is honored for up and down scrolling, where:

PAGE Page is the equivalent of the length of the display field.

DATA Data is the equivalent of the length of the display field minus 1.

HALF Half is half the length of the display field.

CSR CSR will scroll relative to the cursor position.

You can enter **M** in the command line to scroll the maximum distance in the left or right direction. The maximum right position is the field length minus the display length. The maximum left position is 1. You can also enter a number in the command line to specify the number of characters to scroll to the left or right.

Example of ISPF panel services with scrollable fields

This example illustrates the scrollable field, the left and right scroll indicators, the left and right column of the data field in the panel, scrollable field length, scale and the scroll indicator. This example further illustrates:

- The use of LEFT, RIGHT, and EXPAND commands to manipulate the data in the scrollable field.
- Use of EXPAND command to display the pop-up panel to manipulate the entire length of data in EBCDIC or HEX.

Figure 8-2 on page 194, shows the source of an ISPF panel service. The Value field has been defined as scrollable.

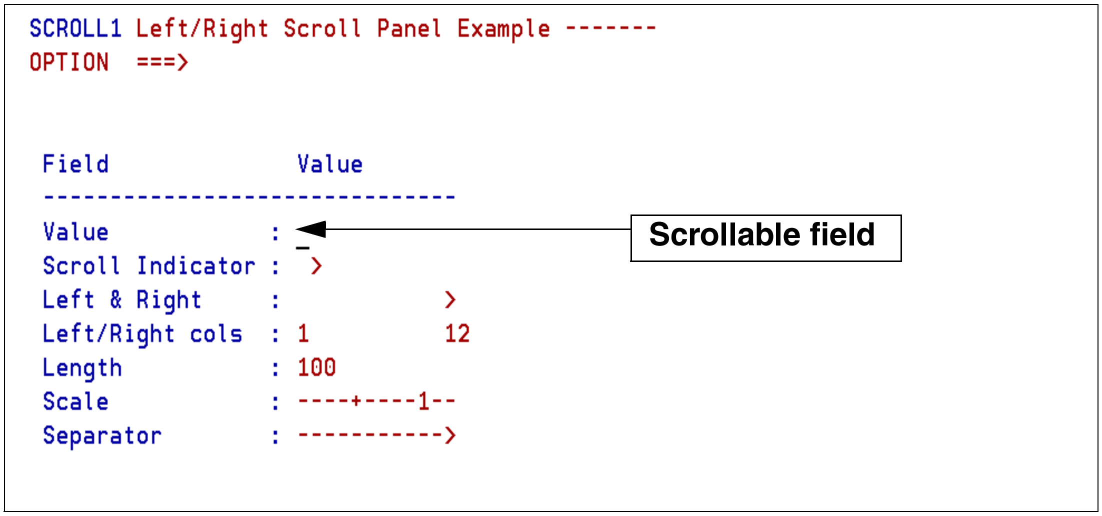
.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR
 | TYPE(OUTPUT) CAPS(OFF) JUST(ASIS )
 _ TYPE(INPUT)  CAPS(OFF) JUST(ASIS ) FORMAT(MIX)
)BODY
%------- Left/Right Scroll Panel Example -------
%OPTION  ===>_ZCMD
%
+
+ Field              Value
+ -------------------------------
+ Value            :_LR1         +
+ Scroll Indicator :|lr1in
+ Left & Right     :|lril      |lrir
+ Left/Right cols  :_Lr1lf     _lr1ri
+ Length           :_lr1ln
+ Scale            :|lr1sc       +
+ Separator        :|lrisp       +
)INIT
  .CURSOR = ZCMD
)FIELD
 Field(LR1)
 Len(100)
 Ind(lr1in,'<>')
 Lind(lril,'<') Rind(lrir,'>')
 Lcol(lr1lf) Rcol(lr1ri)
 Len(lr1ln)
 Scale(lr1sc)
 Sind(lrisp)
)END
```

Figure 8-2 Panel source with scrollable fields

Figure 8-3 displays an initial panel, generated from the ISPF panel source defined in Figure 8-2.

.



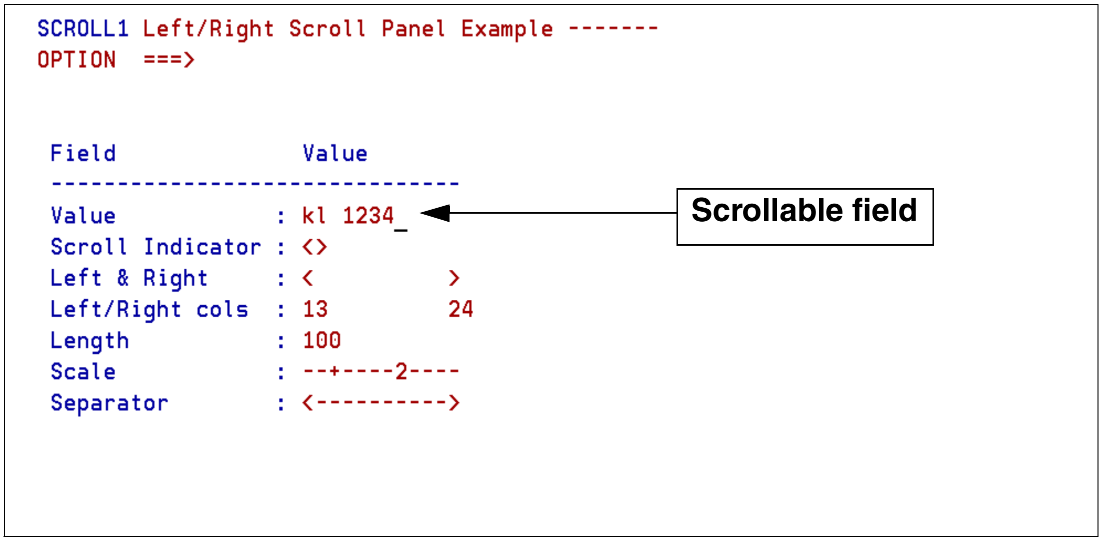
```
SCROLL1 Left/Right Scroll Panel Example -------
OPTION  ===>


 Field              Value
 -------------------------------
 Value            : _
 Scroll Indicator :  >
 Left & Right     :           >
 Left/Right cols  : 1         12
 Length           : 100
 Scale            : ----+----1--
 Separator        : ----------->
```

Figure 8-3 SCROLL1 panel 1 showing scrollable Value field

In this panel, the cursor is in col1 of the Value field and only the right scroll indicator (">") is displayed. Left Col. shows 1 and right Col. shows 12. This is because we have defined panel field size for the Value field as 12. However, we have defined the length of the Value field as 100. In this panel, ASIS, we can input, change, or display data from col1 to col12 of the Value field without scrolling left or right.

In the initial panel, We enter some data in the Value field and then press the pf11(RIGHT) key to scroll to the right. Now the values in the panel change as shown in Figure 8-4. In this panel, both the scroll indicators ("<"and ">") are displayed. The left Col. shows 13 and the right Col. shows 24.This means we can input, change, or display data from col13 to col24 of the Value field.

Thus we can move left and right using the LEFT and RIGHT PF keys in the Value field to input or change or display the data.

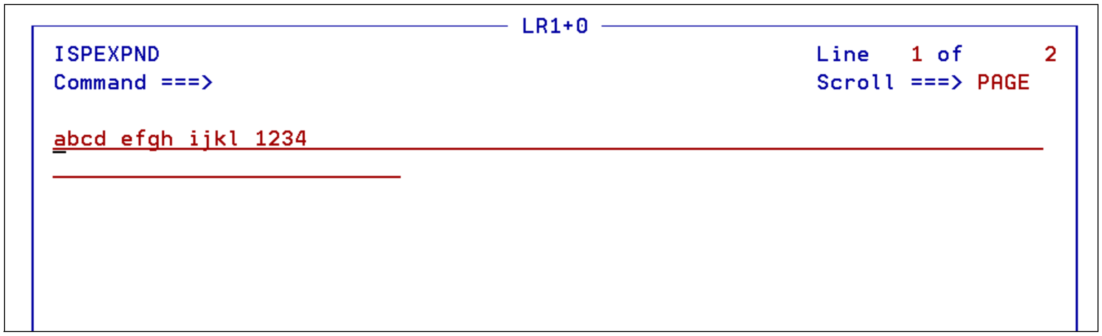


```
SCROLL1 Left/Right Scroll Panel Example -------
OPTION  ===>


 Field              Value
 -------------------------------
 Value            : kl 1234_
 Scroll Indicator : <>
 Left & Right     : <          >
 Left/Right cols  : 13         24
 Length           : 100
 Scale            : --+----2----
 Separator        : <---------->
```

Figure 8-4 SCROLL1 Panel 2 showing Value field as scrollable

Keylist ISRSAB has been assigned for this panel in the panel definition source. When the panel is displayed, we can assign commands to various PF keys in keylist ISRSAB. For example, we assign the EXPAND command to key PF4. Now, keeping the cursor in the Value field, if we press the PF4 key, a pop-up panel ISPEXPND displays the entire scrollable field, in which the cursor is placed. This is shown in Figure 8-5.

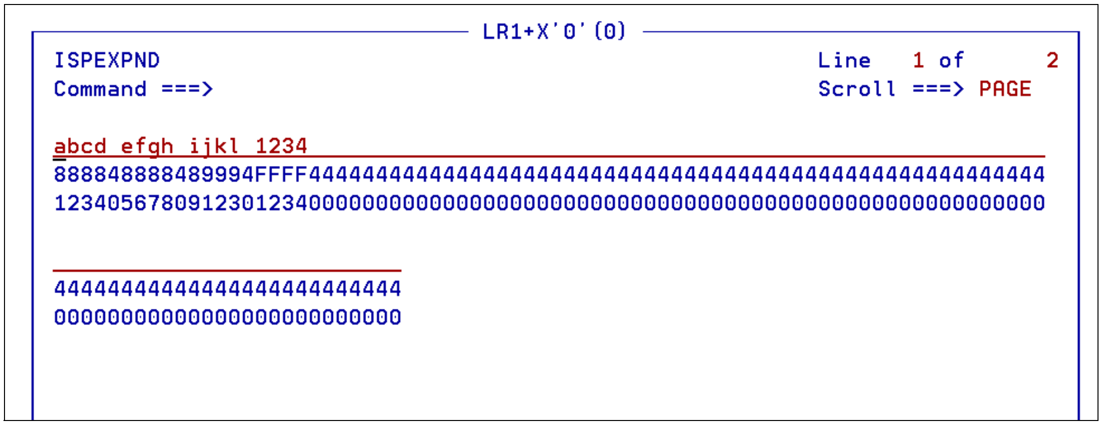
.



```
                                   LR1+0
ISPEXPND                                            Line    1 of      2
Command ===>                                        Scroll ===> PAGE

abcd efgh ijkl 1234
```

Figure 8-5 Pop-up panel showing the complete length of Value field

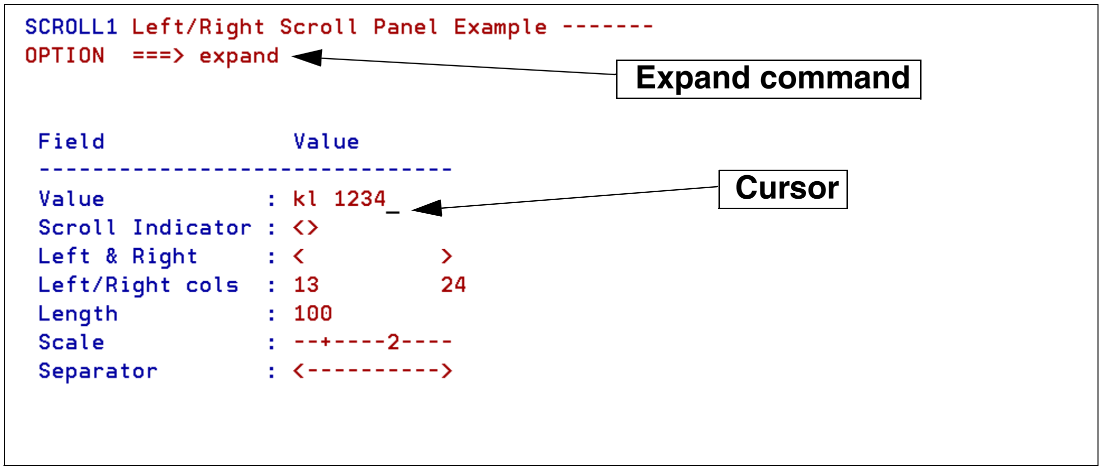
If we want to display the data in HEX, we can enter HEX ON primary command in the pop-up panel shown in Figure 8-5; now the panel shows the HEX data as shown in Figure 8-6.



```
                               LR1+X'0'(0)
ISPEXPND                                                   Line    1 of       2
Command ===>                                               Scroll ===> PAGE

abcd efgh ijkl 1234
888848888489994FFFF44444444444444444444444444444444444444444444444444444444
12340567809123012340000000000000000000000000000000000000000000000000000000

44444444444444444444444444
00000000000000000000000000
```

Figure 8-6   Pop-up panel showing the complete length of Value field in HEX

The same pop-up panel that is shown in Figure 8-5 on page 195 is displayed by entering the EXPAND command in the command line, placing the cursor in the Value field, and pressing enter. This is shown in Figure 8-7.



```
SCROLL1 Left/Right Scroll Panel Example -------
OPTION  ===> expand

 Field               Value
 -------------------------------
 Value             : kl 1234_
 Scroll Indicator  : <>
 Left & Right      : <          >
 Left/Right cols   : 13         24
 Length            : 100
 Scale             : --+-----2----
 Separator         : <---------->
```

Figure 8-7   Panel showing the entry of EXPAND primary command

The data, which is displayed in the pop-up panel, can easily be manipulated.

Note: Scrollable field support is panel-specific. A subsequent panel display that references the same variable but does not define it as scrollable may cause data truncation (depending on the data lengths involved).

8.1.2 DTL support for scrollable fields

The dialog tag language (DTL) is a set of markup language tags that you can use to define dialog elements. You can use DTL tags in addition to or instead of ISPF methods for defining panels, messages, and command tables. In addition, when you define a panel using DTL tags, you can assign a specific keylist to be associated with and displayed on that panel, if requested by the user.

The SCRFLD tag defines a field on an application panel as being scrollable. The panel field is defined using either the DTAFLD or LSTCOL tag. The SCRFLD tag must be nested within either a DTAFLD or LSTCOL tag.

Using the SCRFLD tag with a DTAFLD or LSTCOL tag causes the conversion utility to format an entry in the)FIELD section of the generated panel.

The SCRFLD tag and its parameters are shown in Figure 8-8.

```
<SCRFLD
     [DISPLEN=len] .............................. Length of variables
     [INDVAR=ind-var] [INDVAL='ind-chars'] ....... Left/Right scroll indicator
     [LINDVAR=lind-var] [LINDVAL='lind-char'] .... Left scroll indicator
     [RINDVAR=rind-var] [RINDVAR='rind-chars'] ... Right scroll indicator
     [SINDVAR=sind-var] [SINDVAL='sind-chars'] ... Separator scroll indicator
     [LCOLIND=lcol-var] [LCOLDISP=NO|YES] ........ Left column indicator
     [RCOLIND=rcol-var] [RCOLDISP=NO|YES] ........ Right column indicator
     [SCALE=scale-var] .......................... Scale indicator
     [SCROLL=ON|OFF] ............................ Scroll switch
     [FLDPOS=BELOW|ABOVE] ....................... Scroll indicator position
</SCRFLD>
```

Figure 8-8 SCRFLD tag for scrollable fields

The following are brief descriptions of parameters of the SCRFLD tag:

DISPLEN=n | %varname Specifies a length for the variable displayed in the scrollable field.

INDVAR=ind-var Specifies the name of a dialog variable that contains the left and right scroll indicator.

INDVAL='ind-chars' Overrides the default scroll indicator values of “-” and “+”.

LINDVAR=lind-var Specifies the name of a dialog variable that contains the left scroll indicator.

LINDVAL='lind-char' Overrides the default left-scroll indicator value of “-”.

RINDVAR=rind-var Specifies the name of a dialog variable that contains the right scroll indicator.

RINDVAL='rind-char' Overrides the default right-scroll indicator value of “+”.

SINDVAR=sind-var Specifies the name of a dialog variable that contains the separator scroll indicator.

SINDVAL='sind-chars' Overrides the default separator scroll indicator value of “<->”.

LCOLIND=lcol-var Specifies the name of a dialog variable that contains the value of the left column position for the displayed scrollable field.

LCOLDISP=NO | YES Specifies whether the left column position indicator defined using the LCOLIND attribute is displayed on the panel.

RCOLIND=rcol-var Specifies the name of a dialog variable that contains the value of the right column position for the displayed scrollable field.

RCOLDISP=NO | YES Specifies whether the right column position indicator defined using the LCOLIND attribute is displayed on the panel.

SCROLL=ON | OFF | %varname Specifies whether the field is scrollable or not.

FLDSPOS=BELOW | ABOVE Specifies where the scroll indicator panel fields are positioned in relation to the heading text for a table display field defined using the LSTCOL tag.

Figure 8-9 shows a panel source for our example.

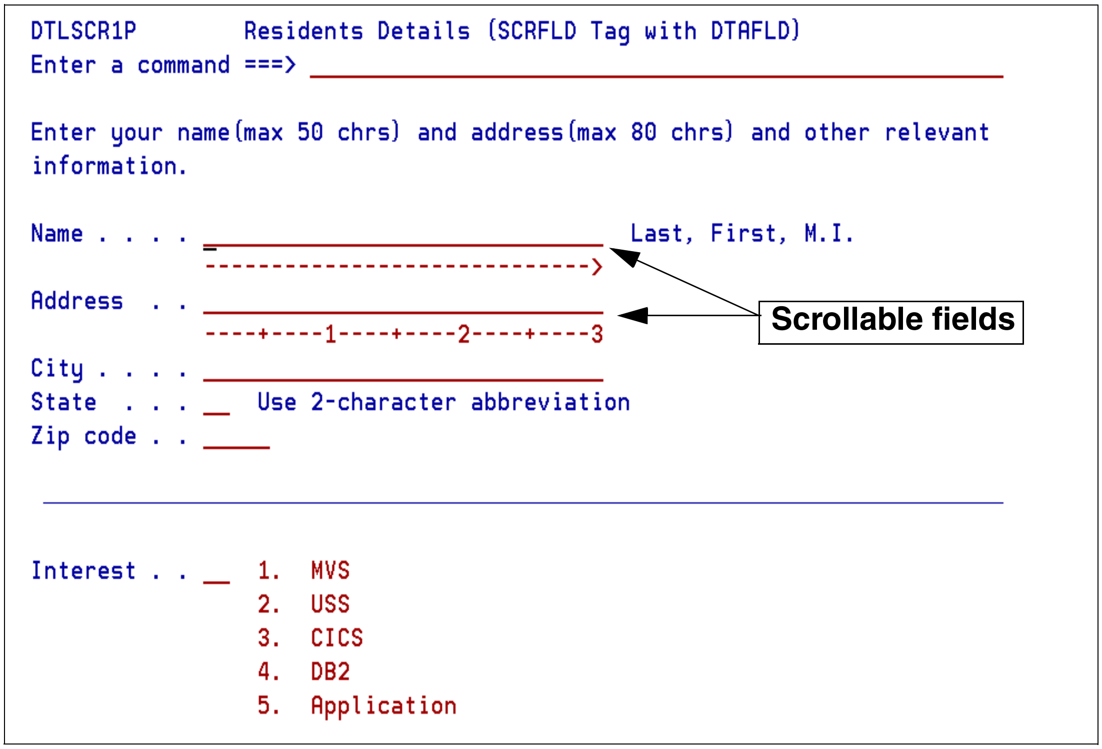
```
<!doctype dm system>
<varclass name=sampcls type ='char 30'>
<varclass name=statcls type ='char 2'>
<varclass name=zipcls  type ='char 5'>
<varclass name=char1cls  type ='char 1'>
<varlist>
<vardcl name=name varclass=sampcls>
  <vardcl name=addr varclass=sampcls>
  <vardcl name=city varclass=sampcls>
  <vardcl name=stat varclass=statcls>
  <vardcl name=day  varclass=char1cls>
  <vardcl name=zipc varclass=zipcls>
 </varlist>
<panel name=dtlscr1p keylist=key01 depth=24>Residents Details
  (SCRFLD Tag with DTAFLD)
  <topinst>Enter your name(max 50 chrs) and address(max
  80 chrs) and other relevant information.
  <area>
    <dtacol pmtwidth=12 entwidth=30 deswidth=29 selwidth=30>
      <dtafld datavar=name>Name
        <dtafldd>Last, First, M.I.
        <scrfld displen=50 sindvar=namesi>
      <dtafld datavar=addr>Address
        <scrfld displen=80 scale=addrsi>
      <dtafld datavar=city>City
      <dtafld datavar=stat entwidth=2>State
        <dtafldd>Use 2-character abbreviation
      <dtafld datavar=zipc entwidth=5>Zip code
      <divider type=solid gutter=3>
      <selfld name=day pmtloc=before>Interest
        <choice>MVS
        <choice>USS
        <choice>CICS
        <choice>DB2
        <choice>Application
      </selfld>
    </dtacol>
  </area>
 <CMDAREA>Enter a command
</panel>
```

Figure 8-9 DTL source using SCRFLD tag with DTAFLD

Example of using SCRFLD tag with DTAFLD

This example shows a panel to manipulate the Residents Details. The Name and Address fields have been defined as scrollable fields. The Name field is displayed with a separator scroll indicator and the Address field is displayed with a scale indicator. The DTL conversion utility automatically generates the separator scroll indicator below the Name field and the scale indicator below the Address field.

Figure 8-10 shows a display of the panel generated from the DTL definition in Figure 8-9 on page 198. A separator scroll indicator is displayed below the Name scrollable field. A scale line is displayed below the Address scrollable field.



```
DTLSCR1P       Residents Details (SCRFLD Tag with DTAFLD)
Enter a command ===> ____________________________________________________

Enter your name(max 50 chrs) and address(max 80 chrs) and other relevant
information.

Name . . . . ______________________________  Last, First, M.I.
             ----------------------------->
Address  . . ______________________________
             ----+----1----+----2----+----3
City . . . . ______________________________
State  . . . __  Use 2-character abbreviation
Zip code . . _____

  ________________________________________________________________________

Interest . . __  1.  MVS
                 2.  USS
                 3.  CICS
                 4.  DB2
                 5.  Application
```

Scrollable fields

Figure 8-10   Panel showing resident details

The name and address fields are scrollable. During data input, change, and display, RIGHT or LEFT PF keys can be used to scroll the data left or right in the field in which the cursor is placed. Also, an EXPAND PF key or EXPAND primary command can be used to display and manipulate the scrollable field in a pop-up panel as described previously, in “Example of ISPF panel services with scrollable fields” on page 193.

Example of using SCRFLD tag with LSTCOL

When the scrollable field is defined using the LSTCOL tag, the conversion utility automatically generates, with the column heading, the output fields for any scroll indicators you specify.

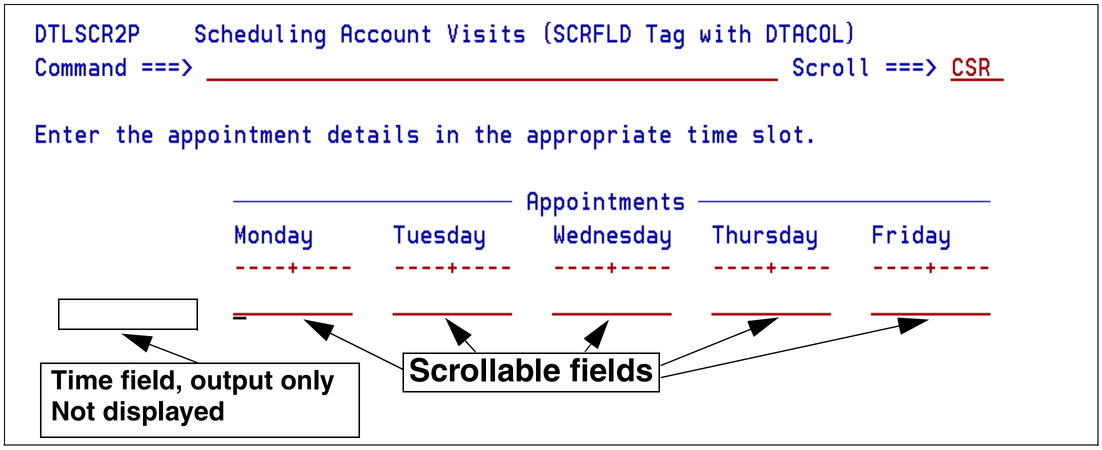
This is an example of the DTL panel definition to manipulate appointment data using scrollable panel. This defines the days field as scrollable.

Figure 8-11 on page 200 shows a panel source for our example.

```
<!doctype dm system>
<varclass name=timecls type='char 13'>
<varclass name=vc1     type ='char 9'>
<varlist>
  <vardcl name=timecol varclass=timecls>
  <vardcl name=moncol  varclass=vc1>
  <vardcl name=tuecol  varclass=vc1>
  <vardcl name=wedcol  varclass=vc1>
  <vardcl name=thrcol  varclass=vc1>
  <vardcl name=fricol  varclass=vc1>
</varlist>
<panel name=dtlscr2p keylist=key01>Scheduling Account Visits
  (SCRFLD Tag with LSTCOL)
 <topinst>Enter the appointment details in the appropriate time slot.
 <area>
  <lstfld scrollvar=scrlamt scrvhelp=scrhelp>
    <lstcol datavar=timecol usage=out colwidth=13>
    <lstgrp headline=yes>Appointments
      <lstcol datavar=moncol colwidth=9>Monday
        <scrfld displen=30 scale=monscl>
      <lstcol datavar=tuecol colwidth=9>Tuesday
        <scrfld displen=30 scale=tuescl>
      <lstcol datavar=wedcol colwidth=9>Wednesday
        <scrfld displen=30 scale=wedscl>
      <lstcol datavar=thrcol colwidth=9>Thursday
        <scrfld displen=30 scale=thrscl>
      <lstcol datavar=fricol colwidth=9>Friday
        <scrfld displen=30 scale=friscl>
    </lstgrp>
  </lstfld>
</area>
<cmdarea>
</panel>
```

Figure 8-11 DTL source using SCRFLD with LSTCOL

Figure 8-12 on page 201 shows the panel generated for the DTL definition in Figure 8-11. The appointment information for each hour of each working day is displayed in column fields which are scrollable. A scale indicator is displayed with the heading for each day's column. The time field is output only, so no value is displayed. The days field is scrollable and has a length of 30 characters. Keeping the cursor in the Days field, the LEFT or RIGHT PF keys can be used to scroll the data left or right. Also the EXPAND primary command or PF key can be used to display and manipulate the data in a pop-up panel as discussed in“Example of ISPF panel services with scrollable fields” on page 193.



```
DTLSCR2P    Scheduling Account Visits (SCRFLD Tag with DTACOL)
Command ===> ____________________________________________ Scroll ===> CSR

Enter the appointment details in the appropriate time slot.

                        ──────────────────── Appointments ────────────────────
                        Monday       Tuesday      Wednesday    Thursday     Friday
                        ----+----    ----+----    ----+----    ----+----    ----+----
```

Figure 8-12 Panel scheduling account visits

Note: For details, refer to chapter 13 of *z/OS ISPF Dialog Tag Language Guide and Reference*, SC34-4824.

8.2 Changes to the ISPF panel processing

The following additional enhancements have been made to ISPF panel processing:

- LENGTH built-in function

  variable = LENGTH(field-name)

  – The LENGTH built-in function is available for use in panel procedures.

  – The LENGTH built-in function can occur on the right side of an assignment statement to evaluate the length of a dialog variable. The variable length returned is the maximum value of the actual length of the variable if it exists and the length specified in the)FIELD section if any.

    • Example: `&A = LENGTH(ABC)`

      The length of dialog variable ABC is stored in &A. If ABC does not exist, zero is returned.

- UPPER built-in function

  variable = UPPER(field-name)

  – The UPPER built-in function is available for use in panel procedures.

  – The UPPER built-in function can occur on the right side of an assignment statement and returns the uppercase value of a variable.

    • Example: `&A = UPPER(ABC)`

      The uppercase value of the ABC dialog variable is returned to variable A.

- Support has been added allowing multi-line input fields to be defined in scrollable areas. This *removes* the following restriction that applied when defining a scrollable area:

  – Fields in the scrollable area or text fields cannot be defined to wrap. A field cannot extend beyond one line of the area.

8.3 Catalog name in data set list

The data set list created by ISPF has been enhanced to return the name of the catalog where the data set is located. This is only applicable when the data set list is built via a catalog search. A new option has been added to the data set list utility panel (ISPF option 3.4) to show the catalog name in the data set list.

This provides the end-user with the name of the catalog in which the data set was located. This is particularly useful when there are duplicate data set names in the Data Set List.

The new shared pool variable ZDLCATNM is available for use by TSO commands, CLISTs, and REXX execs. This variable stores the catalog name in which the data set is located.

The new parameter CATALOG has been added to the LMDDISP (Data Set List) service. This controls whether the catalog name is shown in the Total View for the resulting Data Set List display.

```
ISPEXEC LMDDISP LISTID(dslist-id)
      [VIEW(VOLUME|SPACE|ATTRIB|TOTAL)]
         [COMFIRM(YES|NO)]
      [PANEL(panel-name)]
      [CATALOG(YES|NO)]
```

Figure 8-13 ISPF LMDDISP service

The new option SAVEC has been added to the OPTION parameter on the LMDLIST (List Data Sets) service. This option is similar to the SAVE option, but also causes the catalog name to be written to the output file containing the data set list information.

```
ISPEXEC LMDLISTLISTID(dslist-id)
                  [OPTION(LIST|FREE|SAVE|SAVEC)]
                  [DATASET(dataset-var)]
                  [STATS(YES|NO)]
                  [GROUP(group)]
```

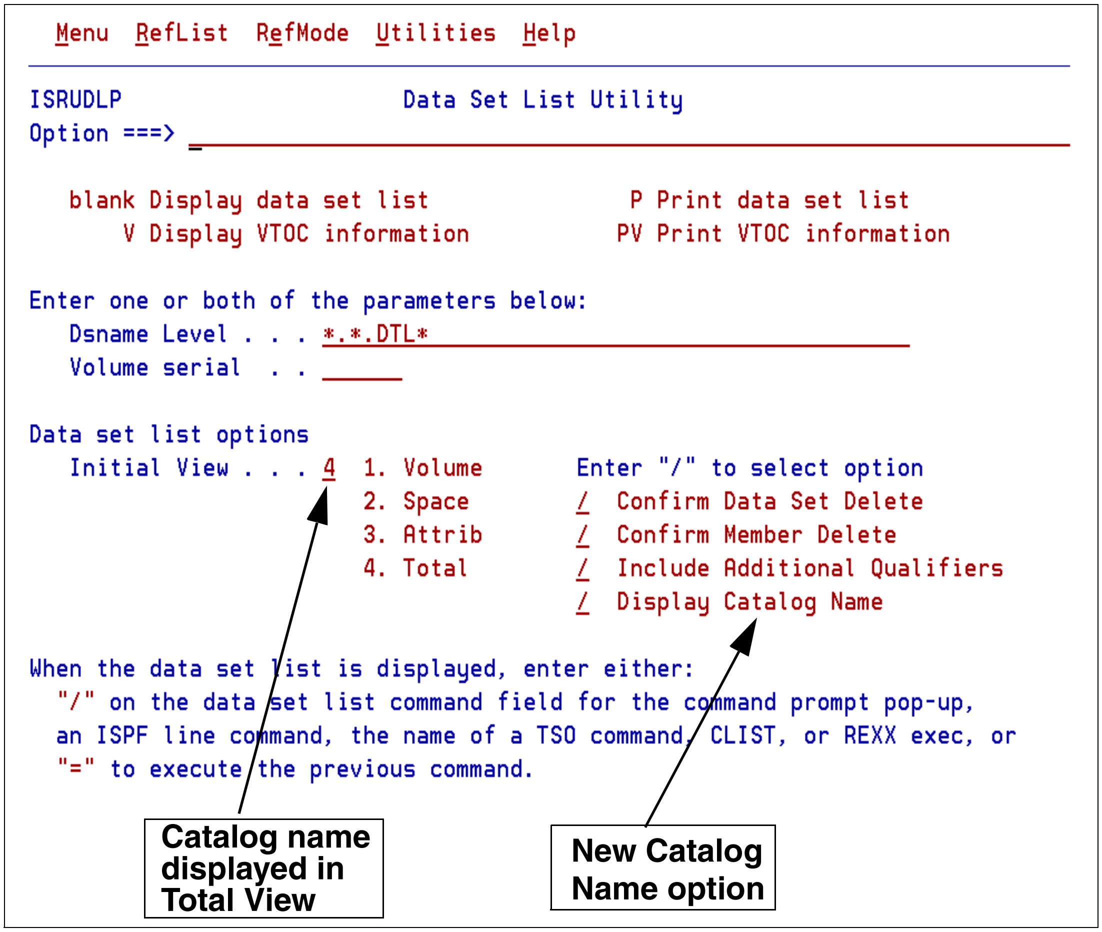
Figure 8-14 ISPF LMDLIST service

8.3.1 Data set list utility panel

The data set list utility panel ISRUDLP has been modified to include an option “Display Catalog Name” to allow the user to display the name of the catalog where each data set in the list was located.

The catalog name is shown in the Data Set List display when the Total View option is selected.

The new Data Set Utility panel is shown in Figure 8-15 on page 203 to display Total View of data sets “*.*.DTL*”.



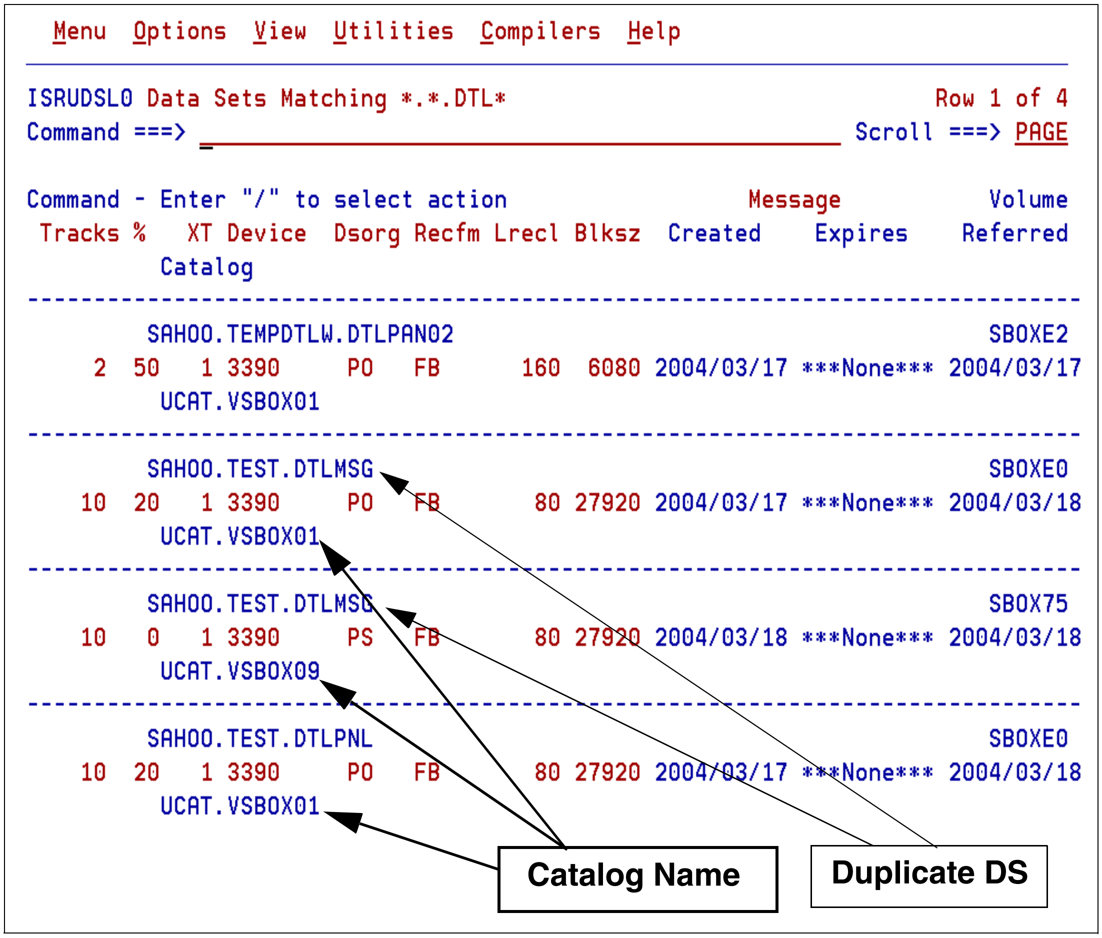
```
  Menu  RefList  RefMode  Utilities  Help
------------------------------------------------------------------------------
ISRUDLP                      Data Set List Utility
Option ===> _

   blank Display data set list                P Print data set list
       V Display VTOC information            PV Print VTOC information

Enter one or both of the parameters below:
   Dsname Level . . . *.*.DTL*
   Volume serial  . . ______

Data set list options
   Initial View . . . 4  1. Volume       Enter "/" to select option
                         2. Space        /  Confirm Data Set Delete
                         3. Attrib       /  Confirm Member Delete
                         4. Total        /  Include Additional Qualifiers
                                         /  Display Catalog Name

When the data set list is displayed, enter either:
  "/" on the data set list command field for the command prompt pop-up,
  an ISPF line command, the name of a TSO command, CLIST, or REXX exec, or
  "=" to execute the previous command.
```

Figure 8-15 Panel ISRUDLP showing new option

Panel showing the catalog name

The data sets are listed in the panel shown in Figure 8-16 on page 204. As we can see in this panel, the catalog name, in which the data set is located, is listed against each data set in the third line of the data set entry.

Interestingly, as shown in the panel, the data set SAHOO.TEST.DTLMSG is cataloged in two different catalogs, UCAT.VSBOX01 and UCAT.VSBOX09. This enhancement is extremely helpful to end users in identifying the duplicate data sets and the catalogs in which the data sets are located. Otherwise, it would be a very time consuming task in determining the catalogs in which the duplicate data sets are cataloged.



```
  Menu  Options  View  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
ISRUDSL0 Data Sets Matching *.*.DTL*                                  Row 1 of 4
Command ===> _                                                 Scroll ===> PAGE

Command - Enter "/" to select action                  Message           Volume
 Tracks %   XT Device  Dsorg Recfm Lrecl Blksz Created    Expires    Referred
          Catalog
-------------------------------------------------------------------------------
         SAHOO.TEMPDTLW.DTLPAN02                                        SBOXE2
    2  50    1 3390     PO    FB      160  6080 2004/03/17 ***None*** 2004/03/17
          UCAT.VSBOX01
-------------------------------------------------------------------------------
         SAHOO.TEST.DTLMSG                                              SBOXE0
   10  20    1 3390     PO    FB       80 27920 2004/03/17 ***None*** 2004/03/18
          UCAT.VSBOX01
-------------------------------------------------------------------------------
         SAHOO.TEST.DTLMSG                                              SBOX75
   10   0    1 3390     PS    FB       80 27920 2004/03/18 ***None*** 2004/03/18
          UCAT.VSBOX09
-------------------------------------------------------------------------------
         SAHOO.TEST.DTLPNL                                              SBOXE0
   10  20    1 3390     PO    FB       80 27920 2004/03/17 ***None*** 2004/03/18
          UCAT.VSBOX01
```

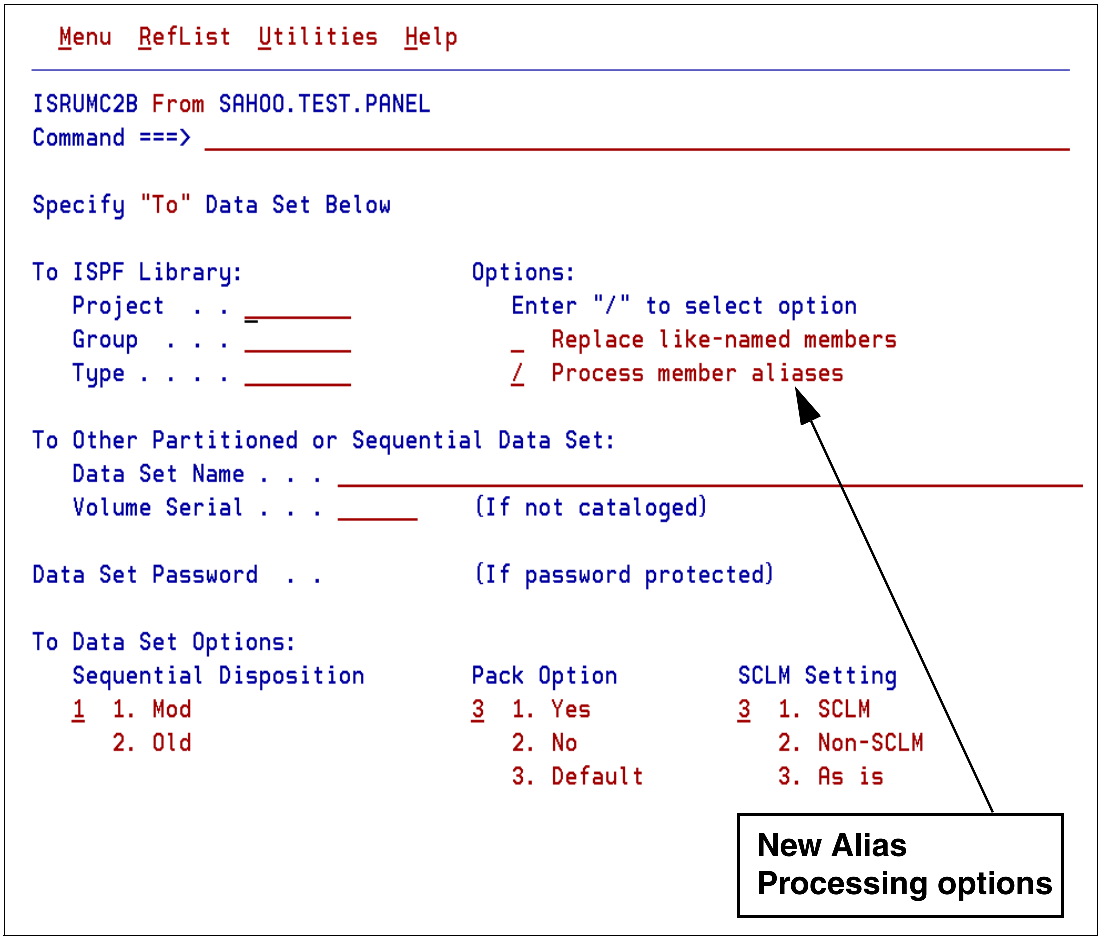
Figure 8-16 Panel ISRUDSL0 showing the catalog name

8.4 MOVE/COPY alias support

In ISPF options 3.3, 3.4, and 11, the move/copy facility now includes an option “Process member aliases.” With this option in effect, the main member and all its aliases are copied. This option is implemented as an additional keyword ALIAS or NOALIAS on the LMMOVE and LMCOPY services.

- In the past, the user had to ensure the primary member was copied first, and *then* the aliases, in order to maintain the correct alias relationships.
- When using this new support, a number of new rules apply, and a number of previous restrictions are relaxed.
  - Either the primary member or any alias member may be selected to copy the primary member and all of its aliases. This occurs even if a single member is specified or some of the members are not displayed in the current member selection list.
  - Alias members are copied for both load and non-load data sets as well as for PDS and PDSE data sets.
  - Copying to the same data set is not supported when aliases are automatically selected. This would result in the from and to member name being the same.
- The new MOVE/COPY alias support is available with the following PDF functions:
  - ISPF Option 3.3 (Move/Copy)

  - ISPF Option 3.4 (Data Set Utility) MO and CO line commands
  - M or C line commands from within a member list created from ISPF Option 3.4, including the Edit (E), View (V), Browse (B), and Member List (M) options
  - ISPF Option 11 (Workplace™) M or C commands, or MOVE or COPY actions from the "File" action bar
- For the PDF MOVE/COPY functions, the panel prompting the user for the target or output data set has been changed to include the new option "Process member aliases." Following is an example of using ISPF option 3.3 to move members from data set SAHOO.TEST.PANEL. As you can see, panel ISRUMC2B is displayed with the new option as shown in Figure 8-17.



```
  Menu  RefList  Utilities  Help
ISRUMC2B From SAHOO.TEST.PANEL
Command ===> ________________________________________________________________

Specify "To" Data Set Below

To ISPF Library:                  Options:
   Project  . . ________             Enter "/" to select option
   Group  . . . ________             _  Replace like-named members
   Type . . . . ________             /  Process member aliases

To Other Partitioned or Sequential Data Set:
   Data Set Name . . . ________________________________________________________
   Volume Serial . . . ______     (If not cataloged)

Data Set Password  . .            (If password protected)

To Data Set Options:
   Sequential Disposition         Pack Option         SCLM Setting
   1  1. Mod                      3  1. Yes           3  1. SCLM
      2. Old                         2. No               2. Non-SCLM
                                     3. Default          3. As is
```

Figure 8-17 Panel ISRUMC2B showing new option

8.5 Changes to ISPF services

The new keywords ALIAS and NOALIAS have been added to the ISPF services LMCOPY and LMMOVE. When ALIAS is specified, the member and all its aliases are copied.

8.5.1 ISPF LMCOPY service

Figure 8-18 on page 206 shows the definition of LMCOPY service with its parameters.

```
ISPEXEC LMCOPY FROMID(from-data-id)
          [FROMMEM(from-member-name)]
          TODATAID(to-data-id)
          [TOMEM(to-member-name)]
          [REPLACE]
          [PACK]
          [TRUNC]
          [LOCK]
          [SCLMSET(Y|N)]
          [ALIAS|NOALIAS]
```

Figure 8-18 ISPF LMCOPY service

ALIAS option

If the ALIAS option is in effect, LMCOPY automatically processes alias members as follows:

- Either the main member or any alias member may be selected to copy the main member and all of its aliases. This occurs even if some of the members are not displayed in the current member selection list.
- Alias members are copied for both load and non-load data sets as well as for PDS and PDSE data sets.

Copying to the same data set is not supported when aliases are automatically selected because it would result in the “from” and “to” member names being the same.

NOALIAS option

If the NOALIAS option is in effect, LMCOPY does not copy alias members unless one of the following is true:

- All members of the data set are selected.
- A member pattern is used and both the main member and the alias member are included in that pattern.

If the NOALIAS option is in effect, copying an alias member by itself results in a new member being created, even if the main member has already been copied.

8.5.2 ISPF LMMOVE service

Figure 8-19 shows the definition of the LMMOVE service with its parameters.

```
ISPEXEC LMMOVE FROMID(from-data-id)
          [FROMMEM(from-member-name)]
          TODATAID(to-data-id)
          [TOMEM(to-member-name)]
          [REPLACE]
          [PACK]
          [TRUNC]
          [LOCK]
          [SCLMSET(Y|N)]
          [ALIAS|NOALIAS]
```

Figure 8-19 ISPF LMMOVE service

ALIAS option

If the ALIAS option is in effect, LMMOVE automatically processes alias members as follows:

- Either the main member or any alias member may be selected to move the main member and all of its aliases. This occurs even if some of the members are not displayed in the current member selection list.
- Alias members are moved for both load and non-load data sets as well as for PDS and PDSE data sets.

Moving to the same data set is not supported when aliases are automatically selected because it would result in the “from” and “to” member names being the same.

NOALIAS option

If the NOALIAS option is in effect, LMMOVE does not move alias members unless one of the following is true:

- All members of the data set are selected.
- A member pattern is used and both the main member and the alias member are included in that pattern.

If the NOALIAS option is in effect, moving an alias member by itself results in a new member being created, even if the main member has already been moved.

8.6 PDS/PDSE member delete by pattern

Support has been added to allow for the deletion of multiple members of a PDS or PDSE with a single command, optionally bypassing the member list display.

This support is based on an enhancement by DFSMS to the STOW macro, which allows a PDS to be reset (that is, delete all members leaving the data set with an empty directory).

This facility is available via the following PDF functions:

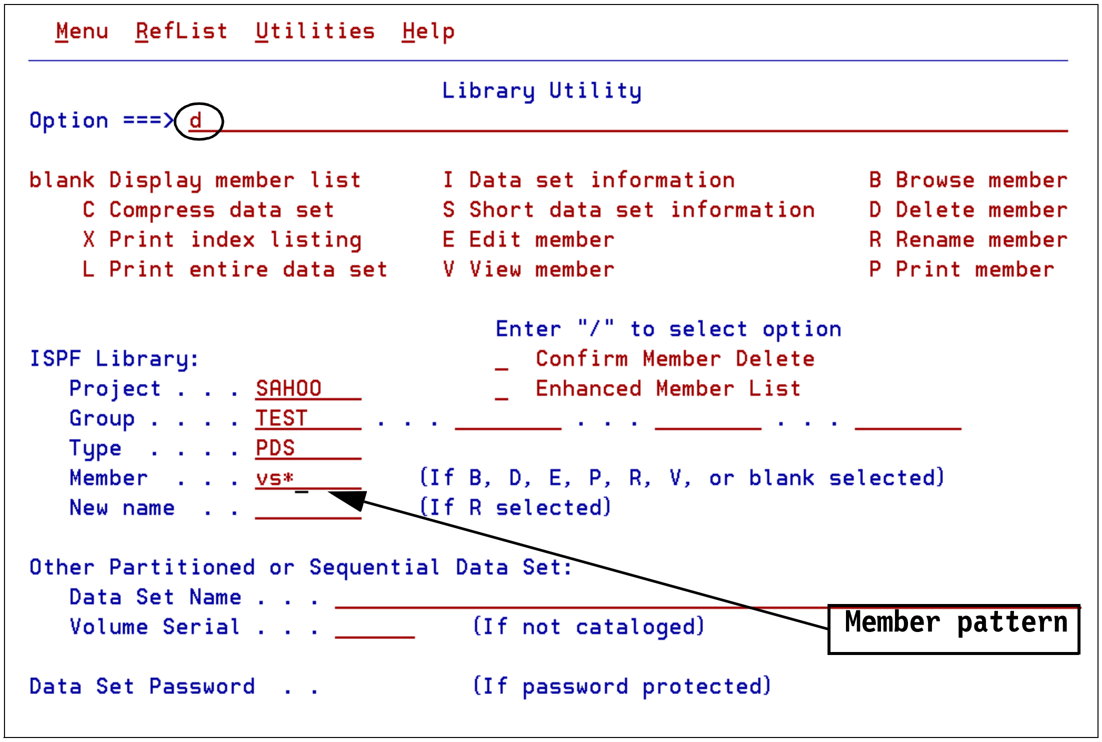
- ISPF Option 3.1 (Library Utility) - “**D**” command
- ISPF Option 3.4 (Data Set Utility) - “**D**” line command
- ISPF Option 11 (Workplace)
  - “D” command
  - DELETE → MEMBER actions from the “File” action bar.

LMMDEL (delete a member of a data set) service now supports specification of a member name pattern with the MEMBER parameter.

- If MEMBER(*) is specified on the LMMDEL service, the associated LMINIT service must specify ENQ(EXCLU).

Example 1: Delete a group of members using ISPF option 3.1

Here is an example of using ISPF option 3.1 to delete a group of members starting with characters VS (vs*) in data set SAHOO.TEST.PDS. This is shown in Figure 8-20 on page 208. If the “Confirm Member Delete” option is selected, it displays another pop-up panel to confirm the delete. If the “Confirm Member Delete” option is deselected, it deletes all members matching the pattern without asking the user for confirmation.

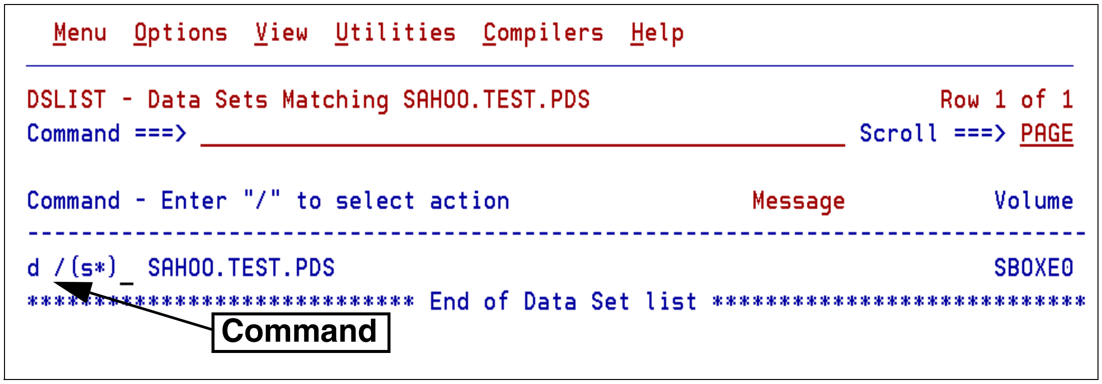


```
  Menu  RefList  Utilities  Help
------------------------------------------------------------------------------
                              Library Utility
Option ===> d

blank Display member list      I Data set information          B Browse member
    C Compress data set        S Short data set information    D Delete member
    X Print index listing      E Edit member                   R Rename member
    L Print entire data set    V View member                   P Print member

                                   Enter "/" to select option
ISPF Library:                      _  Confirm Member Delete
   Project . . . SAHOO             _  Enhanced Member List
   Group . . . . TEST     . . .          . . .          . . .
   Type  . . . . PDS
   Member  . . . vs*_        (If B, D, E, P, R, V, or blank selected)
   New name  . .             (If R selected)

Other Partitioned or Sequential Data Set:
   Data Set Name . . .
   Volume Serial . . .          (If not cataloged)

Data Set Password  . .          (If password protected)
```

Figure 8-20 Delete a group of members using ISPF option 3.1

Example 2: Delete a group of members using ISPF option 3.4

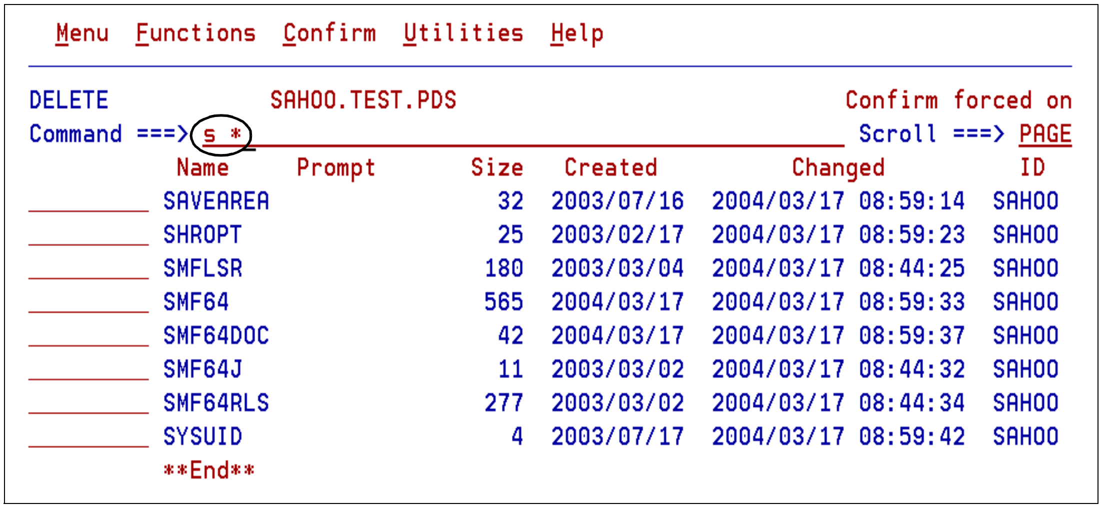
Here is an example of deleting a group of members starting with character S (**s***) using ISPF option 3.4, in data set SAHOO.TEST.PDS. The line command "**d /(s*)**" is entered as shown in Figure 8-21.



```
  Menu  Options  View  Utilities  Compilers  Help
------------------------------------------------------------------------------
DSLIST - Data Sets Matching SAHOO.TEST.PDS                            Row 1 of 1
Command ===>                                                    Scroll ===> PAGE

Command - Enter "/" to select action                 Message           Volume
-------------------------------------------------------------------------------
d /(s*)_ SAHOO.TEST.PDS                                                SBOXE0
***************************** End of Data Set list ****************************
```

Figure 8-21 Delete a group of members using ISPF option 3.4

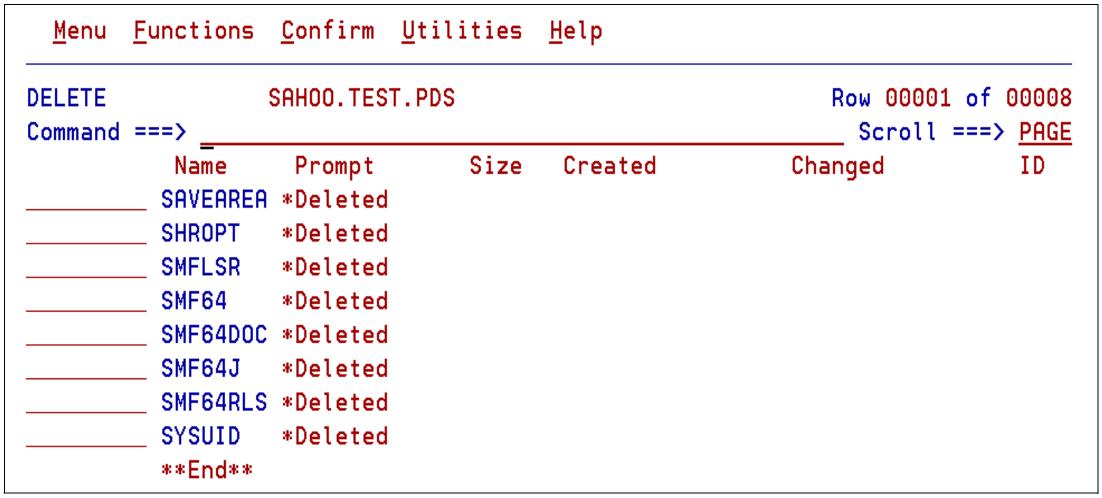
This displays another panel with the members list matching the pattern **s*** as shown in Figure 8-22 on page 209.



```
  Menu  Functions  Confirm  Utilities  Help
─────────────────────────────────────────────────────────────────────────────
DELETE            SAHOO.TEST.PDS                            Confirm forced on
Command ===> s *                                             Scroll ===> PAGE
          Name     Prompt       Size   Created          Changed          ID
_________ SAVEAREA                32  2003/07/16  2004/03/17 08:59:14  SAHOO
_________ SHROPT                  25  2003/02/17  2004/03/17 08:59:23  SAHOO
_________ SMFLSR                 180  2003/03/04  2004/03/17 08:44:25  SAHOO
_________ SMF64                  565  2004/03/17  2004/03/17 08:59:33  SAHOO
_________ SMF64DOC                42  2004/03/17  2004/03/17 08:59:37  SAHOO
_________ SMF64J                  11  2003/03/02  2004/03/17 08:44:32  SAHOO
_________ SMF64RLS               277  2003/03/02  2004/03/17 08:44:34  SAHOO
_________ SYSUID                   4  2003/07/17  2004/03/17 08:59:42  SAHOO
          **End**
```

Figure 8-22 Confirm delete of group of members

Here the **d** command is still active. Now you can delete the selected members from the list by selecting the members with line command **s** or use primary command "**s ***" to delete all selected members. A confirmation panel is returned; once confirmed, the selected members are deleted. Figure 8-23 shows the effect of primary command "**s ***" indicating that all the members selected are deleted.



```
  Menu  Functions  Confirm  Utilities  Help
─────────────────────────────────────────────────────────────────────────────
DELETE            SAHOO.TEST.PDS                          Row 00001 of 00008
Command ===> _                                               Scroll ===> PAGE
          Name     Prompt       Size   Created          Changed          ID
_________ SAVEAREA *Deleted
_________ SHROPT   *Deleted
_________ SMFLSR   *Deleted
_________ SMF64    *Deleted
_________ SMF64DOC *Deleted
_________ SMF64J   *Deleted
_________ SMF64RLS *Deleted
_________ SYSUID   *Deleted
          **End**
```

Figure 8-23 The group of selected members deleted

8.7 PDS/PDSE member list enhancements

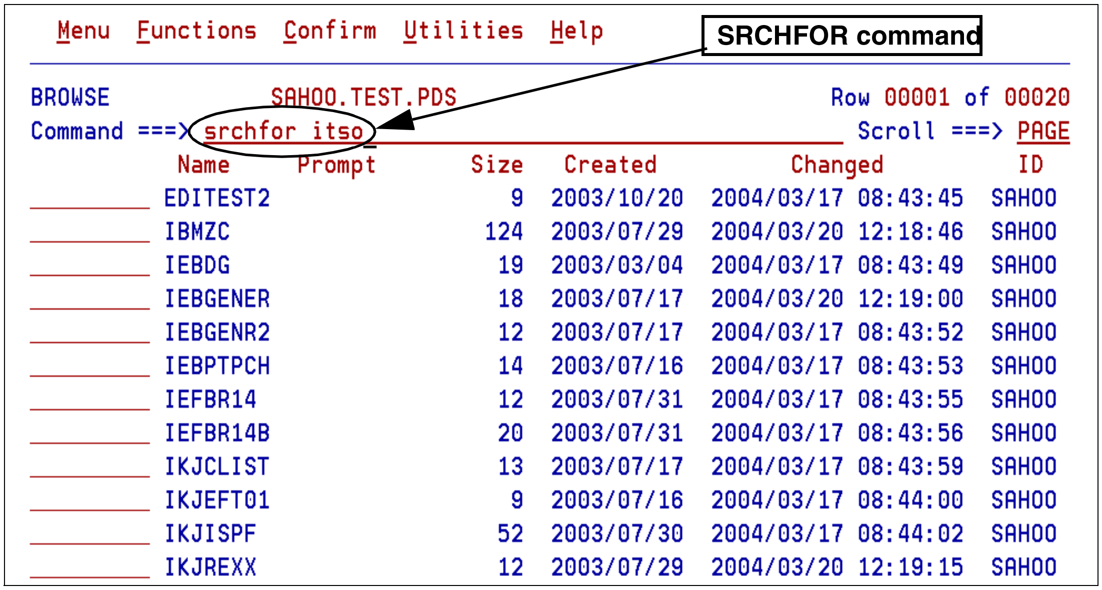
ISPF member list has been enhanced to include a new SRCHFOR primary command in the Data Set List. This allows the users to search the members containing a specific string of data. Also, a new option is available to set member list scroll behavior in the Settings panel. This option allows you to control the scroll behavior of the members in the data set.

8.7.1 SRCHFOR command

A SRCHFOR command has been added to member list, allowing a search of members in the list using SuperC. Under ISPF Data Set List, SRCHFOR is now available as a primary command.

ISPF invokes the SuperC utility to search the contents of the members in the member list for the string specified as a parameter on the SRCHFOR command. If the string is found in a member “*Found” is displayed in the Prompt column.

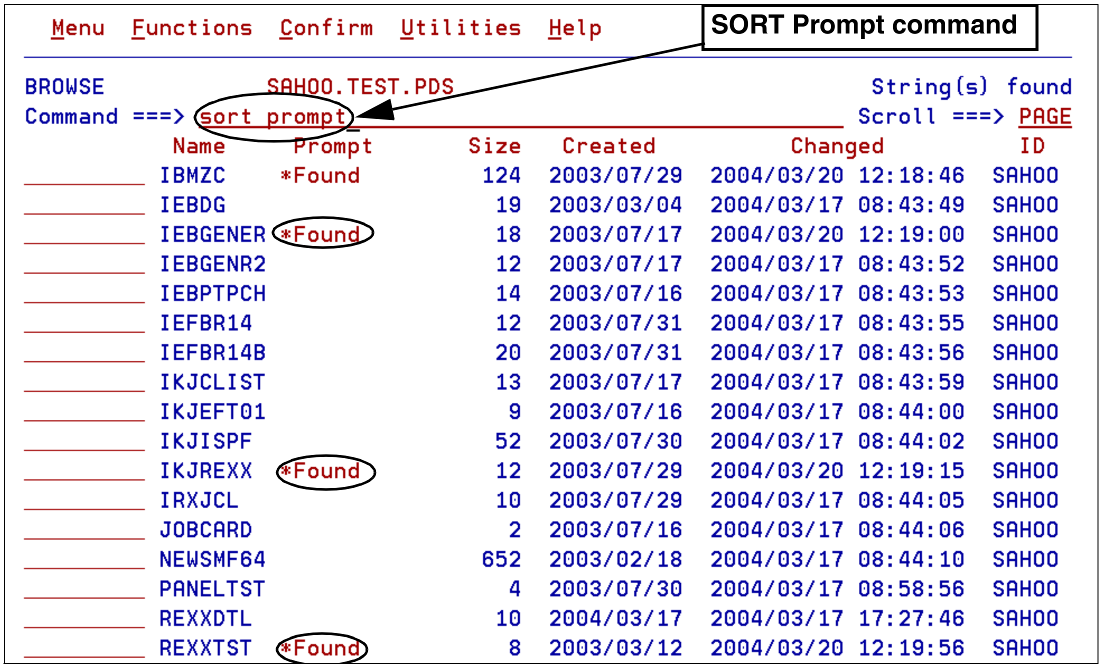
If no string is supplied with the SRCHFOR command, it displays the MEMBER LIST Srchfor Options panel, where you can specify search strings and modify options affecting the SRCHFOR command. An example of issuing the command SRCHFOR ITSO in the member list of data set SAHOO.TEST.PDS is shown in Figure 8-24.



```
BROWSE            SAHOO.TEST.PDS                       Row 00001 of 00020
Command ===> srchfor itso                               Scroll ===> PAGE
           Name     Prompt       Size   Created          Changed          ID
_________ EDITEST2                  9  2003/10/20  2004/03/17 08:43:45  SAHOO
_________ IBMZC                   124  2003/07/29  2004/03/20 12:18:46  SAHOO
_________ IEBDG                    19  2003/03/04  2004/03/17 08:43:49  SAHOO
_________ IEBGENER                 18  2003/07/17  2004/03/20 12:19:00  SAHOO
_________ IEBGENR2                 12  2003/07/17  2004/03/17 08:43:52  SAHOO
_________ IEBPTPCH                 14  2003/07/16  2004/03/17 08:43:53  SAHOO
_________ IEFBR14                  12  2003/07/31  2004/03/17 08:43:55  SAHOO
_________ IEFBR14B                 20  2003/07/31  2004/03/17 08:43:56  SAHOO
_________ IKJCLIST                 13  2003/07/17  2004/03/17 08:43:59  SAHOO
_________ IKJEFT01                  9  2003/07/16  2004/03/17 08:44:00  SAHOO
_________ IKJISPF                  52  2003/07/30  2004/03/17 08:44:02  SAHOO
_________ IKJREXX                  12  2003/07/29  2004/03/20 12:19:15  SAHOO
```

Figure 8-24 Use of `SRCHFOR` *command*

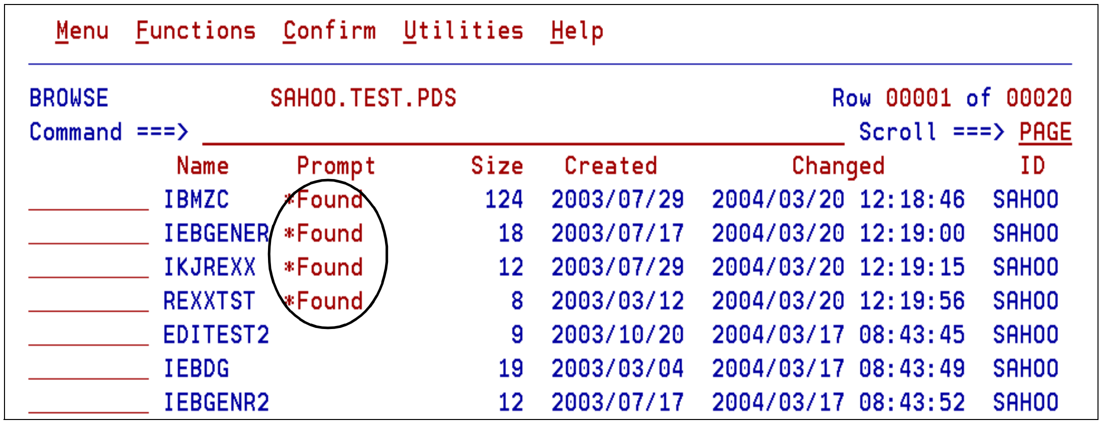
It displays “*Found” against the members in which the string ITSO is found, as shown in Figure 8-25 on page 211.



```
BROWSE            SAHOO.TEST.PDS                              String(s) found
Command ===> sort prompt_                                     Scroll ===> PAGE
          Name      Prompt       Size   Created         Changed           ID
_________ IBMZC     *Found        124  2003/07/29  2004/03/20 12:18:46  SAHOO
_________ IEBDG                    19  2003/03/04  2004/03/17 08:43:49  SAHOO
_________ IEBGENER *Found          18  2003/07/17  2004/03/20 12:19:00  SAHOO
_________ IEBGENR2                 12  2003/07/17  2004/03/17 08:43:52  SAHOO
_________ IEBPTPCH                 14  2003/07/16  2004/03/17 08:43:53  SAHOO
_________ IEFBR14                  12  2003/07/31  2004/03/17 08:43:55  SAHOO
_________ IEFBR14B                 20  2003/07/31  2004/03/17 08:43:56  SAHOO
_________ IKJCLIST                 13  2003/07/17  2004/03/17 08:43:59  SAHOO
_________ IKJEFT01                  9  2003/07/16  2004/03/17 08:44:00  SAHOO
_________ IKJISPF                  52  2003/07/30  2004/03/17 08:44:02  SAHOO
_________ IKJREXX   *Found         12  2003/07/29  2004/03/20 12:19:15  SAHOO
_________ IRXJCL                   10  2003/07/29  2004/03/17 08:44:05  SAHOO
_________ JOBCARD                   2  2003/07/16  2004/03/17 08:44:06  SAHOO
_________ NEWSMF64                652  2003/02/18  2004/03/17 08:44:10  SAHOO
_________ PANELTST                  4  2003/07/30  2004/03/17 08:58:56  SAHOO
_________ REXXDTL                  10  2004/03/17  2004/03/17 17:27:46  SAHOO
_________ REXXTST   *Found          8  2003/03/12  2004/03/20 12:19:56  SAHOO
```

Figure 8-25 List of members found with string ITSO

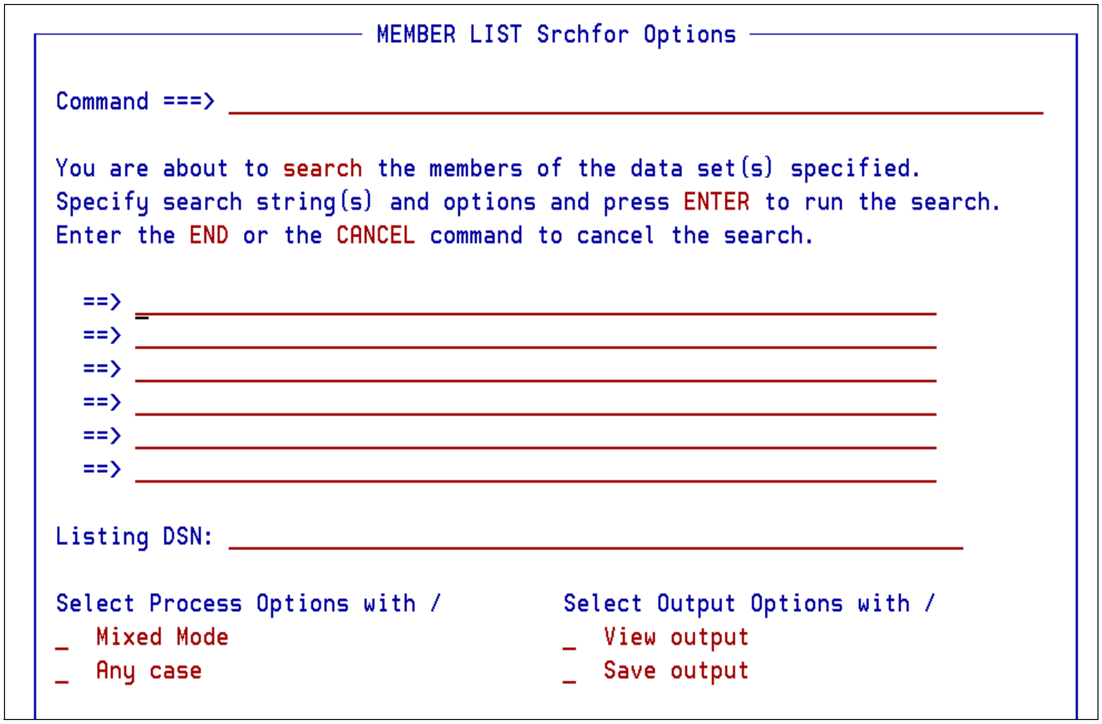
Now, you can issue the primary command **SORT prompt** in the command line and the list of members with message “*Found” in the Prompt column is sorted to the top of the member list. This is shown in Figure 8-26.



```
BROWSE            SAHOO.TEST.PDS                          Row 00001 of 00020
Command ===> ________________________________________________ Scroll ===> PAGE
          Name      Prompt       Size   Created         Changed           ID
_________ IBMZC     *Found        124  2003/07/29  2004/03/20 12:18:46  SAHOO
_________ IEBGENER  *Found         18  2003/07/17  2004/03/20 12:19:00  SAHOO
_________ IKJREXX   *Found         12  2003/07/29  2004/03/20 12:19:15  SAHOO
_________ REXXTST   *Found          8  2003/03/12  2004/03/20 12:19:56  SAHOO
_________ EDITEST2                  9  2003/10/20  2004/03/17 08:43:45  SAHOO
_________ IEBDG                    19  2003/03/04  2004/03/17 08:43:49  SAHOO
_________ IEBGENR2                 12  2003/07/17  2004/03/17 08:43:52  SAHOO
```

*Figure 8-26 Sorted list of members with *Found message in prompt column*

When the SRCHFOR command is issued without any parameter, the panel MEMBER LIST Srchfor Options is displayed to input the search string. This is shown in Figure 8-27 on page 212.



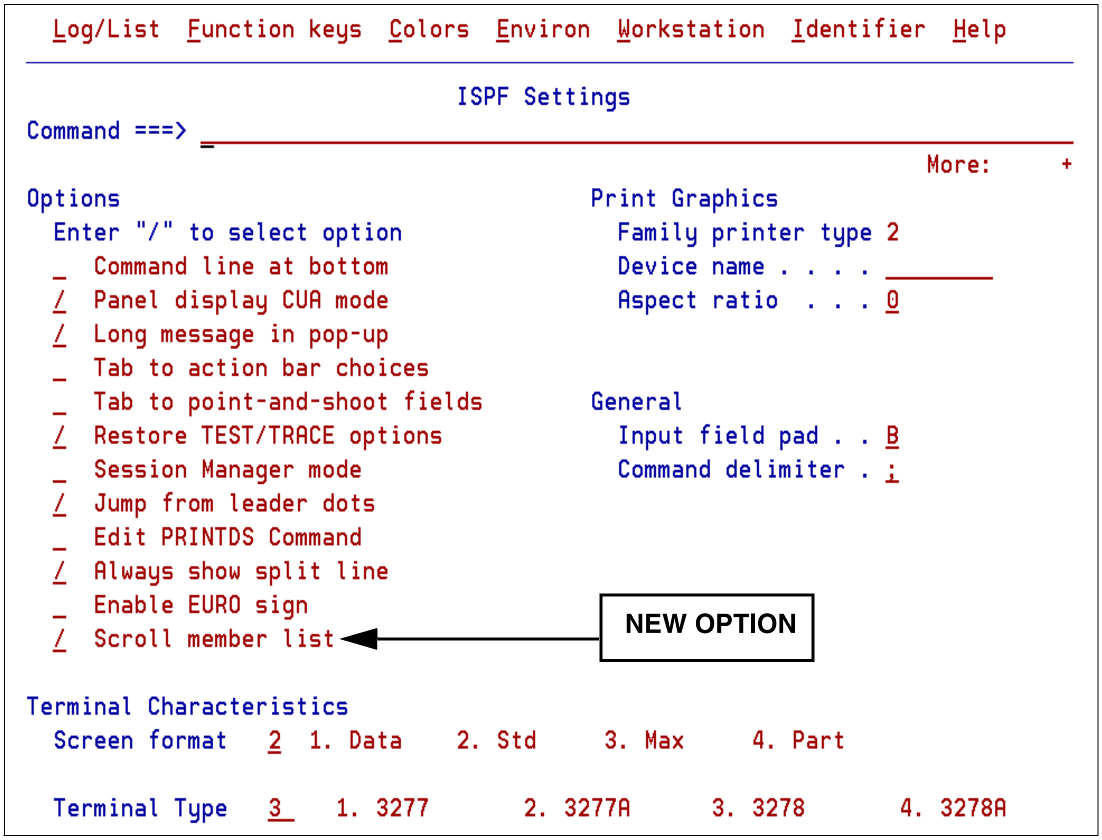
```
                    MEMBER LIST Srchfor Options

Command ===> ________________________________________________________

You are about to search the members of the data set(s) specified.
Specify search string(s) and options and press ENTER to run the search.
Enter the END or the CANCEL command to cancel the search.

  ==> _______________________________________________________
  ==> _______________________________________________________
  ==> _______________________________________________________
  ==> _______________________________________________________
  ==> _______________________________________________________
  ==> _______________________________________________________

Listing DSN: ____________________________________________________

Select Process Options with /         Select Output Options with /
_  Mixed Mode                         _  View output
_  Any case                           _  Save output
```

Figure 8-27 Panel displayed when SRCHFOR command was issued without any parameter

8.7.2 Scroll behavior

A new Option is provided to allow the user to disable member lists from scrolling to the first member selected for processing. The new option is available on the ISPF Settings panel. The setting of this option is reflected in the value of dialog variable ZSCRML. When the option is deselected, member lists will NOT scroll to the first member selected for processing.

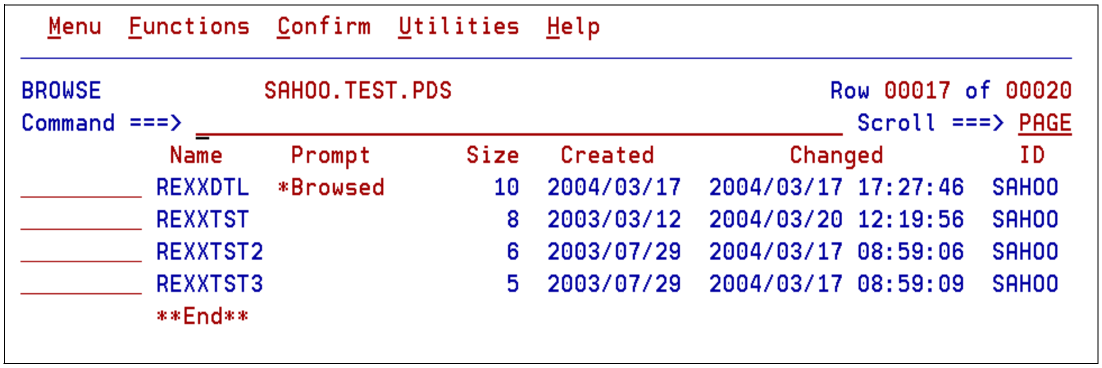
The ISPF configuration table can be used to set an initial default value for this option. The new option is shown in Figure 8-28 on page 213.



```
  Log/List  Function keys  Colors  Environ  Workstation  Identifier  Help
 ------------------------------------------------------------------------------
                                 ISPF Settings
Command ===> _
                                                                 More:     +
Options                                  Print Graphics
  Enter "/" to select option               Family printer type 2
  _  Command line at bottom                Device name . . . . ________
  /  Panel display CUA mode                Aspect ratio  . . . 0
  /  Long message in pop-up
  _  Tab to action bar choices
  _  Tab to point-and-shoot fields       General
  /  Restore TEST/TRACE options            Input field pad . . B
  _  Session Manager mode                  Command delimiter . ;
  /  Jump from leader dots
  _  Edit PRINTDS Command
  /  Always show split line
  _  Enable EURO sign
  /  Scroll member list

Terminal Characteristics
  Screen format   2  1. Data    2. Std     3. Max     4. Part

  Terminal Type   3_   1. 3277      2. 3277A      3. 3278       4. 3278A
```

Figure 8-28 New option in settings panel

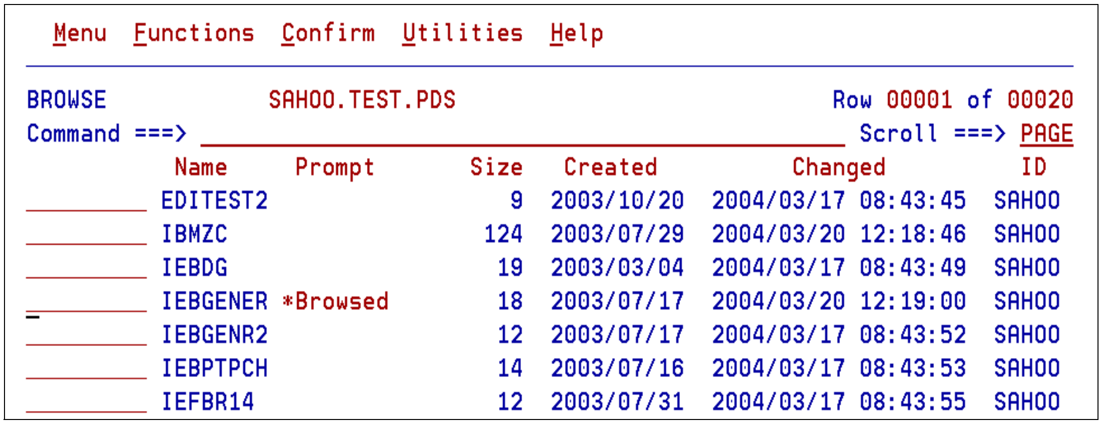
With the “Scroll member list” option selected, the last selected member is scrolled to become the top member in the list after the user exits the member. This behavior was introduced into ISPF for OS/390 2.10. This is shown in Figure 8-29.



```
  Menu  Functions  Confirm  Utilities  Help
 ------------------------------------------------------------------------------
BROWSE            SAHOO.TEST.PDS                          Row 00017 of 00020
Command ===> _                                                Scroll ===> PAGE
           Name     Prompt       Size   Created          Changed          ID
_________ REXXDTL  *Browsed       10  2004/03/17  2004/03/17 17:27:46  SAHOO
_________ REXXTST                  8  2003/03/12  2004/03/20 12:19:56  SAHOO
_________ REXXTST2                 6  2003/07/29  2004/03/17 08:59:06  SAHOO
_________ REXXTST3                 5  2003/07/29  2004/03/17 08:59:09  SAHOO
          **End**
```

Figure 8-29 Scroll member list selected

When the “Scroll member list” option is deselected, the member list does not scroll, as shown in Figure 8-30 on page 214.

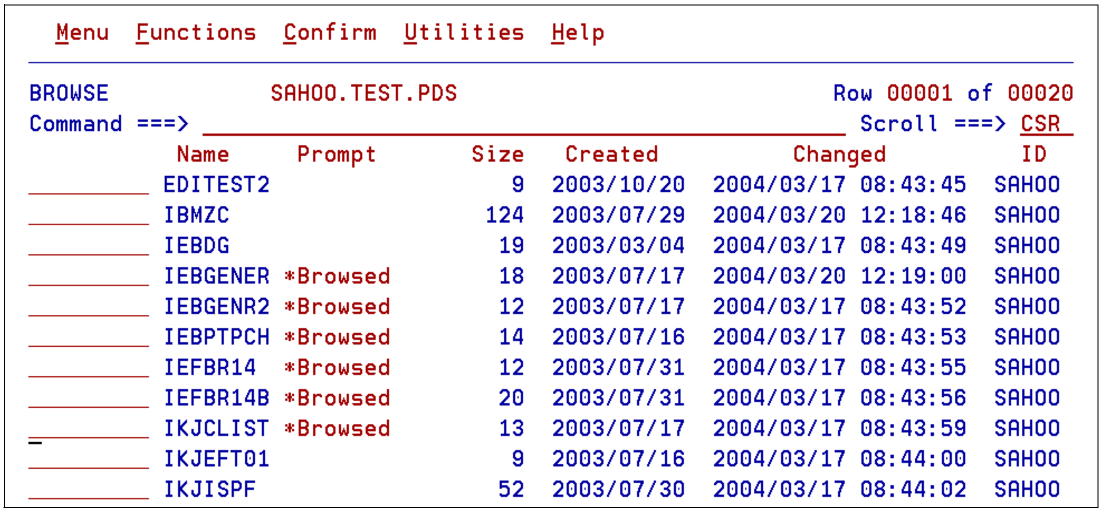


```
  Menu  Functions  Confirm  Utilities  Help
BROWSE          SAHOO.TEST.PDS                     Row 00001 of 00020
Command ===> ________________________________________  Scroll ===> PAGE
          Name     Prompt       Size  Created          Changed          ID
_________ EDITEST2                 9  2003/10/20  2004/03/17 08:43:45  SAHOO
_________ IBMZC                  124  2003/07/29  2004/03/20 12:18:46  SAHOO
_________ IEBDG                   19  2003/03/04  2004/03/17 08:43:49  SAHOO
_________ IEBGENER *Browsed       18  2003/07/17  2004/03/20 12:19:00  SAHOO
_________ IEBGENR2                12  2003/07/17  2004/03/17 08:43:52  SAHOO
_________ IEBPTPCH                14  2003/07/16  2004/03/17 08:43:53  SAHOO
_________ IEFBR14                 12  2003/07/31  2004/03/17 08:43:55  SAHOO
```

Figure 8-30 Scroll member list deselected - single member selected

With the “Scroll member list” option deselected, the scroll behavior depends on the number of members selected for the operation before you press Enter.

- If only single selections are allowed and the selection is contained in the last screen displayed before the Enter key was pressed, the member list is not scrolled and the cursor is positioned in front of the selected member.
- If only single selections are allowed and the selection is not contained in the last screen displayed before the Enter key was pressed, the member list is redisplayed with the last screen containing the selected member displayed at the top and the cursor positioned in front of the selected member.
- If multiple selections are allowed and the last selected member is contained in the last screen displayed before the Enter key was pressed, the member list is not scrolled and the cursor is positioned in front of the last selected member. This is shown in Figure 8-31 on page 214.
- If multiple selections are allowed and the last selection is not contained in the last screen displayed before the Enter key was pressed, the member list is redisplayed with the last selected member scrolled to the top and the cursor is positioned in front of the last selected member.



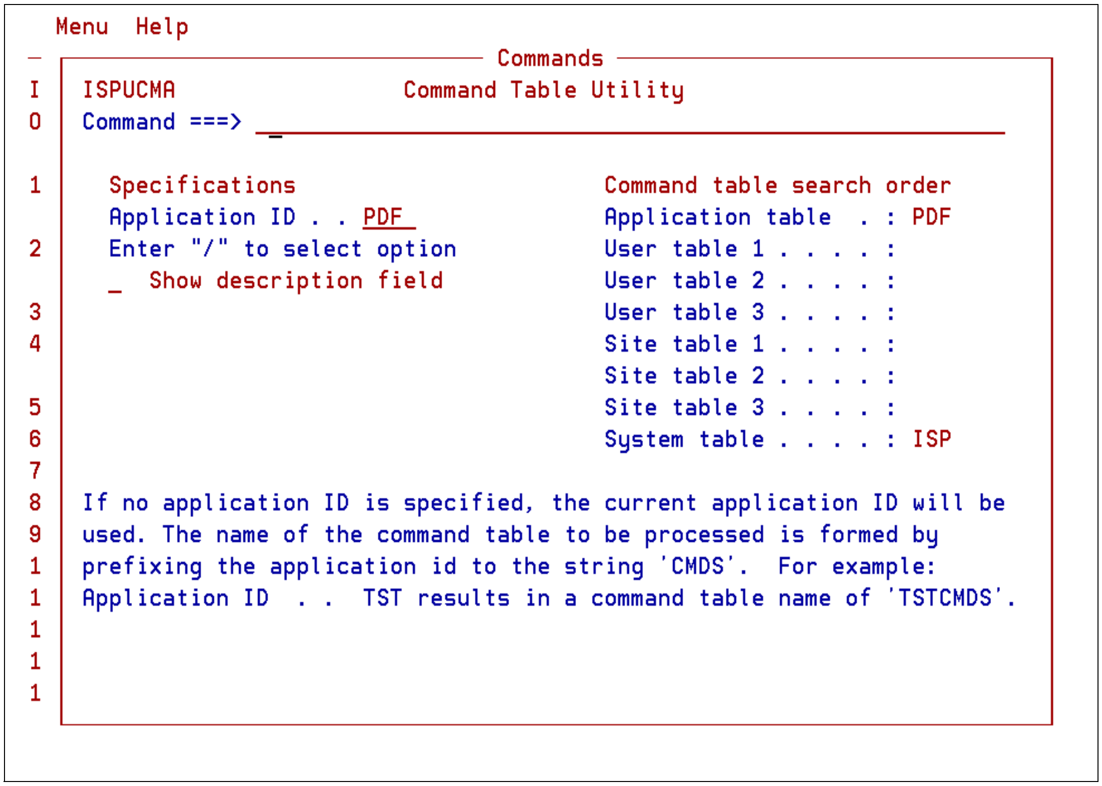
```
  Menu  Functions  Confirm  Utilities  Help
BROWSE          SAHOO.TEST.PDS                     Row 00001 of 00020
Command ===> ________________________________________  Scroll ===> CSR
          Name     Prompt       Size  Created          Changed          ID
_________ EDITEST2                 9  2003/10/20  2004/03/17 08:43:45  SAHOO
_________ IBMZC                  124  2003/07/29  2004/03/20 12:18:46  SAHOO
_________ IEBDG                   19  2003/03/04  2004/03/17 08:43:49  SAHOO
_________ IEBGENER *Browsed       18  2003/07/17  2004/03/20 12:19:00  SAHOO
_________ IEBGENR2 *Browsed       12  2003/07/17  2004/03/17 08:43:52  SAHOO
_________ IEBPTPCH *Browsed       14  2003/07/16  2004/03/17 08:43:53  SAHOO
_________ IEFBR14  *Browsed       12  2003/07/31  2004/03/17 08:43:55  SAHOO
_________ IEFBR14B *Browsed       20  2003/07/31  2004/03/17 08:43:56  SAHOO
_________ IKJCLIST *Browsed       13  2003/07/17  2004/03/17 08:43:59  SAHOO
_________ IKJEFT01                 9  2003/07/16  2004/03/17 08:44:00  SAHOO
_________ IKJISPF                 52  2003/07/30  2004/03/17 08:44:02  SAHOO
```

Figure 8-31 Scroll member list deselected - multiple members selected

8.8 Additional command tables

The Command Table Utility is used to create or change application command tables. A command table contains the specification of general commands that can be entered from any panel during the execution of an application. Command tables are identified by application ID, and are maintained in the ISPF table input library.

The number of both USER and site command tables has been increased from 1 to 3. This provides greater flexibility, especially for customers operating in a sysplex environment. Separate command tables can be defined for individual systems in the sysplex. This pop-up panel is displayed using ISPF option 3.9 and is shown in Figure 8-32.



```
  Menu  Help
 _                                  Commands
 I   ISPUCMA                 Command Table Utility
 O   Command ===> _
 
 1     Specifications                       Command table search order
       Application ID . . PDF               Application table  . : PDF
 2     Enter "/" to select option           User table 1 . . . . :
       _  Show description field            User table 2 . . . . :
 3                                          User table 3 . . . . :
 4                                          Site table 1 . . . . :
                                            Site table 2 . . . . :
 5                                          Site table 3 . . . . :
 6                                          System table . . . . : ISP
 7
 8   If no application ID is specified, the current application ID will be
 9   used. The name of the command table to be processed is formed by
 1   prefixing the application id to the string 'CMDS'.  For example:
 1   Application ID  . .  TST results in a command table name of 'TSTCMDS'.
 1
 1
 1
```

Figure 8-32 Command table utility

- In order to support up to 3 user and 3 site command tables, the Configuration Table keywords APPLID_FOR_USER_COMMAND_TABLE and APPLID_FOR_SITE_COMMAND_TABLE now allow up to 3 applid values to be specified. For example:

  ```
  APPLID_FOR_USER_COMMAND_TABLE = (usr1[,usr2{,usr3]])
  APPLID_FOR_SITE_COMMAND_TABLE = (sit1[,sit2{,sit3]])
  ```

- To provide added flexibility in a sysplex environment, an applid value can be based upon the current system name stored in ISPF dialog variable ZSYSID using the special format:

  *, *m or *m:n

  where “m” and “n” are the start and end positions within the 8 character system name. For example, if `ZSYSID = APSY1ZOS` and `APPLID_FOR_SITE_COMMAND_TABLE = (*3:5)` then `Site command table 1 = SY1CMDS`.

- Since the system name can be up to 8 characters, `m` and `n` are the start and end positions within the system name used to determine the application ID for the user command table.

- The values for m and n must be in the range 1 to 8, with m less than or equal to n and the difference in the values being no more than 3. The default value for m is 1, and n is m+3, to a maximum value of 8.
- The ISPF Configuration Utility ISPCCONF has been updated to support these new options.
- The user and site command table applid values are stored in the following ISPF system variables:

ZUCTPREF	First user command table prefix
ZUCTPRE2	Second user command table prefix
ZUCTPRE3	Third user command table prefix
ZSCTPREF	First site command table prefix
ZSCTPRE2	Second site command table prefix
ZSCTPRE3	Third site command table prefix

- The values that can be specified for the SITE_COMMAND_TABLE_SEARCH ORDER_SETTING keyword continue to be BEFORE and AFTER.
  - When BEFORE is specified the command table search order is:
    i. Application
    ii. User (1 to 3)
    iii. Site (1 to 3)
    iv. System
  - When AFTER is specified the command table search order is:
    i. Application
    ii. User (1 to 3)
    iii. System
    iv. Site (1 to 3

8.8.1 System symbolics in temporary data set names

To allow the same user ID to log on to multiple systems in a sysplex environment, ISPF now supports an additional qualifier in the names used for the temporary log, list, and control data sets.

This new qualifier can be derived from the values in system symbolic variables (&SYSNAME).

This additional qualifier is appended to the ISPF log, list, and temporary control data set names. The qualifier comes after the ISPF assigned prefix, but before the suffix area. If Exit 16 is active, this qualifier is part of the 26-byte prefix area passed to the exit.

The new qualifier is defined using the ISPF Configuration Table option ISPF_TEMPORARY_DATA_SET_QUALIFIER. The valid values for this option can be any of the following:

- NONE (default - no qualifier).
- A valid data set qualifier, comprising 1 to 8 alphanumeric characters, the first being alphabetic (not numeric).
- A string containing 1 or more system symbolic variables. The string may be up to 24 characters in length, but when resolved is truncated to 8 characters. Other characters may be included between the symbolic variables, providing they are alphanumeric characters,

and the first character is always non-numeric. The use of any of the date and time symbols would require an alphabetic character before the symbol name to ensure a valid qualifier.

The ISPF Configuration Utility ISPCCONF has been updated to support this new option.

8.9 Configure min and max scroll amounts

ISPF now allows sites to configure the minimum and maximum values that can be set by users for the scroll amount.

ISPF issues an error message if a user specifies a scroll amount outside the bounds of the minimum and maximum scroll amount values configured for the system.

This was done to help avoid “frozen” screen problems, when the user had accidently set the panel scroll amount to 0 and then saw no change on the screen when they entered a scroll command (UP, DOWN, LEFT, or RIGHT). Configuring a minimum scroll amount of 1 would avoid this type of problem.

The minimum scroll amount value is stored in ISPF system dialog variable ZXSMIN. The minimum scroll amount keyword is SCROLL_MIN and the default value is 0.

The maximum scroll amount value is stored in ISPF system dialog variable ZXSMAX.The maximum scroll amount keyword is SCROLL_MAX and the default value is 9999.

The ISPF configuration utility ISPCCONF has been updated to support these new option.

8.10 Configuration table identification

The ISPF Configuration Table is a load module which is assembled from information stored in a keyword source file. This enhancement allows a user to identify the configuration table they are currently running with, the source file it was assembled from, and when it was assembled.

The information identifying the configuration table is stored in the table. ISPF also places this information in system dialog variables in the SHARED pool. The identification information is stored in the following ISPF system dialog variables:

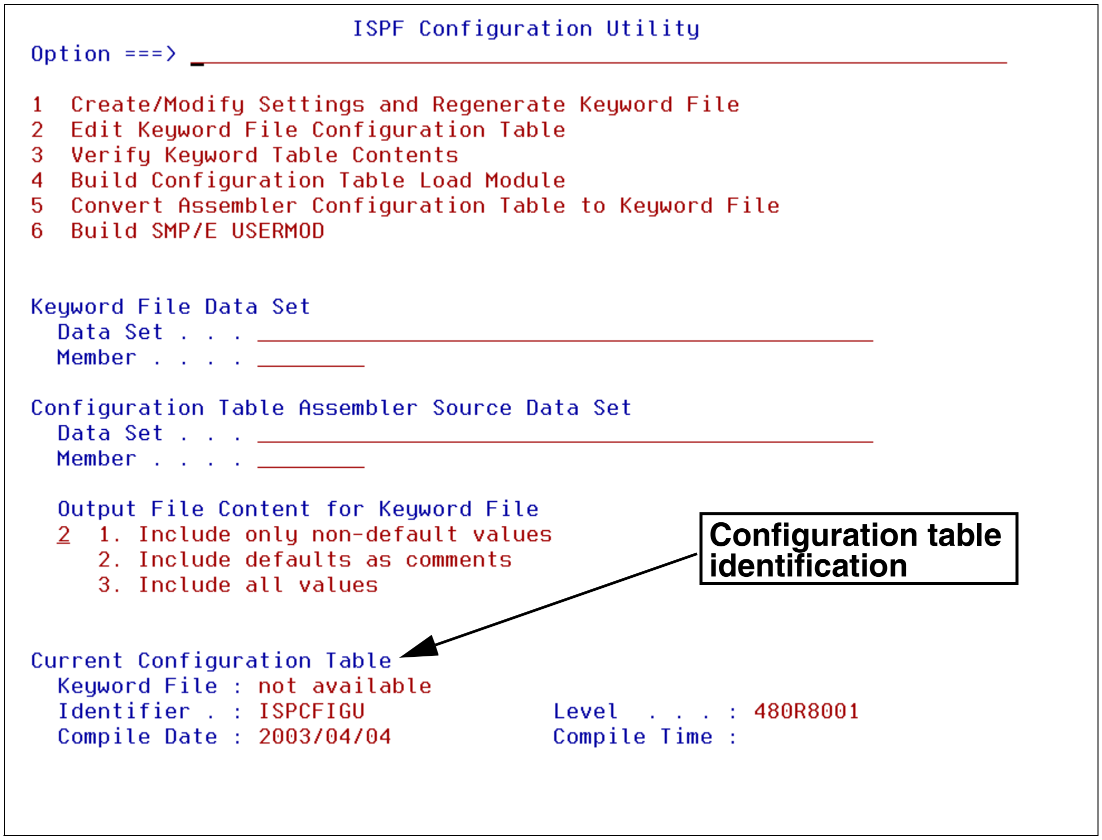
ZCFGMOD	Configuration table module name
ZCFGLVL	Configuration table level
ZCFGKSRC	Keyword source data set and member name
ZCFGCMPD	Compile date in the form YYYY/MM/DD
ZCFCMPT	Compile time in the form HH.MM

The configuration table identification information is written to the trace data set generated by the ISPVCALL diagnostic utility.

```
=========================< PDF Configuration Table >============================
     Table id........... ISPCFIGU      Table level id...... 480R8001
     Keyword Source...... VANDYKE.CONFIG.TABLE(OSDEV)
       Compile date...... 20030707        Compile time...... 13.47
```

Figure 8-33 ISPVCALL trace

The configuration table identification information can be displayed by the ISPF configuration utility ISPCCONF (command **TSO ISPCCONF**) as shown in Figure 8-34.



```
                        ISPF Configuration Utility
Option ===> _

1  Create/Modify Settings and Regenerate Keyword File
2  Edit Keyword File Configuration Table
3  Verify Keyword Table Contents
4  Build Configuration Table Load Module
5  Convert Assembler Configuration Table to Keyword File
6  Build SMP/E USERMOD


Keyword File Data Set
  Data Set . . .
  Member . . . .

Configuration Table Assembler Source Data Set
  Data Set . . .
  Member . . . .

  Output File Content for Keyword File
  2  1. Include only non-default values
     2. Include defaults as comments
     3. Include all values


Current Configuration Table
  Keyword File : not available
  Identifier . : ISPCFIGU              Level   . . . : 480R8001
  Compile Date : 2003/04/04            Compile Time :
```

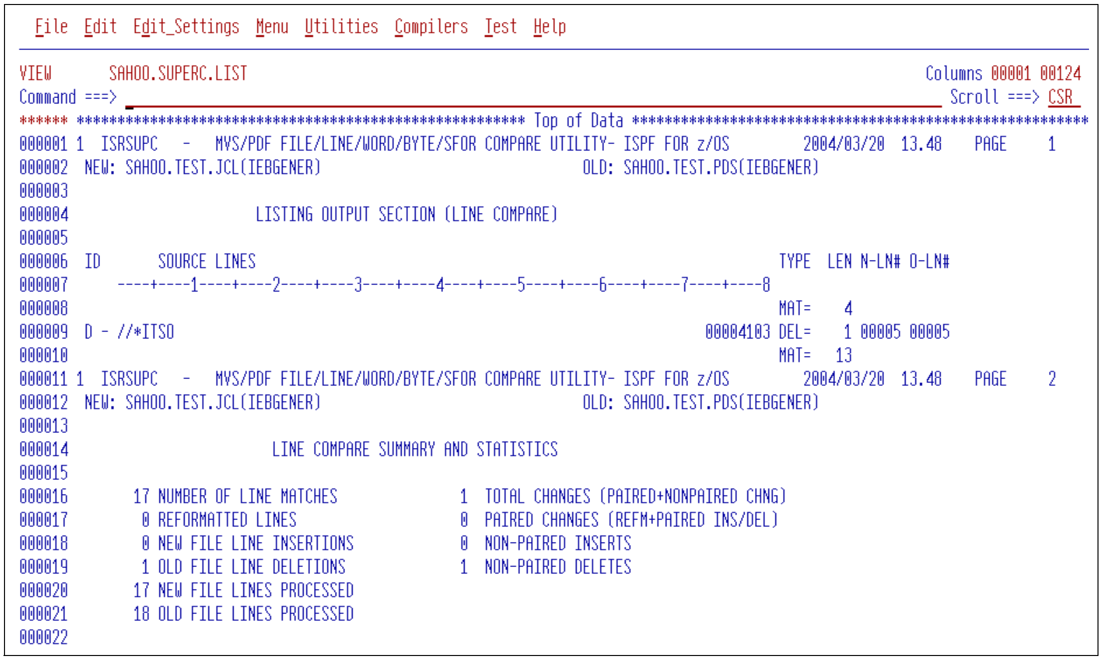
Configuration table identification

Figure 8-34 ISPF configuration utility

8.11 SuperC command output highlighting

The ISPF Edit HILITE command and macro have been enhanced to highlight various elements of the listing generated by the SuperC utility.

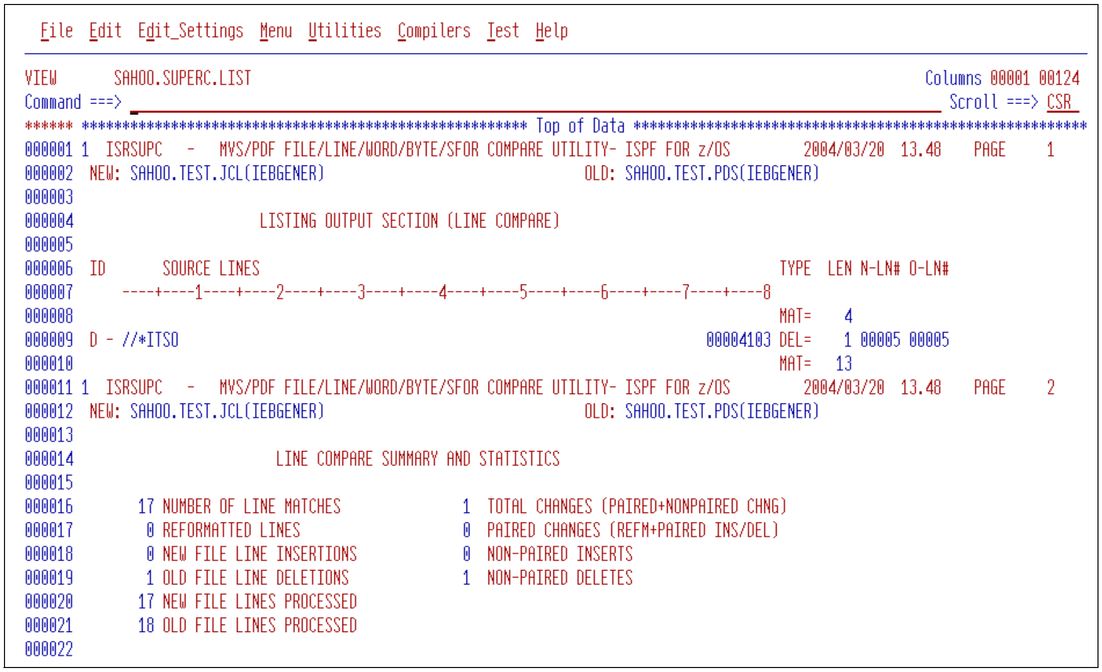
An example of SuperC listing with the HILITE OFF command issued and the output is shown in Figure 8-35 on page 219.



```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW      SAHOO.SUPERC.LIST                                                                    Columns 00001 00124
Command ===> ____________________________________________________________________________________ Scroll ===> CSR
****** ******************************************************* Top of Data *********************************************************
000001 1  ISRSUPC   -   MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF FOR z/OS        2004/03/20  13.48    PAGE     1
000002  NEW: SAHOO.TEST.JCL(IEBGENER)                                OLD: SAHOO.TEST.PDS(IEBGENER)
000003
000004                      LISTING OUTPUT SECTION (LINE COMPARE)
000005
000006  ID       SOURCE LINES                                                        TYPE  LEN N-LN# O-LN#
000007      ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
000008                                                                               MAT=    4
000009  D - //*ITSO                                                         00004103 DEL=    1 00005 00005
000010                                                                               MAT=   13
000011 1  ISRSUPC   -   MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF FOR z/OS        2004/03/20  13.48    PAGE     2
000012  NEW: SAHOO.TEST.JCL(IEBGENER)                                OLD: SAHOO.TEST.PDS(IEBGENER)
000013
000014                        LINE COMPARE SUMMARY AND STATISTICS
000015
000016        17 NUMBER OF LINE MATCHES               1  TOTAL CHANGES (PAIRED+NONPAIRED CHNG)
000017         0 REFORMATTED LINES                    0  PAIRED CHANGES (REFM+PAIRED INS/DEL)
000018         0 NEW FILE LINE INSERTIONS             0  NON-PAIRED INSERTS
000019         1 OLD FILE LINE DELETIONS              1  NON-PAIRED DELETES
000020        17 NEW FILE LINES PROCESSED
000021        18 OLD FILE LINES PROCESSED
000022
```

Figure 8-35 SuperC output with HILITE OFF

Now, we issue the command HILITE ON and the display output changes as shown in Figure 8-36.



```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW      SAHOO.SUPERC.LIST                                                                    Columns 00001 00124
Command ===> ____________________________________________________________________________________ Scroll ===> CSR
****** ******************************************************* Top of Data *********************************************************
000001 1  ISRSUPC   -   MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF FOR z/OS        2004/03/20  13.48    PAGE     1
000002  NEW: SAHOO.TEST.JCL(IEBGENER)                                OLD: SAHOO.TEST.PDS(IEBGENER)
000003
000004                      LISTING OUTPUT SECTION (LINE COMPARE)
000005
000006  ID       SOURCE LINES                                                        TYPE  LEN N-LN# O-LN#
000007      ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
000008                                                                               MAT=    4
000009  D - //*ITSO                                                         00004103 DEL=    1 00005 00005
000010                                                                               MAT=   13
000011 1  ISRSUPC   -   MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF FOR z/OS        2004/03/20  13.48    PAGE     2
000012  NEW: SAHOO.TEST.JCL(IEBGENER)                                OLD: SAHOO.TEST.PDS(IEBGENER)
000013
000014                        LINE COMPARE SUMMARY AND STATISTICS
000015
000016        17 NUMBER OF LINE MATCHES               1  TOTAL CHANGES (PAIRED+NONPAIRED CHNG)
000017         0 REFORMATTED LINES                    0  PAIRED CHANGES (REFM+PAIRED INS/DEL)
000018         0 NEW FILE LINE INSERTIONS             0  NON-PAIRED INSERTS
000019         1 OLD FILE LINE DELETIONS              1  NON-PAIRED DELETES
000020        17 NEW FILE LINES PROCESSED
000021        18 OLD FILE LINES PROCESSED
000022
```

Figure 8-36 SuperC output with HILITE ON

8.12 SCLM enhancements

Software Configuration and Library Manager (SCLM) is used to create, control, maintain, and track software components for a project. SCLM runs in the user's address space and there is no started task. The SCLM project database consists of a series of related ISPF libraries (partitioned data sets). These contain source and non-source software components. SCLM project definition and control information is contained in an assembled and linked PROJECTDEFS data set. SCLM project cross-reference and accounting data sets are VSAM clusters.

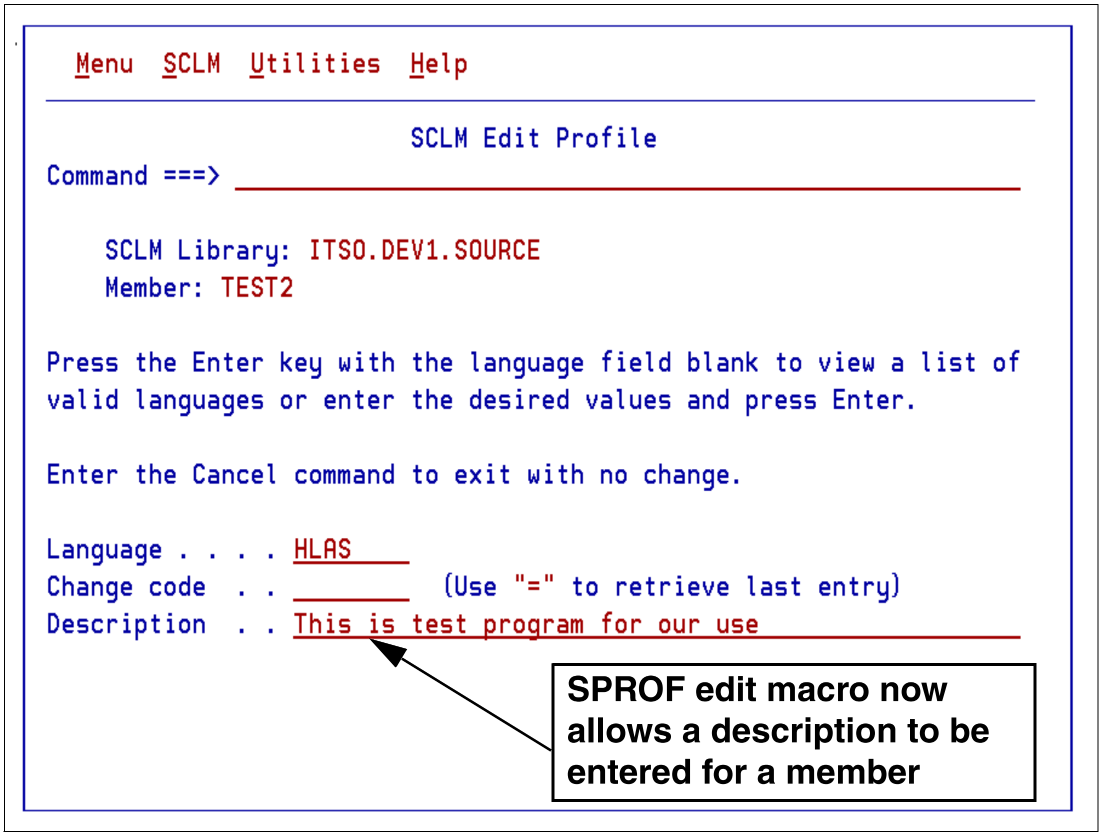
The ISPF SCLM component in z/OS V1R5 contains the following new functions.

8.12.1 Member description

SCLM has been enhanced to allow a description to be entered for members within an SCLM project.

While editing the member the SPROF edit macro is used; this provides the facility to enter member description.

SCLM edit profile now allows a description to be entered for the member name as shown in Figure 8-37. Use option 2 from the SCLM primary option menu to display this panel.

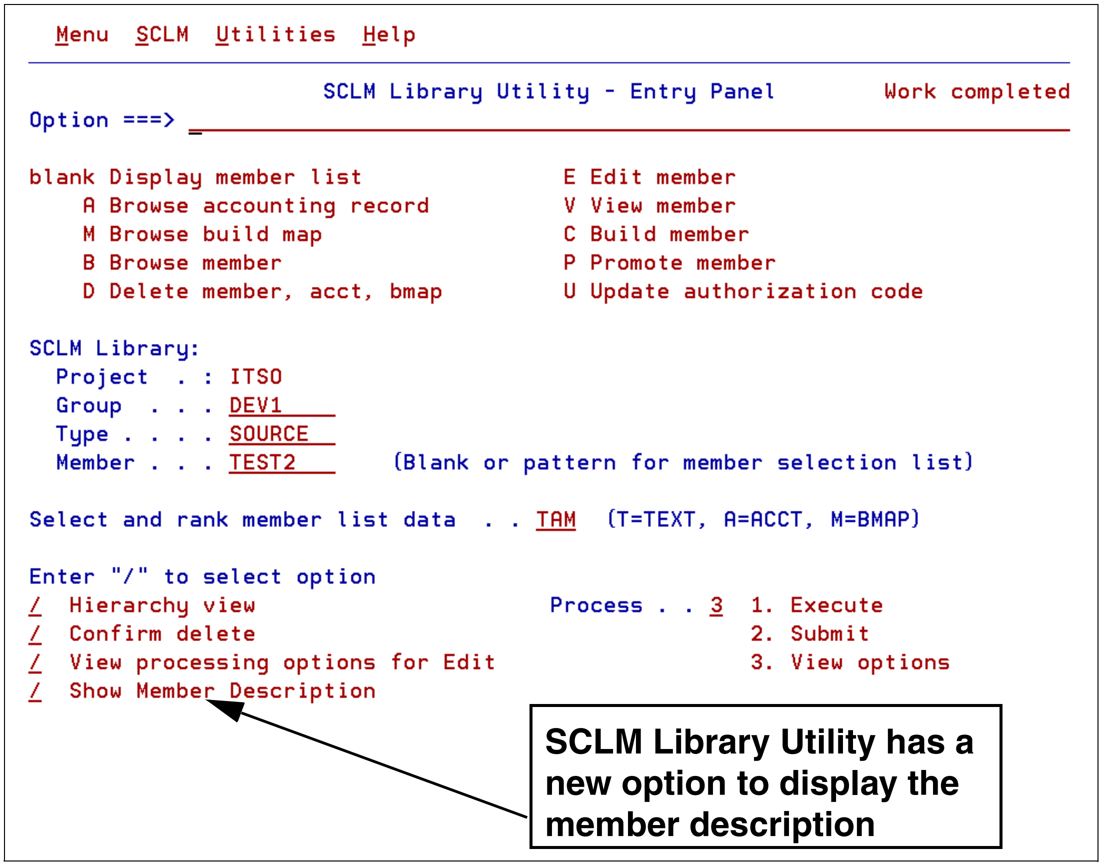


```
                         SCLM Edit Profile
Command ===> ________________________________________________________

    SCLM Library: ITSO.DEV1.SOURCE
    Member: TEST2

Press the Enter key with the language field blank to view a list of
valid languages or enter the desired values and press Enter.

Enter the Cancel command to exit with no change.

Language . . . . HLAS
Change code  . . ________  (Use "=" to retrieve last entry)
Description  . . This is test program for our use
```

SPROF edit macro now allows a description to be entered for a member

Figure 8-37 SCLM edit profile

The SCLM Library Utility panel has been changed to add a new option “Show Member Description” as shown in Figure 8-38 on page 221. Use option 3.1 in the SCLM main panel to display this panel.

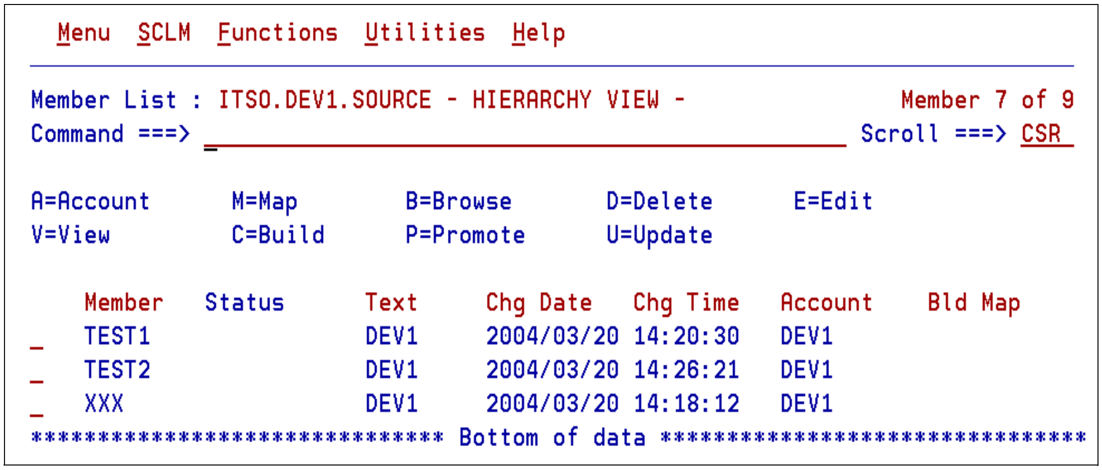


```
  Menu  SCLM  Utilities  Help
 ------------------------------------------------------------------------------
                     SCLM Library Utility - Entry Panel        Work completed
Option ===> _

blank Display member list                  E Edit member
    A Browse accounting record             V View member
    M Browse build map                     C Build member
    B Browse member                        P Promote member
    D Delete member, acct, bmap            U Update authorization code

SCLM Library:
  Project  . : ITSO
  Group  . . . DEV1
  Type . . . . SOURCE
  Member . . . TEST2       (Blank or pattern for member selection list)

Select and rank member list data  . . TAM  (T=TEXT, A=ACCT, M=BMAP)

Enter "/" to select option
/  Hierarchy view                         Process . . 3  1. Execute
/  Confirm delete                                        2. Submit
/  View processing options for Edit                      3. View options
/  Show Member Description
```

SCLM Library Utility has a new option to display the member description

Figure 8-38   SCLM Library Utility - Entry Panel

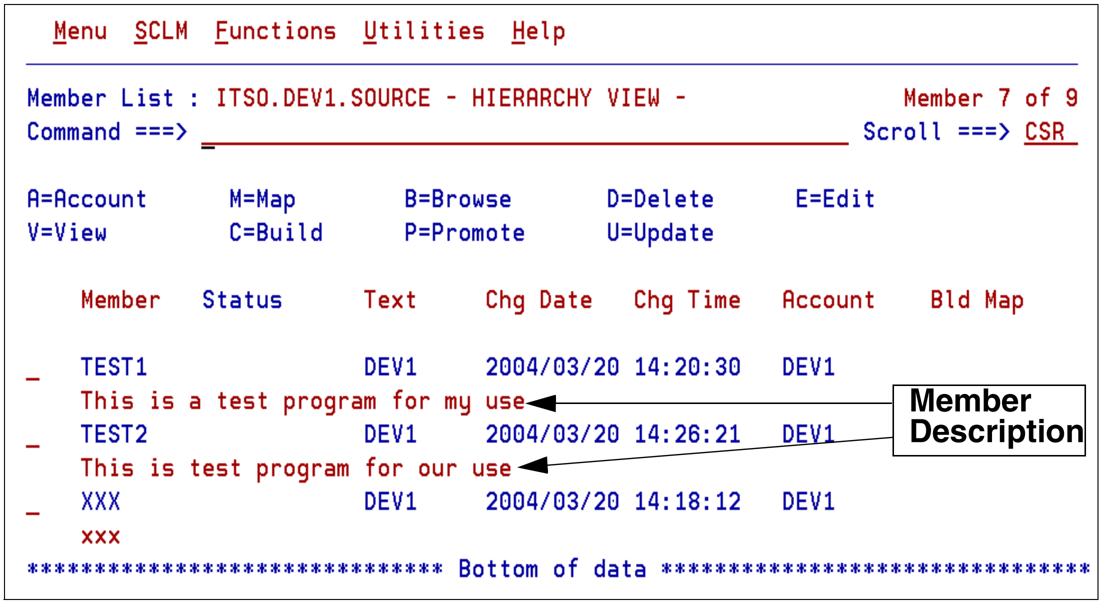
An example of the resulting panel when member list with option “Show Member Description” deselected is shown in Figure 8-39.



```
  Menu  SCLM  Functions  Utilities  Help
 ------------------------------------------------------------------------------
Member List : ITSO.DEV1.SOURCE - HIERARCHY VIEW -                Member 7 of 9
Command ===> _                                                 Scroll ===> CSR

A=Account      M=Map        B=Browse      D=Delete      E=Edit
V=View         C=Build      P=Promote     U=Update

    Member    Status      Text      Chg Date   Chg Time   Account    Bld Map
_   TEST1                 DEV1      2004/03/20 14:20:30   DEV1
_   TEST2                 DEV1      2004/03/20 14:26:21   DEV1
_   XXX                   DEV1      2004/03/20 14:18:12   DEV1
******************************* Bottom of data ********************************
```

Figure 8-39   Member list with option Show Member Description deselected

A display of the member list with option “Show Member Description” selected is shown in Figure 8-40 on page 222.



```
  Menu  SCLM  Functions  Utilities  Help
Member List : ITSO.DEV1.SOURCE - HIERARCHY VIEW -              Member 7 of 9
Command ===> _                                                Scroll ===> CSR

A=Account      M=Map        B=Browse      D=Delete      E=Edit
V=View         C=Build      P=Promote     U=Update

    Member   Status      Text     Chg Date   Chg Time   Account    Bld Map

_   TEST1                DEV1     2004/03/20 14:20:30   DEV1
    This is a test program for my use
_   TEST2                DEV1     2004/03/20 14:26:21   DEV1
    This is test program for our use
_   XXX                  DEV1     2004/03/20 14:18:12   DEV1
    xxx
******************************* Bottom of data ********************************
```

Figure 8-40 Member list with option Show Member Description selected

8.12.2 Project information API

The new API SCLMINFO enables user applications to obtain information about the project currently being processed within SCLM.

Syntax: **FLMCMD SCLMINFO, *project*, [*prj_def*]**

Where:

project	Project name
prj_def	Project definition name

This returns information in the following ISPF dialog variables:

ZSCIPROJ	Project
ZSCIPDEF	Alternate project
ZSCIGRP	List of all the groups specified for the project
ZSCITYPE	List of all the types specified for the project
ZSCILANG	List of all the languages specified for the project
ZSCISVER	SCLM version ID for the project
ZSCITMST	Timestamp (date and time the project was generated)

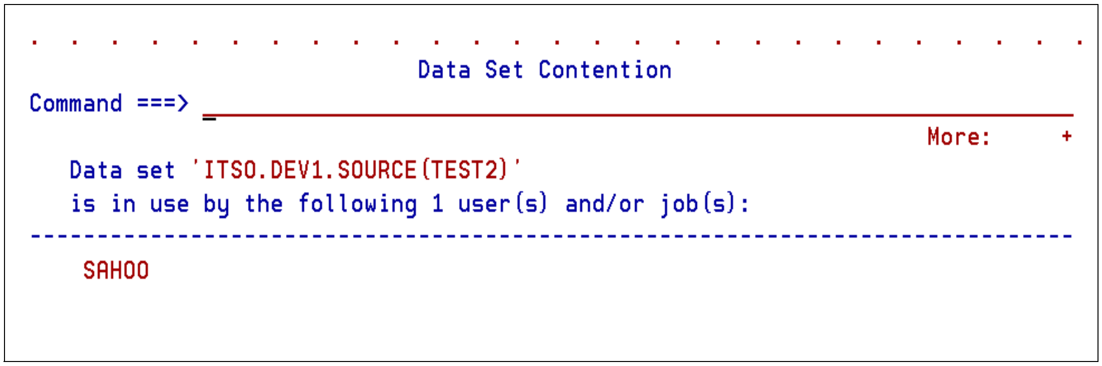
8.12.3 SCLM audit version utility

The audit and version utility enables you to audit SCLM operations on SCLM-controlled members and create versions of editable members. Using the audit and version utility, you can view the audit information for a member and retrieve a version to a sequential data set not controlled by SCLM, to a partitioned data set not controlled by SCLM, or to a SCLM-controlled development group. This utility also enables you to delete audit and version information from the database.

The audit and version utility now allows a wildcard to be entered in the Type field. This enhancement is in line with the SCLM Library Edit and View processes.

8.12.4 Help for “Member in use” in SCLM edit

During SCLM edit session, if a member is being used by another process, additional panels are presented showing exactly which other users have ENQs on the member in question. Figure 8-41 shows that the data set ITSO.DEV1.SOURCE(TEST2) is currently used by user SAHOO.



```
 . . . . . . . . . . . . . . . . . . . . . . . . . . .
                              Data Set Contention
Command ===> _
                                                            More:     +
   Data set 'ITSO.DEV1.SOURCE(TEST2)'
   is in use by the following 1 user(s) and/or job(s):
------------------------------------------------------------------------------
    SAHOO
```

Figure 8-41 Panel showing member use by other process

8.13 Other miscellaneous ISPF changes

There are some miscellaneous changes implemented in z/OS V1R5 ISPF. They are as follows:

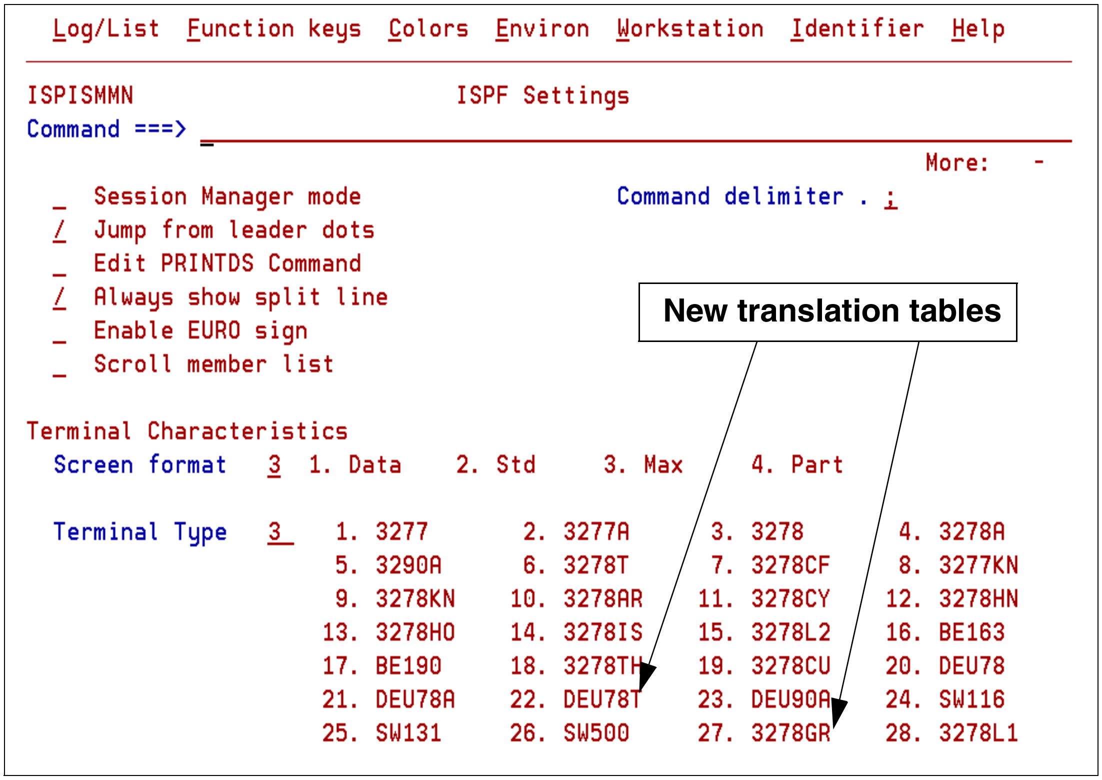
- Changes to translation table
- Changes to EXIT 11
- HFS command support from data set utility
- Remove obsolete command from ISPTCM
- New variable @@FLMCAA

8.13.1 Changes to translation tables

Changes have been done to translation tables for terminal types, including:

- New Greek translation table and associated terminal type 3278GR.
- New TEXT translation table and associated terminal type DEU78T with support for the Euro sign.
- English/Swiss translation table changed to allow for uppercase translations of German umlaut.

Figure 8-42 on page 224 shows the ISPF Settings panel with the new translation tables.



```
  Log/List  Function keys  Colors  Environ  Workstation  Identifier  Help
 ------------------------------------------------------------------------------
 ISPISMMN                        ISPF Settings
 Command ===> _
                                                                    More:    -
    _  Session Manager mode                    Command delimiter . ;
    /  Jump from leader dots
    _  Edit PRINTDS Command
    /  Always show split line
    _  Enable EURO sign
    _  Scroll member list

 Terminal Characteristics
   Screen format  3  1. Data     2. Std     3. Max     4. Part

   Terminal Type  3_    1. 3277       2. 3277A      3. 3278       4. 3278A
                        5. 3290A      6. 3278T      7. 3278CF     8. 3277KN
                        9. 3278KN    10. 3278AR    11. 3278CY    12. 3278HN
                       13. 3278HO    14. 3278IS    15. 3278L2    16. BE163
                       17. BE190     18. 3278TH    19. 3278CU    20. DEU78
                       21. DEU78A    22. DEU78T    23. DEU90A    24. SW116
                       25. SW131     26. SW500     27. 3278GR    28. 3278L1
```

New translation tables

Figure 8-42 ISPF settings panel

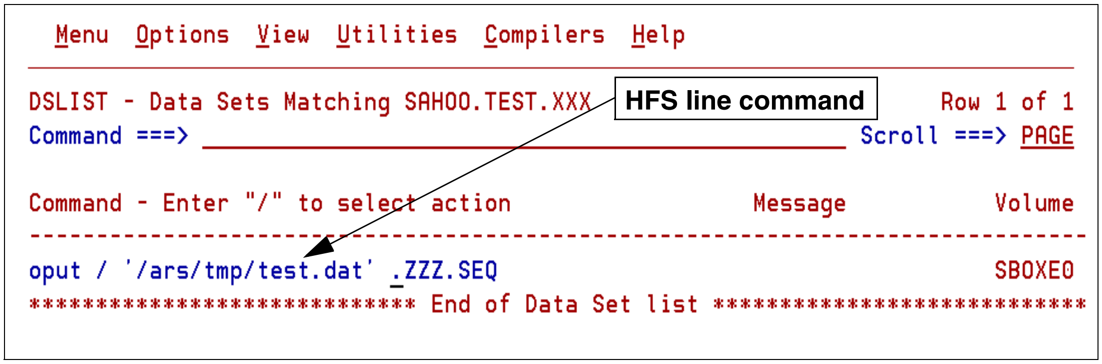
8.13.2 Exit 11

The ISPF installation-wide exit 11 (Logical Screen End) helps to gather accounting and monitoring information for each ISPF logical screen. It gives control for both normal and abnormal termination of logical screens. This exit has been enhanced to now supply the next logical screen to be displayed. This enhancement implements the SUG APAR OW49665.

8.13.3 HFS command support from data set list utility

An HFS command can now be issued as a line command against data sets displayed in the data set list (Option 3.4). This enhancements implements FIN APAR OW51425. Figure 8-43 shows a sample of issuing the **oput** command against a data set under ISPF Option 3.4

.



```
  Menu  Options  View  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 DSLIST - Data Sets Matching SAHOO.TEST.XXX                          Row 1 of 1
 Command ===> ______________________________________________ Scroll ===> PAGE

 Command - Enter "/" to select action                  Message           Volume
 -------------------------------------------------------------------------------
 oput / '/ars/tmp/test.dat'  .ZZZ.SEQ                                    SBOXE0
 ***************************** End of Data Set list ****************************
```

HFS line command

Figure 8-43 Issue HFS command under ISPF option 3.4

8.13.4 Remove obsolete commands from ISPTCM

As of APAR OY23474, IDCAMS CHKLIST and CKLST commands were removed. However, the TSO command table, ISPTCM, still contains entries for these two commands.

The obsolete IDCAMS commands **CHKLIST** and **CKLST** have been removed from the ISPF TSO command table (ISPTCM) in z/OS V1R5. This enhancement implements FIN APAR OW51634.

8.13.5 Variable @@FLMCAA

A currently entered change code is not available, via any SCLM variable, at the parse phase of the translator. It is required by some users to use the entered change code during processes initiated during the parse phase, rather than after the member has been saved.

A new variable, @@FLMCAA, has been added which contains the current change code during the parse phase.

9

z/OS V1R5 Workload Manager (WLM)

The workload manager component of z/OS provides a solution for managing workload distribution, workload balancing, and distributing system resources to competing workloads. z/OS workload management is the combined cooperation of various subsystems such as CICS, IMS/ESA®, JES, APPC, TSO/E, UNIX System Services, DDF, and DB2, with the z/OS workload management (WLM) component. Using the z/OS Workload Manager administrative interface, you define performance goals and assign a business importance to each goal. You define these goals for workloads in business terms, and the system decides how much resource, such as CPU and storage, should be given to the workload to meet its goal.

This chapter describes the enhancements to WLM introduced in z/OS V1 R5. They include:

- WebSphere Dynamic Application Environment API
- High Virtual Shared Area support
- Importance-based initiator dispatch priority control
- WLM Sub-capacity Reporting Tool (SCRT) support
- Native WLM support for Enterprise Workload Manager (eWLM)

9.1 WLM enhancements for z/OS V1R5 overview

Workload manager for z/OS V1R5 responds positively to specific customer requests for enhancements and addresses IBM internally generated requirements by providing:

- A Dynamic Application Environment for WebSphere
- Support for the High Shared Virtual Area
- Importance-based initiator dispatch priority control
- Improvement in sub-capacity reporting statistical generation
- WLM additions and enhancements to support the Enterprise Workload Manager (eWLM)

9.2 WebSphere Dynamic Application Environment API

Prior to z/OS V1R5, WebSphere requires that application environments be defined twice: once to the WebSphere environment itself via its customization dialog, and again to WLM though the administrative application. This requirement leads to duplication of effort, unnecessary complication of implementation tasks, and increased potential for configuration errors and mismatches between the definitions.

Under z/OS V1R5, it is no longer necessary to use the WLM administrative application to define application environments to exploit WLM's Queue and Sever Management facilities. Applications such as WebSphere Enterprise Edition, MQSeries Workflow, and DB2 among others are current exploiters of WLM's Queue and Server Management.

Application environment definitions provide the information WLM needs to start server address spaces and effectively manage the total number of servers in the system. They contain a name for the application environment itself, a designation of subsystem type, the name of the startup JCL procedure for the server address space, and may also contain optional parameters which are passed to the server address space at startup time.

Specifically for WebSphere and other applications that implement this support, the WLM application environments will be built dynamically by the application when they are required, and deleted by the application when they are no longer needed.

9.2.1 Application interfaces

The applications queue managers are now able to use the WebSphere Dynamic Application Environment API provided by z/OS V1R5 to dynamically define application environments after they connect to WLM. The API provides a new definition service, IWMAEDEF, which is exploited by the applications to add or delete dynamic application environments. The IWMAEDEF service defines the application environment using:

- The name to be used for the application environment
- The subsystem type of the application
- The name of the JCL start procedure for the server address spaces
- Optional parameters that will be passed to the server address space at startup
- Optional replication policy that defines if WLM can start multiple server address spaces for the application environment
- Optional work selection policy for requests queued directly to the service class and requests sent to the server address space

IWMAEDEF service

The IWMAEDEF service is used by the application to delete the dynamic application environment when it is no longer needed.

Refer to *z/OS MVS Programming: Workload Management Services*, SA22-7619 for more information on the IWMAEDEF service.

WebSphere runtime environment

The WebSphere runtime environment may also be embedded within another application. This can lead to a problem since the embedded instance could be started many times from within that application, thus attempting to generate identically named application environments. Prior to z/OS V1R5, WLM support does not allow multiple application environments with the same name on the same system.

IWMCONN connect service

The IWMCONN connect service for server manager address spaces has been upgraded to allow the server manager to connect to either a static or a dynamic application environment. An optional new 8 character NODENM parameter of the IWMCONN service is made available to allow the application to specify the entity in which the work manager is executing. This provides a means of distinguishing between multiple copies of a runtime within the same application executing in the same system. An example of this would be one copy of WebSphere embedded in an application and another copy of WebSphere running as a standalone Web application.

Refer to *z/OS MVS Programming: Workload Management Services*, SA22-7619 for more information on the IWMCONN service.

IWMQINS service

The queue manager insert service IWMQINS has been enhanced to allow queue managers to insert requests to static and dynamic application environments through the new DYNAMIC=YES|NO optional subparameter of the APPLENV parameter.

Refer to *z/OS MVS Programming: Workload Management Services*, SA22-7619 for more information on the IWMQINS service.

Security interfaces

The validity of server address spaces can be checked at the time of creation through an SAF product such as RACF. The STARTED and SERVER resource classes may be used to restrict access to the application environment servers. The STARTED class can be used to assign a user ID to the MVS procname used for the server's started procedure. The SERVER class can then be used to authorize that user ID as a valid server for the application. If the queue manager specifies the node name, WLM requires that the value of the new NODENM parameter in the IWMCONN service be appended to the server's profile in the SERVER class. The new format of the SERVER profile is:

```
subsystem_type.subsystem_name.application_environment_name.node_name
```

> **Note:** The value for node_name is appended to the three qualifiers contained in the pre-z/OS V1R5 Server profile to prevent potential conflict.

9.2.2 Operational interfaces

The MVS **DISPLAY** command can be used to list a specific application environment or all application environments on a console. The MVS **VARY** command can be used to alter the operational state of an application environment. While the effect and scope and output of the commands has not changed, they have been enhanced to process dynamic application environments.

DISPLAY command for dynamic application environments

The following command can be used to list all or specific dynamic application environments for the system in which the command is entered.

```
DISPLAY WLM,DYNAPPL=[name|*][,options]
   where options can be:
      STYPE=subsystem_type
      SNAME=subsystem_name
      SNODE=nodename
```

To display a specific dynamic application environment specify its name in the DYNAPPL= parameter.

To display all dynamic applications environments specify “*” in the DYNAPPL= parameter.

A wildcard may be used in the DYNAPPL= parameter by specifying a partial string of a dynamic application environment name followed by an “*”. This will display a list of all dynamic application environments whose names begins with the string entered.

The options parameters may be used to further qualify your search criteria and limit the list of dynamic application environments displayed.

To display all dynamic application environments currently in the system enter:

```
D WLM,DYNAPPL=*
```

If there are dynamic application environments currently in the system, messages such as the following are displayed.

```
D WLM,DYNAPPL=*
IWM029I  09.58.37  WLM DISPLAY 602
  DYNAMIC APPL. ENVIRON. NAME      STATE     STATE DATA
  CLUD6                            AVAILABLE
  ATTRIBUTES: PROC=WAS5DS6  SUBSYSTEM TYPE: CB
  SUBSYSTEM NAME: WASD6     NODENAME: CELD6
```

To display specific dynamic application environments currently in the system enter:

```
D WLM,DYNAPPL=CL*
```

This command displays all dynamic application environments currently in the system whose names begin with the characters **CL**. Messages such as the following are displayed.

```
D WLM,DYNAPPL=CL*
IWM029I  09.59.12  WLM DISPLAY 604
  DYNAMIC APPL. ENVIRON. NAME      STATE     STATE DATA
  CLUD6                            AVAILABLE
  ATTRIBUTES: PROC=WAS5DS6  SUBSYSTEM TYPE: CB
  SUBSYSTEM NAME: WASD6     NODENAME: CELD6
```

In the event a **DISPLAY** command is issued for dynamic application environments and there are none in the system that meet the display criteria, messages such as the following will be displayed.

```
D WLM,DYNAPPL=D*
IWM030I DISPLAY FOR D* REJECTED, APPLICATION ENVIRONMENT NOT DEFINED
```

VARY command for dynamic application environments

The following command can be used to alter the state of dynamic application environments.

```
VARY WLM,DYNAPPL=applname,[REFRESH]
                          [QUIESCE|Q]
                          [RESUME]
                          [,OPTIONS]
   where options can be:
      STYPE=subsystem_type
      SNAME=subsystem_name
      SNODE=nodename
```

The **VARY** command has a scope of *system*; only instances of the application executing in the system where the command is entered will be varied. Message IWM031I is issued indicating that a vary action is in progress.

REFRESH and QUIESCE parameters

Active work requests on all server instances for the specified application are allowed to complete before processing the **REFRESH** and **QUIESCE** commands. Message IWM031I is issued every 3 minutes until the command is completed.

Completion of the command will also be delayed for applications that support affinities to local objects residing in the virtual storage of the server address spaces. When these affinities have been terminated, the command processing will continue. Message IWM031I will be issued every 3 minutes until the command is completed.

The REFRESH parameter directs that the specified dynamic application environment's server address spaces be terminated after the current request has been completed, and that new server address spaces be started.

The QUIESCE parameter directs that the specified dynamic application environment's server address spaces be terminated after the current request has been completed. New server address spaces for the specified application may not be started by either WLM or the operator. New work requests for an application that supports queueing will be queued.

The RESUME parameter directs that the dynamic application environment be restarted. New server address spaces will be allowed to start and queued work requests will become eligible to be processed.

The options parameters can be used to uniquely identify the dynamic application environment to be varied. If the system cannot determine which application environment to act on, it will issue message IWM030I and the command will be rejected.

Message IWM032I will be issued when vary processing is complete.

An example of the **VARY** command to refresh a dynamic application named CLUD6 is as follows:

```
V WLM,DYNAPPL=CLUD6,REFRESH
```

An example of the resulting output follows.

```
V WLM,DYNAPPL=CLUD6,REFRESH
IWM031I VARY REFRESH FOR CLUD6 IN PROGRESS
+BBOO0222I WSVR0217I: Stopping application: adminconsole
+BBOO0222I SRVE0170I: Stopping Web Module: adminconsole.
+BBOO0222I WSVR0220I: Application stopped: adminconsole
+BBOO0222I WSVR0217I: Stopping application: JaasTest
+BBOO0222I SRVE0170I: Stopping Web Module: JaasTestWeb.
+BBOO0222I WSVR0220I: Application stopped: JaasTest
+BBOO0222I WSVR0217I: Stopping application: SWIPE
+BBOO0222I SRVE0170I: Stopping Web Module: IBMAuthSSLClient.
+BBOO0222I SRVE0170I: Stopping Web Module: IBMnoWebApp.
+BBOO0222I SRVE0170I: Stopping Web Module: IBMForms.
+BBOO0222I SRVE0170I: Stopping Web Module: IBMEBizWeb.
+BBOO0222I WSVR0220I: Application stopped: SWIPE
+BBOO0222I WSVR0217I: Stopping application: ivtApp
+BBOO0222I SRVE0170I: Stopping Web Module: IVT Application.
+BBOO0222I WSVR0220I: Application stopped: ivtApp
+BBOO0222I WSVR0024I: Server SERVANT PROCESS wd6nd6cd6sc59 stopped
IWM034I PROCEDURE WAS5DS6 STARTED FOR SUBSYSTEM WASD6 628
APPLICATION ENVIRONMENT CLUD6
PARAMETERS JOBNAME=WASD6S,ENV=CELD6.NODD6.WASD6
IWM032I VARY REFRESH FOR CLUD6 COMPLETED
```

Note: Additionally, you will observe many SYSLOG messages from the resulting address space terminations and restarts which were not included here to preserve clarity.

An example of the **VARY** command to quiesce a dynamic application named CLUD6 is as follows:

```
V WLM,DYNAPPL=CLUD6,QUIESCE
```

An example of the resulting output follows.

```
V WLM,DYNAPPL=CLUD6,QUIESCE
IWM031I VARY QUIESCE FOR CLUD6 IN PROGRESS
+BBOO0222I WSVR0217I: Stopping application: adminconsole
+BBOO0222I SRVE0170I: Stopping Web Module: adminconsole.
+BBOO0222I WSVR0220I: Application stopped: adminconsole
+BBOO0222I WSVR0217I: Stopping application: JaasTest
+BBOO0222I SRVE0170I: Stopping Web Module: JaasTestWeb.
+BBOO0222I WSVR0220I: Application stopped: JaasTest
+BBOO0222I WSVR0217I: Stopping application: SWIPE
+BBOO0222I SRVE0170I: Stopping Web Module: IBMAuthSSLClient.
+BBOO0222I SRVE0170I: Stopping Web Module: IBMnoWebApp.
+BBOO0222I SRVE0170I: Stopping Web Module: IBMForms.
+BBOO0222I SRVE0170I: Stopping Web Module: IBMEBizWeb.
+BBOO0222I WSVR0220I: Application stopped: SWIPE
+BBOO0222I WSVR0217I: Stopping application: ivtApp
+BBOO0222I SRVE0170I: Stopping Web Module: IVT Application.
+BBOO0222I WSVR0220I: Application stopped: ivtApp
+BBOO0222I WSVR0024I: Server SERVANT PROCESS wd6nd6cd6sc59 stopped
IWM032I VARY QUIESCE FOR CLUD6 COMPLETED
```

Note: Additionally, you will see many SYSLOG messages from the resulting address space terminations and restarts which were not included here to preserve clarity.

An example of the **VARY** command to resume a dynamic application named CLUD6 after it has been quiesced is as follows:

```
V WLM,DYNAPPL=CLUD6,RESUME
```

An example of the resulting output follows.

```
V WLM,DYNAPPL=CLUD6,RESUME
IWM032I VARY RESUME FOR CLUD6 COMPLETED
```

Note: Additionally, you will observe many SYSLOG messages from the resulting address space terminations and restarts which were not included here to preserve clarity.

9.3 High virtual shared area support

z/OS V1R5 provides support for shared memory usage above the 2 Gigabyte(GB) bar. This support allows multiple address spaces to share storage areas in the high virtual shared area.

The size of the high virtual shared area is defined by the system programmer through specification of the new HVSHARE parameter of IEASYSxx. The high virtual shared area can range in size from 0GB to 1Exabyte(EB) and will be rounded up to a 2GB boundary. If the maximum size is specified, a 2GB high virtual private area is allocated above and below the high virtual shared area. The size allocation also determines where the shared area will be allocated. If the allocation specifies a value less than 2 Terabytes (TB), the shared virtual area will center on the 4TB boundary with half of the shared area allocated below the boundary and half above it. If the allocation is greater than 2TB the shared area will be allocated beginning at the 2TB boundary. Refer to *z/OS MVS Initialization and Tuning Reference*, SA22-7592 for more information on the specification of the HVSHARE IEASYSxx parameter.

9.3.1 WLM monitoring of shared area

WLM is enhanced to provide monitoring of the high virtual shared area and support for high shared area paging.

The monitoring function reviews usage thresholds and issues operator messages in the event of high shared area utilization.

When utilization of the high shared area reaches 80%, the following message is issued:

```
IRA110E High Shared Virtual Area Shortage
```

If the monitored usage reaches 95% utilization the following message is issued:

```
IRA110E Critical High Shared Virtual Area Shortage
```

When the measured utilization falls below 80% the following message is issued:

```
IRA112I High Shared Virtual Area Shortage Relieved
```

High virtual shared area paging support is added to WLM to allow it to effectively manage the paging environment and provide for reporting of statistical paging detail.

9.3.2 WLM support for C and C++

WLM provides C and C++ based callable interfaces that can be invoked by servers to exploit WLM services. These interfaces are being provided in both 31-bit and 64-bit AMODE callers. This will provide the capability for these and future zSeries server applications to run in 64-bit virtual mode. Servers making use of these interfaces include:

- Web server
- Domain name services
- Intelligent Data Miner (MPI/POE)
- CICS Open Transaction Environment (OTE)
- MQ Series Workflow/390

The interfaces are documented in Part 2. Reference: Workload Management Services of *z/OS MVS Programming: Workload Management Services*, SA22-7619.

9.4 Importance-based initiator dispatch priority control

OS/390 V2R10 introduced CPU protection to ensure that lower priority workloads would generally have a lower dispatch priority than workloads defined as CPU-critical. It is still possible, although unlikely, that a lower priority workload's dispatch priority could be temporarily elevated over CPU-critical workloads in an attempt to relieve enqueue contention that is delaying higher priority work.

9.4.1 WLM CPU protection

Workloads are eligible for CPU protection through assignment to a service class that has been defined with the CPU-critical attribute.

CPU protection is intended primarily for high priority CPU-sensitive work such as DB2 and certain IMS and CICS transactions, whose processing could be significantly delayed by lower priority workload's excessive CPU consumption.

One very common instance where lower priority tasks execute above CPU-critical workloads remains. JES, APPC, and OMVS initiators may run pre-execution installation dependent tasks that must be completed in order to ensure successful job completion. While initiators are performing this setup work, they are executing at a system dispatch priority of 254 in service class SYSSTC, which places them above WLM-defined CPU-critical workloads. They will be assigned to the service class that has been assigned for the batch job when they complete setup processing and are ready to begin job execution. This applies to all JES, APPC, and OMVS initiators.

Pre-execution processing rarely casts more than minimal impact on overall utilization of CPU resources. However, under extraordinary circumstances, this processing could be very CPU-intensive and could negatively impact workloads running in defined service classes.

z/OS V1R5 introduces importance-based initiator dispatch priority control. This enhancement provides the installation with the ability to set dispatch priority for initiators that have not been assigned to a service class, below the dispatch priority for CPU-critical workloads.

INITIMP parmlib member

Importance-based initiator dispatching control is implemented through specification of a new parameter, INITIMP in SYS1.PARMLIB member IEAOPTxx.

The valid parameter settings for INITIMP are:

INITIMP=0 The dispatch priority is set to the dispatch priority of service class SYSSTC. This is the default, and the functional equivalent of what took place prior to z/OS V1R5. The initiator dispatching priority is set to 254.

INITIMP=1 The dispatch priority is set lower than the lowest dispatch priority for any CPU-critical service class with an importance of one.

INITIMP=2 The dispatch priority is set lower than the lowest dispatch priority for any CPU-critical service class with an importance of two.

INITIMP=3 The dispatch priority is set lower than the lowest dispatch priority for any CPU-critical service class with an importance of three.

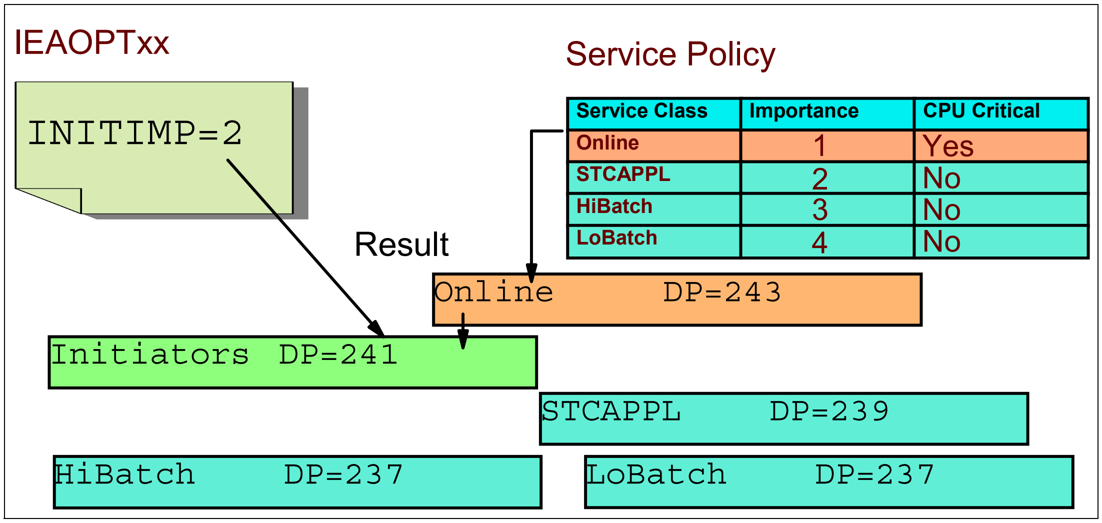
INITIMP=E The dispatch priority is set to the enqueue promotion dispatch priority, which is calculated dynamically and ensures access to the processor. This does not guarantee that CPU-critical work will always have a higher dispatch priority, but it should not be negatively impacted.

Note: In the event that INITIMG is specified as 1, 2, or 3 and there is no service class in the active WLM policy that is defined as CPU-critical with an importance of 1, 2, or 3, dispatch priority will be calculated as if INITIMP=E was specified.

Dispatching priority example 1

The example shown in Figure 9-1 has the following definitions in SYS1.PARMLIB and the WLM service policy:

- The ONLNE service class in the service policy is defined with CPU-critical, importance 1.
- The INITIMP parameter (INITIMP=2) in the IEAOPTxx parmlib member specifies that the initiator should have a dispatching priority (DP) that always has to be lower than CPU-critical and importance 2 work.
- As a result, all work for Online always has a higher DP than initiators before job execution.
- All other work also will have a lower DP than the Online service class work.



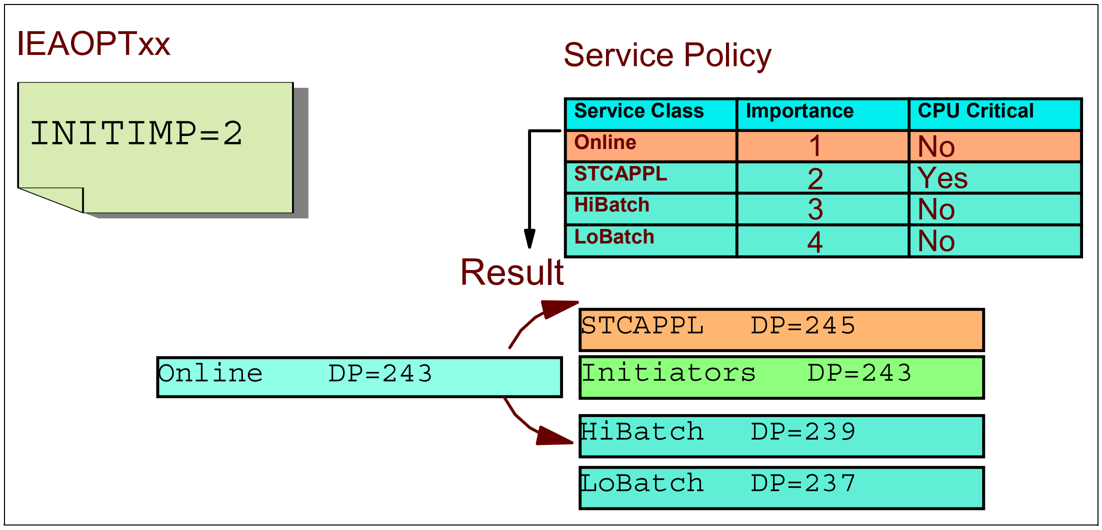
Service Class	Importance	CPU Critical
Online	1	Yes
STCAPPL	2	No
HiBatch	3	No
LoBatch	4	No

Figure 9-1 Dispatching priority example 1

Dispatching priority example 2

The example shown in Figure 9-2 on page 237 has the following definitions in SYS1.PARMLIB and the WLM service policy:

- The STCAPPL service class in the service policy is defined with CPU-critical and importance 2.
- The INITIMP specification, (INITIMP=2), in the IEAOPTxx parmlib member specifies that the initiator should have a dispatching priority that always has to be lower than CPU-critical and importance 2 work.
- As a result, the initiators will always be below STCAPPL.
- Online work may have any dispatching priority, below and above STCAPPL and the other initiators.



Service Class	Importance	CPU Critical
Online	1	No
STCAPPL	2	Yes
HiBatch	3	No
LoBatch	4	No

Figure 9-2 Dispatching priority example 2

Refer to *z/OS MVS Initialization and Tuning Reference*, SA22-7592 for more information on the specification of the INITIMP IEAOPTxx parameter.

9.4.2 Installation considerations

Implementation of this enhancement only affects the dispatching priority. Initial assignment of JES, APPC, and OMVS initiators to service class SYSSTC remains unchanged.

It is recommended that this enhancement not be implemented unless there is already some experience with the problems described and there has been significant negative impact suffered to the environment as a result. It is important to note that the pre-execution processing that occurs prior to job execution is short in duration and allowing it to process at a high dispatching priority ensures that the job completes as quickly as possible.

9.5 WLM Sub-capacity Reporting Tool (SCRT) support

The Sub-Capacity Reporting Tool (SCRT) is used by customers to analyze one month of SMF data from z/800, z/900 and z/990 machines running z/OS in 64-bit mode. The tool will generate a sub-capacity report for each processor analyzed. This report documents the required licensed capacity for each sub-capacity eligible software product. The required license capacity is based on the highest four hour rolling average of processor usage observed during the month. The sub-capacity reports are e-mailed to IBM monthly in order to satisfy the qualification requirements for sub-capacity Workload License Charges (WLC), sub-capacity Entry Workload License Charges (eWLC) and sub-capacity charging for several WebSphere products on z/OS. These pricing structures enable the customer to pay for eligible IBM software based on measured usage, which is usually less than the full rated capacity of the machine, resulting in reduced total cost of ownership. For the latest information on SCRT, sub-capacity eligible products, and the associated pricing strategies refer to:

http://www-1.ibm.com/servers/eserver/zseries/swprice/

9.5.1 z/OS V1R5 and WLC

Prior to z/OS V1R5, sub-capacity pricing was available only to customers with a defined capacity limit set for their LPARs. The enhanced WLM support provided in z/OS V1R5 removes this restriction.The rolling four hour MSU average is calculated for LPARs on all z/800, z/900, z/990 and later processors. If the sole reason for implementing defined capacity limits was to comply with sub-capacity requirements, we recommend it be deactivated by using the LPAR Change Logical Partition Control Panel to dynamically update the running system and specify a defined capacity of zero (0).

Soft-capping is the process used by WLM and PR/SM™ to prevent an LPAR from exceeding its defined capacity. It is implemented dynamically when an LPAR's rolling four hour average exceeds the defined capacity limit. Soft-capping is removed from an LPAR when its rolling four hour average falls below the defined capacity limit. An enhancement has been made to the calculation of the four hour rolling average, which was calculated over too short a time frame during the IPL phase. For customers operating with defined capacity this could result in soft-capping the LPAR during IPL. To prevent this situation from occurring, the solution provided bases the calculation of the rolling four hour average after completion of the IPL, over a period of 4 hours.

10

Console restructure enhancements

This chapter describes the console availability enhancements in z/OS V1R5. The following topics are discussed:

- Console restructure
- System console availability
- One-byte console ID tracker
- Migration/Coexistence considerations

10.1 Console restructure

The original console component exploitation of sysplex works best in a small sysplex filled with homogenous systems. Introducing systems with different capabilities, or greatly increasing the size of the sysplex caused problems.

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 and reduce system outages caused by message floods.

10.1.1 Problems previously encountered

Figure 10-1 on page 241 shows the problem experienced with the message delivery system prior to this restructure. Each system in the sysplex has the same message processing. Messages are sent from one system to another, via XCF, without regard to whether the receiving system is able to consume the messages. The receiving system has no choice except to attempt to consume the messages that it was forced to receive.

Message processing

Message production and the processing of the messages can cause problems. Messages are delivered only once to a system that has multiple interested consoles, but messages must still be individually delivered to each system that has consoles that receive the messages. Consequently, the overhead of delivering messages to other systems, via XCF writes, rises as the number of systems in the sysplex increases. Therefore:

- A runaway application creating messages can cause a system to suffer buffer shortages and out-of-storage conditions that can bring down either a receiving system or the system generating the messages.
- All queuing decisions are made from a single task, represented by the egg-shaded area in Figure 10-1. All messages arrive on a single queue and are queued by a single task to any local consoles, to the SYSLOG, to the OPERLOG, and to the three EMCS console queuers. Any disruption to the initial queuing task affects the queuing to all consoles and to the logs. Once a message has been queued to one of the EMCS queuers, queuing to its consoles proceeds independently of the queuing to the consoles owned by the other two EMCS queuers.

  Because a message can be queued to multiple consoles, the storage that the message occupies cannot be re-used until the last console has displayed the message and the message is de-queued.
- Large systems creating enormous numbers of messages can send so many messages that they might overwhelm smaller systems.
- The only processing done in the user's address space is the MPF exit and the SSI processing, as shown in Figure 10-1.

Message flow

Messages are delivered only once to a system that may have multiple consoles that receive the messages. The more systems that are in a sysplex, the more overhead of delivering messages (the number of XCF writes) increases. The following problems have been addressed by this console restructure:

- Traffic to a particular console

  A problem on a single local console can cause message queue storage shortage problems which affect not only the other local consoles but the log and EMCS consoles as well. Once a message is queued to an EMCS console queuer, the message is copied one or more times into EMCS message data spaces. Queuing or storage problems in one data space do not affect queuing or storage in other EMCS message data spaces.

- Un-ended multiline WTOs

  The way multiple line messages are constructed can cause further message delivery problems since a multiple line message can be queued, and sent to other systems in the sysplex before the entire multiple line message is complete. Queuing the incomplete message does not immediately cause problems, but if the message is still incomplete when a console attempts to display the message, it causes the console to stop displaying other messages until the message it has begun to display is complete.

- All message traffic is paced by the slowest operator console.
- The message is then handed over to the console address space where it must reside in a 24-bit storage buffer (limited by the MLIM parameter) to be processed, as shown in Figure 10-1.
- Any disruption to the initial queueing task affects the queueing to all consoles and the logs.

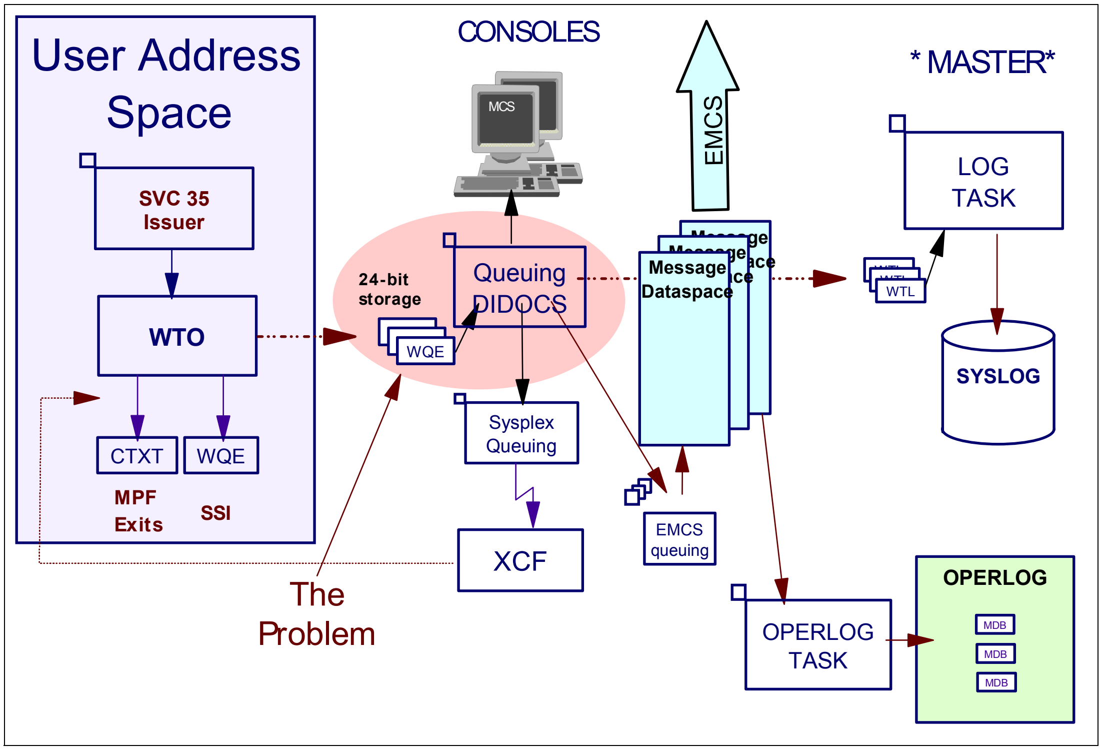


Figure 10-1 Message delivery before the console restructure

10.2 Console restructure enhancements

Design changes made to message processing during the console restructure result in the following enhancements:

- Help to eliminate outages due to a flood of WTOs (write to operator) and DOMs (delete operator messages)

  Introduction of a new message cache data space

- Reduce dependence on a single task

- Establish message queuing independence
  - Deliver messages to the SYSLOG and OPERLOG from the caller's unit of work.
  - Queuing to the SYSLOG and OPERLOG is separated from queuing to the consoles. Queuing to the EMCS consoles is performed (in parallel) prior to queuing to the local consoles.
- Do not queue MLWTOs (multi-line WTOs) for delivery until all messages have been received.

10.2.1 Eliminating message processing outages

All message producers now place their messages and message deletion requests into a single, common pool from which the message consumers can extract their messages and message deletion requests. The pool decouples at the rate at which messages (and message deletion requests) are produced from the rate at which individual message consumers consume them.

At the sysplex level, messages and message deletion requests are written once to a logstream that is common to the entire sysplex. At the system level, messages and message deletion requests are read from the common logstream and written to a data space (the console message cache shown in Figure 10-2 on page 244) common to all of the consoles on the system.

Console message cache data space

The console message cache data space is introduced to decouple message reception from other systems from the queuing of the messages to consoles. The console message cache data space decouples at the rate at which messages (and message deletion requests) are read from the common logstream from the rate at which those messages can be queued (and consumed) by the console class queuers.

Note: Console services now makes use of enhancements to the XCF protocols to request ordered delivery of messages and message deletion requests. This change should ensure that message deletion requests do not arrive before the message that they are intended to delete, and it should ensure that the pieces of a multiple line message arrive in the order in which they were created.

10.2.2 Eliminate single task processing

By moving DIDOCS queuing to the end of the processing rather than the beginning, the dependence on the one main queuing task is eliminated.

Move processing to the user’s address space

Differences to reduce processing by a single task involved the moving of processing into the user address space (illustrated in Figure 10-2 on page 244) as follows:

- Broadcast the message via ordered delivery to all systems (including the one issuing the message).
- Multiple line messages that are constructed incrementally using connect processing are queued in a dataspace and held there until complete.
- Upon receipt of the message from XCF, MPF and SSI is performed for foreign messages and the message is placed in a new console message cache structure.

- EMCS consoles and hardcopy processing are no longer dependent on the ability to queue to MCS consoles.
- Hardcopy has been moved to the front of command processing, and now occurs in the WTO issuer's task. Delivery is no longer dependent on the health or speed of the DIDOCS queuer. Hardcopy queuing sits on the EMCS interface, but receives messages sent directly from the SVC 35 issuance. It doesn't rely on the message cache mechanism for delivery.
- The EMCS queuers request new messages and deliver them to any and all consoles (including the DIDOCS consoles) that are receivers.
- The DIDOCS queuer will now only process messages that are intended for display on an MCS console.

10.2.3 Queuing to SYSLOG and OPERLOG

Queuing to the SYSLOG and OPERLOG is now performed under the message issuer's thread after the message has passed through the message processing facility (MPF) and subsystem interface (SSI) processing. Logging performance is no longer dependent on Communications Task performance and writing to the log will no longer cause console message buffer shortages. (Logging problems now affect individual message producers but not the console function.)

EMCS queuers

The three EMCS message queuers now queue messages from the console message cache data space to the EMCS consoles independently. All of the local consoles are represented as a single EMCS console; messages are then queued from the EMCS message data space to the individual MCS and SMCS consoles.

10.2.4 Multiline messages

Incomplete multiple line messages are gathered after MPF and SSI processing into a data space until they are complete. Once complete, they will be written from the data space to the common logstream. The common logstream and everything else on the message path now only see complete multiple line messages. Multiple line messages being constructed incrementally are passed to the MPF exit and the SSI as they are constructed to preserve compatibility with current behavior. Time-out processing is used to force the completion of multiple line messages which have remained incomplete for long periods of time.

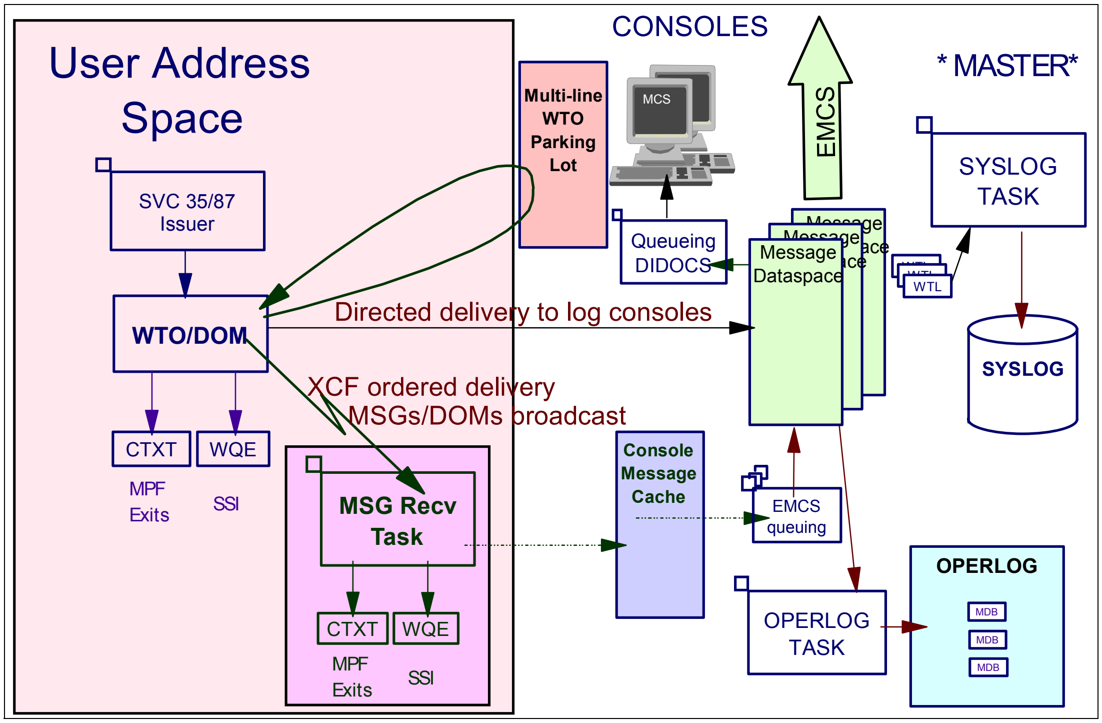


Figure 10-2 Enhanced message delivery

10.3 System console availability

The system console, or Hardware Management Console (HMC), under MVS, was designed for use as an initialization and emergency console. It was intended for those times when MCS consoles could not be active, or were all inactive due to error.

When defining a system console in the current version of z/OS, there are restrictions on system console usage. For example, the operator has to:

- Use the system console as a NIP console.
- Issue the **VARY CN(*),ACTIVATE** command when NIP processing is over, possibly losing a few messages.
- Issue the **VARY CN(*), DEACTIVATE** against the system console.

This procedure is user-unfriendly and error-prone. What is needed is some automatic means of activating the system console when it is needed, and deactivating it when no longer needed. However, it is necessary to know which console(s) the installation considers appropriate to replace the system console.

10.3.1 System console enhancements

A better mode of operation for the operator to be able to use the system console can be achieved through the definition of an AUTOACT console group in SYS1.PARMLIB.

Using AUTOACT with the system console

On the CONSOLE statement in SYS1.PARMLIB, define the system console in the CONSOLxx member as follows:

```
CONSOLE   DEVNUM(SYSCONS) AUTOACT(groupname)
```

If AUTOACT (groupname) is specified in CONSOLxx on the CONSOLE statement, groupname is the name of a console group, as defined in CNGRPxx.

Automatic activate group

AUTOACT specifies the "automatic activate group" for the system console. If an automatic activate group (AUTOACT) is active for the system console, the system automatically issues the **VARY CN(syscons),ACTIVATE** and **VARY CN(syscons),DEACTIVATE** commands.

While the AUTOACT group is defined and not suspended:

- The system console is automatically placed into problem determination (PD) mode when all of the consoles in AUTOACT are inactive.
- The system console is automatically removed from PD mode when any console in the AUTOACT group becomes active.

To suspend AUTOACT processing, issue a **VARY CN(*), DEACTIVATE** command from the system console.

- This manual intervention overrides automatic processing until the opposite command is issued.

Note: AUTOACT is only valid when DEVNUM(SYSCONS) is specified.

AUTOACT console group commands

Use the following commands in support of the system console availability enhancements:

- To find out if an AUTOACT group is in place, issue:

  ```
  D EMCS,I,CN=syscons name
  ```

  or

  ```
  D C,CN=syscons name
  ```

- To display the consoles in the AUTOACT group, issue:

  ```
  D CNGRP
  ```

- To add or change the AUTOACT value for the system console, issue:

  ```
  VARY CN(syscons name),AUTOACT=groupname
  ```

- To add or change a console group, change the parmlib member CNGRPxx, then issue:

  ```
  SET CNGRP=xx
  ```

- To delete the AUTOACT value for the system console, issue:

  ```
  VARY CN(syscons name),AUTOACT=*NONE*
  ```

System console considerations

If an operator is using the system console during NIP, he can continue using it without interruption or manual intervention. When a real console (a member of the AUTOACT group) comes up, the system console is deactivated automatically. The operator does not have to remember to turn it off. If all of the consoles in the group become inactive, the system console comes up automatically.

10.4 One-byte console ID tracker

The Console ID Tracking facility is designed to assist with the identification and removal of one-byte console IDs and one-byte migration IDs. In future releases, only four-byte IDs will be accepted. While four-byte IDs have generally replaced one-byte IDs, some services still accept one-byte IDs. The users of services that still accept one-byte console and migration IDs are now known as violators, and instances of one-byte ID usage are known as violations.

Because no interfaces are being changed, the Console ID Tracking facility does not present any compatibility issues. It tracks 1-byte users on the following macro services:

```
WTO, MPF, SSI, MGCR/MGCRE, CONVCON, and MCSOPER
```

To prepare for the removal of one-byte console IDs, the Console ID Tracking facility provides the following new functions:

- The **SETCON** operator command, which is used to activate and deactivate the Console ID Tracking facility
- The **DISPLAY OPDATA,TRACKING** operator command, which is used to display the current status of the Console ID Tracking facility, along with any recorded instances of violations
- The CNIDTRxx parmlib member, which is used to list violations that have already been identified in order to prevent them from being recorded again
- The CNZTRKR macro, which is used to invoke the Console ID Tracking facility

10.4.1 The SETCON operator command

The **SETCON** operator command allows the Console ID Tracking facility to be activated, activated with an ABEND option, or deactivated, as follows.

```
SETCON TRACKING=ON
```

- Use this command to activate the tracking facility.
- You can issue this command manually, or it can be automated by placing the command in a COMMNDxx member of parmlib.

  ```
  SETCON TRACKING=ONWITHABEND
  ```

- Use this command to activate the tracking facility and cause violators to be ABENDed with ABEND code 077, reason code 0034.
- This ABEND is designed to allow an installation to set a SLIP trap to obtain a dump, as follows:

  ```
  SLIP SET,ENABLE,ID=TRAK,COMP=077, REASON=34,ACTION=SVCD,END
  ```

- If no SLIP is set, an entry in Logrec will be made, but no dump will be taken. You can toggle between **SETCON TRACKING=ON** and **SETCON TRACKING=ONWITHABEND** without any loss of recorded instances.

> **Note:** If the track value is 0 or 128, no ABEND is issued even when you specify ONWITHABEND.

- Use this command to deactivate the tracking facility.

  ```
  SETCON TRACKING=OFF
  ```

- Before turning the facility off, the **SETCON TRACKING=OFF** command issues a **DISPLAY OPDATA,TRACKING** command to capture, in the hardcopy log, all recorded instances of violations.

- The tracking facility then deletes all recorded violations and does not record any more violations.
- After issuing a **SETCON TRACKING=OFF** command, wait for message IEE7121 SETCON PROCESSING COMPLETE to appear, to ensure that the facility has had time to fully deactivate before you attempt to reactivate it.
- If you do not wait, the activation command may complete before the deactivation command finishes, leaving the facility off when you expect it to be on.

Note: The maximum number of unique instances that the Console ID Tracking facility can record is 1,000.

10.4.2 Using the DISPLAY OPDATA,TRACKING command

The **DISPLAY OPDATA,TRACKING** command displays the current status of the Console ID Tracking facility, along with information about each recorded instance of one-byte console ID or migration ID usage.

The **DISPLAY OPDATA,TRACKING** command displays the following information about the Console ID Tracking facility:

- Whether the facility is ON, ONWITHABEND, or OFF.
- The number of unique violations that have been recorded.
- The maximum number of unique violations that is accepted.
- The suffix of the active CNIDTRxx parmlib member (if any).
- The number of violations that were not recorded because they were excluded.
- The number of violations that were not recorded because the facility was full, because of timing issues, or because serialization of the facility could not be obtained.
- An indication if the facility is full.

The **DISPLAY OPDATA,TRACKING** command displays information about a violation, as shown in Figure 10-3 on page 248. The information displayed in this example of the command response when tracking is active and violations have been recorded is as follows:

- The tracking information (up to 28 characters) that was provided when the tracking request was made.
- The four-byte track value (typically a console ID) that was provided when the tracking request was made.
- This value is displayed in hexadecimal, with leading zeros suppressed.
- If no value was provided, zero is displayed, since the zero could be valid (for instance, if the console ID is zero).
- The name of the job from which the tracking request was made.
- The name of the program from which the tracking request was made.
- The offset into the program where the violation occurred and ASID of the violator.

```
CNZ1001I 10.53.40 TRACKING DISPLAY
STATUS=ON,ABEND NUM=19   MAX=1000 MEM=n/a EXCL=0     REJECT=0
----TRACKING INFORMATION---- -VALUE-- JOBNAME  PROGNAME+OFF-- ASID NUM
Parmlib Reader: ADYSET00           00 *MASTER* ADYSETP   1BD8   01   1
Parmlib Reader: COFVLF04           00 VLF      COFMINIT  2EFE   18   1
Parmlib Reader: IEFSSN00           00 *MASTER* IEEMB860  9E2A   01   1
Parmlib Reader: SMFPRM00           00 SMF      IFASMF    ECBE   19   1
WTO: $HASP000 OK                   00 JES2     HASJES20 1B0AC   14   2
WTO: $HASP003         SPECIF       00 JES2     HASJES20 1B0AC   14   2
WTO: $HASP003 RC=(52),             00 JES2     HASJES20 1B0AC   14   2
WTO: $HASP003 RC=(52),S1-999       00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP003 RC=(52),T1-999       00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP604 ID 0007 T=***.       00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP604 ID 0008 T=***.       00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP604 ID 0010 T=***.       00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP604 ID 0011 T=***.       00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP646 0.5714 PERCENT       00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP650 Q,Q=W     INVA       00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP893                      00 JES2     HASJES20 1B0AC   14   1
WTO: $HASP893 VOLUME(SPOOL1)       00 JES2     HASJES20 1B0AC   14   2
WTO: IEC350I CATALOG ADDRESS       00 CATALOG  IGG0CLX0 1BBC2   1B   1
WTO: IEF677I WARNING MESSAGE       00 JES2     IEFNB903  BF12   14   1
```

Figure 10-3 Example of the display tracking command

10.4.3 Using the CNIDTRxx parmlib member

The CNIDTRxx parmlib member allows you to exclude specific violation instances of one-byte console IDs from being recorded by the Console ID Tracking facility. This exclusion allows the facility to ignore violations that have already been reported but not yet corrected.

Use the **`SET CNIDTR=xx`** command to activate the CNIDTRxx parmlib member.

If no CNIDTRxx parmlib member is active, then all instances of one-byte console ID violations will be recorded. Customers are expected to update the CNIDTRxx parmlib member with reported violations. Violations should be reported to IBM, and updates to CNIDTRxx must also be made available.

Data to add to CNIDTRxx member

CNIDTRxx is used to list the following information about each tracking instance for exclusion processing (the wild cards * and ? are supported):

- Tracking information (up to 28 characters)
- Job name (up to 8 characters)
- Program name (up to 8 characters)

Be aware that once a violation is recorded, changing the exclusion list in CNIDTRxx does not remove the instance from the list of recorded instances displayed by the DISPLAY OPDATA,TRACKING command.

- The Console ID Tracking facility must be restarted to exclude any new additions to the CNIDTRxx parmlib member.

> **Note:** Many of the services that track console IDs also invoke the CONVCON service. This means that one violation might cause two instances to be recorded in the Console ID Tracking facility: one for CONVCON and one for the service that called CONVCON.

Example output using CNIDTRxx to exclude violations

Figure 10-4 is an example of using the CNIDTRxx parmlib member.

```
*                                Jobname   Pgmname
* Tracking Information Mask      Mask      Mask      Comments (ignored)
*+-----------------------------+ +-------+ +-------+ +--------------------------+
 WTO: $HASP*                     JES*      HAS*      Ignore JES2 messages
```

Figure 10-4 CNIDTRxx parmlib member example

Using the **`d opdata,tracking`** command after the parmlib member is created is shown in Figure 10-5. See the contrast between this and Figure 10-3 on page 248.

```
CNZ1001I 10.53.40 TRACKING DISPLAY
STATUS=ON       NUM=6    MAX=1000 MEM=00  EXCL=13    REJECT=0
----TRACKING INFORMATION---- -VALUE-- JOBNAME  PROGNAME+OFF-- ASID NUM
Parmlib Reader: ADYSET00           00 *MASTER* ADYSETP   1BD8   01   1
Parmlib Reader: COFVLF04           00 VLF      COFMINIT  2EFE   18   1
Parmlib Reader: IEFSSN00           00 *MASTER* IEEMB860  9E2A   01   1
Parmlib Reader: SMFPRM00           00 SMF      IFASMF    ECBE   19   1
WTO: IEC350I CATALOG ADDRESS       00 CATALOG  IGG0CLX0 1BBC2   1B   1
WTO: IEF677I WARNING MESSAGE       00 JES2     IEFNB903  BF12   14   1
```

Figure 10-5 D OPDATA,TRACKING command after CNIDTRxx parmlib member created

> **Note:** After you issue the SET CNIDTR=xx command to activate the CNIDTRxx member, JES2 violations are no longer recorded or displayed.

10.4.4 Using the CNZTRKR macro

You can use the CNZTRKR macro to invoke the Console ID Tracking facility, which records violations of 1-byte console ID usage.

Before issuing the CNZTRKR macro, you must do the following:

- Include the CNZTRPL mapping macro in your program.
- Obtain storage for the CNZTRKR parameter list.
- TRPL_LEN in CNZTRPL contains the length of the parameter list. The parameter list can be in any type of storage.
- Clear the entire parameter list by setting it to binary zeros.
- Initialize the following fields in the parameter list mapped by macro CNZTRPL:

  TRPL_Acro The TRPL acronym.

  TRPL_Version The current version level of the parameter list. The CNZTRPL mapping macro contains the current version level in TRPL_K_Curr_Version.

TRPL_Track_Info Text that describes the occurrence of this instance. This text can be from 1 to 28 characters in length. Any EBCDIC value is allowed, although you should use displayable characters because undisplayable characters may be changed to blanks when displayed on an operator’s console or in the hardcopy log. The text cannot be all blank or all hexadecimal zeros.

TRPL_Track_Data Four bytes of data associated with this track instance. This data could be the one-byte console ID that was used by the violator. Zero is a valid value. The DISPLAY OPDATA operator command will display this value as a hexadecimal number.

TRPL_Violators_Addr While optional, this field should contain the address where the violation occurred (perhaps the address to which the service invoking CNZTRKR will return). If set to zero, the Console ID Tracking facility will attempt to determine the violation address but may not be able to determine the exact violation location. This address is assumed to be a 31-bit address. If a 24-bit address is provided, you must ensure that the high-order byte of the address is zero.

10.5 Configuration requirements

Utilizing the console restructure is integral to running the operating system. To utilize system console availability:

- Create a group in CNGRPxx containing the consoles which can *replace* the system console.
- Specify AUTOACT(group_name) on the CONSOLE statement in CONSOLxx for the system console.
- Alternatively, the group can be activated dynamically through the use of the **`SET CNGRP`** and **`V CN(syscons),AUTOACT`** command.

Utilizing the one-byte ID tracker:

- To track usages on the console component interfaces:
  - Operator command to control the activation of the service.
  - **`SETCON TRACKING=ON/ONWITHABEND/OFF`**.
  - Display “violators” via the **`DISPLAY OPDATA,TRACKING`** command.
  - Known instances can be ignored through the specification of CNIDTRxx member in SYS1.PARMLIB.

To track other instances of 1-byte console IDs:

- Initialize the CNZTRPL parameter list.
  - Includes data that describes what is being tracked
  - Includes the “bad” console ID being used
- Invoke the CNZTRKR service, passing the CNZTRPL parameter list.

10.5.1 Migration and coexistence considerations

The ALTGRP keyword has been available since MVS/ESA 4.2.0. It is now the required method to specify backup consoles. The ALTERNATE keyword on the CONSOLE statement in CONSOLxx is no longer accepted.

All messages will still appear in the log, but the code no longer takes extra effort determining if a message has been delivered to a console, or in redirecting the message to a console. Undeliverable messages (UD) are no longer detected.

- The UD keyword on the CONSOLE and HARDCOPY statements in the CONSOLxx parmlib member is no longer accepted.
- The UD keyword is no longer supported on the VARY CONSOLE and VARY HARDCPY commands.

HARDCOPY statement

Hardcopy must now be either SYSLOG or OPERLOG. The HCPYGRP keyword on the HARDCOPY statement in CONSOLxx is no longer accepted. Printer devices can still be used, but they cannot be specified as the hardcopy medium.

The DEVNUM keyword on the HARDCOPY statement in CONSOLxx parmlib member is no longer accepted.

CONTROL Q command

Messages queued to a console (but not yet displayed) can no longer be redirected to another console. The **`CONTROL Q`** command can still be used to indicate that they don't need to be seen. They will still appear in hardcopy.

The R= parameter on the **`CONTROL Q`** command is no longer supported. It was used to redirect backed up messages to another console.

CONSOLE statement

The NAME keyword on the CONSOLE statement in CONSOLxx parmlib member is now required.

- All consoles must be explicitly named (except for the system console, where the code will continue to create a name if none is supplied).
- The console definition will be rejected if no name is specified.

MSCOPE keyword

The MSCOPE keyword on the CONSOLE statement now defaults to * instead of *ALL.

Compatibility requirements

All systems in a sysplex must be at either:

- z/OS V1R5
- z/OS V1R4 with SDSF APAR PQ73805 installed
- z/OS V1R4 with APAR OW56244 and SDSF APAR PQ73805 installed
- Any level between OS/390 V2R10 and z/OS V1R3 with APAR OW56244 installed

11

z/OS V1R5 RMF

Resource Measure Facility (RMF) is an optional feature of z/OS that provides monitoring, measurement, and reporting of system performance data. This chapter describes enhancements for z/OS V1R5 RMF, including the following:

- PCIX cryptographic support
- z/990 exploitation
- RMF and multilevel security
- RMF storage report enhancements
- RMF and msys for setup
- RMF Spreadsheet Reporter Java Technology Edition
- RMF Performance Monitoring - Java Webstart enabling
- Sysplex data services for 64-bit environments
- Monitoring support for WLM enqueue management and performance block (PB) state reporting

11.1 PCIX cryptographic support

Cryptography can be described as an algorithmic enciphering or scrambling of data based on a unique encryption key. A complementary description key and associated algorithm may be used to decipher or unscramble the data. Cryptographic services provide the basis for overall security on the Internet, and provide the means necessary to ensure a secure e-business environment.

Cryptographic services can be exploited through either software- or hardware-based facilities. Application software-based solutions for public key cryptography tend to be compute-intensive processes that compete with business applications for cpu resources. Hardware-based encryption provided by cryptographic processors removes this competition.

Updated cryptographic support for RMF is provided as an SPE with APAR OW56656 and with z/OS V1R5.

11.1.1 Cryptographic hardware support

Cryptographic support is provided through Peripheral Component Interconnect (PCI) cards. PCI cards work in parallel with and asynchronously to zSeries processors to provide a secure environment for e-business.

G5/G6, z/800 and z/900 cryptographic support

Two types of cryptographic cards are supported:

- PCICC is a PCI Cryptographic Coprocessor data encrypting card designed to provide a high security environment; it includes hardware to support an array of cryptographic operations such as RSA public key encryption, Data Encryption Standard (DES) for secret key encryption, and Secure Hash Algorithm (SHA-1) for message encryption.
- PCICA is a PCI Cryptographic Accelerator that provides high performance cryptographic services for RSA public key algorithms which are required for Web applications, but does not include the high security options of the PCICC card.

Cryptographic Coprocessor Facility (CCF) is also available as an integrated feature of these processors and provides basic cryptographic services including RSA public key generation.

z/990 cryptographic support

Two types of cryptographic cards are supported:

- PCIxCC is a PCI extended Cryptographic Coprocessor, a replacement for both the PCICC card and the Cryptographic Coprocessor Facility. It is designed to work in conjunction with the Integrated Cryptographic Support Facility (ICSF) and IBM's Resource Access Control Facility (RACF) to provide RSA, DES, SHA-1, and Triple Data Encryption Standard (TDES) cryptographic services.
- PCICA support is maintained.
- The PCICC card is no longer supported.
- CCF is no longer supported.

CP Assist for Cryptograph Function (CPACF) is new with the z990 and provides hardware acceleration for RSA, DES, SHA-1, and TDES cryptographic services.

11.1.2 Cryptographic workload reporting support

Prior to z/OS V1R5, RMF reports were available to assist capacity planning efforts by presenting utilization data for the supported active cryptographic features of the installed system.

Specifically, RMF reports cryptographic activity for the PCICC Cryptographic Coprocessor, the PCICA Cryptographic Accelerator and ICSF activity from the Cryptographic Coprocessor Facility.

The new PCIX support provided through SPE APAR OW56656 and with z/OS V1R5 adds reporting capability for activity on the new PCIXCC card and for ICSF services that are executed on the PCIXCC card.

11.1.3 RMF monitor Crypto Hardware Activity report

Cryptographic activity statistics can be produced via the RMF post processor RMF. The REPORTS parameter CRYPTO can be used to request the Crypto Hardware Activity report.

Figure 11-1 is an example of the JCL and control statements necessary to produce the RMF Crypto Hardware Activity report.

```
//RMFJCL JOB (0),'RMF CRYPTO',CLASS=A,MSGCLASS=X
//RMF      EXEC PGM=ERBRMFPP
//MFPINPUT DD DSN=SAMPLE.Z990.RMF,DISP=SHR
//MFPMSGDS DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
  REPORTS(CRYPTO)
  SYSOUT(X)
//*
```

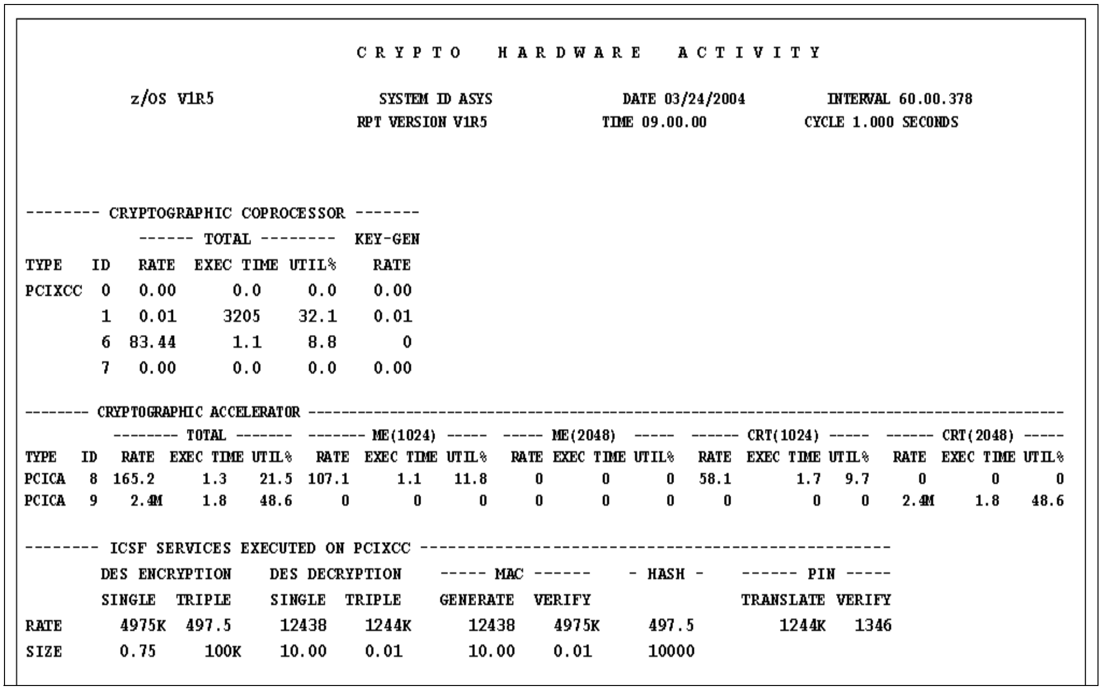
Figure 11-1 RMF JCL for the Crypto report

RMF report

Figure 11-2 on page 256 presents an example of the generated RMF Crypto Hardware Activity report. The updated sections of the report include the column type of PCIXCC to support the new PCI extended Cryptographic Coprocessor card, and the modified report header that is displayed for ICSF Services if a PCIXCC card is installed.

The section of the report dedicated to cryptographic coprocessors contains both the total rate of all operations and the rate of key generation operations. The rates displayed are rates per second. This data allows the user to calculate the processor capacity impact of key generation operations, which tend to be large consumers of coprocessor resources. This will enable more accurate predictions of coprocessor impact for projected business growth.

The section of the report for cryptographic accelerators also presents the total rate for all operations as well as rates for each of the available algorithms. This data will enable the user to calculate the accelerator utilization impact for each algorithm.



```
                                    C R Y P T O    H A R D W A R E    A C T I V I T Y

            z/OS V1R5                     SYSTEM ID ASYS             DATE 03/24/2004          INTERVAL 60.00.378
                                          RPT VERSION V1R5           TIME 09.00.00            CYCLE 1.000 SECONDS



-------- CRYPTOGRAPHIC COPROCESSOR -------
             ------ TOTAL --------  KEY-GEN
TYPE    ID    RATE  EXEC TIME UTIL%    RATE
PCIXCC   0    0.00       0.0    0.0    0.00
         1    0.01      3205   32.1    0.01
         6   83.44       1.1    8.8       0
         7    0.00       0.0    0.0    0.00

------- CRYPTOGRAPHIC ACCELERATOR ------------------------------------------------------------------------------
           -------- TOTAL -------  ------- ME(1024) -----  ----- ME(2048)  -----  ------ CRT(1024) -----  ------ CRT(2048) -----
TYPE    ID   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%
PCICA    8  165.2        1.3  21.5  107.1        1.1  11.8      0         0     0   58.1        1.7   9.7      0         0     0
PCICA    9   2.4M        1.8  48.6      0          0     0      0         0     0      0          0     0   2.4M       1.8  48.6

-------- ICSF SERVICES EXECUTED ON PCIXCC --------------------------------------------------
        DES ENCRYPTION    DES DECRYPTION    ----- MAC ------    - HASH -    ------ PIN -----
        SINGLE  TRIPLE    SINGLE  TRIPLE    GENERATE  VERIFY                TRANSLATE VERIFY
RATE      4975K  497.5     12438   1244K       12438   4975K      497.5         1244K   1346
SIZE      0.75    100K     10.00   0.01        10.00   0.01       10000
```

Figure 11-2   Crypto Hardware Activity report

Note: The rates displayed for both coprocessors and accelerators include all activity in the CPC. The ICSF Services section reported rates only contain the individual LPAR rate.

If an installation does not have one of the support cryptographic features, that feature is excluded from the RMF Crypto Hardware Activity report.

Refer to *z/OS Resource Measurement Facility (RMF) Report Analysis*, SC33-7991 for more information on interpreting the data presented in the RMF Crypto Hardware Activity report.

11.2 z/990 exploitation

The z/990 is IBM's next generation enterprise server that is based on 64-bit z/architecture. It is available in four models that support up to 30 LPARS, 32 physical processors, 256GB of processor memory, 512 channels, and relieves the 64K subchannel/device constraint. The z/990 provided enhancements affect the following specific areas of RMF through internal control block changes. The visible effects are described in more detail in the following sections of this chapter.

- Device Activity reports
- I/O Queuing reports
- Channel Activity reports

Updated RMF z/990 support is provided as an SPE with APAR OW56656 and with z/OS V1R5.

11.2.1 Device Activity reports

The primary impact on RMF Device Activity report processing occurs in device data gathering due to the creation of a new control block to support 64-bit z/architecture. This has resulted in several field level changes in the Device Activity reports for Monitor I, II, and III. These changes are discussed in the following sections.

Monitor I changes

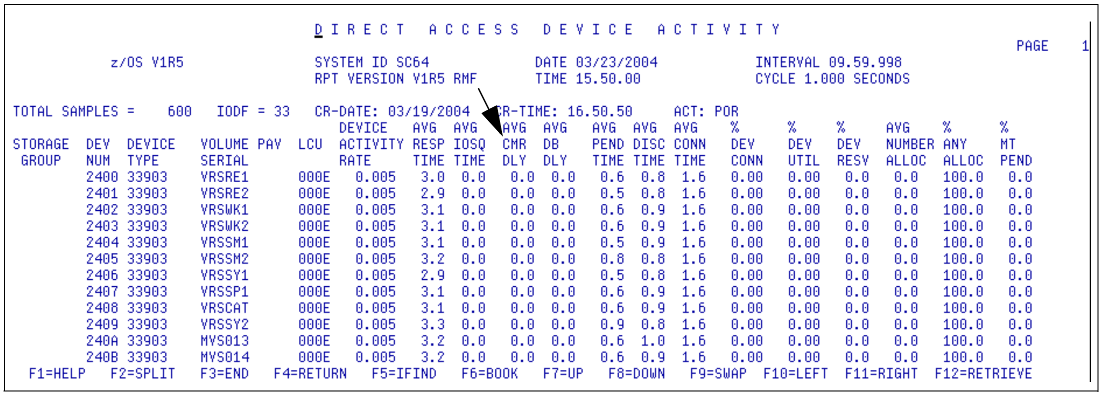
The Columns AVG CUB DLY and AVG DPB DLY have been removed from the Monitor I Postprocessor Device Activity report. The following new field shown in Figure 11-3 has been added:

AVG CMR DLY This field has been added to the report to indicate the average command response time, in milliseconds, that a successfully initiated start or resume command needs until it is accepted by a device.

Command response time delay (CMR)

CMR is the command response time delay, which is the percentage of time during the report interval when the first command of an I/O instruction of the channel program is sent to the device, until the device indicates it has accepted the command.

```
                  Accumulated CMR Delay Time
     DLY CMR% = -------------------------- * 10
                         Range Time
```



```
                                  D I R E C T   A C C E S S   D E V I C E   A C T I V I T Y
                                                                                                      PAGE    1
             z/OS V1R5                SYSTEM ID SC64           DATE 03/23/2004          INTERVAL 09.59.998
                                      RPT VERSION V1R5 RMF     TIME 15.50.00            CYCLE 1.000 SECONDS

 TOTAL SAMPLES =    600   IODF = 33   CR-DATE: 03/19/2004   CR-TIME: 16.50.50     ACT: POR
                                         DEVICE   AVG  AVG   AVG  AVG   AVG  AVG  AVG    %      %     %    AVG    %      %
 STORAGE  DEV  DEVICE    VOLUME PAV  LCU  ACTIVITY RESP IOSQ  CMR  DB    PEND DISC CONN   DEV    DEV   DEV  NUMBER ANY    MT
  GROUP   NUM  TYPE      SERIAL           RATE     TIME TIME  DLY  DLY   TIME TIME TIME   CONN   UTIL  RESV ALLOC  ALLOC  PEND
          2400 33903     VRSRE1      000E   0.005   3.0  0.0   0.0  0.0   0.6  0.8  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2401 33903     VRSRE2      000E   0.005   2.9  0.0   0.0  0.0   0.5  0.8  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2402 33903     VRSWK1      000E   0.005   3.1  0.0   0.0  0.0   0.6  0.9  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2403 33903     VRSWK2      000E   0.005   3.1  0.0   0.0  0.0   0.6  0.9  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2404 33903     VRSSM1      000E   0.005   3.1  0.0   0.0  0.0   0.5  0.9  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2405 33903     VRSSM2      000E   0.005   3.2  0.0   0.0  0.0   0.8  0.8  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2406 33903     VRSSY1      000E   0.005   2.9  0.0   0.0  0.0   0.5  0.8  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2407 33903     VRSSP1      000E   0.005   3.1  0.0   0.0  0.0   0.6  0.9  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2408 33903     VRSCAT      000E   0.005   3.1  0.0   0.0  0.0   0.6  0.9  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          2409 33903     VRSSY2      000E   0.005   3.3  0.0   0.0  0.0   0.9  0.8  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          240A 33903     MVS013      000E   0.005   3.2  0.0   0.0  0.0   0.6  1.0  1.6   0.00   0.00   0.0   0.0  100.0   0.0
          240B 33903     MVS014      000E   0.005   3.2  0.0   0.0  0.0   0.6  0.9  1.6   0.00   0.00   0.0   0.0  100.0   0.0
   F1=HELP    F2=SPLIT    F3=END    F4=RETURN    F5=IFIND    F6=BOOK    F7=UP    F8=DOWN    F9=SWAP  F10=LEFT  F11=RIGHT  F12=RETRIEVE
```

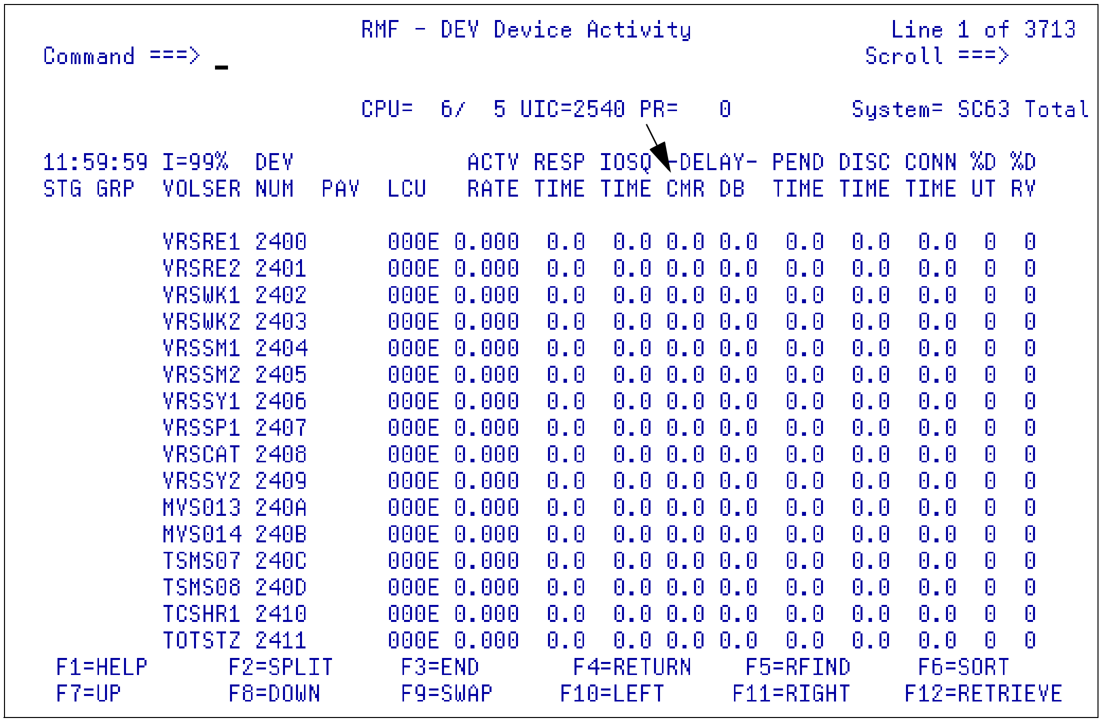
Figure 11-3 Monitor I report showing AVG CMR DLY

> **Note:** The same changes have also been made to the Monitor I Postprocessor Shared Device Activity report.

Monitor II changes

The Columns CUB Delay and DPB Delay have been removed from the Monitor II Device Activity Report. The following new field has been added as shown in Figure 11-4 on page 258:

CMR Delay This field has been added to the report to indicate the average command response time, in milliseconds, that a successfully initiated start or resume command needs until it is accepted by a device.



```
                        RMF - DEV Device Activity          Line 1 of 3713
Command ===> _                                             Scroll ===>

                        CPU=  6/  5 UIC=2540 PR=   0        System= SC63 Total

11:59:59 I=99%  DEV              ACTV RESP IOSQ -DELAY- PEND DISC CONN %D %D
STG GRP  VOLSER NUM  PAV  LCU    RATE TIME TIME CMR DB  TIME TIME TIME UT RV

         VRSRE1 2400      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSRE2 2401      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSWK1 2402      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSWK2 2403      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSSM1 2404      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSSM2 2405      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSSY1 2406      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSSP1 2407      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSCAT 2408      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         VRSSY2 2409      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         MVS013 240A      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         MVS014 240B      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         TSMS07 240C      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         TSMS08 240D      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         TCSHR1 2410      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
         TOTSTZ 2411      000E 0.000  0.0  0.0 0.0 0.0  0.0  0.0  0.0  0  0
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=SORT
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 11-4 Monitor II report showing CMR DELAY

Monitor III changes - Job Report Device Delay Variation

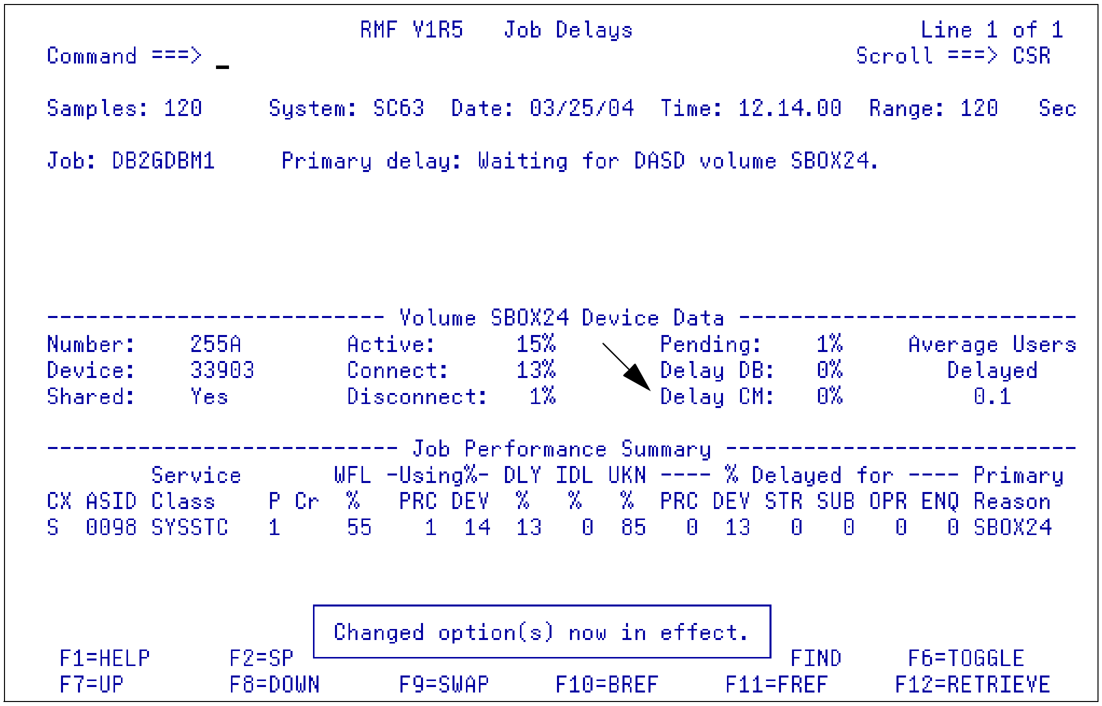
The fields Delay CU and Delay DP in the Pending % have been removed from the Monitor III Device Job Report. The following new field has been added, as shown in Figure 11-5 on page 259:

Delay CM This field has been added to the report to indicate the percentage of time during the range period that a successfully initiated start or resume command needs until it is accepted by the device.

Command response time delay (CM)

Command response time delay, which is the percentage of time during the report interval, when the first command of an I/O instruction of the channel program is sent to the device, until the device indicates it has accepted the command.

```
             Accumulated Command Response Delay Time
Delay CM% = ---------------------------------------
                           Range Time
```



```
                         RMF V1R5   Job Delays                      Line 1 of 1
Command ===> _                                                     Scroll ===> CSR

Samples: 120     System: SC63  Date: 03/25/04  Time: 12.14.00  Range: 120   Sec

Job: DB2GDBM1     Primary delay: Waiting for DASD volume SBOX24.




-------------------------- Volume SBOX24 Device Data --------------------------
Number:    255A        Active:       15%        Pending:    1%     Average Users
Device:    33903       Connect:      13%        Delay DB:   0%        Delayed
Shared:    Yes         Disconnect:    1%        Delay CM:   0%          0.1

--------------------------- Job Performance Summary ---------------------------
        Service          WFL -Using%- DLY IDL UKN ---- % Delayed for ---- Primary
CX ASID Class     P Cr   %   PRC DEV  %   %   %  PRC DEV STR SUB OPR ENQ Reason
S  0098 SYSSTC    1      55    1  14  13   0  85   0  13   0   0   0   0 SBOX24


                       Changed option(s) now in effect.
 F1=HELP      F2=SP                                     FIND      F6=TOGGLE
 F7=UP        F8=DOWN      F9=SWAP     F10=BREF     F11=FREF     F12=RETRIEVE
```

Figure 11-5 Monitor III report showing Delay CM

Monitor III changes - Device Resource Delays report

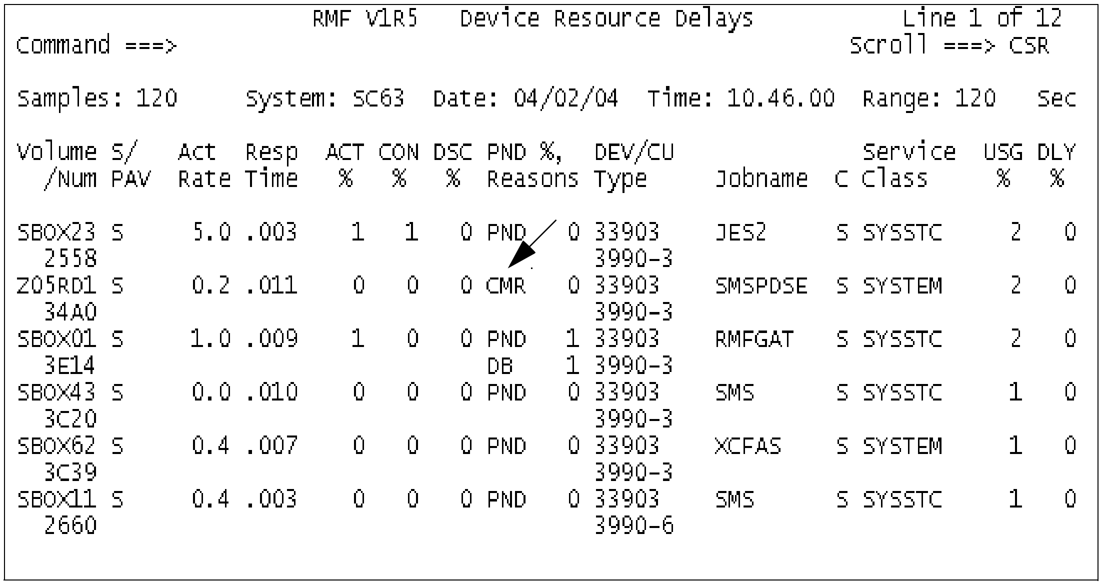
The pending reasons CUB for control unit busy and DPB for director port busy have been removed from the Monitor III Device Resource Delays report. The pending reason CMR for initial command response time has been added to the report, as shown in Figure 11-6 on page 260.

PND % Reasons The first entry is always the pending percentage (PND). See the description under % ACT. RMF calculates the value as follows:

```
             Accumulated Pending Time
PND % = ----------------------- * 100
                   Range Time
```

DLY DB % and DLY CU % are included in pending time.

Below PND % are the pend reasons that contribute to the total pending percentage. A value appears only when there is a non-zero delay percentage.

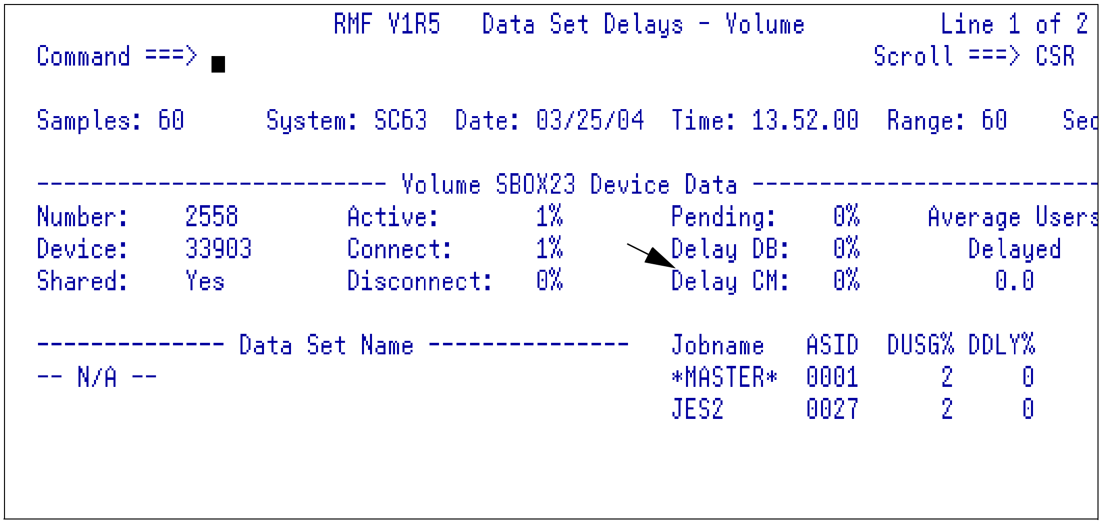


```
                          RMF V1R5   Device Resource Delays          Line 1 of 12
Command ===>                                                     Scroll ===> CSR

Samples: 120     System: SC63  Date: 04/02/04  Time: 10.46.00  Range: 120   Sec

Volume S/   Act  Resp  ACT CON DSC PND %,  DEV/CU            Service   USG DLY
  /Num PAV  Rate Time   %   %   %  Reasons Type     Jobname  C Class     %   %

SBOX23 S     5.0 .003   1   1   0 PND    0 33903    JES2     S SYSSTC    2   0
  2558                                     3990-3
Z05RD1 S     0.2 .011   0   0   0 CMR    0 33903    SMSPDSE  S SYSTEM    2   0
  34A0                                     3990-3
SBOX01 S     1.0 .009   1   0   0 PND    1 33903    RMFGAT   S SYSSTC    2   0
  3E14                              DB     1 3990-3
SBOX43 S     0.0 .010   0   0   0 PND    0 33903    SMS      S SYSSTC    1   0
  3C20                                     3990-3
SBOX62 S     0.4 .007   0   0   0 PND    0 33903    XCFAS    S SYSTEM    1   0
  3C39                                     3990-3
SBOX11 S     0.4 .003   0   0   0 PND    0 33903    SMS      S SYSSTC    1   0
  2660                                     3990-6
```

Figure 11-6 Monitor III report showing PND % Reasons - CMR

Monitor III Changes - Data Set Delays Volume report

The pending reasons CU for control unit busy and DP for director port busy have been removed from the Monitor III Data Set Delays Volume report. The pending reason CMR for initial command response time has been added to the report, as shown in Figure 11-7.

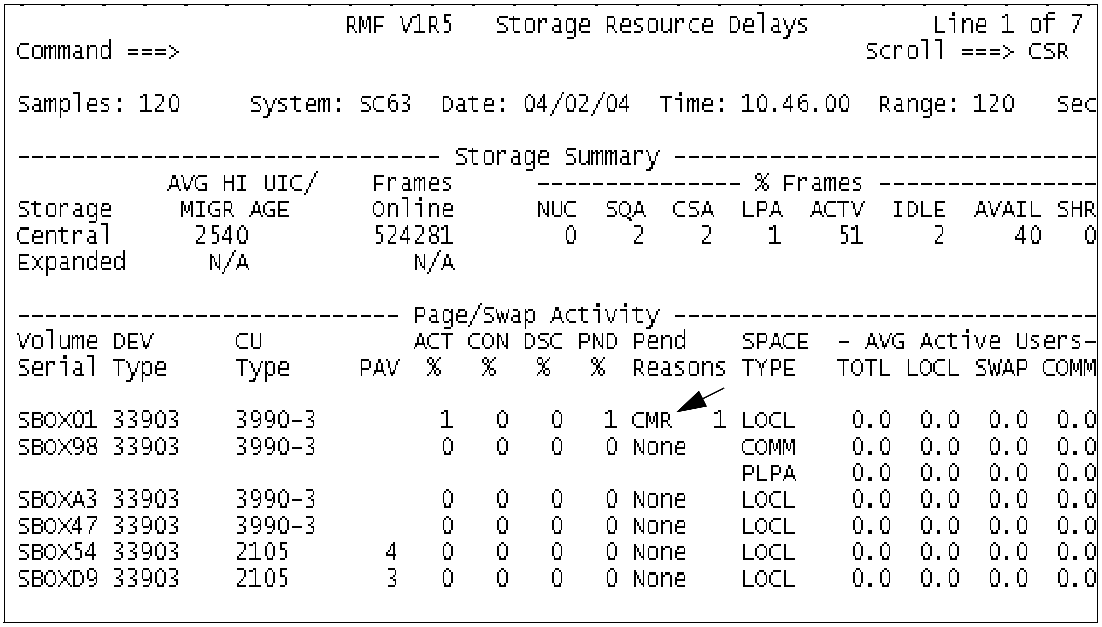


```
                      RMF V1R5   Data Set Delays - Volume         Line 1 of 2
Command ===>                                                  Scroll ===> CSR

Samples: 60      System: SC63  Date: 03/25/04  Time: 13.52.00  Range: 60    Sec

-------------------------- Volume SBOX23 Device Data --------------------------
Number:    2558       Active:       1%       Pending:   0%     Average Users
Device:    33903      Connect:      1%       Delay DB:  0%        Delayed
Shared:    Yes        Disconnect:   0%       Delay CM:  0%          0.0

-------------- Data Set Name ---------------  Jobname   ASID  DUSG% DDLY%
-- N/A --                                     *MASTER*  0001      2     0
                                              JES2      0027      2     0
```

Figure 11-7 Monitor III Data Set Delays - Volume report showing Delay CMR

Monitor III Changes - Storage Resource Delays report

The pending reasons CU for control unit busy and DP for director port busy have been removed from the Monitor III Storage Resource Delays report. The pending reason CMR for initial command response time has been added to the report, as shown in Figure 11-8 on page 261.



```
                        RMF V1R5   Storage Resource Delays        Line 1 of 7
Command ===>                                                    Scroll ===> CSR

Samples: 120     System: SC63  Date: 04/02/04  Time: 10.46.00  Range: 120   Sec

------------------------------- Storage Summary -------------------------------
           AVG HI UIC/    Frames      --------------- % Frames ---------------
Storage     MIGR AGE      Online      NUC  SQA  CSA  LPA  ACTV  IDLE  AVAIL SHR
Central      2540         524281        0    2    2    1    51     2     40   0
Expanded      N/A            N/A

---------------------------- Page/Swap Activity ------------------------------
Volume DEV      CU             ACT CON DSC PND Pend     SPACE  - AVG Active Users-
Serial Type     Type       PAV  %   %   %   %  Reasons  TYPE   TOTL LOCL SWAP COMM

SBOX01 33903    3990-3           1   0   0   1 CMR    1 LOCL    0.0  0.0  0.0  0.0
SBOX98 33903    3990-3           0   0   0   0 None     COMM    0.0  0.0  0.0  0.0
                                                        PLPA    0.0  0.0  0.0  0.0
SBOXA3 33903    3990-3           0   0   0   0 None     LOCL    0.0  0.0  0.0  0.0
SBOX47 33903    3990-3           0   0   0   0 None     LOCL    0.0  0.0  0.0  0.0
SBOX54 33903    2105         4   0   0   0   0 None     LOCL    0.0  0.0  0.0  0.0
SBOXD9 33903    2105         3   0   0   0   0 None     LOCL    0.0  0.0  0.0  0.0
```

Figure 11-8 Monitor III Storage Resource Delays report showing CMR

11.2.2 I/O Queuing reports

The primary impact in RMF I/O Queueing Activity processing also occurs in device data gathering due to the creation of the new control block to support 64-bit z/architecture. This has resulted in several field level changes in the I/O Queueing reports for Monitor I, II, and III. These changes are described in the following sections.

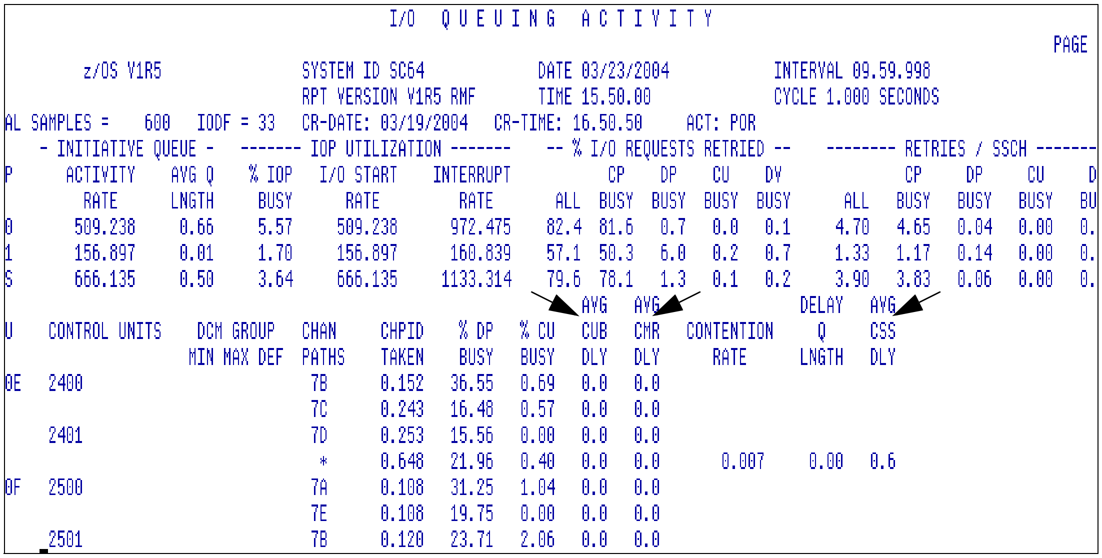
Monitor I changes

Three new columns have been added to the I/O Queuing report. They are the AVG CSS DLY, the AVG CUB DLY and the AVG CMR DLY, as shown in Figure 11-9 on page 262.

AVG CSS DLY This is the average number of milliseconds of delay an I/O request experienced after acceptance of the start or resume function at the subchannel for the LCU, until the channel subsystem attempted to initiate the operation.

AVG CUB DLY This is the average number of milliseconds of delay an I/O request experienced for the channel path due to a busy control unit.

AVG CMR DLY This is the average command response time, in milliseconds, that a successfully initiated start or resume command needs until it is accepted by a device.

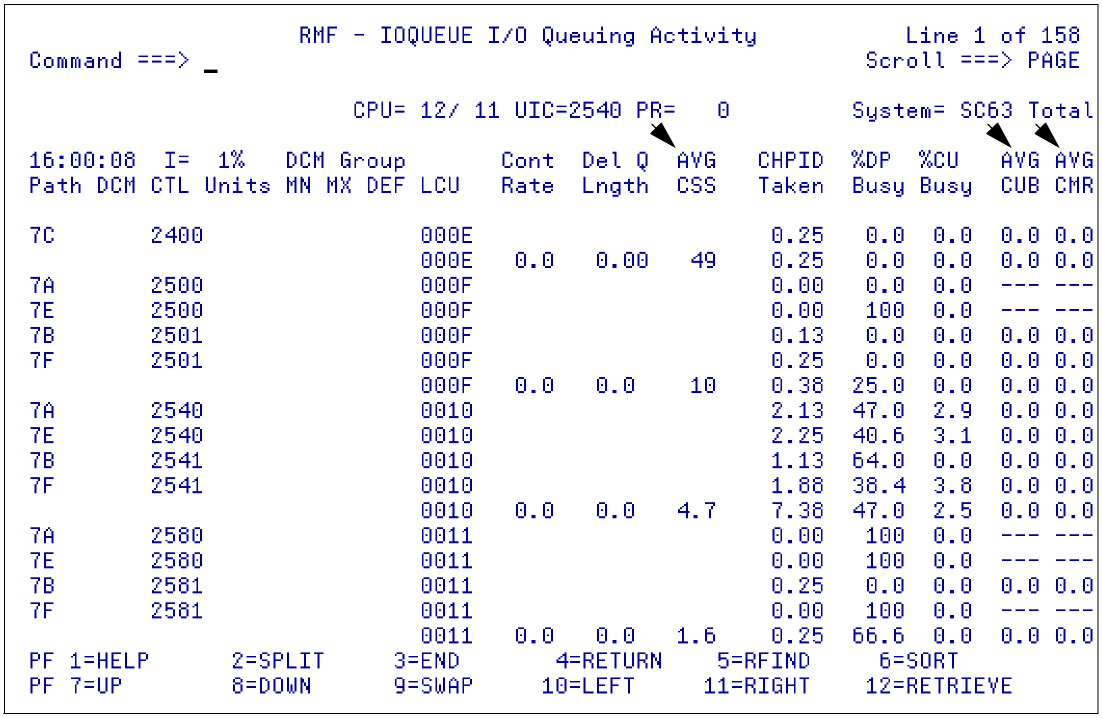


```
                                          I/O   Q U E U I N G   A C T I V I T Y
                                                                                               PAGE
        z/OS V1R5               SYSTEM ID SC64            DATE 03/23/2004           INTERVAL 09.59.998
                                RPT VERSION V1R5 RMF      TIME 15.50.00             CYCLE 1.000 SECONDS
AL SAMPLES =     600   IODF = 33   CR-DATE: 03/19/2004   CR-TIME: 16.50.50     ACT: POR
    - INITIATIVE QUEUE -   ------- IOP UTILIZATION -------    -- % I/O REQUESTS RETRIED --    -------- RETRIES / SSCH -------
P       ACTIVITY    AVG Q    % IOP    I/O START    INTERRUPT          CP    DP    CU    DV             CP     DP     CU     D
          RATE      LNGTH     BUSY      RATE         RATE      ALL  BUSY  BUSY  BUSY  BUSY      ALL  BUSY   BUSY   BUSY   BU
0        509.238     0.66     5.57    509.238      972.475    82.4  81.6   0.7   0.0   0.1      4.70  4.65   0.04   0.00   0.
1        156.897     0.01     1.70    156.897      160.839    57.1  50.3   6.0   0.2   0.7      1.33  1.17   0.14   0.00   0.
S        666.135     0.50     3.64    666.135     1133.314    79.6  78.1   1.3   0.1   0.2      3.90  3.83   0.06   0.00   0.
                                                                        AVG   AVG                   DELAY   AVG
U    CONTROL UNITS    DCM GROUP    CHAN     CHPID    % DP    % CU   CUB   CMR   CONTENTION     Q      CSS
                     MIN MAX DEF   PATHS    TAKEN    BUSY    BUSY   DLY   DLY      RATE       LNGTH   DLY
0E   2400                           7B      0.152   36.55    0.69   0.0   0.0
                                    7C      0.243   16.48    0.57   0.0   0.0
     2401                           7D      0.253   15.56    0.00   0.0   0.0
                                     *      0.648   21.96    0.40   0.0   0.0       0.007      0.00   0.6
0F   2500                           7A      0.108   31.25    1.04   0.0   0.0
                                    7E      0.108   19.75    0.00   0.0   0.0
     2501                           7B      0.120   23.71    2.06   0.0   0.0
```

Figure 11-9 Monitor I I/O Queuing report with new fields

Monitor II changes

Three new columns have been added to the Monitor II I/O Queuing report. They are the AVG CSS, the AVG CUB, and the AVG CMR, as shown in Figure 11-10.

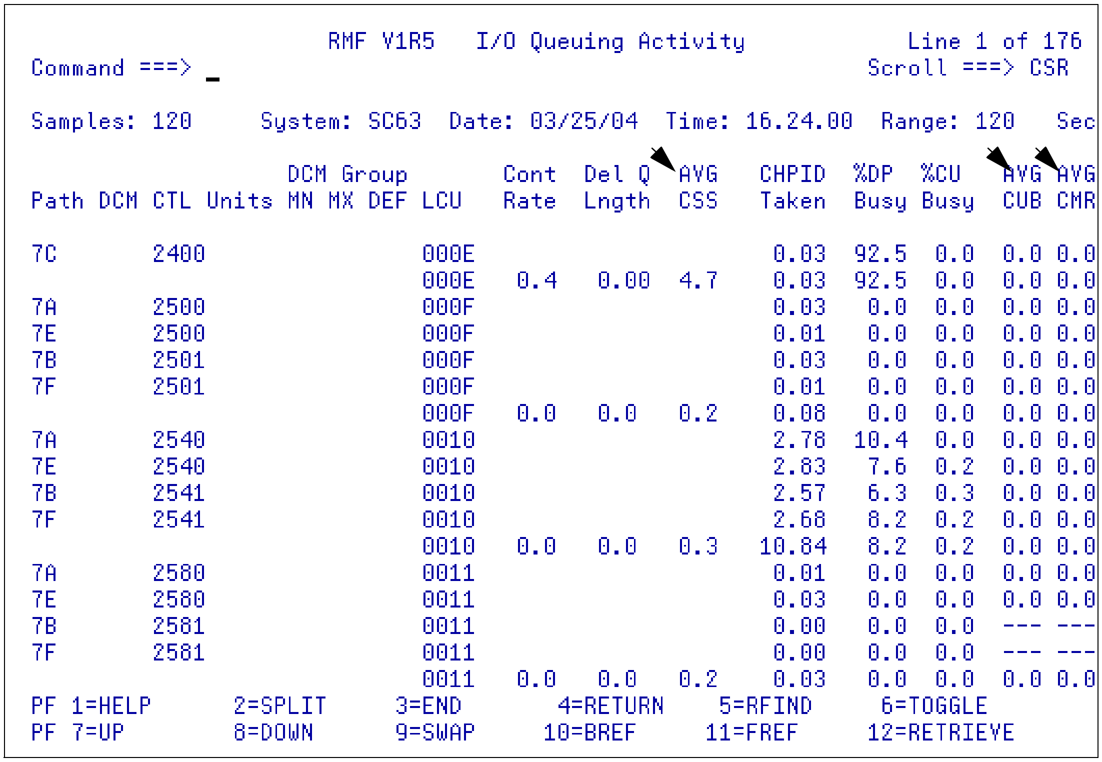


```
                    RMF - IOQUEUE I/O Queuing Activity           Line 1 of 158
Command ===> _                                                Scroll ===> PAGE

                        CPU= 12/ 11 UIC=2540 PR=   0         System= SC63 Total

16:00:08  I=  1%   DCM Group      Cont  Del Q  AVG   CHPID  %DP  %CU   AVG AVG
Path DCM CTL Units MN MX DEF LCU  Rate  Lngth  CSS   Taken  Busy Busy  CUB CMR

7C       2400                000E                      0.25   0.0  0.0  0.0 0.0
                             000E   0.0   0.00   49    0.25   0.0  0.0  0.0 0.0
7A       2500                000F                      0.00   0.0  0.0  --- ---
7E       2500                000F                      0.00   100  0.0  --- ---
7B       2501                000F                      0.13   0.0  0.0  0.0 0.0
7F       2501                000F                      0.25   0.0  0.0  0.0 0.0
                             000F   0.0   0.0    10    0.38  25.0  0.0  0.0 0.0
7A       2540                0010                      2.13  47.0  2.9  0.0 0.0
7E       2540                0010                      2.25  40.6  3.1  0.0 0.0
7B       2541                0010                      1.13  64.0  0.0  0.0 0.0
7F       2541                0010                      1.88  38.4  3.8  0.0 0.0
                             0010   0.0   0.0   4.7    7.38  47.0  2.5  0.0 0.0
7A       2580                0011                      0.00   100  0.0  --- ---
7E       2580                0011                      0.00   100  0.0  --- ---
7B       2581                0011                      0.25   0.0  0.0  0.0 0.0
7F       2581                0011                      0.00   100  0.0  --- ---
                             0011   0.0   0.0   1.6    0.25  66.6  0.0  0.0 0.0
PF 1=HELP      2=SPLIT     3=END       4=RETURN    5=RFIND     6=SORT
PF 7=UP        8=DOWN      9=SWAP     10=LEFT     11=RIGHT    12=RETRIEVE
```

Figure 11-10 Monitor II I/O queuing Activity report

Monitor III changes

Three new columns have been added to the Monitor III I/O Queuing Activity report. They are the AVG CSS, the AVG CUB, and the AVG CMR, as shown in Figure 11-11.

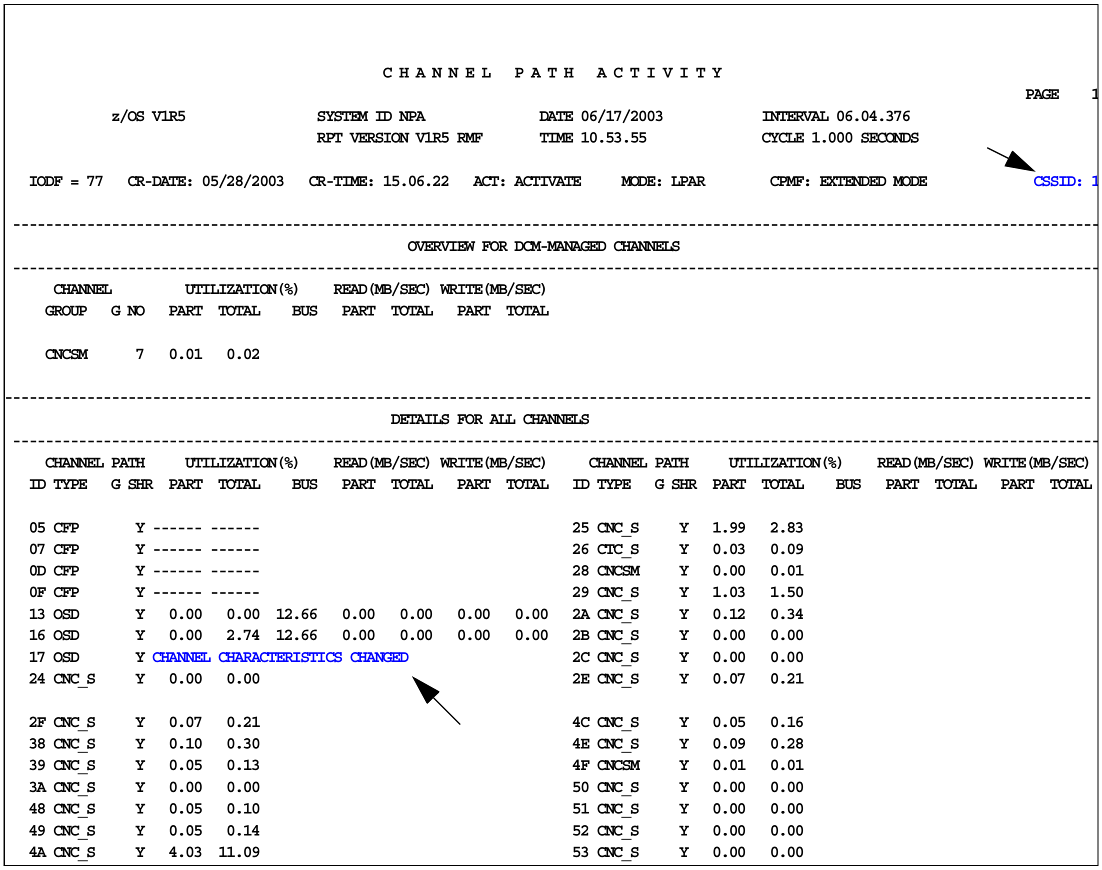


```
                      RMF V1R5   I/O Queuing Activity            Line 1 of 176
Command ===> _                                                Scroll ===> CSR

Samples: 120     System: SC63  Date: 03/25/04  Time: 16.24.00  Range: 120   Sec

                   DCM Group       Cont  Del Q  AVG   CHPID  %DP  %CU   AVG AVG
Path DCM CTL Units MN MX DEF LCU   Rate  Lngth  CSS   Taken  Busy Busy  CUB CMR

7C       2400                000E                       0.03  92.5  0.0  0.0 0.0
                             000E   0.4   0.00  4.7     0.03  92.5  0.0  0.0 0.0
7A       2500                000F                       0.03   0.0  0.0  0.0 0.0
7E       2500                000F                       0.01   0.0  0.0  0.0 0.0
7B       2501                000F                       0.03   0.0  0.0  0.0 0.0
7F       2501                000F                       0.01   0.0  0.0  0.0 0.0
                             000F   0.0   0.0   0.2     0.08   0.0  0.0  0.0 0.0
7A       2540                0010                       2.78  10.4  0.0  0.0 0.0
7E       2540                0010                       2.83   7.6  0.2  0.0 0.0
7B       2541                0010                       2.57   6.3  0.3  0.0 0.0
7F       2541                0010                       2.68   8.2  0.2  0.0 0.0
                             0010   0.0   0.0   0.3    10.84   8.2  0.2  0.0 0.0
7A       2580                0011                       0.01   0.0  0.0  0.0 0.0
7E       2580                0011                       0.03   0.0  0.0  0.0 0.0
7B       2581                0011                       0.00   0.0  0.0  --- ---
7F       2581                0011                       0.00   0.0  0.0  --- ---
                             0011   0.0   0.0   0.2     0.03   0.0  0.0  0.0 0.0
PF 1=HELP      2=SPLIT     3=END       4=RETURN    5=RFIND     6=TOGGLE
PF 7=UP        8=DOWN      9=SWAP     10=BREF     11=FREF     12=RETRIEVE
```

Figure 11-11 Monitor III I/O Queuing Activity report

11.2.3 Channel Activity reports

The RMF Monitor I Postprocessor Channel Activity report has been updated to display the Channel Subsystem Identifier (CSSID); the Channel Characteristics Changed indication is shown if the characteristics of the channel have changed during the displayed interval.



```
                                           C H A N N E L   P A T H   A C T I V I T Y
                                                                                                                    PAGE    1
         z/OS V1R5               SYSTEM ID NPA            DATE 06/17/2003          INTERVAL 06.04.376
                                 RPT VERSION V1R5 RMF     TIME 10.53.55            CYCLE 1.000 SECONDS

 IODF = 77   CR-DATE: 05/28/2003   CR-TIME: 15.06.22   ACT: ACTIVATE     MODE: LPAR       CPMF: EXTENDED MODE            CSSID: 1

------------------------------------------------------------------------------------------------------------------------------
                                           OVERVIEW FOR DCM-MANAGED CHANNELS
------------------------------------------------------------------------------------------------------------------------------
    CHANNEL         UTILIZATION(%)    READ(MB/SEC) WRITE(MB/SEC)
   GROUP   G NO   PART  TOTAL    BUS   PART  TOTAL   PART  TOTAL

   CNCSM      7   0.01   0.02

------------------------------------------------------------------------------------------------------------------------------
                                         DETAILS FOR ALL CHANNELS
------------------------------------------------------------------------------------------------------------------------------
   CHANNEL PATH     UTILIZATION(%)    READ(MB/SEC) WRITE(MB/SEC)    CHANNEL PATH     UTILIZATION(%)    READ(MB/SEC) WRITE(MB/SEC)
 ID TYPE   G SHR  PART  TOTAL    BUS   PART  TOTAL   PART  TOTAL  ID TYPE   G SHR  PART  TOTAL    BUS   PART  TOTAL   PART  TOTAL

 05 CFP      Y ------ ------                                      25 CNC_S    Y   1.99   2.83
 07 CFP      Y ------ ------                                      26 CTC_S    Y   0.03   0.09
 0D CFP      Y ------ ------                                      28 CNCSM    Y   0.00   0.01
 0F CFP      Y ------ ------                                      29 CNC_S    Y   1.03   1.50
 13 OSD      Y   0.00   0.00  12.66   0.00   0.00   0.00   0.00  2A CNC_S    Y   0.12   0.34
 16 OSD      Y   0.00   2.74  12.66   0.00   0.00   0.00   0.00  2B CNC_S    Y   0.00   0.00
 17 OSD      Y CHANNEL CHARACTERISTICS CHANGED                    2C CNC_S    Y   0.00   0.00
 24 CNC_S    Y   0.00   0.00                                      2E CNC_S    Y   0.07   0.21

 2F CNC_S    Y   0.07   0.21                                      4C CNC_S    Y   0.05   0.16
 38 CNC_S    Y   0.10   0.30                                      4E CNC_S    Y   0.09   0.28
 39 CNC_S    Y   0.05   0.13                                      4F CNCSM    Y   0.01   0.01
 3A CNC_S    Y   0.00   0.00                                      50 CNC_S    Y   0.00   0.00
 48 CNC_S    Y   0.05   0.10                                      51 CNC_S    Y   0.00   0.00
 49 CNC_S    Y   0.05   0.14                                      52 CNC_S    Y   0.00   0.00
 4A CNC_S    Y   4.03  11.09                                      53 CNC_S    Y   0.00   0.00
```

11.2.4 RMF Spreadsheet Reporter

The RMF Spreadsheet Reporter has been updated to work with all of the changed RMF Monitor I Postprocessor reports as of version 4.8.2 that is shipped with APAR OW56656.

The RMF Spreadsheet Reporter is available for download via the RMF homepage:

`http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/`

11.2.5 RMF Performance Monitor

RMF Performance Monitor provides the new Distributed Data Server (DDS) that ships with APAR OW56656. The DDS is the host server component that provides the performance data for RMF PM, as well as other applications that may want to use the data collected by RMF. The DDS also provides TCP/IP interfaces that can be used by networked applications to connect to the server and retrieve z/OS-based RMF performance data.

Similar to the other reporting facilities of RMF, RMF PM no longer supports the use of the following device metrics:

- Percentage of delay for director port busy
- Percentage of delay for CU busy

The following LCU metrics have also been removed from RMF PM:

- Percentage of delay of CU by volume
- Percentage of delay of director port busy by volume

The following device metrics have been added to RMF PM to support the z/990:

- Percentage of delay due to device command response time

The following LCU metrics have been added to RMF in support of the z/990.

- Average CU busy time by channel path and LCU
- Average command response time by channel path and LCU
- Average channel subsystem delay time by LCU
- Percentage of delay to device command response time by volume

11.3 RMF and multilevel security

Multilevel security provides for the classification of users and data using a system of hierarchical security levels in combination with non-hierarchical security categories. z/OS V1R5 includes several enhancements to support MLS.

11.3.1 Name hiding

Access to and usage of RMF is limited to users who have been granted authority to do so by the security administrator. RMF depends on the existence of this level of security and as such does not perform security label dominance checking for any information presented in Monitor II or Monitor III tabular report displays. Additionally, dataset names are not hidden in any RMF report. There are several opportunities for unauthorized access to dataset names from different areas of RMF. Enhancements have been provided to effectively block unauthorized access of dataset name information.

Prior to z/OS V1R5, RMF stored control blocks containing dataset names in common unprotected storage in the SQA. This allowed unauthorized users and unauthorized programs access to the dataset names. RMF at z/OS V1R5, has closed this exposure and secured the dataset name information from unauthorized users and programs by moving the control blocks containing this information to common fetch protected storage.

RMF sysplex data services

RMF sysplex data services ERBDSQRY and ERBDSREC access to SMF data is controlled by the RACF FACILITY class ERBSDS.SMFDATA. RMF Monitor II sysplex data gathering and interface services do not require an authorized caller. The RMF Monitor III sysplex data retrieval service also does not require an authorized caller. This creates an exposure for Monitor II SMF type 79 data and Monitor III data that contain dataset name information in that they may be accessed by an unauthorized user or program.

FACILITY class services

New RACF Facility class profiles have been introduced to control access to Monitor II and Monitor III data. The Monitor II services, ERB2XDGS and ERBSMFI are protected by the new RACF resource ERBSDS.MON2DATA. The Monitor III Service ERB3XDRS is protected by the new RACF resource ERBSDS.MON3DATA.

If this level of protection is not required, the resource profile should not be defined.

If these resource profiles are defined by the customer, every user of the associated services must be authorized to the appropriate resource. Users of these services may include RMF PM users, users executing programs that invoke RMF services ERB2XDGS, ERBSMFI, or ERB3XDRS, users of Monitor III sysplex-wide reports, and users of cross-system reporting capabilities in Monitor II or Monitor III.

To activate the RACF resource class:

```
SETROPTS CLASSACT(FACILITY) GENCMD(FACILITY) GENERIC(FACILITY)
```

To define the profile for the RMF sysplex data services and permit read access to the userid of the application program:

```
RDEFINE FACILITY ERBSDS.SMFDATA UACC(NONE)
PERMIT ERBSDS.SMFDATA CLASS(FACILITY) ID(userid) ACC(READ)
```

To define the profile for the RMF Monitor II sysplex data gathering and interface services and permit read access to the userid of the application program:

```
RDEFINE FACILITY ERBSDS.MON2DATA UACC(NONE)
PERMIT ERBSDS.MON2DATA CLASS(FACILITY) ID(userid) ACC(READ)
```

To define the profile for the RMF Monitor III sysplex data retrieval service and permit read access to the userid of the application program:

```
RDEFINE FACILITY ERBSDS.MON3DATA UACC(NONE)
PERMIT ERBSDS.MON3 CLASS(FACILITY) ID(userid) ACC(READ)
```

To generically define the profiles for all of the new RMF resources and permit read access to the userid of the application program:

```
RDEFINE FACILITY ERBSDS.* UACC(NONE)
PERMIT ERBSDS.* CLASS(FACILITY) ID(userid) ACC(READ)
```

To activate the changes you have made:

```
SETROPTS REFRESH RACLIST(FACILITY)
```

11.3.2 Protection of z/OS performance data

Prior to z/OS V1R5, access to z/OS performance data by RMF Performance Monitor Java Client users could not be protected. The introduction of the RACF Facility class profile ERBSDS.MON3DATA provides the ability to secure z/OS performance data and to permit access to authorized users only.

To enable this level of security, the RMF PM users should be granted read access to the ERBSDS.MON3DATA resource:

```
PERMIT ERBSDS.MON3 CLASS(FACILITY) ID(RMF_PM_userid) ACC(READ)
```

If access to the data is not granted, the RMF PM user will receive the following message:

```
GPM0456I - The userid <uid> is not authorized to retrieve RMF performance data
```

Note: Those users wishing to implement security authorization for Monitor II and Monitor III data who are using LDAP to access Monitor III data should be aware that no authorization check can be done for the ERBSDS.MON3DATA resource. The only option to prevent unwanted access through the LDAP interface is to disable RMF LDAP requests in general. This can be accomplished by removing the RMF LDAP backend ERB6LBCK in the SLAPD.CONF z/OS LDAP server configuration file.

11.4 RMF storage reporting enhancements

RMF storage reporting has been enhanced in z/OS V1R5 to support reporting of shared memory usage in a 64-bit environment. These enhancements have occurred in the RMF Monitor I Overview report and the Virtual Storage report. Monitor III STORR and STORS reports have also been updated to include shared page values.

11.4.1 Monitor I Overview report

System-wide Overview reporting has been enhanced with the addition of Overview conditions that provide usage counts for all Shared Memory segments.

Table 11-1 displays the new overview conditions and their descriptions for RMF Shared Memory support below the bar.

Table 11-1 New RMF overview conditions for shared memory support below the bar

Condition name	Condition
SHRPT	Average total number of shared page groups in the system
SHRPC	Average number of shared page groups in central storage
SHRPA	Average number of shared page groups in auxiliary storage
SHRPF	Average number of shared pages fixed
SHRPB	Average number of shared pages fixed below 16MB
SHRPI	Number of page-ins from auxiliary storage for shared pages
SHRPO	Number of page-outs to auxiliary storage for shared pages

Table 11-2 displays the new overview conditions and their descriptions for RMF Shared Memory support above the bar.

Table 11-2 New RMF overview conditions for shared memory support above the bar

Condition name	Condition
SHRPTH	Average total number of shared pages in the system with a virtual address above the bar
SHRPCH	Average number of shared pages in central storage with a virtual address above the bar
SHRPAH	Average number of shared pages in auxiliary storage with a virtual address above the bar
SHRPBLG	High water mark for number of shared bytes from large virtual memory in memory object for entire system
SHRPIH	Number of page-ins from auxiliary storage for shared pages with a virtual storage address above the bar
SHRPOH	Number of page-outs to auxiliary storage for shared pages with a virtual storage address above the bar

Report JCL

Figure 11-12 on page 268 is an example of JCL and control statements that can be used to generate an RMF Monitor I Postprocessor Overview report.

```
//OVERVIEW JOB (0),'RMF PAGING OVERVIEW',CLASS=A,MSGCLASS=X
//RMF      EXEC PGM=ERBRMFPP
//MFPINPUT DD DSN=SORTED.RMF,DISP=SHR
//MFPMSGDS DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
  OVW(PSYS(SHRPT))
  OVW(PSYSHI(SHRPTH))
  OVW(PCEN(SHRPC))
  OVW(PCENHI(SHRPCH))
  OVW(PAUX(SHRPA))
  OVW(PAUXHI(SHRPAH))
  OVW(PFIX(SHRPF))
  OVW(PLOW(SHRPB))
  OVW(BYTEHI(SHRPBLG))
  OVERVIEW(REPORT)
  SYSOUT(X)
//*
```

Figure 11-12 RMF Monitor I Postprocessor Overview report JCL

Monitor Overview report

Figure 11-13 is the resulting RMF Monitor I Postprocessor Overview report generated from the preceding JCL.

Figure 11-13 Monitor Overview report

Postprocessor report JCL

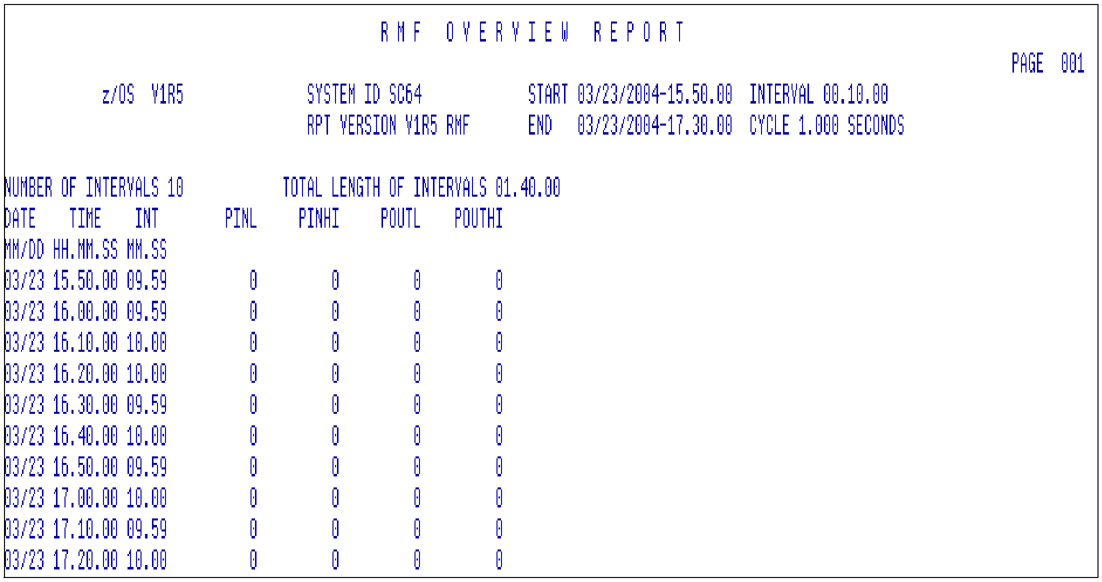
Figure 11-14 on page 269 is an example of JCL and control statements that can be used to generate an RMF Monitor I Postprocessor Overview report showing paging counts.

```
//OVERVIEW JOB (0),'RMF PAGING OVERVIEW',CLASS=A,MSGCLASS=X
//RMF      EXEC PGM=ERBRMFPP
//MFPINPUT DD DSN=SORTED.RMF,DISP=SHR
//MFPMSGDS DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
  OVW(PINL(SHRPI))
  OVW(PINHI(SHRPIH))
  OVW(POUTL(SHRPO))
  OVW(POUTHI(SHRPOH))
  OVERVIEW(REPORT)
  SYSOUT(X)
//*
```

Figure 11-14 Monitor I Postprocessor report JCL

Postprocessor report

Figure 11-15 is the resulting RMF Monitor I Postprocessor Overview report generated from the preceding JCL.



```
                              R M F   O V E R V I E W   R E P O R T
                                                                                        PAGE  001
        z/OS  V1R5            SYSTEM ID SC64          START 03/23/2004-15.50.00  INTERVAL 00.10.00
                              RPT VERSION V1R5 RMF    END   03/23/2004-17.30.00  CYCLE 1.000 SECONDS

NUMBER OF INTERVALS 10        TOTAL LENGTH OF INTERVALS 01.40.00
DATE   TIME    INT       PINL    PINHI    POUTL    POUTHI
MM/DD HH.MM.SS MM.SS
03/23 15.50.00 09.59        0        0        0        0
03/23 16.00.00 09.59        0        0        0        0
03/23 16.10.00 10.00        0        0        0        0
03/23 16.20.00 10.00        0        0        0        0
03/23 16.30.00 09.59        0        0        0        0
03/23 16.40.00 10.00        0        0        0        0
03/23 16.50.00 09.59        0        0        0        0
03/23 17.00.00 10.00        0        0        0        0
03/23 17.10.00 09.59        0        0        0        0
03/23 17.20.00 10.00        0        0        0        0
```

Figure 11-15 Monitor I Postprocessor Overview report

11.4.2 Monitor I VSTOR report

The Monitor I Virtual storage (VSTOR) report has four unique sections. They are:

- Common storage summary
- Common storage detail
- Private area summary
- Private area detail

Common area summary

The common area summary section of the VTSOR report is generated by default and is a system-wide overview. To produce the common storage summary section of the VSTOR report use the following Postprocessor option:

```
REPORTS(VSTOR)
```

Common area detail

To produce the common area summary and common storage detail sections of the VSTOR report use the following Postprocessor option:

```
REPORTS(VSTOR(D))
```

Private area summary

The private area sections of the report are only generated when specific address spaces are specified in the RMF Monitor I VSTOR gathering and reporting options. To produce the Common Area Summary, Common Storage Detail, and Private Area Summary sections of the VSTOR report for a specific address space use the following Postprocessor option:

```
REPORTS(VSTOR(address_space_name))
```

Private area detail

The detail sections of the report are only produced if the Detail parameter is specified in the RMF Monitor I VSTOR gathering and reporting options. To produce all summary sections and the detail sections of the VSTOR report for a specific address space use the following Postprocessor option:

```
REPORTS(VSTOR(D,address_space_name))
```

VSTOR report JCL

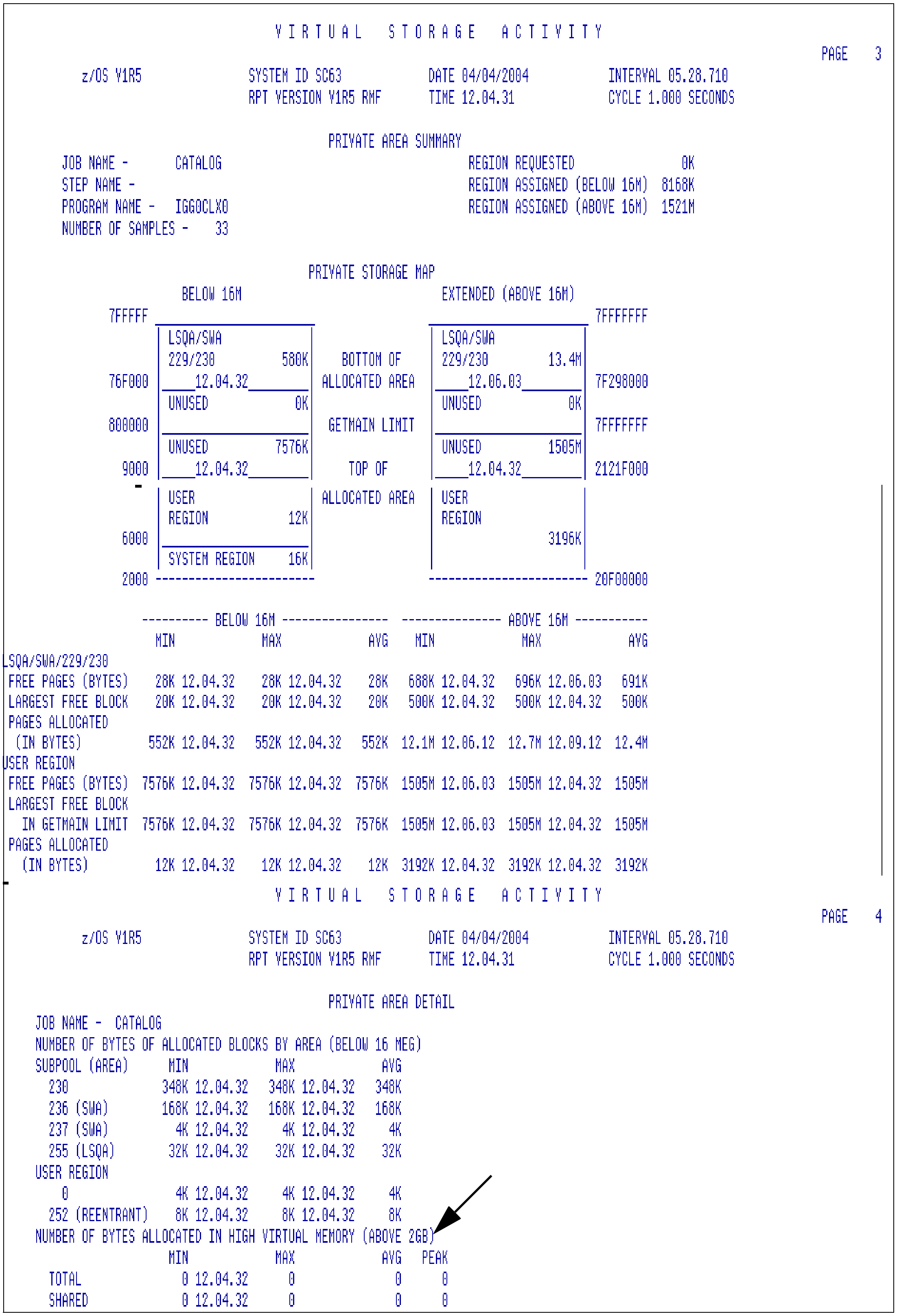
Figure 11-16 displays a JCL example of how to generate the VSTOR summary and detail sections of the VSTOR report for a specific address space, which in this example is the CATALOG address space.

```
//RMFJCL JOB (O),'RMF VSTOR',CLASS=A,MSGCLASS=X
//RMF      EXEC PGM=ERBRMFPP
//MFPINPUT DD DSN=SORTED.RMF,DISP=SHR
//MFPMSGDS DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
  REPORTS(VSTOR(D,CATALOG))
  SYSOUT(X)
//*
```

Figure 11-16 VSTOR report JCL

VSTOR report

Figure 11-17 on page 271 contains an excerpt from the report generated which shows the Private Area Detail section of the VSTOR report that has been extended to support shared memory usage in a 64-bit environment.



```
                              V I R T U A L   S T O R A G E   A C T I V I T Y
                                                                                           PAGE    3
      z/OS V1R5             SYSTEM ID SC63          DATE 04/04/2004         INTERVAL 05.28.710
                            RPT VERSION V1R5 RMF    TIME 12.04.31           CYCLE 1.000 SECONDS

                                        PRIVATE AREA SUMMARY
    JOB NAME -      CATALOG                                  REGION REQUESTED             0K
    STEP NAME -                                              REGION ASSIGNED (BELOW 16M)  8168K
    PROGRAM NAME -  IGG0CLX0                                 REGION ASSIGNED (ABOVE 16M)  1521M
    NUMBER OF SAMPLES -    33

                                     PRIVATE STORAGE MAP
                  BELOW 16M                              EXTENDED (ABOVE 16M)
         7FFFFF ________________________          ________________________ 7FFFFFFF
                | LSQA/SWA             |          | LSQA/SWA             |
                | 229/230         580K |  BOTTOM OF  | 229/230       13.4M |
         76F000 |____12.04.32__________| ALLOCATED AREA |____12.06.03__________| 7F298000
                | UNUSED            0K |          | UNUSED            0K |
         800000 |______________________|  GETMAIN LIMIT |______________________| 7FFFFFFF
                | UNUSED         7576K |          | UNUSED         1505M |
           9000 |____12.04.32__________|   TOP OF    |____12.04.32__________| 2121F000
                | USER                 | ALLOCATED AREA | USER                 |
                | REGION           12K |          | REGION               |
           6000 |                      |          |                3196K |
                |______________________|          |                      |
                | SYSTEM REGION    16K |          |                      |
           2000 ------------------------          ------------------------ 20F00000

               ---------- BELOW 16M ----------------  -------------- ABOVE 16M -----------
                 MIN             MAX            AVG    MIN             MAX            AVG
LSQA/SWA/229/230
FREE PAGES (BYTES)    28K 12.04.32    28K 12.04.32    28K   688K 12.04.32   696K 12.06.03   691K
LARGEST FREE BLOCK    20K 12.04.32    20K 12.04.32    20K   500K 12.04.32   500K 12.04.32   500K
PAGES ALLOCATED
 (IN BYTES)          552K 12.04.32   552K 12.04.32   552K  12.1M 12.06.12  12.7M 12.09.12  12.4M
USER REGION
FREE PAGES (BYTES)  7576K 12.04.32  7576K 12.04.32  7576K  1505M 12.06.03  1505M 12.04.32  1505M
LARGEST FREE BLOCK
  IN GETMAIN LIMIT  7576K 12.04.32  7576K 12.04.32  7576K  1505M 12.06.03  1505M 12.04.32  1505M
PAGES ALLOCATED
  (IN BYTES)          12K 12.04.32    12K 12.04.32    12K  3192K 12.04.32  3192K 12.04.32  3192K
                              V I R T U A L   S T O R A G E   A C T I V I T Y
                                                                                           PAGE    4
      z/OS V1R5             SYSTEM ID SC63          DATE 04/04/2004         INTERVAL 05.28.710
                            RPT VERSION V1R5 RMF    TIME 12.04.31           CYCLE 1.000 SECONDS

                                        PRIVATE AREA DETAIL
  JOB NAME -  CATALOG
  NUMBER OF BYTES OF ALLOCATED BLOCKS BY AREA (BELOW 16 MEG)
  SUBPOOL (AREA)       MIN             MAX            AVG
    230              348K 12.04.32   348K 12.04.32   348K
    236 (SWA)        168K 12.04.32   168K 12.04.32   168K
    237 (SWA)          4K 12.04.32     4K 12.04.32     4K
    255 (LSQA)        32K 12.04.32    32K 12.04.32    32K
  USER REGION
     0                 4K 12.04.32     4K 12.04.32     4K
    252 (REENTRANT)    8K 12.04.32     8K 12.04.32     8K
  NUMBER OF BYTES ALLOCATED IN HIGH VIRTUAL MEMORY (ABOVE 2GB)
                       MIN             MAX            AVG   PEAK
    TOTAL              0 12.04.32      0              0      0
    SHARED             0 12.04.32      0              0      0
```

Figure 11-17   VSTOR report excerpt

New fields for VSTOR report

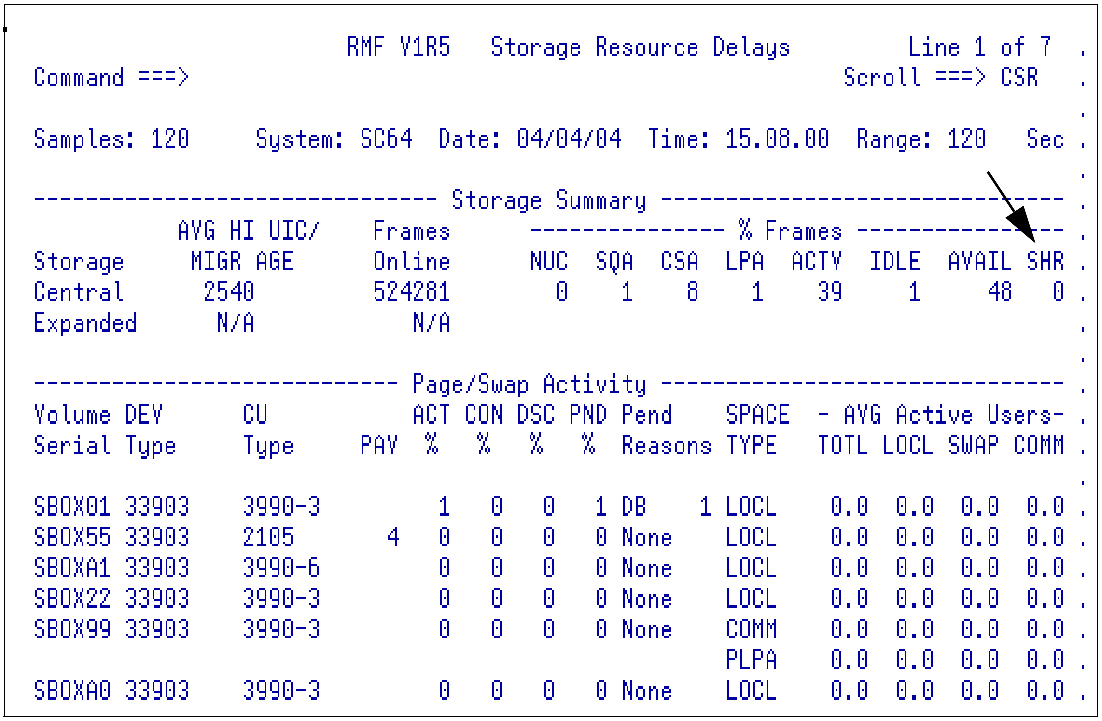
The new fields added to the Extended Private Area Detail Section of the VSTOR report are defined in Table 11-3.

Table 11-3 New fields for RMF VSTOR report

New field heading	Meaning
TOTAL	Total number of byes allocated **MIN** - the minimum value for the interval **MAX** - the maximum value for the interval **AVG** - the average value for the interval **PEAK** - the high watermark since address space initiation
SHARED	Number of byes of shared memory **MIN** - the minimum value for the interval **MAX** - the maximum value for the interval **AVG** - the average value for the interval **PEAK** - the high watermark since address space initiation

11.4.3 Monitor III Shared Page support

The RMF Monitor III Storage Resource Delay report has been updated to display the percentage of frames shared, as shown in Figure 11-18.

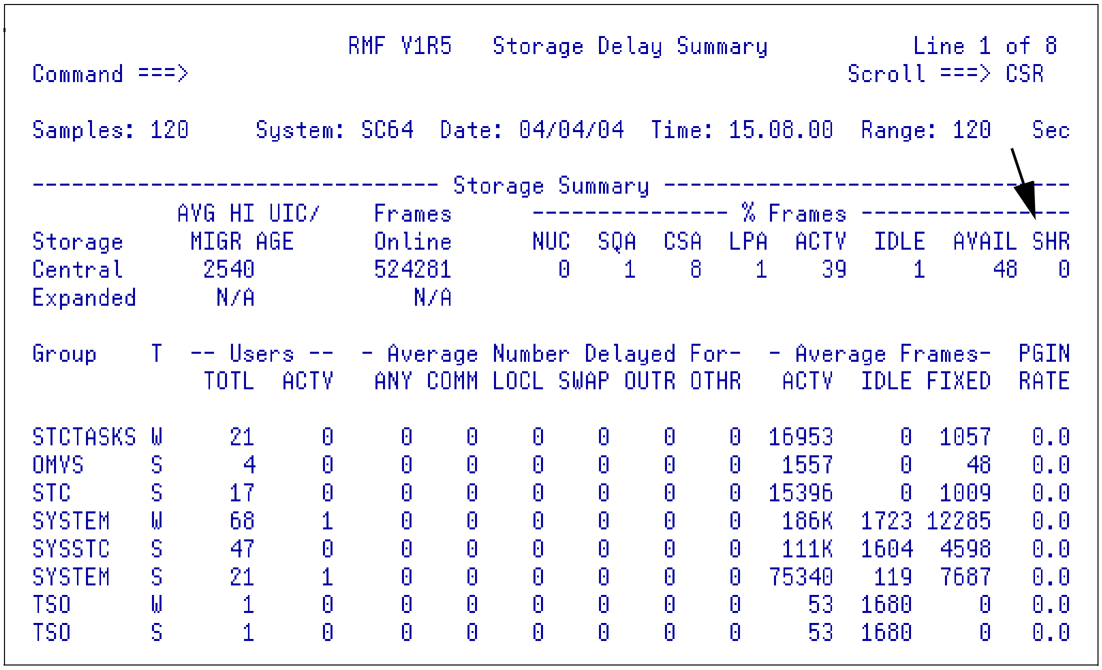


```
                        RMF V1R5   Storage Resource Delays        Line 1 of 7  .
Command ===>                                                 Scroll ===> CSR   .
                                                                               .
Samples: 120     System: SC64  Date: 04/04/04  Time: 15.08.00  Range: 120   Sec .
                                                                               .
------------------------------- Storage Summary ------------------------------- .
           AVG HI UIC/    Frames       --------------- % Frames --------------- .
Storage     MIGR AGE      Online       NUC  SQA  CSA  LPA  ACTY  IDLE  AVAIL SHR .
Central      2540         524281         0    1    8    1    39     1     48   0 .
Expanded      N/A            N/A                                                .
                                                                               .
---------------------------- Page/Swap Activity ------------------------------- .
Volume DEV      CU             ACT CON DSC PND Pend    SPACE  - AVG Active Users- .
Serial Type     Type       PAV  %   %   %   %  Reasons TYPE    TOTL LOCL SWAP COMM .
                                                                               .
SBOX01 33903    3990-3          1   0   0   1 DB     1 LOCL     0.0  0.0  0.0  0.0 .
SBOX55 33903    2105        4   0   0   0   0 None     LOCL     0.0  0.0  0.0  0.0 .
SBOXA1 33903    3990-6          0   0   0   0 None     LOCL     0.0  0.0  0.0  0.0 .
SBOX22 33903    3990-3          0   0   0   0 None     LOCL     0.0  0.0  0.0  0.0 .
SBOX99 33903    3990-3          0   0   0   0 None     COMM     0.0  0.0  0.0  0.0 .
                                                       PLPA     0.0  0.0  0.0  0.0 .
SBOXA0 33903    3990-3          0   0   0   0 None     LOCL     0.0  0.0  0.0  0.0 .
```

Figure 11-18 Monitor III Storage Resource Delay report

Monitor III Storage Delay Summary report

The RMF Monitor III Storage Delay Summary report has been updated to display the percentage of frames shared, as shown in Figure 11-19 on page 273.



```
                          RMF V1R5   Storage Delay Summary           Line 1 of 8
Command ===>                                                    Scroll ===> CSR

Samples: 120     System: SC64  Date: 04/04/04  Time: 15.08.00  Range: 120   Sec

------------------------------- Storage Summary -------------------------------
           AVG HI UIC/    Frames      --------------- % Frames ---------------
Storage     MIGR AGE      Online      NUC  SQA  CSA  LPA  ACTV  IDLE  AVAIL SHR
Central      2540         524281        0    1    8    1    39     1     48   0
Expanded      N/A            N/A

Group     T  -- Users --  - Average Number Delayed For-  - Average Frames-  PGIN
              TOTL  ACTV   ANY COMM LOCL SWAP OUTR OTHR   ACTV  IDLE FIXED  RATE

STCTASKS W     21     0     0    0    0    0    0    0  16953     0  1057   0.0
OMVS     S      4     0     0    0    0    0    0    0   1557     0    48   0.0
STC      S     17     0     0    0    0    0    0    0  15396     0  1009   0.0
SYSTEM   W     68     1     0    0    0    0    0    0   186K  1723 12285   0.0
SYSSTC   S     47     0     0    0    0    0    0    0   111K  1604  4598   0.0
SYSTEM   S     21     1     0    0    0    0    0    0  75340   119  7687   0.0
TSO      W      1     0     0    0    0    0    0    0     53  1680     0   0.0
TSO      S      1     0     0    0    0    0    0    0     53  1680     0   0.0
```

Figure 11-19 Monitor III Storage Delay Summary report

11.5 RMF and msys for Setup

An RMF plug-in has been provided for msys for Setup with z/OS V1R5. The intent of the plug-in is to make use of the facilities of msys for Setup to ease the customization, migration, and maintenance tasks necessary to ready RMF for production and to maintain production operation.

11.5.1 RMF plug-in wizard

A wizard is provided with the RMF plug-in that guides the user through the following tasks:

- Customization of the system environment
- Specifying access definitions
- Setup for RMF control session including Monitor I
- Setup for RMFGAT, the Monitor III Gatherer session
- Setup for GPMSERVE, the Distributed Data Server
- Synchronization of SMF recording intervals
- Storing Gatherer options

Refer to chapter 2 of *z/OS V1R5.0 Resource Measurement Facility (RMF) User's Guide*, SC33-7990 for more information on customization tasks.

Infrastructure preparation for msys for Setup requires configuration of an LDAP server, definition of a DB2 database, and RACF security definitions as well as installation and configuration of the workstation component.

At the time of writing, this team's attempts to install and implement msys for Setup to test the RMF plug-in were unsuccessful due to problems with product installation and customization documentation and with the “Add Product Set” dialog. These problems have been reported to the product developers and should be addressed in the near future.

At the current time, the benefits intended for RMF users through the use of msys for Setup simply do not justify the manpower cost of implementing the enabling infrastructure. We recommend that it’s use be delayed for now and re-assessed in the future.

11.6 RMF Spreadsheet Reporter Java Technology Edition

The RMF Spreadsheet Reporter is a powerful workstation-based tool designed to provide graphical spreadsheet presentation of RMF Postprocessor data.

Performance data extracted from SMF records is the basis for z/OS performance analysis and capacity planning. The RMF Postprocessor tool extracts performance measurements from selected SMF records and produces Report Listings and Overview records. The design goal of the RMF Spreadsheet Reporter is to exploit the graphical presentation facilities inherent in a workstation for z/OS performance analysis and reporting. By converting RMF Postprocessor Report Listings and Overview records into spreadsheets, this program offers a complete solution of enhanced graphical presentation of RMF reports and significantly improves the capability of long-term reporting and trend analysis on RMF data. Support is included in the tool for interaction with Lotus® 1-2-3® and Microsoft Excel spreadsheet programs.

11.6.1 RMF Spreadsheet Reporter features

The features of the RMF Spreadsheet Reporter include:

- Ease of use: RMF Spreadsheet reporter resources are managed by way of a familiar Explorer-like GUI.
- Fast path to graphical presentation: SMF data is prepared for spreadsheet usage in one single step.
- Batch mode: Input files for the spreadsheets are generated without any GUI interaction.

The RMF Spreadsheet Reporter, prior to version 5, was a platform-dependent C++ based implementation that had been functionally stabilized by IBM in December of 2000.

With the introduction of the RMF Spreadsheet Reporter Version 5, the GUI is now a platform-independent Java-based implementation that corrects several shortcomings of the previous versions of the product. Specifically:

- The products Collector task does not allow the specification of variable FTP parameters nor does it allow the user to receive individual datasets.
- The products GUI dialog must be used for extraction; batch extracts are no longer supported. Further, use of the conversion GUI will require the user to make explicit selections.It is no longer possible to convert all reports by default.
- Spreadsheet Reporter tasks are now isolated and must be executed serially.

11.6.2 Spreadsheet Reporter overview

The version 5 GUI employs a data object oriented concept. One common application window contains both Local and Remote data objects. The following data items are accessible from a Spreadsheet Reporter common window:

- Remote (host-based) resources
  - SMF Dump data
  - Report listings
  - Overview records
- Local (workstation based) resources
  - Report listings
  - Overview records
  - Working sets
  - Spreadsheets

Data transition tasks

Spreadsheet Reporter data transition tasks are now executed implicitly. These tasks include Remote RMF Postprocessor execution or Collection, Download, Extraction, Conversion, and Working set creation. This will allow the user to prepare spreadsheet data in a single step and have maximum flexibility in choosing simple or complex operations. Table 11-4 illustrates the data transitions that occur in RMF Spreadsheet Reporter processing and their associated tasks.

Table 11-4 RMF Spreadsheet Reporter data transitions and associated processing tasks

Data source	Target data	Processing task
SMF Dump Data Set	Remote Listing	RMF Postprocessor Execution
SMF Dump Data Set	Local Listing	RMF Postprocessor Execution Spreadsheet Reporter Transfer
SMF Dump Data Set	Working Set	RMF Postprocessor Execution Spreadsheet Reporter Transfer Spreadsheet Reporter Extraction Spreadsheet Reporter Conversion Spreadsheet Reporter Working Set Creation
Remote Listing	Local Listing	Spreadsheet Reporter Transfer
Remote Listing	Working Set	Spreadsheet Reporter Transfer Spreadsheet Reporter Extraction Spreadsheet Reporter Conversion Spreadsheet Reporter Working Set Creation
Local Listing	Working Set	Spreadsheet Reporter Extraction Spreadsheet Reporter Conversion Spreadsheet Reporter Working Set Creation

11.6.3 Download and install the tool

The RMF Spreadsheet Reporter Version 5.1 (March 01, 2004) is available for download from the RMF tools Web site. The executable download file rmfsrv5.exe can be found on the Web at:

```
http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/rmftools.htm#spr_win
```

Download and save the rmfsrv5.exe installation file to the workstation. It is roughly 41MB in size, so the download may take a while to complete. Make note of the folder on the workstation where the file will be downloaded to.

To begin the installation, which is accomplished through a standard Installshield wizard, locate the downloaded rmfsrv5.exe file and double-click on its icon.

Begin the Installshield dialog by clicking the **Next** button. You must then select what type of install is to be done. Selection of a “Typical” install will install all of the Spreadsheet Reporter application data, as well as spreadsheet support for Microsoft Excel. The total space requirement is approximately 75MB.

Lotus 1-2-3 support

If Lotus 1-2-3 support is required choose the “Custom” install option. This allows the specification of which components of the product to install. Selection of the application files and Lotus 1-2-3 support will require approximately 70MB. If support for both spreadsheet products is required the total space requirement is 83MB.

If available hard drive capacity is very limited, select the “Compact” install option to install the minimum required components.

The standard directories for the install are:

- For the product itself:

  C:\Program Files\RMF\RMF Spreadsheet Reporter

- For listings, macros and working sets.

  C:\Documents and Settings\user\Application Data\RMF\RMF Spreadsheet Reporter

If these are unacceptable, click **Browse** to change the directory specification and if necessary build the new directory. When all of the installation specifications have been made, click **Install**.

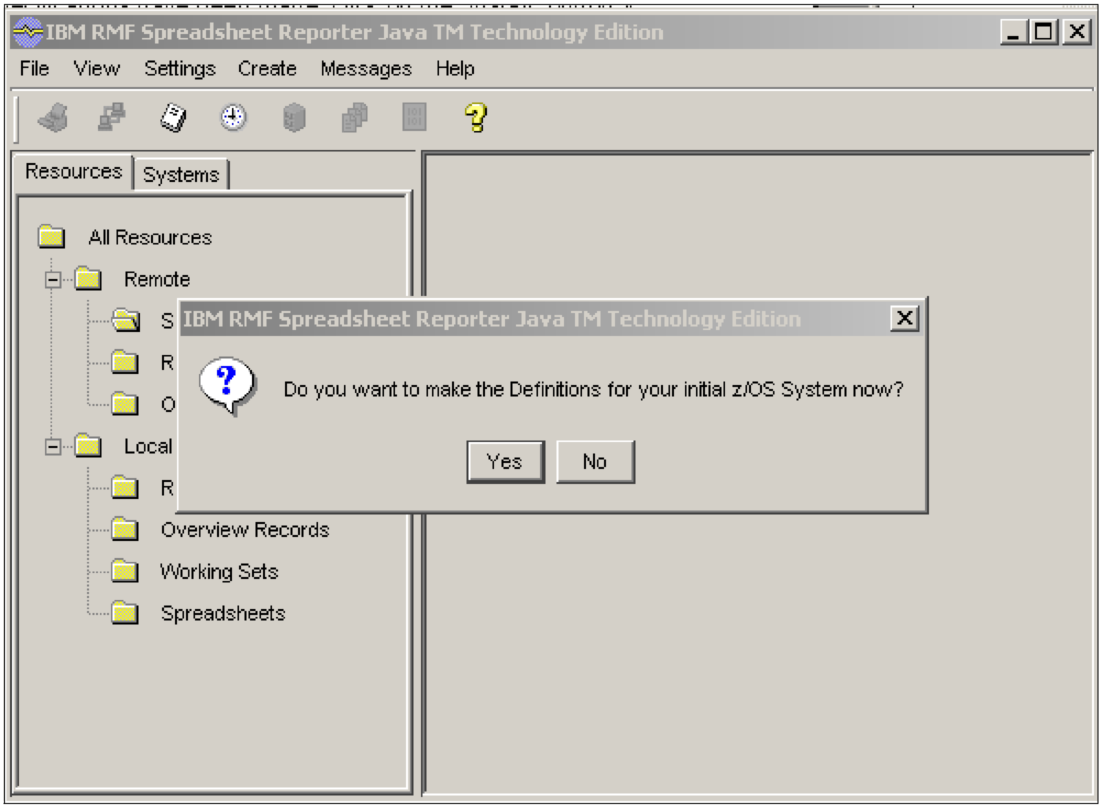
The Installshield then displays a status window depicting the progress of the install and informing you when the installation has completed successfully. Click **Finish** to complete the installation process.

The installation process does not place new icons on the workstation desktop. The tool can be located by clicking the Windows Start button and selecting the Programs icon. Look for a new folder titled IBM RMF Performance Management and select it. This folder has an icon for RMF Spreadsheet Reporter. Simply double-click this icon to initiate the tool.

11.6.4 Initial product setup

When you double-click the RMF Spreadsheet reporter icon for the first time, you are presented with the window shown in Figure 11-20 on page 277, and prompted to define the z/OS system.

We recommend that the best course of action to take at this time is to click **No**, and to proceed with a thorough review of the documentation available either through the product’s “Help” facility, or in chapter 7 of *z/OS Resource Measurement Facility (RMF) User’s Guide*, SC33-7990. Developing an understanding of the product setup, administration, and operation is key to successful implementation of the tool. Also, the F1 key can be used at any time in the RMF Spreadsheet Reporter to access the Help files for the current screen.


IBM RMF Spreadsheet Reporter Java TM Technology Edition
File View Settings Create Messages Help
Resources Systems
All Resources
Remote
S
R
O
Local
R
Overview Records
Working Sets
Spreadsheets
IBM RMF Spreadsheet Reporter Java TM Technology Edition
Yes
No

Do you want to make the Definitions for your initial z/OS System now?

Figure 11-20 RMF Spreadsheet report main menu panel

To re-drive to the System definition dialog, you can simply terminate the tool and restart it. The main menu displayed in Figure 11-20 is presented again and system definition can proceed with the following step.

- Click **Yes** to define your z/OS environment. The Define New System window shown in Figure 11-21 on page 278 is displayed.

Define new System panel

The System parameters shown in Figure 11-21 have the following definitions:

System ID A system identifier of your choice.

Hostname The TCP/IP name of the host system, which can be a symbolic name that can be resolved via a name-server, or your /etc/hosts file, or an IP address.

> **Note:** If you do not know your system's hostname, you can retrieve this hostname and the system's TCP/IP address with the TSO command `hometest`.

Dataset HLQ The dataset high level qualifier. This is a required parameter for remote postprocessor job executions. During this process, several data sets must be allocated.

Userid Your TSO user ID for FTP logon. The default is the same value as the dataset HLQ and is automatically filled in. You can overtype it if your installation has special conventions for HLQs and user IDs.

Password Your password for FTP logon.

Account The Spreadsheet Reporter requires this parameter for Postprocessor JCL generation. This may be, for example, your department number.

Jobclass This required parameter specifies the jobclass to be used in Postprocessor JCL generation.

OVW This button opens a file dialog. You can browse for a file containing overview control statements that you can attach to the current system. The file name is then displayed in the entry field. By default there are two files displayed: one for WLM compatibility mode and one for WLM goal mode operation. Select the appropriate file. For information about the purpose of this file, read "How to use overview control statements" in the RMF Spreadsheet Reporter Help facility.

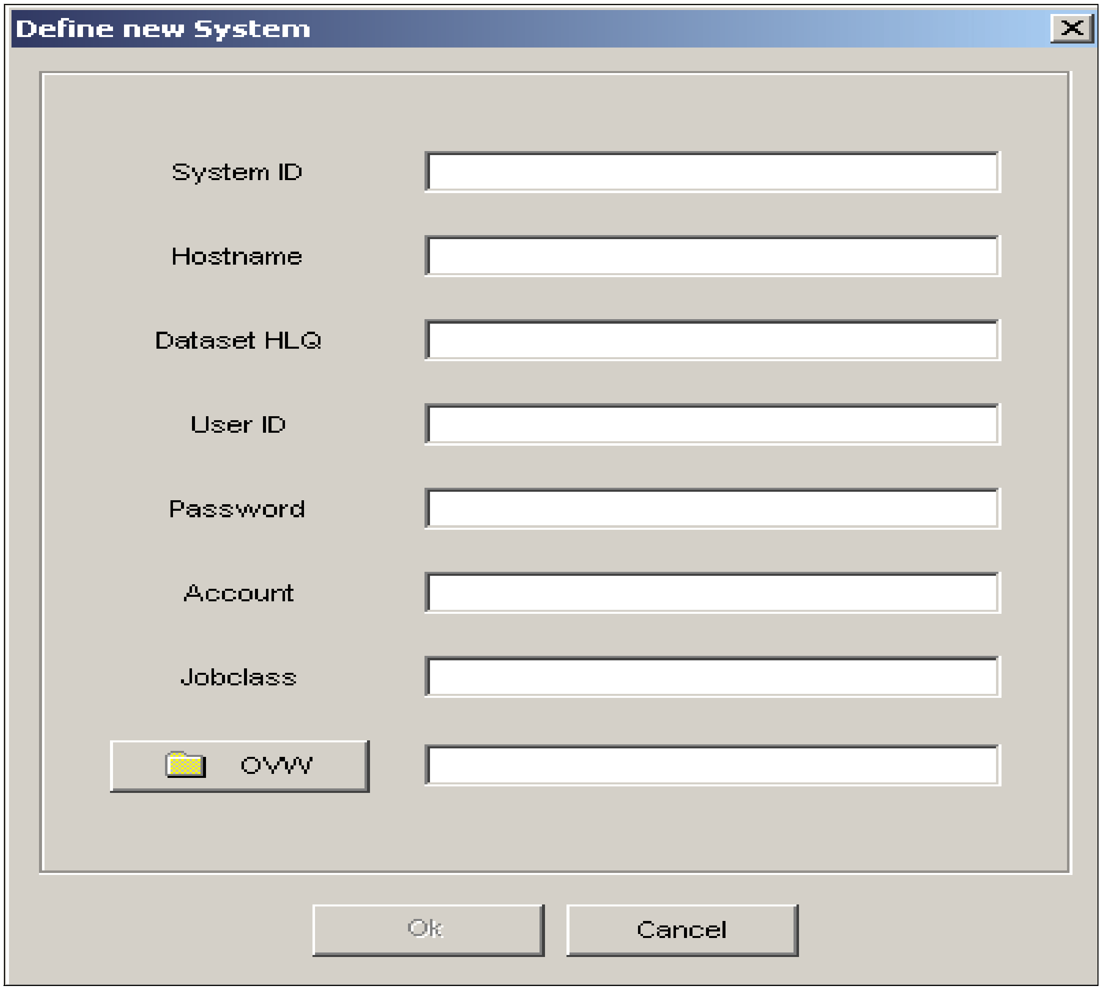


Figure 11-21 Define new System panel

Note: If you want to work with multiple sets of overview control statements for the same system, you can define multiple copies of the same system with a different System ID but the same Hostname. Thus you can work with fixed attachments instead of changing the system properties all the time.

Updating panel information

If a system or a parameter is entered incorrectly or needs to be changed at a later date, right-click the name of the system that needs to be changed. This will result in a popup window displaying options for Rename, Delete, and Properties, which can be used as follows:

Rename To supply a new name for the selected previously defined system.

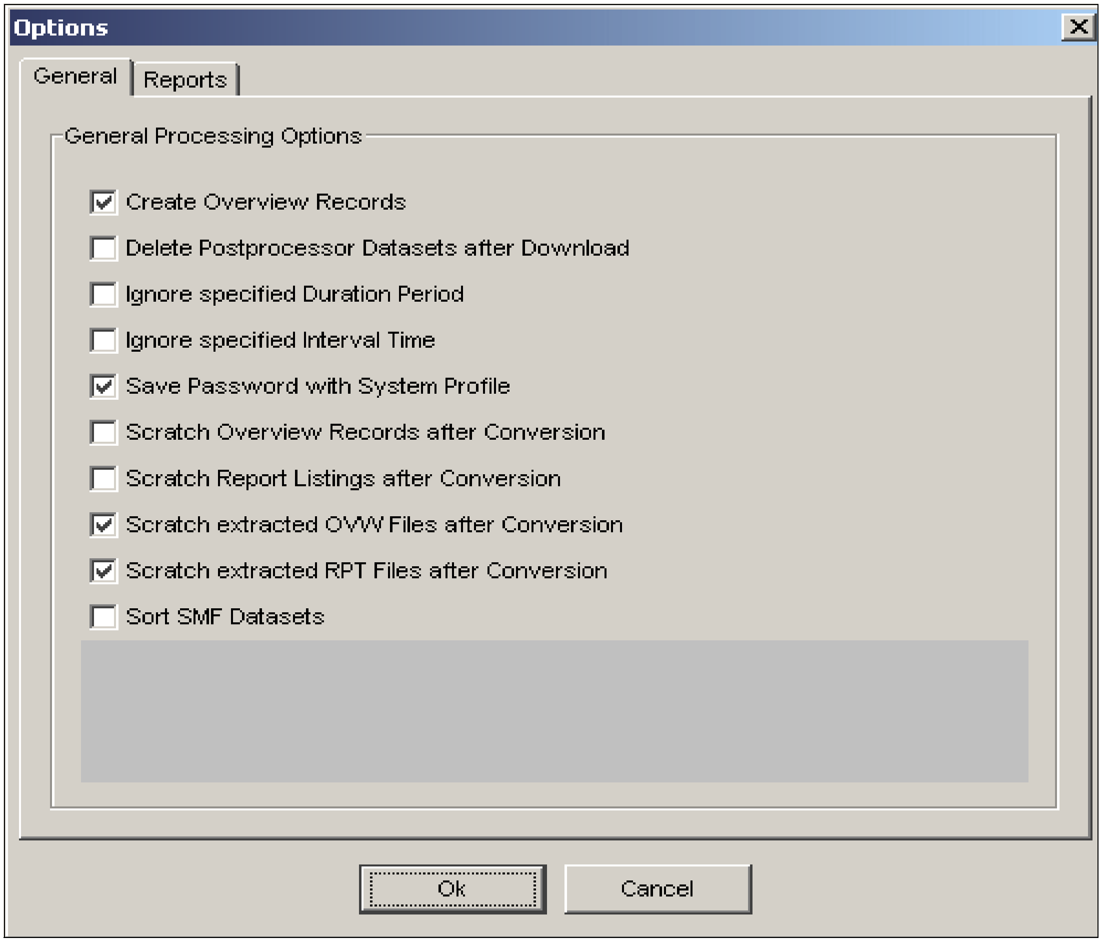
Delete To remove the selected previously defined system from the tool.

Properties To change a parameter specification for a previously defined system.

After you successfully add the first system, more systems can be added by right-clicking in the gray area of the view pane. A new window will be displayed which includes the selection New. Clicking **New** results in the Define New System panel being re-displayed. Define each system whose performance data is processed by the tool.

11.6.5 Specification of General processing options

After the desired systems have been defined to the Spreadsheet Reporter, the general processing option specifications should be reviewed and adjusted as required. From the menu bar on the main menu, shown in Figure 11-20, select **Settings** → **Options**. The General Options screen shown in Figure 11-22 is displayed under the General tab. Select the desired options on this screen.



General Processing Options

☑ Create Overview Records

□ Delete Postprocessor Datasets after Download

□ Ignore specified Duration Period

□ Ignore specified Interval Time

☑ Save Password with System Profile

□ Scratch Overview Records after Conversion

□ Scratch Report Listings after Conversion

☑ Scratch extracted OVW Files after Conversion

☑ Scratch extracted RPT Files after Conversion

□ Sort SMF Datasets

Figure 11-22 Options panel displaying the General options

General options

The options available, their descriptions and usage are shown in Table 11-5, and in the product's Help facility. Simply select each option to be activated by clicking its associated box; click **OK** when all required options have been selected.

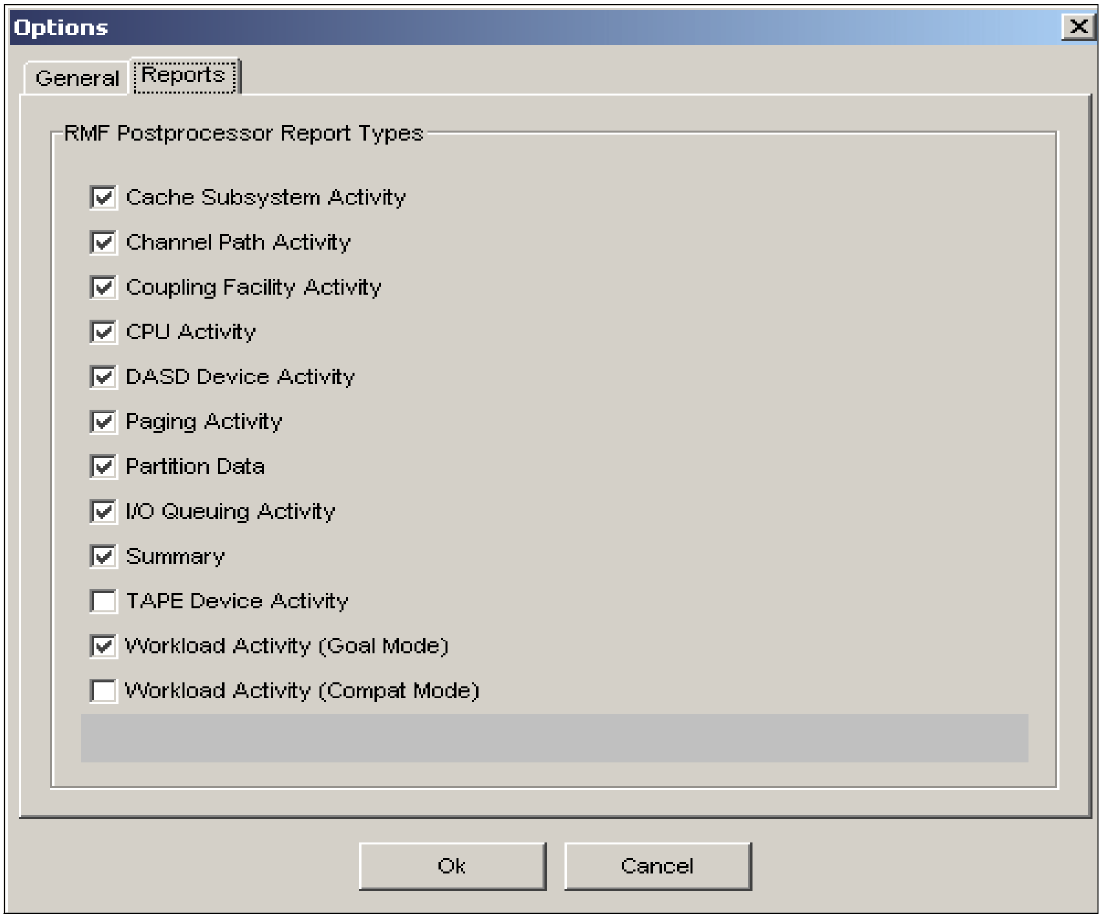
Table 11-5 RMF Spreadsheet Reporter options

General processing option	Description
Create Overview Records	Select this option to create Overview Records using the overview control statements contained in a file that is attached to the current system. If a file with overview control statement is attached to the current system, but this option is not selected, then a readable Postprocessor Overview report is generated according to the overview control statements contained in this file. However, you cannot process this report with spreadsheets. If no file containing overview control statements is attached, then this option is ignored and the Working Set is generated with the selected RMF Postprocessor report types.
Delete Postprocessor Datasets after Download	Remote Report Listings or Overview Records on the host are deleted after a successful download to the workstation.
Ignore specified Duration Period	No DINTV control statement is generated from the interval options so that no duration reports will be created. (See 11.6.7, "Specify report intervals and duration periods" on page 281 for more information.)
Ignore specified Interval Time	No RTOD control statement is generated from the interval options. The default from 00:00 to 24:00 is used as date interval.
Save Password with System Profile	The password that you specified for a system in dialogs Define new System or System Properties is saved but not encrypted. Otherwise you are prompted for the password for all actions that require a host logon.
Scratch Overview Records after Conversion	The local Overview Records (*.rec files) are deleted after Working Set generation.
Scratch Report Listings after Conversion	The local Report Listings (*.lis files) are deleted after Working Set generation.
Scratch extracted OVW Files after Conversion	When generating a Working Set from Overview Records, the Spreadsheet Reporter deletes the local OVW files after Working Set generation.
Scratch extracted RPT Files after Conversion	When generating a Working Set from Report Listings, the Spreadsheet Reporter deletes the local RPT files after Working Set generation.
Sort SMF Datasets	You can specify whether the SMF data should be sorted. To ensure correct reports, the records in an SMF dataset must be sorted by interval start date and interval start time.

11.6.6 Specification of Report Options

From the main menu, select **Settings** → **Options** → **Reports** to display the Report option screen shown in Figure 11-23 on page 281, where you can select Postprocessor report types supported by the Spreadsheet Reporter. A REPORTS or SYSRPTS control statement is generated for each selected report type and at least one report type must be selected.

Select each report to be activated by clicking on its associated box; click **OK** when all required reports have been selected.



RMF Postprocessor Report Types

☑ Cache Subsystem Activity
☑ Channel Path Activity
☑ Coupling Facility Activity
☑ CPU Activity
☑ DASD Device Activity
☑ Paging Activity
☑ Partition Data
☑ I/O Queuing Activity
☑ Summary
□ TAPE Device Activity
☑ Workload Activity (Goal Mode)
□ Workload Activity (Compat Mode)

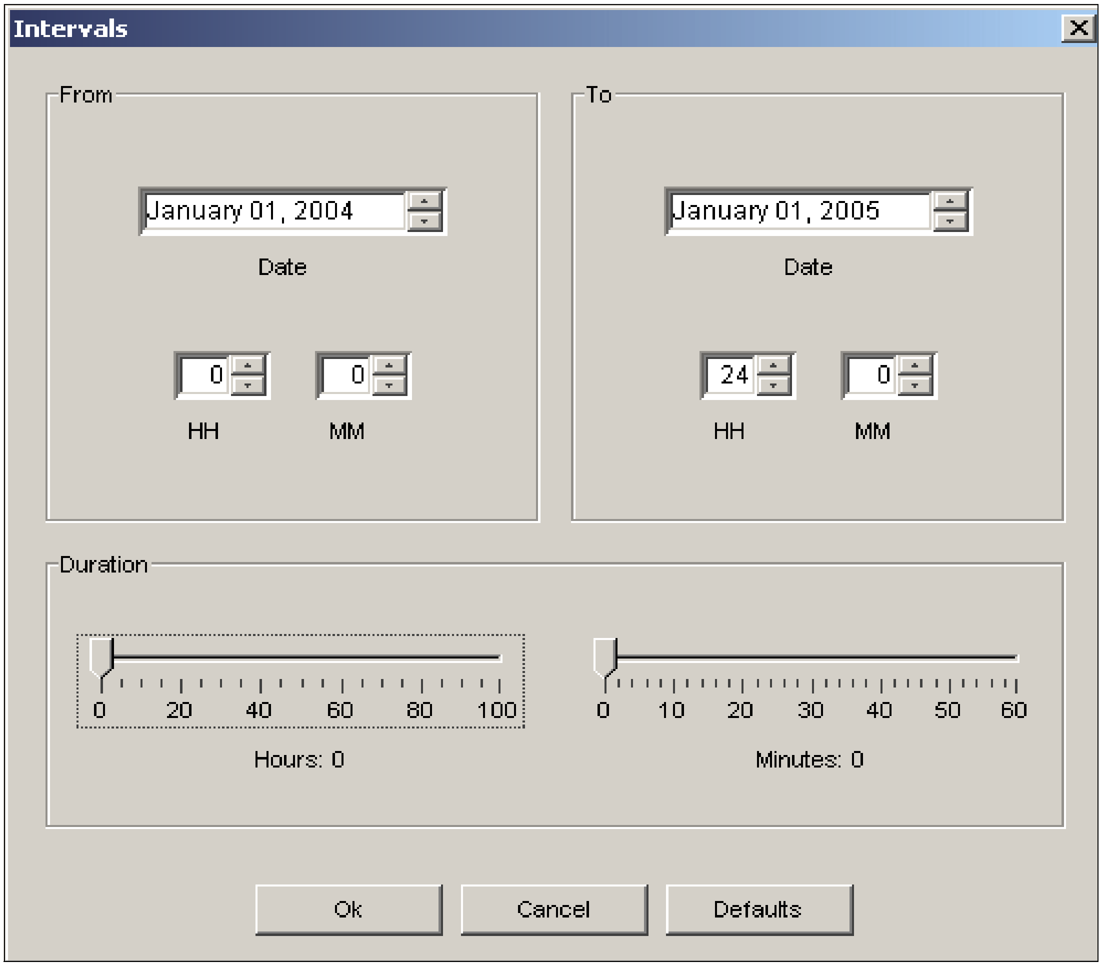
Figure 11-23 Reports options specifications

11.6.7 Specify report intervals and duration periods

To specify the start and end time for data collection select **Settings** → **Intervals** from the main menu, shown in Figure 11-20, to display the Intervals pane, shown in Figure 11-24 on page 282. The specifications made in this dialog are converted to corresponding postprocessor DATE and RTOD control statements. The Duration sliders can also be used to generate a DINTV control statement to produce duration reports.

Specify the beginning and ending dates and times required, as well as the length of time to be used in duration reports and click **OK** to save your specifications.

For more information about DATE, RTOD, and DINTV refer to *z/OS Resource Measurement Facility (RMF) User's Guide*, SC33-7990.



From: Date January 01, 2004, HH 0, MM 0

To: Date January 01, 2005, HH 24, MM 0

Hours: 0

Minutes: 0

Figure 11-24 Intervals panel

11.6.8 Usage scenario

A typical usage scenario for the Spreadsheet Reporter is to run an RMF Postprocessor job on the z/OS host, download the resulting Postprocessor data set (Report Listing or Overview Records) to the workstation, and convert it into a “Working Set.” The working set would then contain the performance measurement data provided by the report.

Generation of a working set is accomplished with a single action without the user needing to take care of the involved data preparation tasks. A spreadsheet macro is selected for graphical performance analysis. This macro is then fed the selected working set. This illustrates the fast path available with the Spreadsheet Reporter for graphical presentation of RMF performance data.

Most of the transitions between the resource types can be performed in all variations. This allows the Spreadsheet Reporter to be used as a remote RMF Postprocessor execution and download utility. As an example, postprocessor job execution could be started from a workstation and the resulting Report Listing output could be stored on the host without downloading. The user could then download these RMF Postprocessor datasets at a later time using the **File** and **Transfer** selections from the main menu.

11.6.9 RMF Spreadsheet Reporter JCL requirements

It may be necessary to adapt the standard JCL generated by the spreadsheet reporter to successfully execute the RMF Postprocessor in the user's environment.

While the System Properties panel may be used to provide several basic site-dependent JCL parameters, it does not cover all possible JCL changes that may be required.

The JCL skeleton that is used to generate the Postprocessor JCL can be found in the installation directory at:

C:\Program Files\RMF\RMF Spreadsheet Reporter\Connect\RMFPP1.JCL

We recommend making a backup copy of the RMFPP1.JCL file prior to making any changes that may be required.

Several changes will need to be considered immediately:

- The JCL will be generated using a JOBNAME that is created by appending a "$" to the end of the USERID specified in the System Properties panel. If this is inconsistent with the installation's enforced jobname standards, change the JOBNAME specification <USER>$ in the JCL skeleton as appropriate.
- The MSGCLASS specified on the JOB card defaults to class "H." If this is not the installation's defined held output class two items may need to be changed. The first is the MSGCLASS specification itself. In addition, the RMFPP step in the Postprocessor jobstream also contains the SYSIN statement SYSOUT(H), which can be changed as required.
- The tool will submit a batch job that builds work datasets that will be allocated using the USERID specified in the System Properties panel as the high level qualifier. The JCL assumes, and we highly recommend, that these be SMS-managed datasets. If they are not SMS-managed, the REFDD= *.MSG parameters in the ALLOC step will have to be deleted and an additional JCL statement added to each of these DD statements that specifies:

```
// UNIT=SYSDA,SPACE=(TRK,(5,5)),DCB=(LRECL=137,RECFM=VBA,BLKSIZE=1693)
```

11.6.10 Creating a working set

The following series of panels are included to provide familiarity with common Spreadsheet Reporter activities.

Using SMF data

The first series of panels is used to create a working set from SMF records on the host. To accomplish this:

1. Select the Systems tab, shown in Figure 11-20 on page 277. The screen shown in Figure 11-25 on page 284 is displayed.

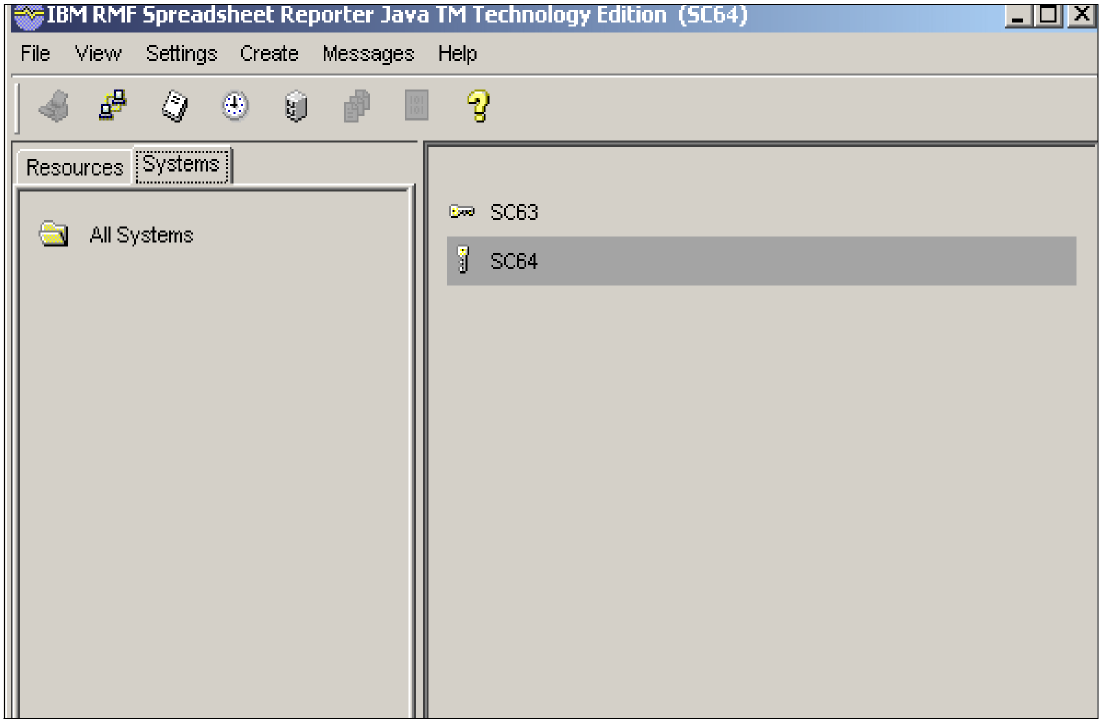


Figure 11-25 Systems panel from main menu

2. Right-click the name of the system you want to process; the screen shown in Figure 11-26 is displayed.

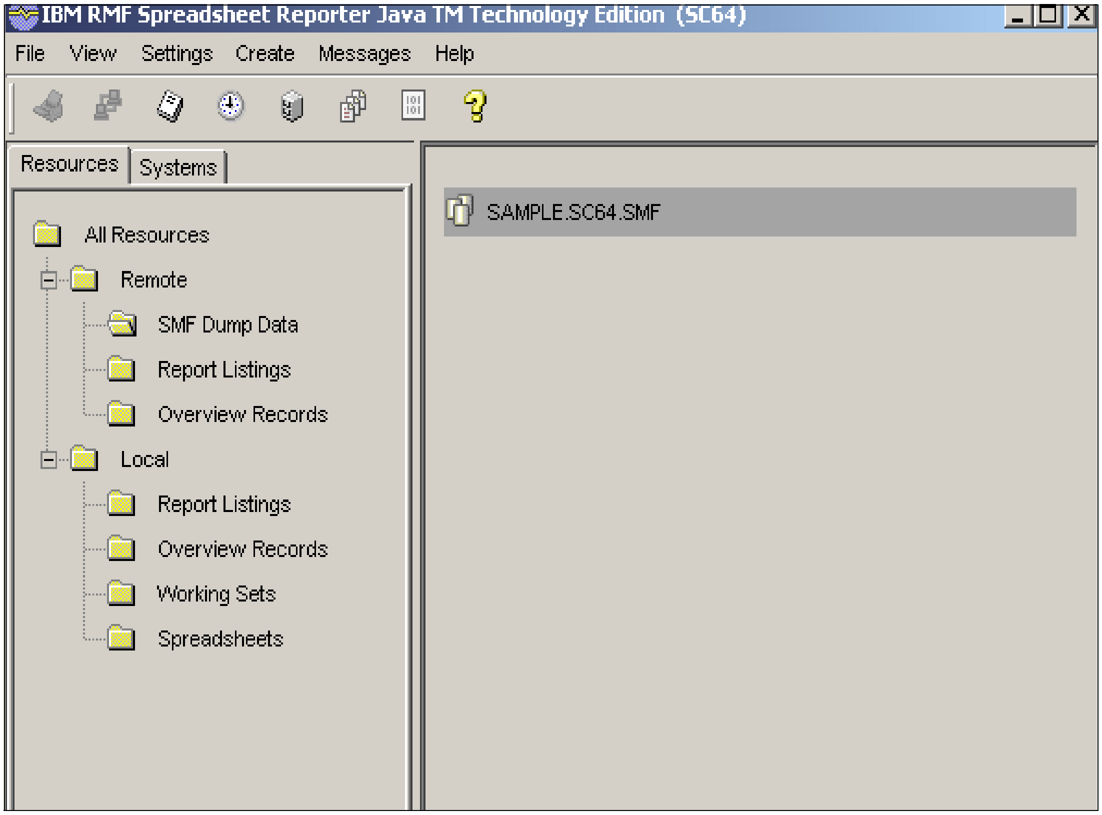
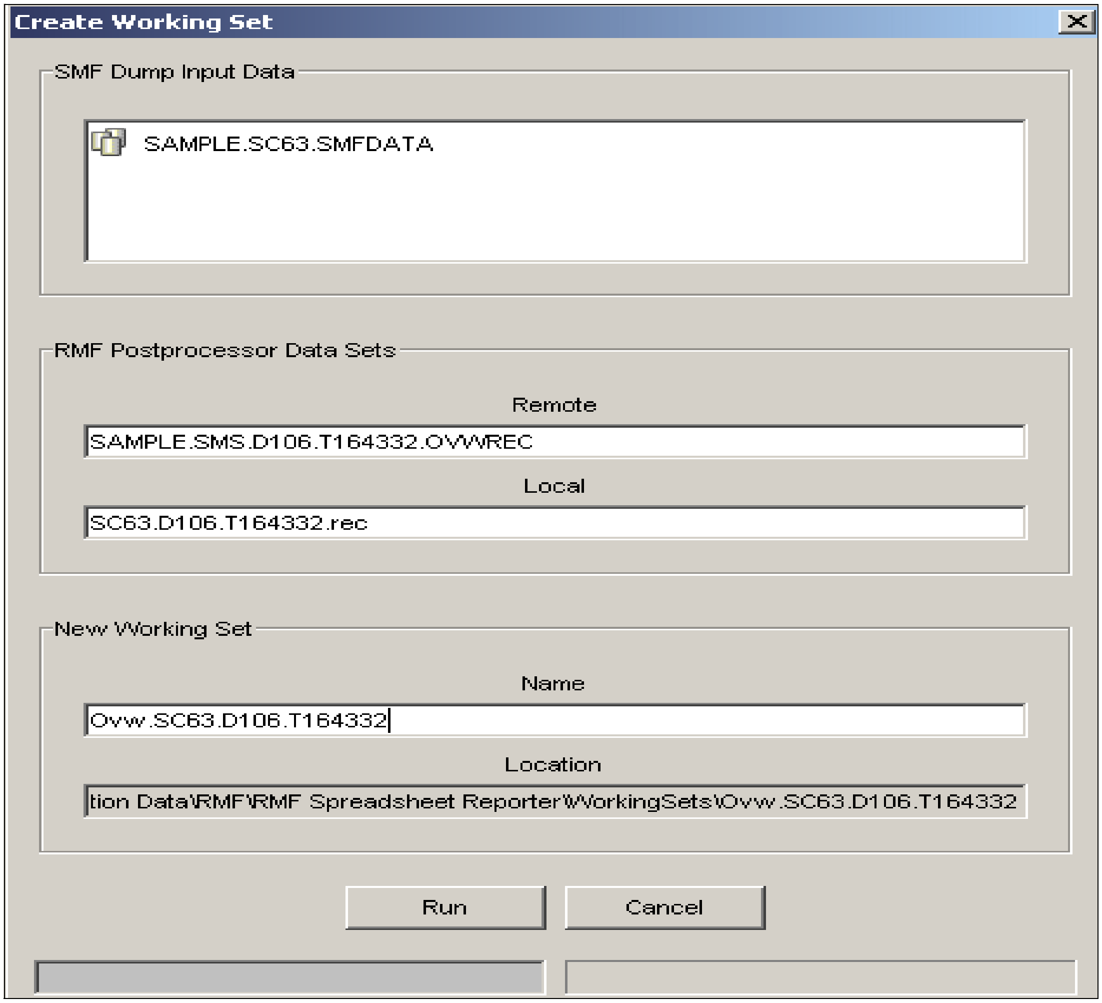


Figure 11-26 System panel display showing the SMF data sets

3. Click the Resource tab, then click the Remote SMF Dump Data folder shown in Figure 11-26. Click the SMF dump data set you want to process.
4. Select the **Create** menu bar option and click **Working Set**; the screen shown in Figure 11-27 is displayed. The Spreadsheet Reporter automatically provides a data set name for all data sets to be created.
5. Click **Run** to start the process. A job is submitted to the host system and the output from the job is returned to the workstation when the job has completed.



SAMPLE.SC63.SMFDATA

Remote: SAMPLE.SMS.D106.T164332.OVWREC

Local: SC63.D106.T164332.rec

Name: Ovw.SC63.D106.T164332

Location: tion Data\RMF\RMF Spreadsheet Reporter\WorkingSets\Ovw.SC63.D106.T164332

Figure 11-27 Create Working Set panel

Producing the other reports and data sets

Use the same procedure to produce reports for the other resources shown in Figure 11-26, such as the following:

- Remote Report Listings
- Remote Overview Records
- Local Report Listings
- Local Overview records
- Local Working Sets
- Local Spreadsheets

11.6.11 Transfer process

There are times that it may be necessary to manually transfer remote Report Listing files or Remote Overview Records datasets to the workstation. The following series of screens illustrates the process to do so using a Report Listing data set as an example.

1. Select the **Systems** tab and right-click the system you want to process, as shown in Figure 11-28.

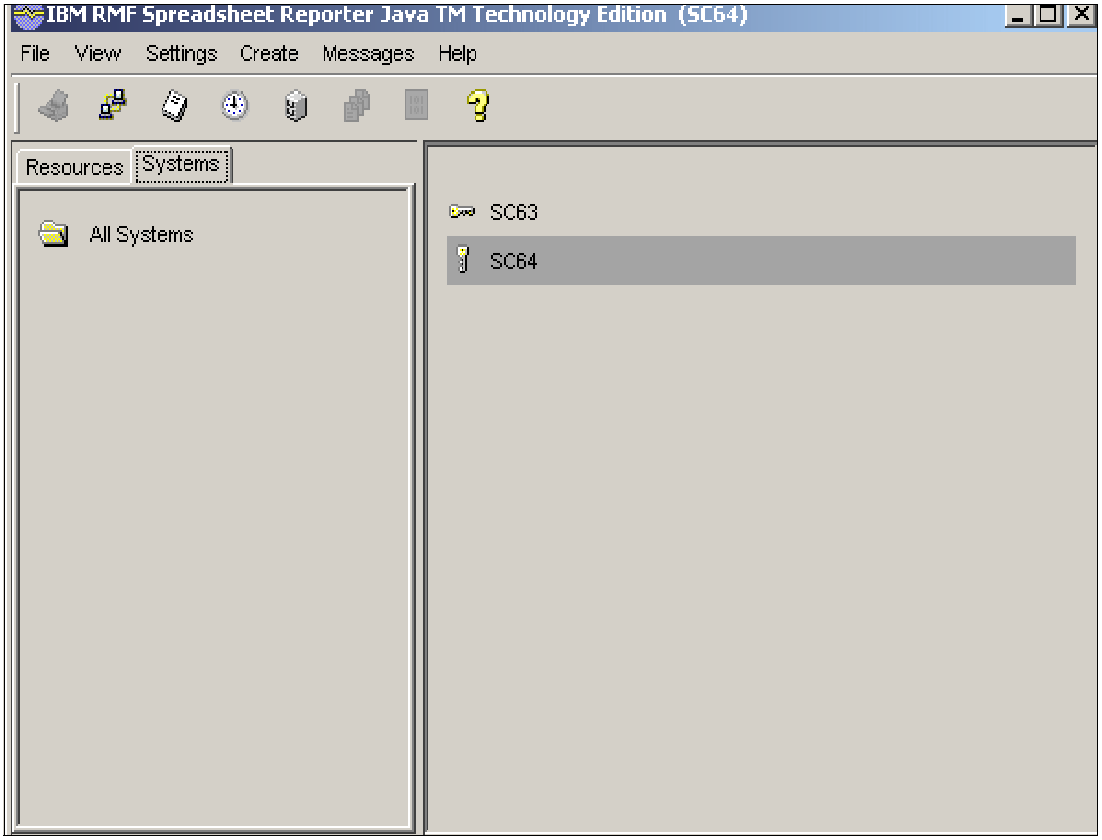


Figure 11-28 Systems panel from main menu

2. Click the **Resources** tab, then click the Remote Report Listing folder. Click the Report Listing data set you want to process, and select **File** from the main menu, as shown in Figure 11-29 on page 287.

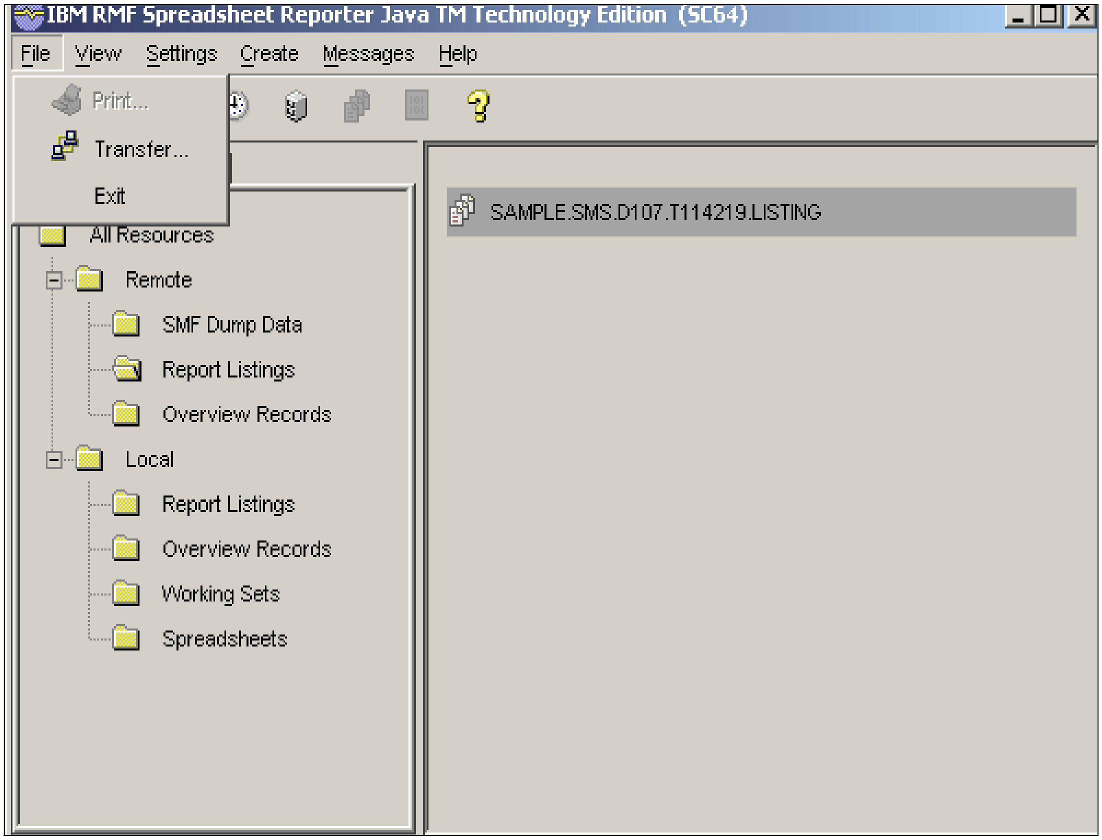
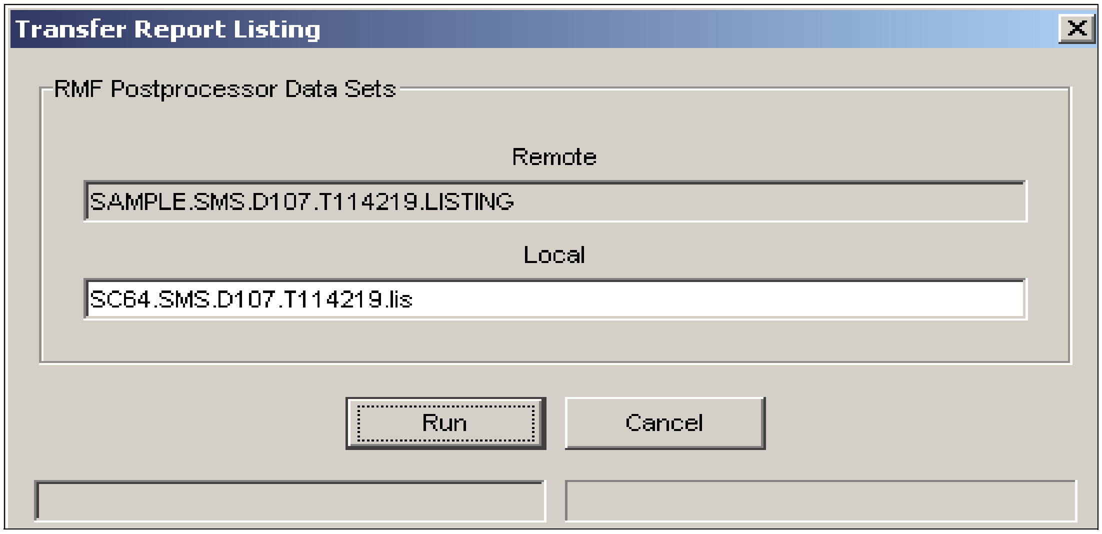


Figure 11-29 Selecting the Report Listing data set you want to process

3. Select the **Transfer** option. the screen shown in Figure 11-30 is displayed. The Spreadsheet Reporter automatically provides a data set name for the local copy.



Remote: SAMPLE.SMS.D107.T114219.LISTING

Local: SC64.SMS.D107.T114219.lis

Figure 11-30 Transfer Report Listing panel

4. Click **Run** to start the process. A message area at the bottom of the screen will be updated to indicate the success of the process.

11.6.12 Batch mode processing

All RMF Spreadsheet Reporter actions also can be performed in batch mode. This includes the following actions:

- Generation of JCL from a skeleton file
- Transmit JCL to a remote system
- Execute the RMF Postprocessor
- Retrieve the JES joblog for a specific job
- Retrieve RMF Postprocessor messages from a specific job
- Transfer a Report Listing from the Host to the Workstation
- Migrate Working Sets from a previous version of the Spreadsheet Reporter
- Migrate Overview Records from a previous version of the Spreadsheet Reporter

Table 11-6 identifies the batch mode files and their functionality.

Table 11-6 Batch mode files and their functionality

Batch File Name	Function
Jclgen.bat	Generates execution JCL from skeleton file rmfpp1.jcl
Collect.bat	Generates execution JCL from skeleton file rmfpp1.jcl Transmits JCL to remote system Executes the RMF Postprocessor Retrieves the JES joblog Retrieves RMF Postprocessor messages Transfer Report Listing from the host to the workstation
CreateRptWSet.bat	Generates the Working Set for Report Listings
CreateOvwWset.bat	Generates the Working Set for Overview Records
MigrateReportWorkSet.bat	Migrates Reports from previous version of Spreadsheet Reporter
MigrateOvwWorkSet.bat	Migrates Overview Records from previous version of Spreadsheet Reporter

Batch mode files

The batch mode files are stored in the Spreadsheet Reporter directory at:

C:\Program Files\RMF\RMF

The Jclgen.bat and Collect.bat procedures need to be edited prior to use to supply the correct variables for each installation. When executed, these procedures are used to modify the skeleton JCL stored in:

C:\Program Files\RMF\RMF Spreadsheet Reporter\Work\rmfpp1.jcl

They use the variables supplied by the user, and generate execution JCL to:

C:\Program Files\RMF\RMF Spreadsheet Reporter\Work\rmfpp2.jcl.

The user is required to supply the directory name of the input and the desired name for the output files as execution options for the CreateRptWSet.bat and CreateOvwWset.bat procedures.

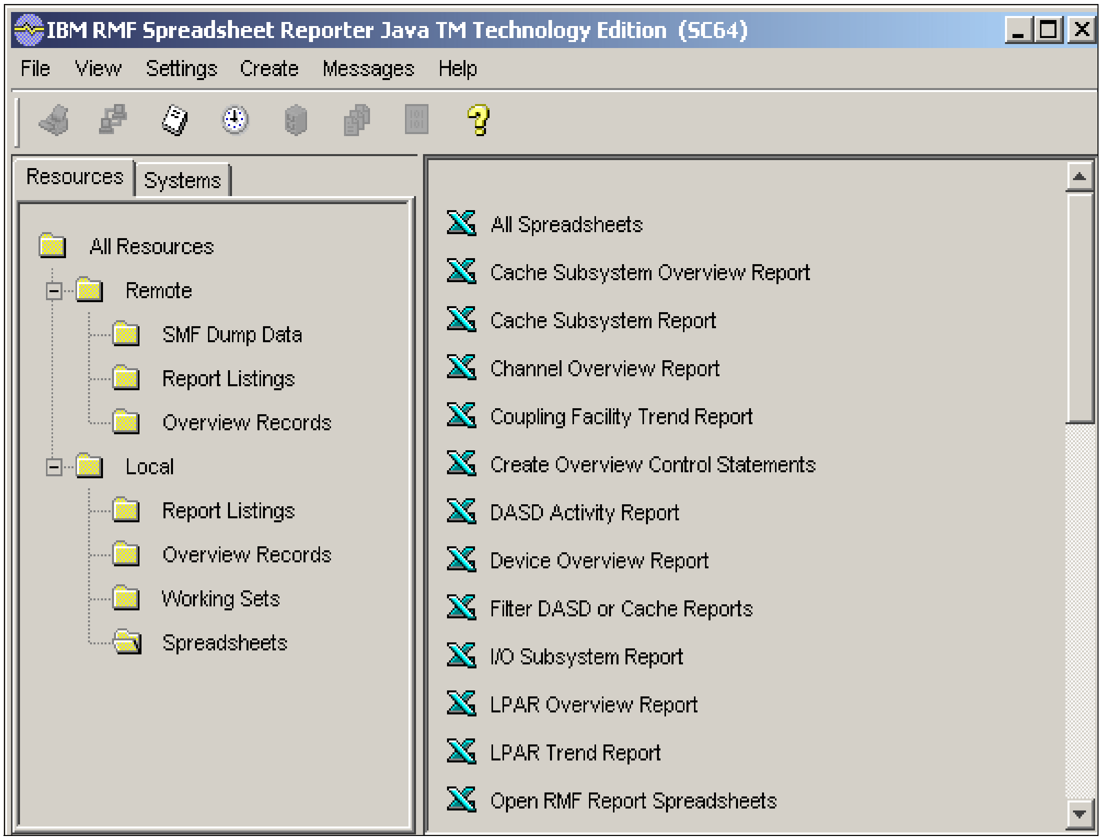
Likewise, the user is required to supply the directory names of the input and output files as execution options for the following procedures:

MigrateReportWorkSet.bat and MigrateOvwWorkSet.bat

Refer to *z/OS Resource Measurement Facility (RMF) User's Guide*, SC33-7990 for detailed information on the use of these procedures.

11.6.13 Spreadsheet Reporting

Upon successful creation of the required local Working Set and Overview Records data sets, the spreadsheet component of the RMF Spreadsheet Reporter can be accessed for productive use. To access the spreadsheets, click the Local Spreadsheets folder in the left pane; the right-hand pane is changed as shown in Figure 11-31.



IBM RMF Spreadsheet Reporter Java TM Technology Edition (SC64)
File
View
Settings
Create
Messages
Help
Resources
Systems

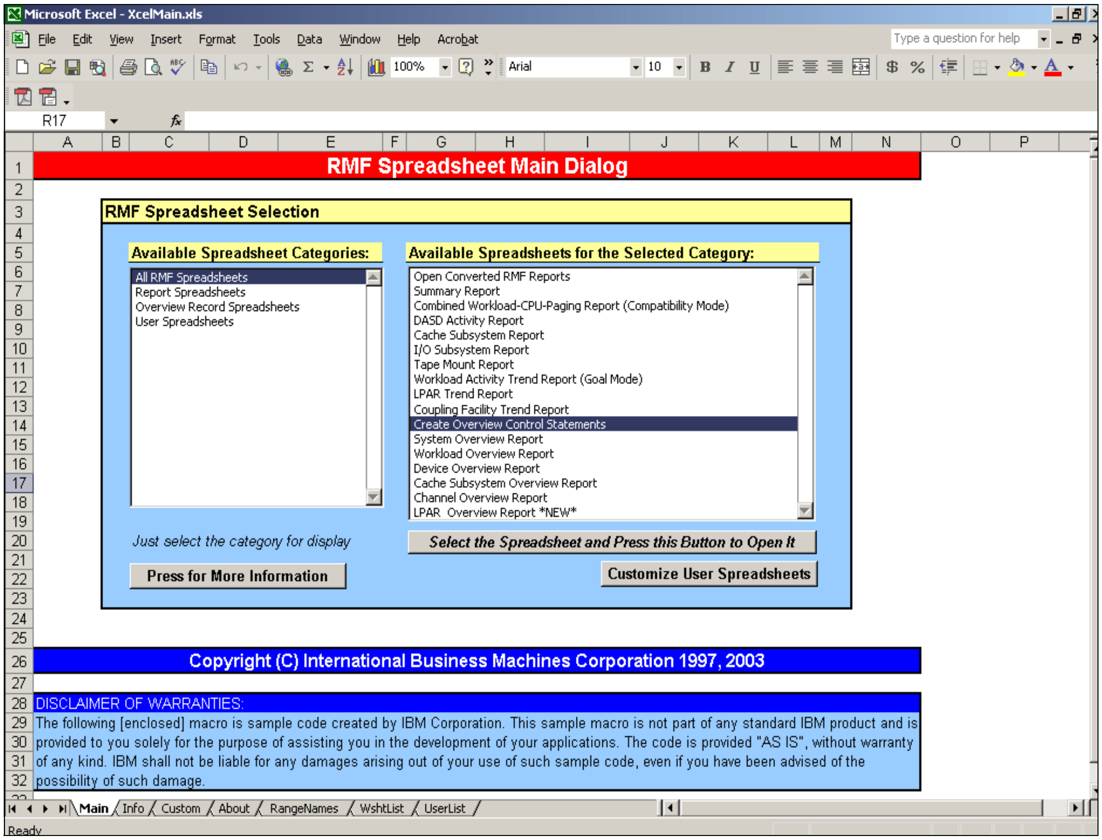
- All Resources
  - Remote
    - SMF Dump Data
    - Report Listings
    - Overview Records
  - Local
    - Report Listings
    - Overview Records
    - Working Sets
    - Spreadsheets

- All Spreadsheets
- Cache Subsystem Overview Report
- Cache Subsystem Report
- Channel Overview Report
- Coupling Facility Trend Report
- Create Overview Control Statements
- DASD Activity Report
- Device Overview Report
- Filter DASD or Cache Reports
- I/O Subsystem Report
- LPAR Overview Report
- LPAR Trend Report
- Open RMF Report Spreadsheets

Figure 11-31 Panel displaying the Spreadsheet folder

To access a spreadsheet, simply double-click its name in the list provided on the screen.

A good starting point to begin exploring the tool is with the All Spreadsheets selection at the top. Double-click that entry to view the screen shown in Figure 11-31 on page 289, from which any of the available spreadsheets can be selected.



Microsoft Excel - XcelMain.xls
File Edit View Insert Format Tools Data Window Help Acrobat
Type a question for help
100%
Arial
10
R17
RMF Spreadsheet Main Dialog
RMF Spreadsheet Selection
Available Spreadsheet Categories:
Available Spreadsheets for the Selected Category:
Just select the category for display
Select the Spreadsheet and Press this Button to Open It
Press for More Information
Customize User Spreadsheets
Main
Info
Custom
About
RangeNames
WshtList
UserList
Ready

All RMF Spreadsheets
Report Spreadsheets
Overview Record Spreadsheets
User Spreadsheets

Open Converted RMF Reports
Summary Report
Combined Workload-CPU-Paging Report (Compatibility Mode)
DASD Activity Report
Cache Subsystem Report
I/O Subsystem Report
Tape Mount Report
Workload Activity Trend Report (Goal Mode)
LPAR Trend Report
Coupling Facility Trend Report
Create Overview Control Statements
System Overview Report
Workload Overview Report
Device Overview Report
Cache Subsystem Overview Report
Channel Overview Report
LPAR Overview Report *NEW*

Copyright (C) International Business Machines Corporation 1997, 2003

DISCLAIMER OF WARRANTIES:
The following [enclosed] macro is sample code created by IBM Corporation. This sample macro is not part of any standard IBM product and is provided to you solely for the purpose of assisting you in the development of your applications. The code is provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of such sample code, even if you have been advised of the possibility of such damage.

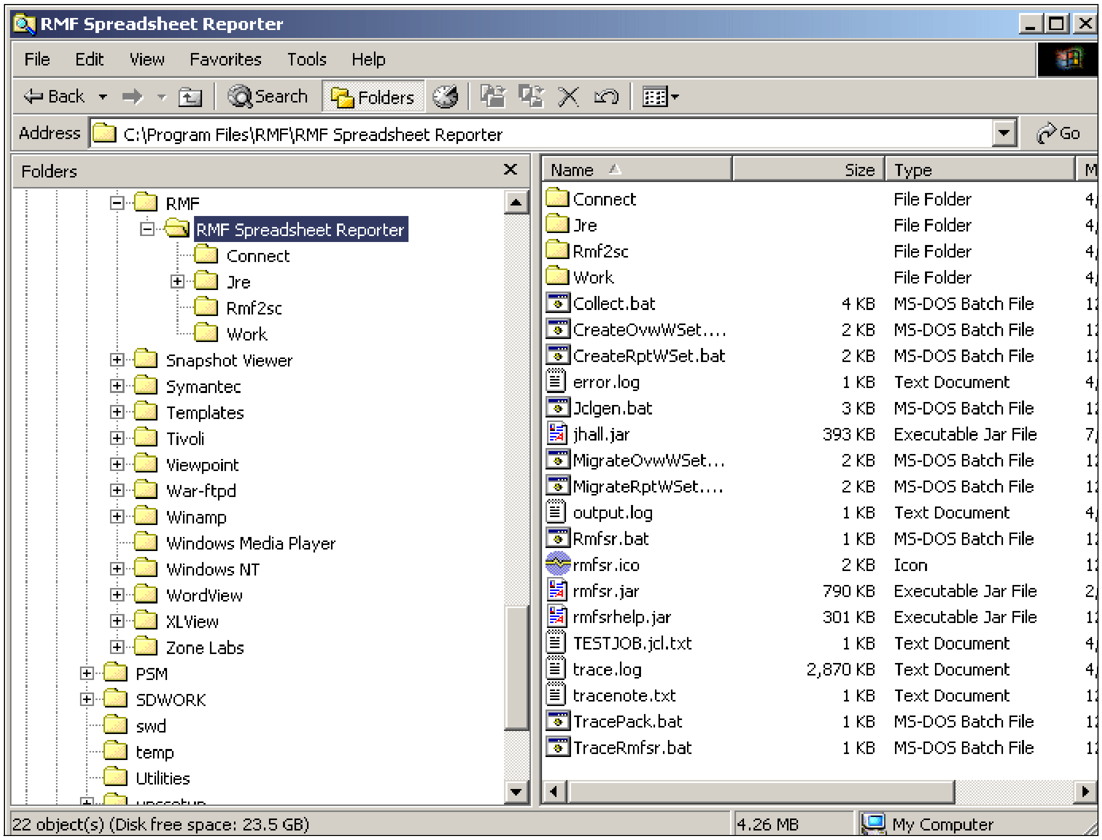
Figure 11-32 All Spreadsheets selection panel

11.6.14 Spreadsheet Reporter installation directories

RMF Spreadsheet Reporter program files can be found at:

C:\Program Files\RMF\RMF Spreadsheet Reporter

As shown in Figure 11-33 on page 291, these directories contain the command procedures, remote job execution, Java runtime, and spreadsheet converter program files, and all required work files.



Address C:\Program Files\RMF\RMF Spreadsheet Reporter

Name	Size	Type
Connect		File Folder
Jre		File Folder
Rmf2sc		File Folder
Work		File Folder
Collect.bat	4 KB	MS-DOS Batch File
CreateOvwWSet....	2 KB	MS-DOS Batch File
CreateRptWSet.bat	2 KB	MS-DOS Batch File
error.log	1 KB	Text Document
Jclgen.bat	3 KB	MS-DOS Batch File
jhall.jar	393 KB	Executable Jar File
MigrateOvwWSet...	2 KB	MS-DOS Batch File
MigrateRptWSet....	2 KB	MS-DOS Batch File
output.log	1 KB	Text Document
Rmfsr.bat	1 KB	MS-DOS Batch File
rmfsr.ico	2 KB	Icon
rmfsr.jar	790 KB	Executable Jar File
rmfsrhelp.jar	301 KB	Executable Jar File
TESTJOB.jcl.txt	1 KB	Text Document
trace.log	2,870 KB	Text Document
tracenote.txt	1 KB	Text Document
TracePack.bat	1 KB	MS-DOS Batch File
TraceRmfsr.bat	1 KB	MS-DOS Batch File

Figure 11-33 RMF Spreadsheet Reporter files display

Application data files

Application data files used by the RMF Spreadsheet Reporter can be found at:

C:\Documents and Settings\youruserid\Application Data\RMF\RMF Spreadsheet Reporter

As shown in Figure 11-34 on page 292, the RMF folder contains the following:

- Macros - stored in the Macros directory
- Local Report Listings - stored in the RmfListings directory
- Local Overview Records - stored in the RmfRecords directory
- Overview control statements - stored in the Text directory
- Working Sets - stored in the WorkingSets directory

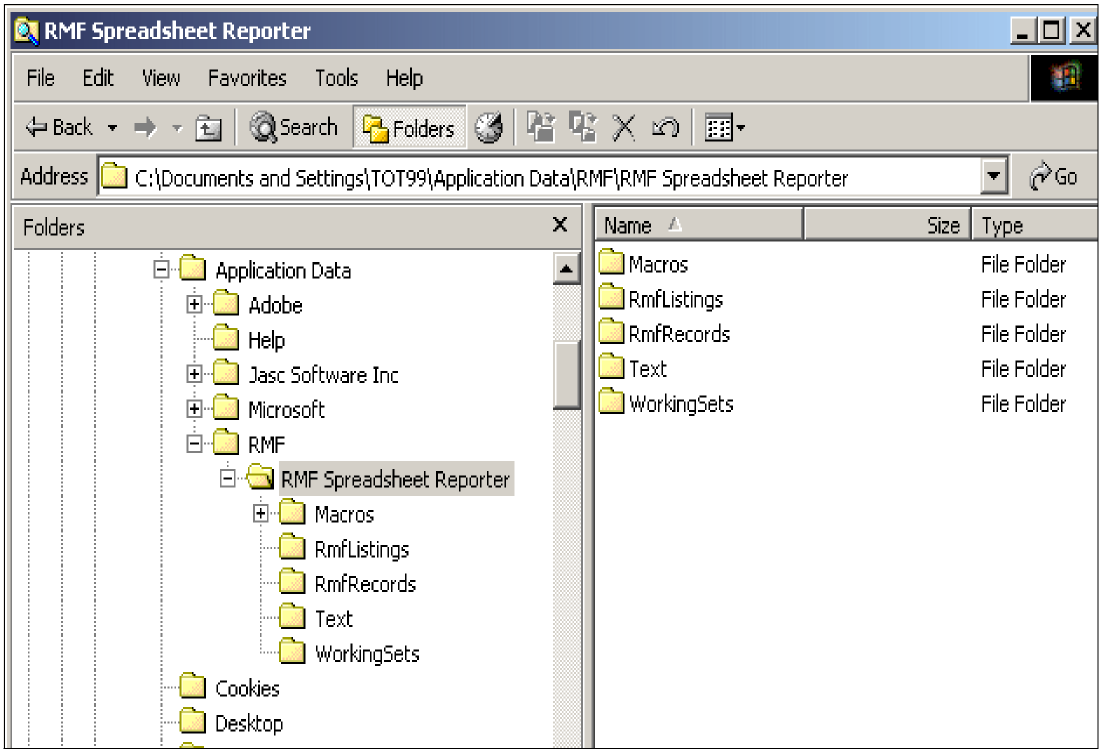


Figure 11-34 Application data files display

11.6.15 Spreadsheet Reporter component messages

The JES joblog, RMF Postprocessor messages, and a log of FTP activity are retrieved by the workstation upon job completion. These logs can be located and reviewed by selecting the **Messages** menu bar item, which result in display of the drop-down menu shown in Figure 11-35.

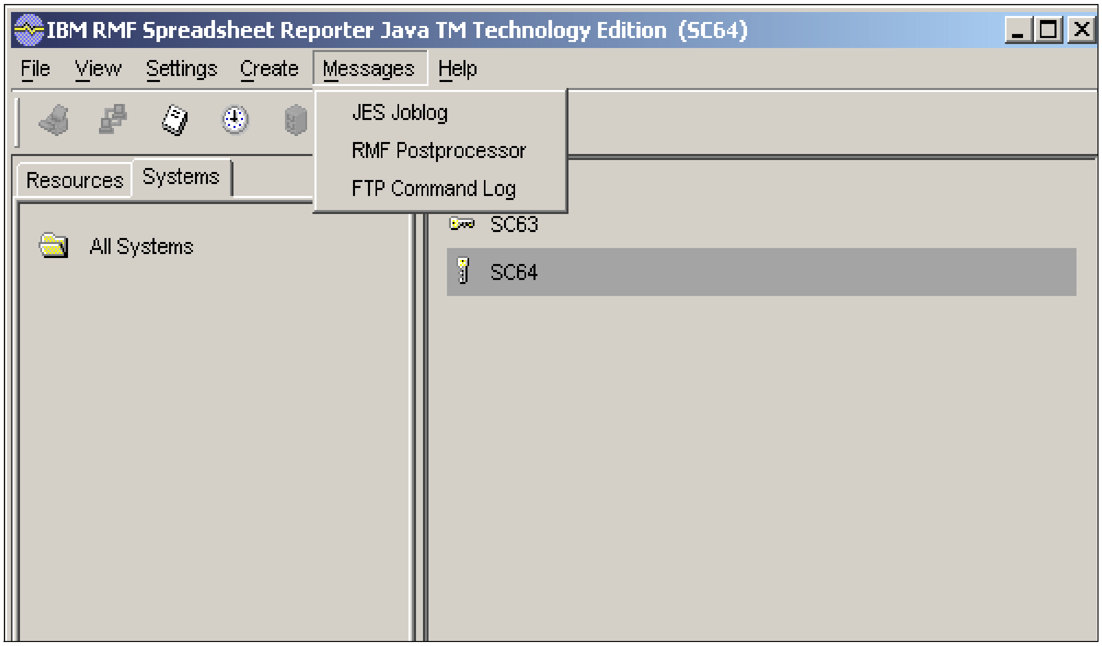

IBM RMF Spreadsheet Reporter Java TM Technology Edition (SC64)
File View Settings Create Messages Help
JES Joblog
RMF Postprocessor
FTP Command Log
Resources
Systems
All Systems
SC63
SC64

Figure 11-35 Systems panel from the RMF Spreadsheet Reporter main menu panel

From the drop-down menu, you can click to view the following items:

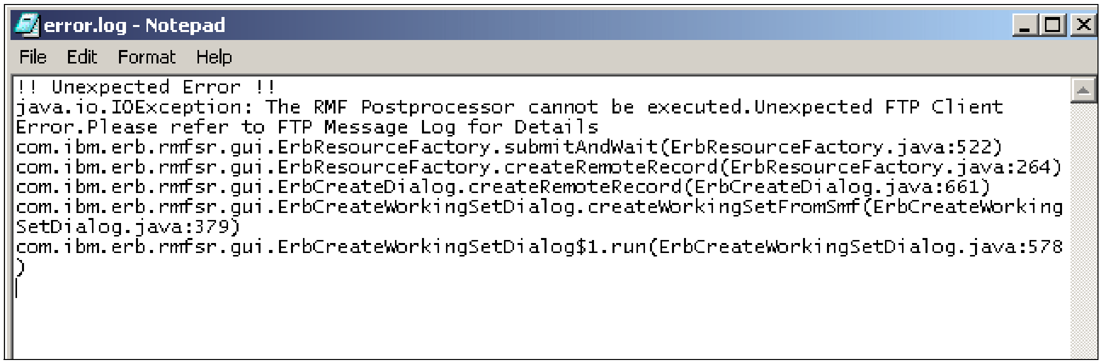
- The JES Joblog
- The RMF Postprocessor messages
- The FTP command log, which contains messages produced during file transfer activities

11.6.16 Error logging

Exceptions detected by the Spreadsheet Reporter are recorded as problem records that consist of an error message and a stack trace entry. These application-written problem records are written to:

C:\Program Files\RMF\RMF Spreadsheet Reporter\error.log

Figure 11-36 is an example of the contents of the error.log file.



```
!! Unexpected Error !!
java.io.IOException: The RMF Postprocessor cannot be executed.Unexpected FTP Client
Error.Please refer to FTP Message Log for Details
com.ibm.erb.rmfsr.gui.ErbResourceFactory.submitAndWait(ErbResourceFactory.java:522)
com.ibm.erb.rmfsr.gui.ErbResourceFactory.createRemoteRecord(ErbResourceFactory.java:264)
com.ibm.erb.rmfsr.gui.ErbCreateDialog.createRemoteRecord(ErbCreateDialog.java:661)
com.ibm.erb.rmfsr.gui.ErbCreateWorkingSetDialog.createWorkingSetFromSmf(ErbCreateWorking
SetDialog.java:379)
com.ibm.erb.rmfsr.gui.ErbCreateWorkingSetDialog$1.run(ErbCreateWorkingSetDialog.java:578
)
```

Figure 11-36 Notepad for the error.log

Using trace options

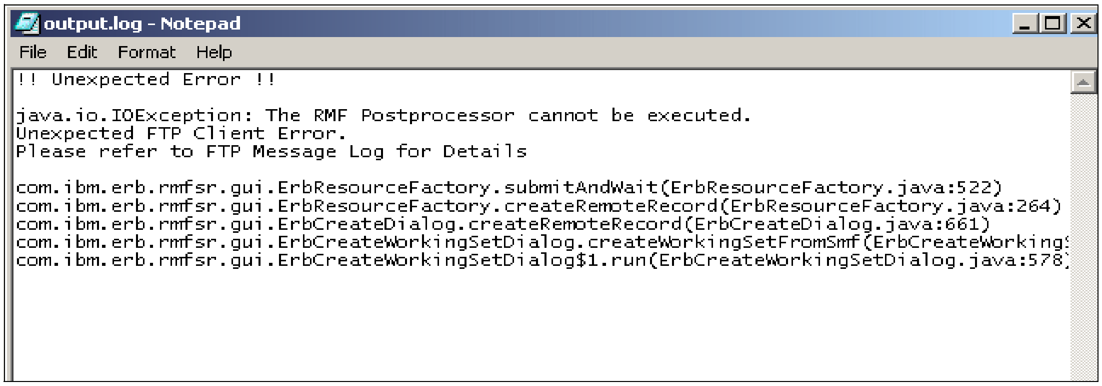
The Spreadsheet Reporter can be started with extended trace options to trap errors the application is otherwise unable to detect. This can be accomplished by starting the trace program from a command line:

C:\Program Files\RMF\RMF Spreadsheet Reporter\TraceRmfsr.bat

The resulting error messages can be viewed using:

C:\Program Files\RMF\RMF Spreadsheet Reporter\output.log

The results are shown in Figure 11-37.

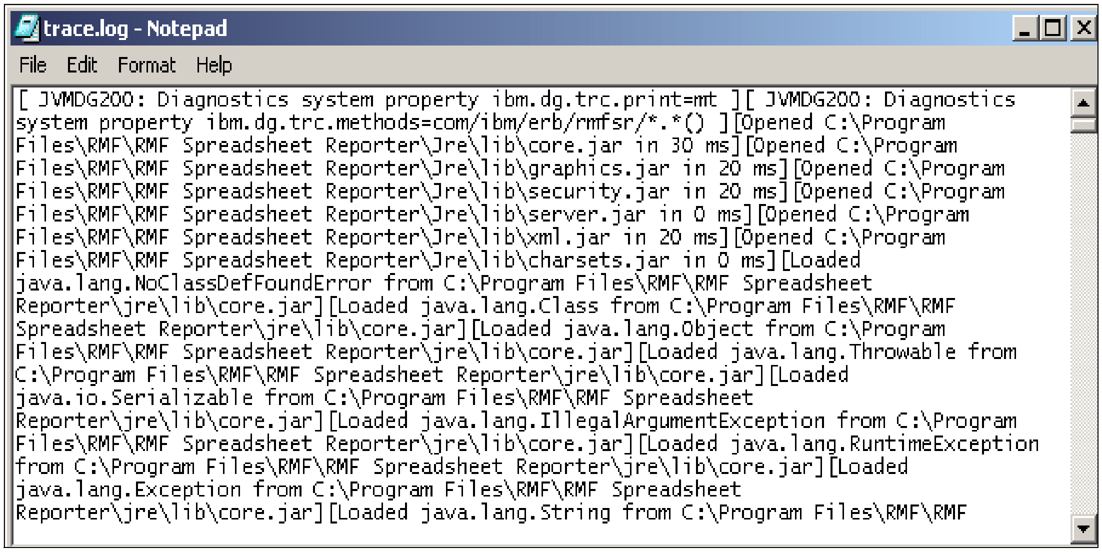


```
!! Unexpected Error !!

java.io.IOException: The RMF Postprocessor cannot be executed.
Unexpected FTP Client Error.
Please refer to FTP Message Log for Details

com.ibm.erb.rmfsr.gui.ErbResourceFactory.submitAndWait(ErbResourceFactory.java:522)
com.ibm.erb.rmfsr.gui.ErbResourceFactory.createRemoteRecord(ErbResourceFactory.java:264)
com.ibm.erb.rmfsr.gui.ErbCreateDialog.createRemoteRecord(ErbCreateDialog.java:661)
com.ibm.erb.rmfsr.gui.ErbCreateWorkingSetDialog.createWorkingSetFromSmf(ErbCreateWorking
com.ibm.erb.rmfsr.gui.ErbCreateWorkingSetDialog$1.run(ErbCreateWorkingSetDialog.java:578
```

Figure 11-37 RMF Spreadsheet Reporter output.log

The resulting trace entries, shown in Figure 11-38, were obtained using:

C:\Program Files\RMF\RMF Spreadsheet Reporter\trace.log



```
[ JVMDG200: Diagnostics system property ibm.dg.trc.print=mt ][ JVMDG200: Diagnostics
system property ibm.dg.trc.methods=com/ibm/erb/rmfsr/*.*() ][Opened C:\Program
Files\RMF\RMF Spreadsheet Reporter\Jre\lib\core.jar in 30 ms][Opened C:\Program
Files\RMF\RMF Spreadsheet Reporter\Jre\lib\graphics.jar in 20 ms][Opened C:\Program
Files\RMF\RMF Spreadsheet Reporter\Jre\lib\security.jar in 20 ms][Opened C:\Program
Files\RMF\RMF Spreadsheet Reporter\Jre\lib\server.jar in 0 ms][Opened C:\Program
Files\RMF\RMF Spreadsheet Reporter\Jre\lib\xml.jar in 20 ms][Opened C:\Program
Files\RMF\RMF Spreadsheet Reporter\Jre\lib\charsets.jar in 0 ms][Loaded
java.lang.NoClassDefFoundError from C:\Program Files\RMF\RMF Spreadsheet
Reporter\jre\lib\core.jar][Loaded java.lang.Class from C:\Program Files\RMF\RMF
Spreadsheet Reporter\jre\lib\core.jar][Loaded java.lang.Object from C:\Program
Files\RMF\RMF Spreadsheet Reporter\jre\lib\core.jar][Loaded java.lang.Throwable from
C:\Program Files\RMF\RMF Spreadsheet Reporter\jre\lib\core.jar][Loaded
java.io.Serializable from C:\Program Files\RMF\RMF Spreadsheet
Reporter\jre\lib\core.jar][Loaded java.lang.IllegalArgumentException from C:\Program
Files\RMF\RMF Spreadsheet Reporter\jre\lib\core.jar][Loaded java.lang.RuntimeException
from C:\Program Files\RMF\RMF Spreadsheet Reporter\jre\lib\core.jar][Loaded
java.lang.Exception from C:\Program Files\RMF\RMF Spreadsheet
Reporter\jre\lib\core.jar][Loaded java.lang.String from C:\Program Files\RMF\RMF
```

Figure 11-38 RMF Spreadsheet Reporter trace.log

11.7 RMF Performance Monitoring - Java Webstart enabling

RMF Performance Monitoring or RMF PM was initially added to the RMF product with OS/390 V1R3. It is an integral component of RMF that provides the capability to monitor the performance of z/OS systems in one or more Sysplexes from one monitoring focal point.

The monitoring focal point consists of either a Windows-, Linux-, or an OS/2®-based workstation connected via TCP/IP to one or more z/OS sysplexes.

Version 2 of the RMF PM workstation client provided with z/OS V1R5 has been rewritten in Java. It has more functionality than previous releases, enhanced usability, and a greatly improved look and feel.

RMF PM uses performance data recorded by the Distributed Data Server (DDS), which is a single data server executing in one system in the sysplex. If multiple sysplexes are to be monitored, each individual sysplex is required to have an active DDS. The DDS gathers data from each system in the sysplex's RMF Monitor III instance. As such, a subset of RMF Monitor III based data, such as general performance, job performance, and WLM performance data for workloads, service classes, reporting periods, and report classes are available. Current data, as well as data from the recent past can be monitored and analyzed.

The most recent version of RMF PM can be downloaded from the RMF PM wesbite at:

`http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmweb.htm`

The name of the file that should be downloaded for a Windows workstation is gpmwinv2.exe. Alternatively, the host data set SYS1.SERBPWSV(GPMWINV2) can be downloaded in binary to gpmwinv2.exe. Simply execute this self-extracting program to install all of the RMF PM components on the workstation. The program will be installed in the directory:

C:\Program Files\RMF\pm390

Its icons are added to the IBM RMF performance management folder.

The name of the file that should be downloaded to a Linux workstation is gpmlinpm.tgz. RMF PM then can be installed on the Linux workstation using the following command:

```
tar xvfz gpmlinpm.tgz
```

The installation process will install RMF PM in a directory named RMFPM, which will also contain the IBM Java 2 Runtime Engine. To start RMF PM under Linux the user simply enters the name of the directory that the product was installed to:

```
./rmfpm
```

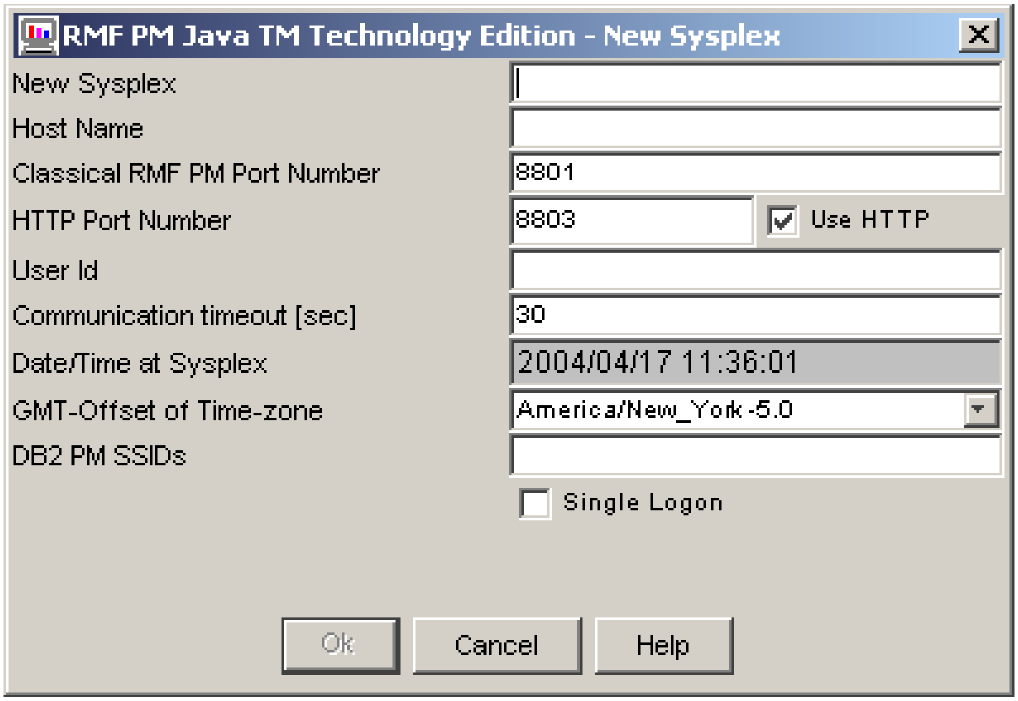
When the program is started the screen shown in Figure 11-39 is briefly displayed.

Loading RMF PM, please wait ...

IBM RMF PM
Java (TM) Technology Edition

Figure 11-39 RMF PM primary panel

The installer is then prompted to define the first sysplex. Detailed help describing each of the required inputs is available by clicking the Help button. The Define New Sysplex screen is displayed in Figure 11-40 on page 296.



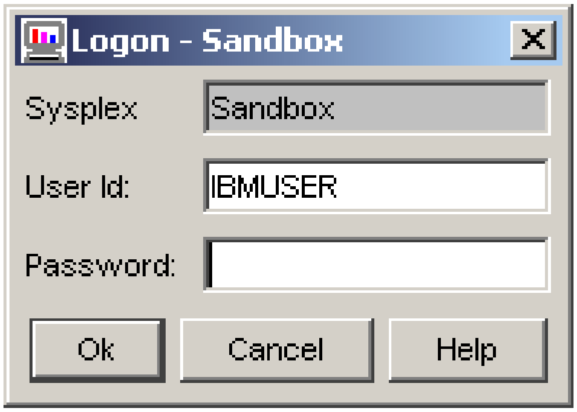
New Sysplex	
Host Name	
Classical RMF PM Port Number	8801
HTTP Port Number	8803 ☑ Use HTTP
User Id	
Communication timeout [sec]	30
Date/Time at Sysplex	2004/04/17 11:36:01
GMT-Offset of Time-zone	America/New_York -5.0
DB2 PM SSIDs	
	□ Single Logon

Figure 11-40 Define New Sysplex screen

When all required information has been specified, click **OK** to build the new sysplex and initiate RMF PM startup activities.

Note: Additional sysplexes can be defined after installation by using the RMF PM Sysplex dialog. Initial settings specified for a sysplex during installation can also be modified after installation using the Change Sysplex dialog.

The Logon screen is then displayed, as shown in Figure 11-41.



Sysplex	Sandbox
User Id:	IBMUSER
Password:	

Figure 11-41 RMF PM logon screen

The User ID (same as TSO user ID) is taken from the information provided when the sysplex was defined. Supply the password and click **OK**. If the logon is successful, the main panel for RMF PM is displayed as shown in Figure 11-42.

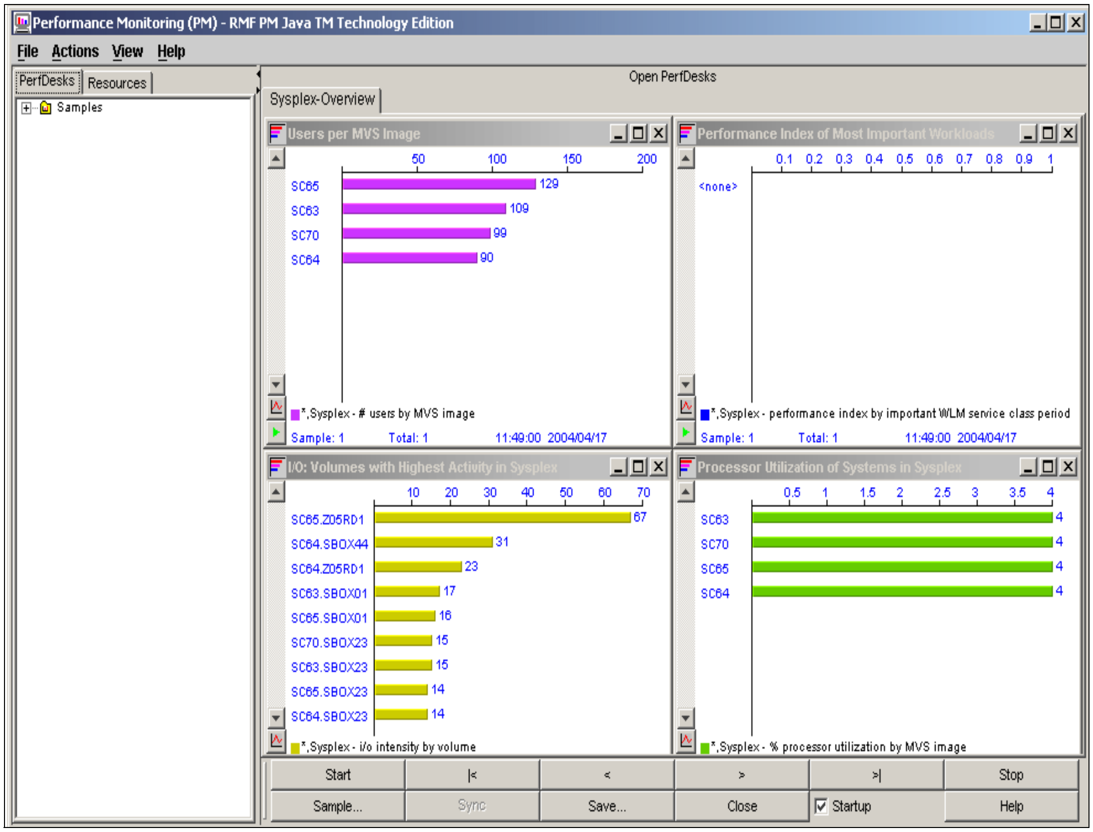

Performance Monitoring (PM) - RMF PM Java TM Technology Edition
File Actions View Help
PerfDesks Resources
Samples
Open PerfDesks
Sysplex-Overview
Users per MVS Image
50 100 150 200
SC65 129
SC63 109
SC70 99
SC64 90
*,Sysplex - # users by MVS image
Sample: 1 Total: 1 11:49:00 2004/04/17
Performance Index of Most Important Workloads
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
<none>
*,Sysplex - performance index by important WLM service class period
Sample: 1 Total: 1 11:49:00 2004/04/17
I/O: Volumes with Highest Activity in Sysplex
10 20 30 40 50 60 70
SC65.Z05RD1 67
SC64.SBOX44 31
SC64.Z05RD1 23
SC63.SBOX01 17
SC65.SBOX01 16
SC70.SBOX23 15
SC63.SBOX23 15
SC65.SBOX23 14
SC64.SBOX23 14
*,Sysplex - i/o intensity by volume
Processor Utilization of Systems in Sysplex
0.5 1 1.5 2 2.5 3 3.5 4
SC63 4
SC70 4
SC65 4
SC64 4
*,Sysplex - % processor utilization by MVS image
Start
|<
<
>
>|
Stop
Sample...
Sync
Save...
Close
☑ Startup
Help

Figure 11-42 RMF PM main panel

RMF PM monitoring

To begin monitoring the initial sysplex you created, click the **Resource** tab, then click the IBM z/OS resource. A screen shown in Figure 11-43 on page 298 is presented, providing the basis for your monitoring activities.

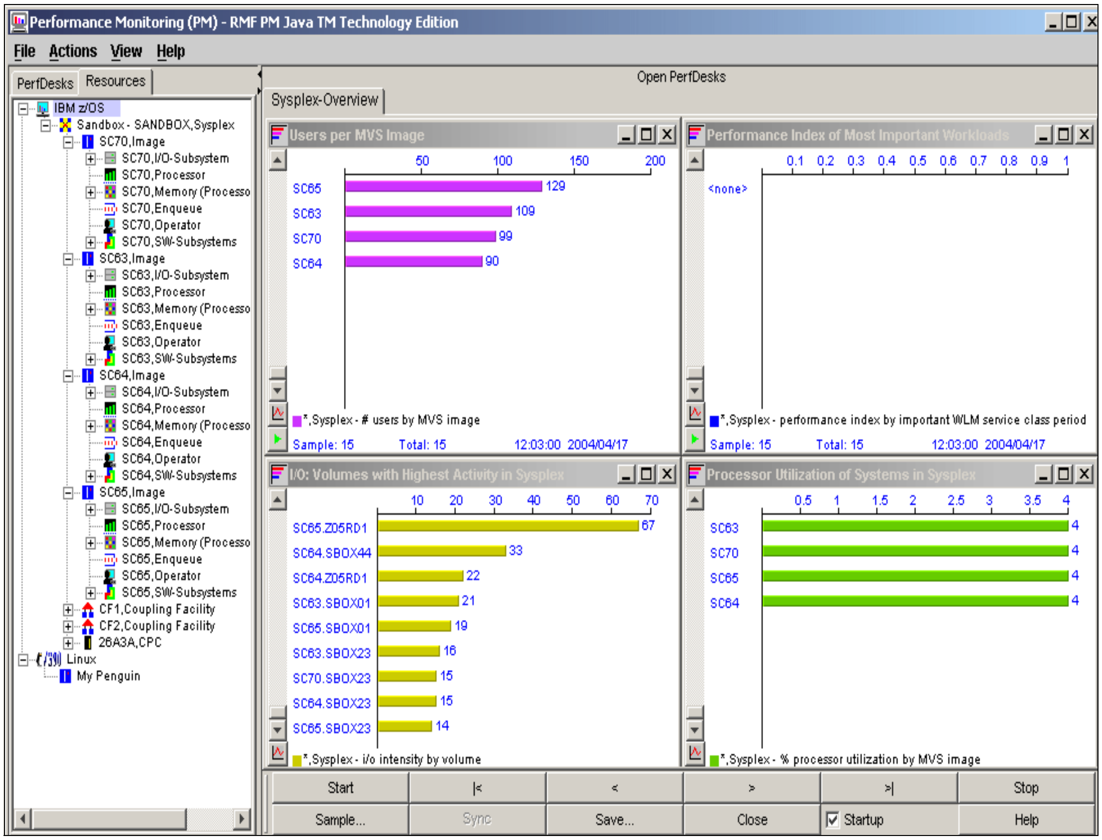


Figure 11-43 Monitoring activity panel

11.7.1 RMF PM summary

A summary of metrics in RMF PM that are available to monitor your system and your sysplex resources at the following Web site:

```
http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pm_metrics.htm
```

At the time of this writing, this information is only available via the internet. Therefore, the content of the Web site has been replicated for the user's convenience in Appendix A, "RMF Performance Monitor metrics" on page 467.

Installation, setup, and usage instruction for RMF PM is available in the chapter titled "Analysis on the Workstation" in the publication *z/OS Resource Measurement Facility (RMF) User's Guide*, SC33-7990.

Additionally, detailed Help facilities are available from RMF PM through the workstation's browser.

11.8 Sysplex data services for 64-bit environments

RMF provides a set of APIs that are available for application programs to use to retrieve performance data for their own use. With the advent of z/Architecture, these applications can be written to take advantage of 64-bit addressing. Since the RMF APIs are implemented as callable services, these too have been enabled for 64-bit callers.

11.8.1 31-bit callable services

The following 31-bit RMF callable services are affected:

ERBDSQRY	RMF query available sysplex SMF data service
ERBDSREC	RMF request sysplex SMF record data service
ERB2XDGS	RMF monitor II sysplex data gathering service
ERB3XDRS	RMF monitor III sysplex data retrieval service

Alternate entry points have been created for 31- and 64-bit callers. The majority of component code is shared between the multiple entry points. The separate entry points provide for normalization of input parameters so that the common code can process data above and below the bar.

11.8.2 64-bit callable services

The following are the 64-bit RMF callable services:

ERBDSQ64	RMF query available sysplex SMF data service
ERBDSR64	RMF request Sysplex SMF record data service
ERB2XD64	RMF monitor II Sysplex data gathering service
ERB3XD64	RMF monitor III Sysplex data retrieval service

11.8.3 64-bit parameters

Parameters required to use the 64-bit APIs are identical to the parameters used for the 31-bit APIs.

The 64-bit services perform the following functions:

- For storage areas based on pointers that are included in the passed parameter list, storage is obtained below the bar and the content of the above the bar storage is copied to the storage below the bar.
- All of the other input parameters are copied from the caller's storage areas to the API's own storage areas.
- Switch modes from AMODE64 to AMODE31.
- Build the parameter list.
- Load and call the related 31-bit service.
- Switch modes back to AMODE64.
- Copy the output parameters from the 31-bit service to the locations defined in the caller's parameter list.
- Return control to the caller.

It should also be noted that the invocation and operation of the existing 31-bit services remains unchanged.

For detailed descriptions of the RMF callable services and their parameters refer to chapter 2 of *z/OS Resource Measurement Facility (RMF) Programmer's Guide*, SC33-7994.

11.9 RMF monitoring support for WLM

RMF has been enhanced at z/OS V1R5 to provide monitoring support for performance block (PB) reporting and enqueue contention management.

11.9.1 Performance block reporting

Performance blocks (PBs), which are also called monitoring environments, are entities that represent performance management and measurement for transaction-based workloads, such as CICS.

WLM has extended support for monitoring environments to provide PB state sampling for multi-period service and report classes, and for service classes with goal types other than response time-based goals.

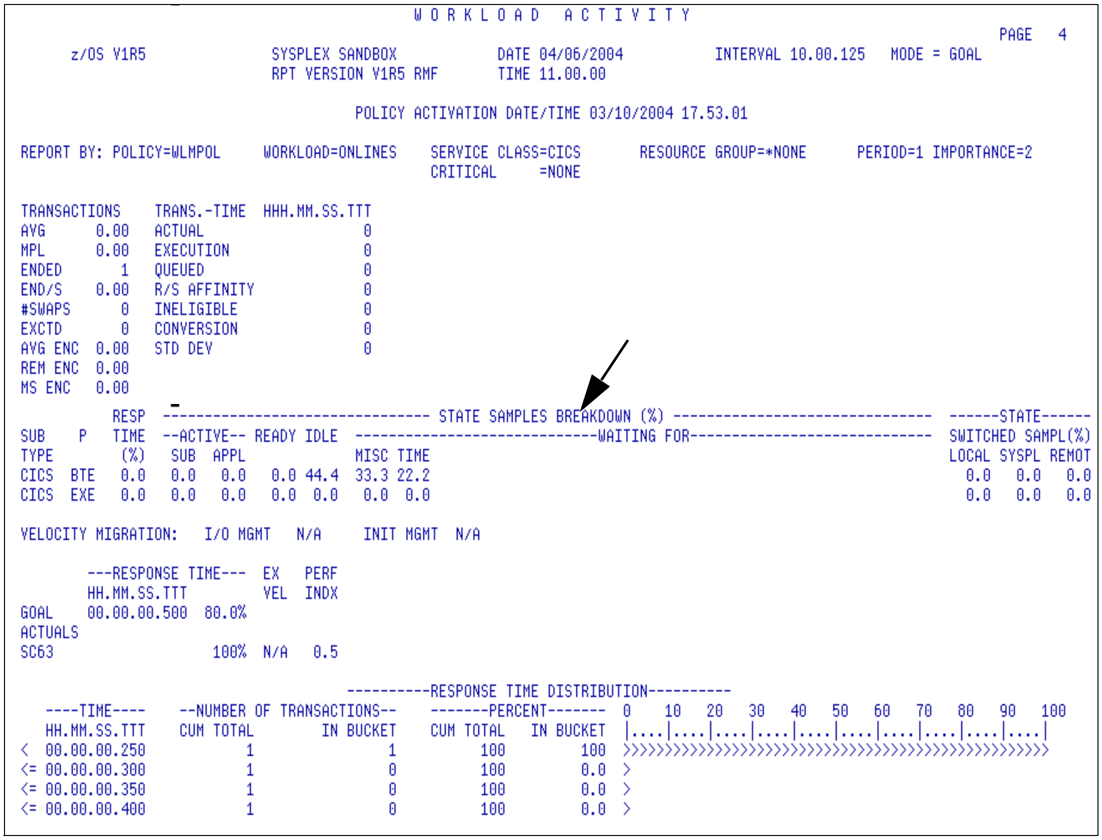
WLM also now provides support for WebSphere EE for z/OS by adding a new active state for monitoring environments that will provide the ability to distinguish between an active subsystem and an active application. Additionally, new WebSphere EE waiting states for monitoring environments have also been added to Monitor I and Monitor III reports. They are:

- Waiting for SSL thread
- Waiting for regular thread
- Waiting for registration to work table
- Waiting for resource type 1
- Waiting for resource type 2
- Waiting for resource type 3
- Waiting for resource type 4
- Waiting for resource type 5

Figure 11-44 is a JCL example of how to generate the RMF Monitor I Postprocessor Workload Activity report. Figure 11-45 on page 301 displays the RMF Monitor I Postprocessor Workload Activity report generated from this JCL.

```
//WLMGL JOB (0),'RMF WLMGL',CLASS=A,MSGCLASS=X
//RMF      EXEC PGM=ERBRMFPP
//MFPINPUT DD DSN=SORTED.VSTOR.RMF,DISP=SHR
//MFPMSGDS DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
  SYSRPTS(WLMGL(SCPER))
  SYSOUT(X)
//*
```

Figure 11-44 JCL to generate the RMF Monitor I Postprocessor Workload Activity report



```
                                               W O R K L O A D   A C T I V I T Y
                                                                                                                      PAGE   4
      z/OS V1R5               SYSPLEX SANDBOX            DATE 04/06/2004          INTERVAL 10.00.125   MODE = GOAL
                              RPT VERSION V1R5 RMF       TIME 11.00.00

                                        POLICY ACTIVATION DATE/TIME 03/10/2004 17.53.01

REPORT BY: POLICY=WLMPOL     WORKLOAD=ONLINES    SERVICE CLASS=CICS       RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=2
                                                 CRITICAL     =NONE

TRANSACTIONS    TRANS.-TIME  HHH.MM.SS.TTT
AVG      0.00   ACTUAL                  0
MPL      0.00   EXECUTION               0
ENDED       1   QUEUED                  0
END/S    0.00   R/S AFFINITY            0
#SWAPS      0   INELIGIBLE              0
EXCTD       0   CONVERSION              0
AVG ENC  0.00   STD DEV                 0
REM ENC  0.00
MS ENC   0.00
           RESP  -------------------------------- STATE SAMPLES BREAKDOWN (%) -------------------------------  ------STATE------
SUB    P   TIME  --ACTIVE-- READY IDLE  -----------------------------WAITING FOR-----------------------------  SWITCHED SAMPL(%)
TYPE        (%)   SUB  APPL             MISC TIME                                                              LOCAL SYSPL REMOT
CICS  BTE   0.0   0.0   0.0   0.0 44.4  33.3 22.2                                                                0.0   0.0   0.0
CICS  EXE   0.0   0.0   0.0   0.0  0.0   0.0  0.0                                                                0.0   0.0   0.0

VELOCITY MIGRATION:   I/O MGMT   N/A     INIT MGMT  N/A

        ---RESPONSE TIME---  EX   PERF
        HH.MM.SS.TTT         VEL  INDX
GOAL    00.00.00.500  80.0%
ACTUALS
SC63                   100%  N/A   0.5

                                       ----------RESPONSE TIME DISTRIBUTION----------
   ----TIME----     --NUMBER OF TRANSACTIONS--     -------PERCENT-------  0    10   20   30   40   50   60   70   80   90   100
   HH.MM.SS.TTT     CUM TOTAL        IN BUCKET     CUM TOTAL   IN BUCKET  |....|....|....|....|....|....|....|....|....|....|
<  00.00.00.250             1                1           100         100  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<= 00.00.00.300             1                0           100         0.0  >
<= 00.00.00.350             1                0           100         0.0  >
<= 00.00.00.400             1                0           100         0.0  >
```

Figure 11-45 RMF Monitor I Postprocessor Workload Activity report

The subsystem section of the report has been changed to split the ACTIVE column into SUB for active subsystem and APPL for active application state samples.

Additionally, the new PB wait states are displayed in WAITING FOR if they are among the highest non-zero values.

The RESPONSE TIME BREAKDOWN has effectively been replaced by the STATE SAMPLES BREAKDOWN. The report expresses single states as a percentage of the total transaction state samples reported for the subsystem phase. The TOTAL column has been renamed to RESP TIME % but is still indicative of the percentage of total response time.

The new fields of the WLMGL report for the STATE SAMPLES BREAKDOWN are shown in Table 11-7.

Table 11-7 New fields in the RMF Workload Activity report

Field heading	Description
RESP TIME %	The percentage of transaction response time in either the BEGIN-TO-END or the EXECUTION phase.
ACTIVE SUB	The active subsystem state sample percentage indicating program execution on behalf of the work request from the perspective of the work manager.
ACTIVE APPL	The active application state sample percentage which when contrasted with APPL SUB allows a subsystem to differentiate between work requests processed by the subsystem and requests processed by an application that was invoked by the subsystem.

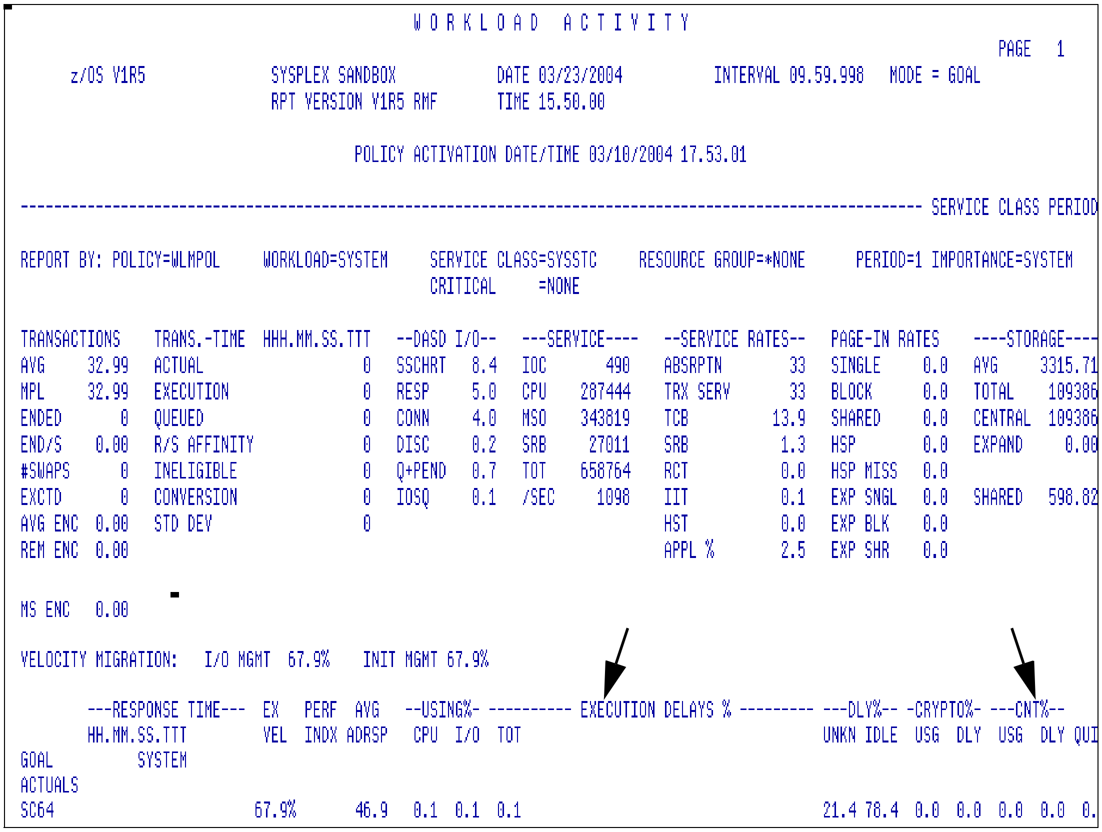
Field heading	Description
WAITING FOR	The fourteen highest non-zero wait values will be displayed. The new subsystem state samples are: SSLT - wait % for SSL thread state samples REGT - wait % for regular thread state samples WORK - wait % for registration to work tables TYP1 - wait % for resource type 1 state samples TYP2 - wait % for resource type 2 state samples TYP3 - wait % for resource type 3 state samples TYP4 - wait % for resource type 4 state samples TYP5 - wait % for resource type 5 state samples

The MONITOR III Sysplex Work Manager Delay SYSWKM report has also been updated to reflect the values previously described for the PM state changes.

11.9.2 Enqueue Contention Management

The RMF Monitor I Postprocessor Workload Activity report (WLMGL) has been updated to include a new section in the Service Class period report to enhance WLM contention monitoring. Specifically, resource contention using CNT USG% and resource contention delay CNT DLY% states have been added to the GOAL and ACTUALS section of the report. Additionally, the column TOTAL in the EXECUTION DELAYS section is renamed to TOT. Only the seven highest non-zero values contributing to TOT are shown in EXECUTION DELAYS.

Figure 11-46 shows the updated WLMGL report.



```
                                               W O R K L O A D   A C T I V I T Y
                                                                                                                  PAGE   1
        z/OS V1R5               SYSPLEX SANDBOX           DATE 03/23/2004          INTERVAL 09.59.998   MODE = GOAL
                                RPT VERSION V1R5 RMF      TIME 15.50.00

                                          POLICY ACTIVATION DATE/TIME 03/10/2004 17.53.01

 ------------------------------------------------------------------------------------------------------------ SERVICE CLASS PERIOD

 REPORT BY: POLICY=WLMPOL     WORKLOAD=SYSTEM     SERVICE CLASS=SYSSTC     RESOURCE GROUP=*NONE     PERIOD=1 IMPORTANCE=SYSTEM
                                                  CRITICAL     =NONE

 TRANSACTIONS    TRANS.-TIME  HHH.MM.SS.TTT   --DASD I/O--   ---SERVICE----   --SERVICE RATES--   PAGE-IN RATES    ----STORAGE----
 AVG     32.99   ACTUAL                   0   SSCHRT   8.4   IOC       490    ABSRPTN        33   SINGLE     0.0   AVG     3315.71
 MPL     32.99   EXECUTION                0   RESP     5.0   CPU    287444    TRX SERV       33   BLOCK      0.0   TOTAL    109386
 ENDED       0   QUEUED                   0   CONN     4.0   MSO    343819    TCB          13.9   SHARED     0.0   CENTRAL  109386
 END/S    0.00   R/S AFFINITY             0   DISC     0.2   SRB     27011    SRB           1.3   HSP        0.0   EXPAND     0.00
 #SWAPS      0   INELIGIBLE               0   Q+PEND   0.7   TOT    658764    RCT           0.0   HSP MISS   0.0
 EXCTD       0   CONVERSION               0   IOSQ     0.1   /SEC     1098    IIT           0.1   EXP SNGL   0.0   SHARED   598.82
 AVG ENC  0.00   STD DEV                  0                                   HST           0.0   EXP BLK    0.0
 REM ENC  0.00                                                                APPL %        2.5   EXP SHR    0.0


 MS ENC   0.00

 VELOCITY MIGRATION:   I/O MGMT   67.9%    INIT MGMT 67.9%

         ---RESPONSE TIME---  EX    PERF  AVG   --USING%- ---------- EXECUTION DELAYS % --------- ---DLY%-- -CRYPTO%- ---CNT%--
         HH.MM.SS.TTT         VEL   INDX ADRSP   CPU  I/O  TOT                                    UNKN IDLE  USG  DLY  USG  DLY QUI
 GOAL          SYSTEM
 ACTUALS
 SC64                        67.9%       46.9   0.1  0.1  0.1                                     21.4 78.4  0.0  0.0  0.0  0.0  0.
```

Figure 11-46   WLMGL report

The new field in the Goals versus Actuals section of the WLMGL report is described in Table 11-8.

Table 11-8 New field in Goals versus Actuals section of RMF Workload Activity report

Field name	Description
CNT%	Resource Contention using and delay states as reported to WLM by the resource Manager. USG - Work is HOLDING a resource DLY - Work is WAITING for a resource

12

SMP/E for z/OS and OS/390 Version 3 Release 2

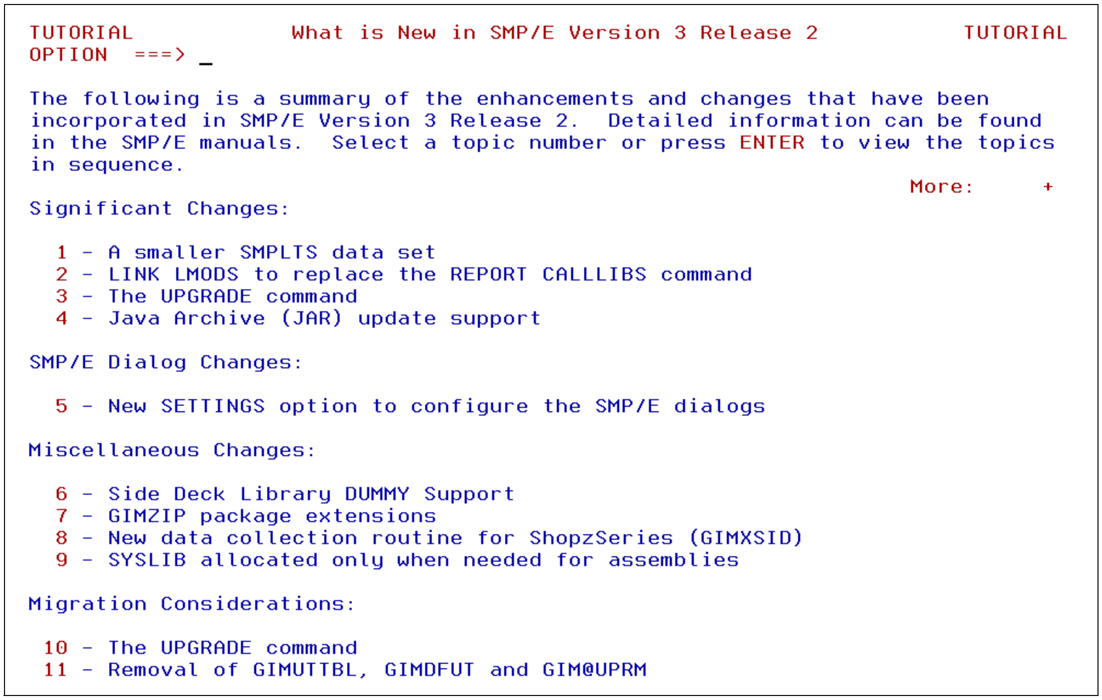
This chapter describes the enhancements and changes that have been incorporated into SMP/E V3R2.

The following topics are discussed:

- What’s new in SMP/E V3R2
- A smaller SMPLTS data set
- The LINK LMODS command
- The UPGRADE command
- Java Archive (JAR) update support
- ShopzSeries data collection
- SMPCSI allocation sample
- SYSDEFSD (side deck library) DUMMY support
- Internet package extensions
- Optional migration actions
- Summary for SMP/E V3R2

12.1 What is new in SMP/E V3R2

For a quick reference of what's new in SMP/E V3R2 go to the SMP/E primary option menu and select option **w**. The panel shown in Figure 12-1 is displayed.



```
TUTORIAL          What is New in SMP/E Version 3 Release 2          TUTORIAL
OPTION  ===> _

The following is a summary of the enhancements and changes that have been
incorporated in SMP/E Version 3 Release 2.  Detailed information can be found
in the SMP/E manuals.  Select a topic number or press ENTER to view the topics
in sequence.
                                                                More:     +
Significant Changes:

  1 - A smaller SMPLTS data set
  2 - LINK LMODS to replace the REPORT CALLLIBS command
  3 - The UPGRADE command
  4 - Java Archive (JAR) update support

SMP/E Dialog Changes:

  5 - New SETTINGS option to configure the SMP/E dialogs

Miscellaneous Changes:

  6 - Side Deck Library DUMMY Support
  7 - GIMZIP package extensions
  8 - New data collection routine for ShopzSeries (GIMXSID)
  9 - SYSLIB allocated only when needed for assemblies

Migration Considerations:

 10 - The UPGRADE command
 11 - Removal of GIMUTTBL, GIMDFUT and GIM@UPRM
```

Figure 12-1 SMPE V3R2 online tutorial

12.2 A smaller SMPLTS data set

The SMPLTS data set is used by SMP/E to save the base version of a product's load modules that use callable services. The SMPLTS data set in prior releases of SMP/E is very large and is used to manage all load modules that have a CALLLIBS subentry (load modules that exploit the link edit autocall facility). In SMP/E V3R2 the goal is to reduce the space requirements for the SMPLTS.

To reduce SMPLTS space requirements, SMP/E now saves a base version of a load module in the SMPLTS data set only if it contains both CALLLIBS and XZMOD (cross-zone modules) subentries. If a load module contains CALLLIBS subentries, but no XZMOD subentries, this load module is not saved in the SMPLTS.

The result is a smaller SMPLTS. However, the use of the SMPLTS is as follows:

- The SMPLTS is still required for load modules that use CALLLIBS and contain cross-zone modules. Cross-zone modules are added to a load module using the **LINK FROMZONE** command.
- The SMPLTS is still used to resolve certain special cases that would otherwise result in a failed link edit operation.

In a typical system there are none, or very few cross-zone load modules, thus the SMPLTS will be empty or very small.

12.2.1 SMP/E and CALLLIBS

For load modules that use CALLLIBS, SMP/E builds them from scratch by including all modules that use the following:

- Single-csect modules in the target libraries
- Modules from PTFs in the SMPPTS
- Modules in the SMPTLIB data set
- Modules from the distribution libraries

There will no longer be two link edit operations for load modules that use CALLLIBS to first link edit into the SMPLTS and to then link edit into the target library.

Now there will be only one link edit into the target library, but:

- This includes all modules rather than a previous copy from SMPLTS.
- One link edit means less SYSPRINT output to go through.

SMPLTS cleanup

What happens to existing load modules in the SMPLTS?

- SMP/E deletes unneeded load modules from the SMPLTS data set during APPLY, RESTORE, and CLEANUP command processing.
- However, since such an SMPLTS data set is not compatible with prior releases of SMP/E, this cleanup of existing load modules occurs only after you use the new **UPGRADE** command to indicate your desire to exploit new functions.

> **Note:** The building of load modules from scratch requires the dlib data sets to be available during APPLY processing.

12.3 The LINK LMODS command

The new **LINK LMODS** command takes the place of the **REPORT CALLLIBS** command. It provides all the functionality of the REPORT CALLIBS command with some additional benefits. LINK LMODS is used to link edit any load module. Load modules may be specified individually by name or like the REPORT CALLLIBS command, by reference to a particular CALLLIBS ddname. Once the load modules are identified, LINK LMODS will find copies of each module included in the load modules and will then directly link edit load modules into their target libraries, just like APPLY and RESTORE. LINK LMODS even has a CHECK mode so you can see what it will do before it actually does it, and it can attempt recovery from an out-of-space condition by compressing the target library and trying the link edit operation again.

Figure 12-2 shows the LINK LMODS command syntax.

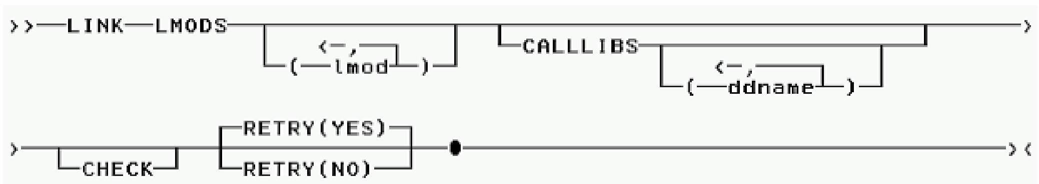


Figure 12-2 LINK LMODS command syntax

Figure 12-3 shows two examples of the LINK LMODS command. The first example link edits all load modules that use SCEELKED as a CALLLIBS.

The second example link edits the load module named *lmodname*.

```
SET BOUNDARY(tzone).
LINK LMODS CALLLIBS(SCEELKED)
     RETRY(YES)
     CHECK.
SET BOUNDARY(tzone).
LINK LMODS CALLLIBS(lmodname)
     RETRY(YES)
     CHECK.
```

Figure 12-3 Examples of the LINK LMODS command

12.4 The UPGRADE command

New SMP/E functions must sometimes make changes to SMP/E data sets that cannot be properly processed by prior SMP/E releases. In the past, such changes were made automatically, without any warning. Afterwards the zone was not usable by previous releases of SMP/E. For example, a new type of element requires a new entry type in the SMPCSI data sets and these new entry types are typically not understood or processed correctly by SMP/E levels that have not been specifically updated to do so.

This is addressed by the new **UPGRADE** command, which allows you to make a trade-off between fully exploiting new functions and preserving compatibility with previous releases. The UPGRADE command is now required before making incompatible changes to existing zones and other data sets. It forces you to decide when incompatible changes may or may not be made, as follows:

- If the UPGRADE command is *not* used, existing zones and data sets will remain compatible with prior SMP/E releases.
- If the UPGRADE command is used, SMP/E is then authorized to exploit new functions and to make incompatible changes to existing zones and data sets.

Figure 12-4 shows the UPGRADE command syntax.

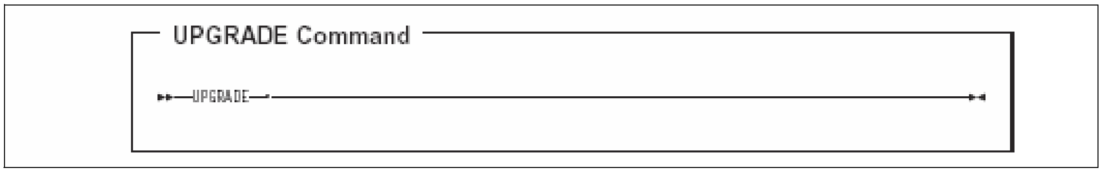


Figure 12-4 The SMP/E V3R2 UPGRADE command

Figure 12-5 on page 309 shows an example of the UPGRADE command. Although the entire traditional SMP/E level indicator is stored and displayed for this subentry (version, release, and PTF level), currently only the version and release levels are used for comparisons.

The UPGRADE command does the following:

- The scope of the upgrade level is the SET-to zone.

- The new UPGLEVEL subentry for a zone indicates the highest SMP/E release level allowed (32.00 in Figure 12-5), to make incompatible changes to the zone and related data sets.
- Sets an "upgrade level" for a zone which becomes the highest SMP/E release level allowed to make incompatible changes to the zone.

```
SET BDY(TGT).
  UPGRADE.
  LIST TARGETZONE.

TGT     ZONE ENTRIES

TGT       TZONE          = TGT
          UPGLEVEL       = SMP/E  32.00
          RELATED        = DLIB
          SREL           = Z038
```

Figure 12-5   Example of the UPGRADE command

12.4.1 SMPLTS cleanup

For SMP/E V3.2 the SMPLTS cleanup activities are not performed unless the UPGLEVEL of the current zone is at least SMP/E 32.00. This means, until you run the UPGRADE command to set the upgrade level of the zone to SMP/E 32.00, SMP/E does not clean up the SMPLTS data set during APPLY, RESTORE, or CLEANUP. SMP/E continues to use and maintain the SMPLTS data set in the traditional fashion until such time, thus ensuring an environment that can be processed by prior SMP/E release levels.

In addition, during APPLY and ACCEPT command processing, if a SYSMOD that contains a ++HFS element with a long LINK value is attempted to be installed, SMP/E does not allow its installation unless the upgrade level of the zone is at least SMP/E 32.00. Also, APPLY and ACCEPT do not allow a SYSMOD to be installed that contains a new ++JAR element unless the upgrade level is SMP/E 32.00.

12.5 Java Archive (JAR) update support

The discussion that follows is beneficial to a developer who wishes to use the new capabilities when performing the SMP/E packaging for a product or application that uses JAR files. Java ARchive (JAR) is a popular method to aggregate many Java class files into one. Traditionally, z/OS and OS/390 JAR files are huge!

More granular control is needed over the files within a JAR file (component files) in order to reduce the size of PTFs:

The following operations can be done on JAR files using the **jar** command (part of the JDK):

- Create JAR files.
- View the contents of JAR files.
- Extract component files from JAR files.
- Update the contents of JAR files. This means replacing a subset of the component files or adding additional component files, or both.

Note: There is no capability to delete component files using the JAR update—only add or replace. Use a JAR replacement with the subject files removed instead.

12.5.1 New element types to describe JAR files

SMP/E Version 3 Release 2 introduces two new element types to describe Java Archive files, and SMP/E replace and update such files. There are two new element types:

- **++JAR** element type is used to add and replace JAR files:
  - Complete replacement for a JAR file
  - Intended for FUNCTIONs and occasional PTFs
  - Constructed in JAR file format
  - Treated just like all other ++HFS elements; copied to target directory
- **++JARUPD** element type is used to provide an update for a JAR file. To update a JAR file means to add or replace component files within a Java Archive, rather than replacing the entire Java Archive file. This then allows much smaller and more granular PTFs for products that use Java and JAR files. This element type:
  - Contains new and changed component files only
  - Is intended for PTFs, APARs, and USERMODs
  - Is constructed in JAR file format
  - Is used by SMP/E with the **`jar`** command to update a JAR file with the contents of the update

12.5.2 ++JAR command usage

Figure 12-6, is an example of the ++JAR command usage. It uses a simple TicTacToe applet as an example.

```
/u/harry/TicTacToe/TicTacToe.class
                  /audio/beep.au
                  /audio/ding.au
                  /audio/yahoo.au
                  /images/cross.gif
                  /images/not.gif
```

Figure 12-6 TicTacToe example

Create a JAR file

Access the directory where the file resides and then create a JAR file using the **`jar`** command:

```
cd /u/harry/TicTacToe/
> jar cf ABCTTT.jar *
```

To view the file’s contents, issue a **`jar`** command and the contents are displayed, as follows:

```
jar tf ABCTTT.jar
TicTacToe.class
audio/beep.au
audio/ding.au
audio/yahoo.au
images/cross.gif
images/not.gif
```

When you create the JAR file for the applet using the **`jar`** command as discussed, and then package it as a ++JAR element in a PTF, as in Figure 12-7 on page 311, the MCS used for file ABCTTT.jar looks very much like the MCS for Hierarchical File System ++HFS elements.

The JAR replacement file, ABCTTT.jar, is simply copied by SMP/E into its target directory during the installation operation, just like all other HFS-type elements.

> **Note:** ++JAR is intended for FUNCTIONs and some PTFs to add or replace an entire JAR file.

```
++JAR(ABCTTT) DISTLIB(AABCBIN) SYSLIB(SABCBIN) RELFILE(2)
  PARM(PATHMODE(0,6,4,4))
  LINK(‘../TicTacToe.jar’)
  SYMLINK(‘../../../../../usr/lib/TicTacToe.jar’)
  SYMPATH(‘../../usr/lpp/abc/bin/TicTacToe.jar’).
```

Figure 12-7 Example of **`++JAR`** *command*

12.5.3 Example of the ++JARUPD command usage

Assume an APAR causes a change to the applet for one of the following reasons:

- Replace the class file
- Add a new image file

Suppose now you must update the /audio/beep.au file within the archive. Again, you can use the ++JARUPD MCS to describe an update to the archive.

Assuming the replacement audio file resides in a directory as follows:

/u/apars/ow54321/TicTacToe/audio/beep.au

The following **`jar`** command could create the necessary JAR update:

```
cd /u/apars/ow54321/TicTacToe/
  jar cvf ABCTTT.jarupd *
```

The resulting JAR file ABCTTT.jarupd is packaged as a ++JARUPD in a PTF as shown in Figure 12-8.

```
++PTF(UW54321).
++VER(Z038) FMID(fmid) PRE(UW12345).
++JARUPD(ABCTTT)
    PARM(PATHMODE(0,6,4,4)) JARPARM(0M)
    LINK('../TicTacToe.jar')
    SYMLINK('../../../../../usr/lib/TicTacToe.jar')
    SYMPATH('../../usr/lpp/abc/bin/TicTacToe.jar').
```

Figure 12-8 ++JARUPD PTF

Unlike the ++JAR element, the ++JARUPD is not copied to the target directory by SMP/E during installation. The existing JAR file is “updated” with the contents of the ++JARUPD.

> **Note:** ++JARUPD is intended for PTFs, APARs, and USERMODs to update an existing JAR file.

12.5.4 ++JAR command extract option

To update a JAR file, SMP/E first *extracts* the new and changed component files from the ++JARUPD element. These files are extracted into a temporary directory. Then SMP/E uses the *update* option of the **`jar`** command to add or replace these component files in the existing JAR file.

SMP/E uses the extract (**`x`**) option of the **`jar`** command to extract the component files from the JAR update as shown in Figure 12-9.

```
cd /tmp/smpe/
jar xf ABCTTT.jarupd *
```

Figure 12-9 The extract option of the JAR command

12.5.5 The update option of the JAR command

PTFs that update or replace the same JAR file must have a PRE or SUP relationship.

- ++JARUPD must PRE previous ++JAR
- ++JARUPD must PRE or SUP previous ++JARUPDs
- ++JAR must PRE or SUP previous ++JAR and ++JARUPDs

SMP/E uses the:

- Extract (**`x`**) option of the **`jar`** command to extract the new and changed component files from ++JARUPDs
- Update (**`u`**) option of the **`jar`** command to update JAR files with the new and changed component files

Java 2 Technology Edition

Java 2 Technology Edition supports the **`u`** option and is therefore a driving system requirement:

- The official name is *IBM Developer Kit for OS/390, Java 2 Technology Edition.*
- It is a no-charge product available for OS/390 V2.8 and above, and in WebSphere 4.0.1 and above.
- SMP/E also requires additional support in BPXCOPY. APAR OW57210 is available on OS/390 V2.10 and up.

Example of the update option

The update (**`u`**) option of the **`jar`** command, shown in Figure 12-10, is used to replace TicTacToe.class and add new.gif to the existing JAR file, as shown in Figure 12-11.

```
cd /tmp/smpe/
jar uf /u/harry/TicTacToe/ABCTTT.jar *
```

Figure 12-10 Update option on jar command

> **Note:** Java 2 Technology Edition is required in order to use the update option of the **`jar`** command.

Replace a file

```
jar tf ABCTTT.jar
TicTacToe.class
audio/beep.au
audio/ding.au
audio/yahoo.au
images/cross.gif
images/not.gif
images/new.gif
```

Figure 12-11 The update options of the **`jar`** *command*

12.6 ShopzSeries data collection

The new data collection routine for ShopzSeries simplifies and consolidates data collection tasks. Previously the existing GIMXTRX routine was used to collect PTF data (bitmaps) and the **`LIST FEATURE`** command was used to collect FEATURE data. Data collection was also done for just one global zone.

The new data collection routine is **GIMXSID**, which does the following:

- Collects both PTF and FEATURE data in one step and stores in one inventory file.
- The inventory file can be written directly to a file in the UNIX file system as well as sequential and partitioned data sets.
- Supports multiple global zones.

Figure 12-12 is an example of the GIMXSID routine. The options are:

WAIT= This option indicates how long to wait for CSI data sets in use. The default is 60 minutes.

SMPXTOUT The output data set may be sequential, member of PDS, or a file in the z/OS UNIX file system.

CSI The data set name for global zone.

TARGET Names of target zones is optional and all target zones are used if not specified.

```
//BITMAP   EXEC PGM=GIMXSID,PARM='WAIT=60MIN'
//SMPOUT   DD SYSOUT=*
//SMPXTOUT DD DSN=USER.GIMXSID.OUTPUT,DISP=(NEW,CATLG),
//            SPACE=(TRK,(1,1)),UNIT=SYSALLDA
//SYSIN    DD *
CSI=SMPE.ZOS.CSI
TARGET=ZOS14,JES214
CSI=SMPE.DB2.CSI
CSI=SMPE.CICS.CSI
/*
```

Figure 12-12 Sample of how to use GIMXSID

Figure 12-13 on page 314 shows the new data collection routine for ShopzSeries.

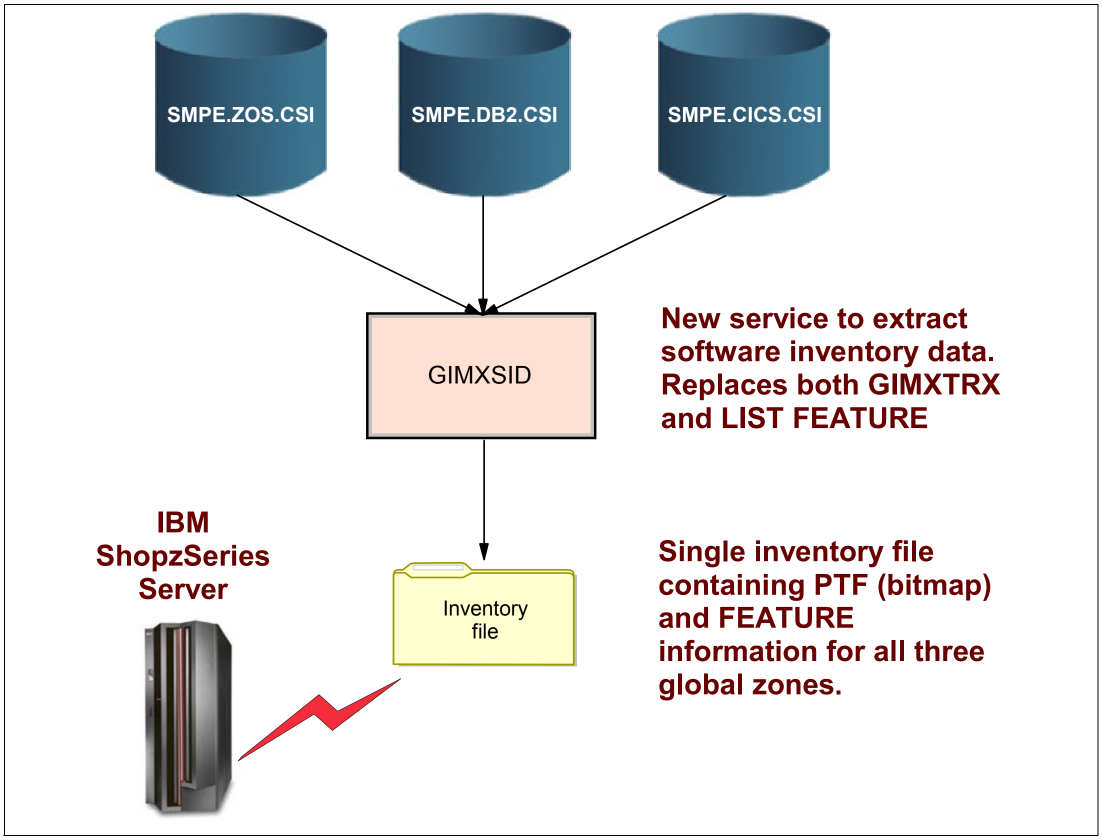

SMPE.ZOS.CSI
SMPE.DB2.CSI
SMPE.CICS.CSI
GIMXSID
New service to extract software inventory data. Replaces both GIMXTRX and LIST FEATURE
IBM ShopzSeries Server
Inventory file
Single inventory file containing PTF (bitmap) and FEATURE information for all three global zones.

Figure 12-13 ShopzSeries Data Collection

12.6.1 Process to receive an internet file into multiple global zones

You can now RECEIVE an internet order for multiple global zones as shown in Figure 12-14 on page 315. To RECEIVE the single order into each global zone independently, but download the order only once over the internet, do the following:

- The first operation uses `RECEIVE FROMNETWORK` to transfer the order into a common SMPNTS.
- Subsequent operations use `RECEIVE FROMNTS` to use the order stored in the common SMPNTS.

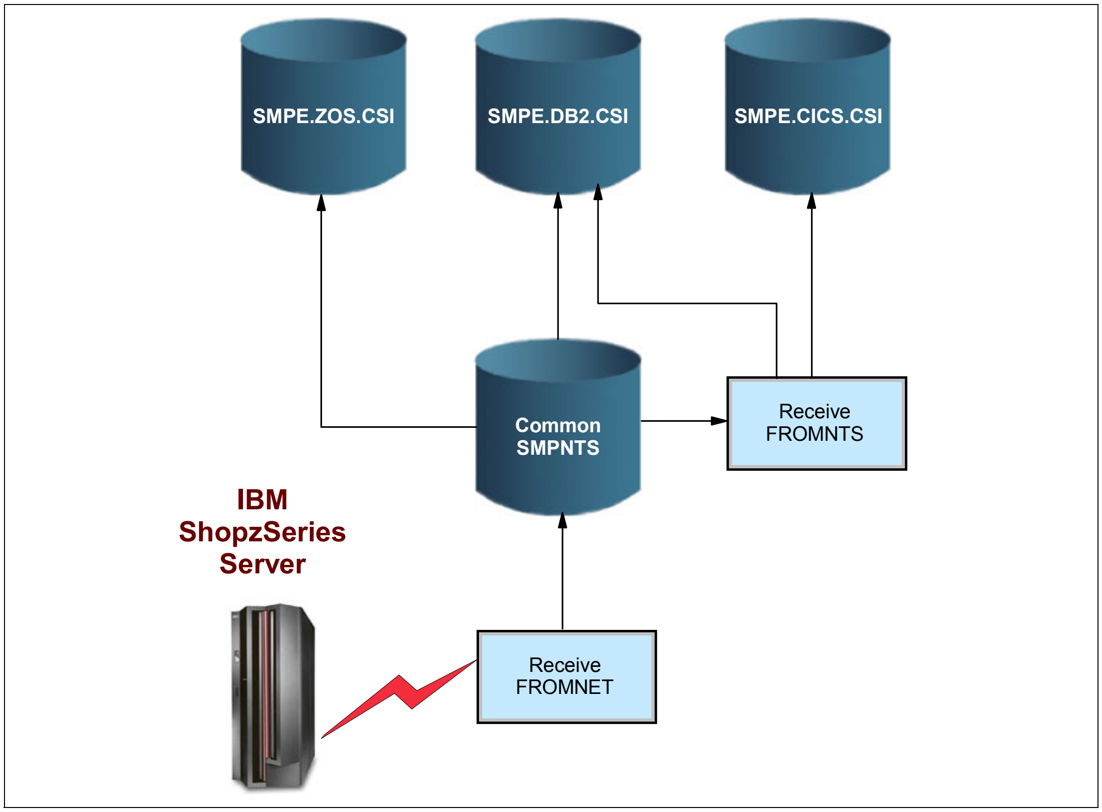


Figure 12-14 Process for receiving an internet order

12.7 SMPCSI allocation sample

When installing individual products, as opposed to installing a system replacement via ServerPac, it is sometimes necessary to create a unique SMP/E environment. To aid in this endeavor, an existing SMP/E sample job, GIMSAMPU, has been modernized to contain steps to create such an environment. The updated sample now contains three steps as follows:

- Allocate SMPCSI data sets for the global, target, and dlib zones. This step also primes the data sets with the GIMZPOOL seed record.
- Allocate the operational data sets required by SMP/E, as follows:
  - SMPPTS (recommend PDSE)
  - SMPMTS
  - SMPSTS
  - SMPLTS
  - SMPSCDS
  - SMPLOG and SMPLOGA (separate LOG and LOGA data sets for each zone)
- To prime the SMPCSI data sets:
  - Define global, target, and dlib zones.
  - Prime with a sample OPTIONS entry.
  - Prime each zone with DDDEF entries for the following:
    - Operational data sets defined in previous step
    - Temporary work data sets
    - Output data sets (SMPOUT, SYSPRINT, and so forth)

12.8 SYSDEFSD (side deck library) DUMMY support

SMP/E V3R2 allows product developers to use a DUMMY data set specification for the SIDE DECK LIBRARY (SYSDEFSD) of DLL load modules.

- Binder exports symbols when link editing DLLs.
  - Exported symbols get defined by IMPORT statements.
  - IMPORT statements are written to SYSDEFSD ddname.
- If SYSDEFSD is not defined, the binder ends with return code 4, which means IMPORT statements could not be written.
- To eliminate the binder return code 4, a SYSDEFSD must be provided if IMPORT statements are not wanted. Developers today provide target libraries for SYSDEFSD, but the generated side deck is not wanted.
- SMP/E allows “dummy” side deck libraries for DLL load modules and supports DD “dummy” for SYSDEFSD.
- When invoking the binder SMP/E allocates SMPDUMMY as a dummy data set and points SYSDEFSD at SMPDUMMY.
- The binder writes IMPORT statements to SMPDUMMY and return code 0 from the binder.

Figure 12-15 shows examples of three ways to define dummy SYSDEFSD statements. The fourth SYSDEFSD statement identifies a real target library.

```
++JCLIN.
//LINK      EXEC PGM=IEBLINK,PARM='DYNAM(DLL)'
//SYSLMOD   DD DSN=SYS1.SABCMOD,DISP=SHR
//SYSDEFSD  DD DUMMY
  -or-
//SYSDEFSD  DD DSN=NULLFILE
  -or-
//SYSDEFSD  DD DSN=SMPDUMMY
  …..............................................................................
//SYSDEFSD  DD DSN=ABC.SABCSDLB,DISP=SHR
//SYSLIN    DD *
  INCLUDE DISTLIB(DLLMOD)
  ENTRY DLLMOD
  NAME DLLLMOD  RC=0
/*
```

Figure 12-15 JCLIN processing specifications for SYSDEFSD

Note: A value of SMPDUMMY is stored in the LMOD entry for each of the three cases.

12.9 Internet package enhancements

Usually archive files are large, which inhibits network transmission and retry attempts. The solution is to make smaller files. This process is called Archive file segmentation.

Archive file segmentation:

- GIMZIP splits large archive files into smaller segments for transmission.
- SMP/E RECEIVE reconstitutes segments later into their original archive.

The archive subdirectory specification does the following:

- Allows archive files to be grouped within an internet package.
- Manufacturing specifies a subdirectory for similar archives.
- GIMZIP creates the subdirectory and stores archives within.

To create an Internet package, use the SMP/E service routine GIMZIP. Figure 12-16 is a sample job to invoke GIMZIP to build such a package. This sample job also shows the following new options:

SEGMENT This option is used to indicate the segment size for archive files.

subdir This option is used to define a subdirectory into which certain archive files are to be stored.

```
//ZIPSTEP  EXEC PGM=GIMZIP,PARM='SEGMENT=100M'
//SYSUT2   DD UNIT=SYSALLDA,SPACE=(CYL,(50,10))
//SYSUT3   DD UNIT=SYSALLDA,SPACE=(CYL,(50,10))
//SYSUT4   DD UNIT=SYSALLDA,SPACE=(CYL,(25,5))
//SMPOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPDIR   DD PATH='/package directory/'
//SYSIN    DD *
  <GIMZIP>
    <FILEDEF name="LOTS.O.PTFS.SMPMCS"
             type="SMPPTFIN">
    </FILEDEF>
    …
    <FILEDEF name="USER.PDO.DOCLIB"
             subdir="DocDir">
    </FILEDEF>
    <FILEDEF name="USER.PDO.PGMDIR"
             subdir="DocDir">
    </FILEDEF>
    …
  </GIMZIP>
/*
```

Figure 12-16 Internet package enhancements with GIMZIP

12.10 Migration actions

To eliminate the need for migration actions, we must separate customization information from the parts supplied by SMP/E. This prevents the regression of a user's changes after installing a replacement release, or even sometimes service.

- Restricting program execution
  - Previously you could restrict which utility programs SMP/E could invoke.
  - Macro GIMDFUT and module GIMUTTBL, previously used to define the allowed programs, have been removed from SMP/E V3R2.
  - You must now use z/OS Security Server (RACF) instead to define profiles for programs in the PROGRAM general resource class to control which user IDs can execute certain programs.

- Customizing settings for SMP/E dialogs

  Previously you could update panel GIM@UPRM to customize certain dialog settings such as:

  - Space information for temporary data set DD statements in generated jobs.
  - Default JOB statements.
  - Unit and volume for temporary data sets used by the dialog.

  You must now use the new SETTINGS option of the SMP/E primary option menu instead. Using the new SETTINGS option means the information is persistent and that you do not need to update panel GIM@UPRM every time a new release of SMP/E is installed.

Note: Macro GIMDFUT, module GIMUTTBL, and panel GIM@UPRM have been removed from SMP/E V3R2.

13

UNIX System Services enhancements in z/OS V1R5

In this chapter we discuss the following enhancements introduced in z/OS V1R5 UNIX System Services:

- Multilevel security enhancements
- Shared file systems in a sysplex
- Enhancements for BPXPRMxx parmlib
- Changed operator commands and TSO/E commands
- Symlink symbolics

13.1 UNIX shells and utilities enhancements

Many of IBM's z/OS customers have requested that historical UNIX functions be made available on z/OS. Therefore, updates were made to several UNIX functions and utilities in this release of z/OS V1R5.

Also, we know that access to z/OS UNIX resources used to be based on POSIX permissions and access control lists (ACLs). To provide additional security it is now possible to perform authorization checks for security labels as well. Using multilevel security enhancements, the customer can:

- Set security labels on UNIX files and directories
- Display and test security labels
- Query, activate, inactivate writedown mode

All together the advantage is enhanced security. In addition, especially for multilevel security, the set of callable services was updated. For a complete list of these callable services, see the *z/OS UNIX System Services Programming: Assembler Callable Services Reference*, SA22-7803.

13.1.1 Multilevel security

UNIX System Services z/OS V1R5 was enhanced with support for multilevel security labels. Multilevel security (MLS) arose from the classified data processing needed by government installations. However, the functions implemented for MLS in this release of z/OS and upcoming enhancements in future releases should be relevant to commercial installations as well.

MLS is a security policy that allows the classification of data and users based on a system of hierarchical security levels combined with a system of non-hierarchical security categories. So in a sense, MLS provides a way to segregate users and their data from other users and their data regardless of access lists, UACC, and so forth.

The two primary goals for a multilevel security policy are:

1. To prevent individuals from accessing information at a higher classification level than their authorization
2. To prevent individuals from declassifying information

How does it work?

z/OS UNIX files and directories can be protected by security labels. The security label for a file or directory is stored in the file security packet (FSP) for the file or directory. z/OS UNIX controls access to files and directories based on security labels in addition to POSIX permissions, access control lists (ACLs) and profiles in the RACF UNIXPRIV class. When the SECLABEL class is active, z/OS UNIX can assign a security label to zFS files and directories when they are created, depending on the security labels of the parent directory and the user.

The security administrator controls each user's access to information by specifying which security labels the user can use. In other words, the user can only access the information in the UNIX directory when the user's security label allows the access.

For more information about multilevel security labels the following documentation is recommended:

- *z/OS V1R5 Security Server RACF Security Administrator's Guide*, SA22-7683
- *z/OS V1R5 Planning for Multilevel Security*, GA22-7509

chlabel command

The new shell command **chlabel** can be used to set security labels for UNIX files and directories. You need RACF SPECIAL authorization to perform this task. It is recommended to run this command *before* multilevel security is activated. Once a SECLABEL has been set it cannot be changed. The format of this command is:

```
chlabel [–cqR] [–h|–L] seclabel pathname
```

By specifying the **-R** parameter, labels are set “recursively” on file subdirectories.

> **Note:** Only the zFS file system supports the setting of seclabels.

Displaying labels

ls -M To display the multilevel security (MLS) label for a file or directory use this command.

id -M On current user address space, use this command.

ipcs -M On inter process communication resources, use this command.

Finding files with labels

To find files that have multilevel security labels, you can use the following command:

```
find dir -seclabel pattern
```

The command used in Figure 13-1 allows you to find files under the current directory with security label “SECRET.”

```
find . -seclabel SECRET
```

Figure 13-1 Find command

Using the command syntax shown in Figure 13-2 will help you to find files under /u directory with *no* security label

```
find /u/ ! -seclabel “*”
```

Figure 13-2 Find command

Testing file labels

Test is a built-in shell command that can be used to compare variables. It checks for various properties of files, strings, and integers, and returns the result of the test. The **test** command is generally used in shell scripts. You can perform the following checks for multilevel security labels:

file **-MI** *seclabel* This one is true if the file has an matching security label.

test **-Ma** *file* True if the file has any security label.

-m *file* Returns security label, or false.

Writedown command

The new command, **writedown** sets or displays the user’s write-down mode for the current address space. For this command, multilevel security has to be activated and the user that is setting the write-down mode must have “writedown” privileges.

```
writedown –a | –d | –i [–p]
```

When activated, it allows the user to write to a resource with lower security label classifications.

Note: Using write-down in a multilevel security environment may declassify information.

To provide for the proper authorization, writedown requires:

- That the invoking user ID has READ access to profile IRR.WRITEDOWN.BYUSER in the FACILITY class.
- SETR MLS, multilevel security has to be activated.

13.1.2 SU login shell

The **su** command can be employed by users who have superuser authority to start a new shell and operate in it with the privileges of a superuser or another user.

```
su [–] [–s][userid [arg ...]]
```

New to z/OS V1R5 is an **su** option that starts a new shell as a login shell.

Using the **su** command as shown in Figure 13-3 starts a child shell with the login environment of the **admin** user ID. This example will make sure you get:

- Admin’s default shell
- Admin’s HOME directory
- Ability to run /etc/profile and admin’s .profile to set the environment variables

```
su admin
```

Figure 13-3 su command

The command shown in Figure 13-4 runs the /usr/lib/backup shell script under the **admin** user ID and returns to the invoker when the shell script ends.

```
su admin /usr/lib/backup
```

Figure 13-4 su command

The command shown in Figure 13-5 runs the remove **rm** shell command under the **admin** user ID and returns to the invoker when the command ends.

```
su admin -c “rm -rf /tmp/”
```

Figure 13-5 su command

Surrogat RACF facility class

Not entirely new to z/OS UNIX, but still very useful to know about, are the following two options of the **su** command:

- Start the new shell as a login shell. Set the shell variables SHELL, HOME, and LOGNAME according to the new user’s profile and append a “-” to the shell to indicate that the shell should read it’s login profiles.

-s Does not prompt for password if a user ID is specified and you do not have read access to the SURROGAT facility class profile, BPX.SRV.uuuuuuuu (where uuuuuuuu is the MVS user ID associated with the target UID).

Figure 13-6 shows what happens if someone tries to switch to a user ID with the **-s** option.

```
PATRICK @ SC64:/u/patrick>su -s naidoo
FSUM5027 su: User is not a surrogate of "naidoo".
```

Figure 13-6 Switching a user ID with the su command

To authorize a user to switch to another user without entering a password, grant them RACF SURROGAT authority as shown in Figure 13-7.

```
RDEFINE SURROGAT BPX.SRV.NAIDOO UACC(NONE)
PERMIT BPX.SRV.NAIDOO CLASS(SURROGAT) ID(PATRICK) ACCESS(READ)
SETROPTS RACLIST(SURROGAT) REFRESH
```

Figure 13-7 Granting SURROGAT authority to a user

The example show in Figure 13-7 gives user ID PATRICK authorization to switch to user ID NAIDOO without entering a password.

Tip: It can be very useful to use the **man** command to display help information about a shell command. It works from within the z/OS UNIX shell, but before you can use the man pages (help files), you have to enable them.

13.1.3 IPCS command

New functions were added to the **ipcs** command in this release of z/OS V1R5. The shell command **ipcs** displays status information about active inter-process communication resources.

Using the new options the user can display active __map memory segment information. One of the advantages of this is that it enables debugging of server applications.

The new options for **ipcs** are:

-S Shows information for each __map memory area:
Creator’s pid
User pid
User name
Group name
Shutdown status

-B Shows additional __map memory details:
Block size
blocks in use
Blocks in the __map object
Blocks mapped by this process

See also “Displaying labels” on page 321 for **ipcs** enhancements to support multilevel security.

13.2 BPXPRMxx parmlib enhancements

For this release of UNIX System Services two enhancements were made to the BPXPRMxx parmlib member. The BPXPRMxx member contains customization values for z/OS UNIX

System Services. To review the latest enhancements, browse your local BPXPRMxx member in SYS1.SAMPLIB and search for FMID HBB7708.

13.2.1 SWA above

In general, when a task is initiated there are control blocks created that contain information about the job and its job steps. These control blocks reside in the Scheduler Work Area (SWA) within the user’s address space until task termination.

The SWA control blocks for the UNIX System Services address space are by default allocated below the 16 megabyte line. Having the control blocks allocated below the 16 megabyte line can cause storage constraints when a large number for file systems are mounted.

To be able to resolve this storage constraint, a new BPXPRMxx parmlib parameter was added to z/OS V1R5. It is now possible to specify where the SWA control blocks are allocated, above or below the 16 megabyte line. The new BPXPRMxx parmlib parameter you can use to put the SWA above the 16 megabyte line is:

```
SWA(ABOVE)
```

To put the SWA below the 16 megabyte line, is the same as the default:

```
SWA(BELOW)
```

Use the following operator command to display your current setting:

```
D OMVS,O
```

Note: Be aware that changes to this SWA setting in your system’s BPXPRMxx is only available when starting OMVS during system initialization.

13.2.2 MKDIR() statement for parmlib member

A new **MKDIR()** statement was added that can be used on either the ROOT or MOUNT statements within the BPXPRMxx parmlib member. This allows you to specify a directory, or mountpoint, which is to be created during parmlib processing. Parmlib processing is supported during z/OS UNIX System Services initialization.

This option can be very useful for those of you who have experience with failed parmlib mounts, because the mountpoint did not preexist. This enables you to specify multiple MKDIR statements on each of the ROOT or MOUNT statements. So now it is possible to create one or more directory entries in the file system associated with the ROOT or MOUNT parameter, or to create directory entries in another file system that is already mounted.

```
MOUNT FILESYSTEM('OMVS.&SYSNAME..CLUBS')
      MOUNTPOINT('/u')
      TYPE(HFS)  MODE(RDWR)
      MKDIR(‘ajax’)

MOUNT FILESYSTEM('OMVS.&SYSNAME..AJAX')
      MOUNTPOINT('/u/ajax')
      TYPE(HFS)  MODE(RDWR)
      MKDIR(‘players’)
      MKDIR(‘sponsors’)
```

Figure 13-8 MKDIR use in BPXPRMxx

With this MKDIR support it is important to know that permissions are set to 755 (-rwxr-xr-x). Also it should not be used with file systems that mount asynchronously, like NFS Clients.

Note: Make sure that the total length of the MKDIR and its mountpoint is less than 1023 characters in total. If sharing parmlib members between shared HFS members is being used, this MKDIR() statement should be omitted unless all are running at V1R5 or above.

13.3 Changed operator commands

Some of the operator commands that affect z/OS UNIX where changed to provide for additional support. In this section you will find a description of those changes.

13.3.1 SETOMVS SYNTAXCHECK

Parmlib mounts can fail if the HFS data set or zFS aggregate to be mounted does not exist. In this release of z/OS V1R5, a new keyword was added to the **SETOMVS** operator command. This enhancement allows you to identify parmlib statements whose data set or compatibility mode aggregate does not exist.

To use this enhancement just execute the following operator command:

```
SETOMVS SYNTAXCHECK=(xx)
```

Where **xx** stands for the BPXPRM suffix.

If an error exists in the syntax of your BPXPRMxx member, you will receive the following message after you issue the command:

```
SETOMVS SYNTAXCHECK=(GU)
RESPONSE=SC64
 BPXO023I THE PARMLIB MEMBER BPXPRMGU CONTAINS SYNTAX ERRORS.
 REFER TO HARD COPY LOG FOR MESSAGES.
```

13.3.2 Avoid wait-state in shared HFS

It is very important for a shared HFS configuration to have compatible software levels on all systems of the Shared HFS group. Prior to release V1R5 of z/OS, it could happen that UNIX System Services got into a wait-state on a system with incompatible software level. The behavior of UNIX System Services is modified to provide a solution for this problem.

Therefore, there are two new messages added to z/OS V1R5.

The message shown in Figure 13-9 appears in the log when z/OS UNIX is configured with Shared HFS support and cannot initialize due to a software service incompatibility between this system and another active system in the Shared HFS configuration.

.

```
BPXF079S - UNIX SYSTEM SERVICES CANNOT EXECUTE IN THE ACTIVE SHARED HFS CONFIGURATION.
THE SOFTWARE SERVICE LEVEL OF ONE OR MORE SYSTEMS IS INCOMPATIBLE WITH THIS SYSTEM.
```

Figure 13-9 Message BPXF079S

The message shown in Figure 13-10 on page 326 is issued in conjunction with message BPXXF079S. It lists the systems that are configured for z/OS UNIX Shared HFS support and are executing at a software service level that is incompatible with the software service level of this system.

.

```
BPXF080I - THE SOFTWARE SERVICE LEVEL OF THE FOLLOWING SYSTEMS ARE INCOMPATIBLE WITH
THIS SYSTEM:
```

Figure 13-10 Message BPXF080I

z/OS UNIX System Services has been modified to disable itself using the following command:

```
F OMVS,SHUTDOWN
```

This change in the behavior of UNIX System Services avoids disabling the whole system. Once UNIX System Services is disabled, you are given time to apply the correct maintenance to that system.

Note: Be sure to check the list of z/OS UNIX coexistence and fallback PTFs that must be applied on each system that is configured with Shared HFS support.

13.4 TSO/E commands

This section describes the new or changed TSO/E commands that you can use to work with the z/OS UNIX System Services file system.

13.4.1 Remount for Shared HFS

The **UNMOUNT** TSO/E command can be used to remove a file system from the file hierarchy. The syntax for this command is as follows:

```
UNMOUNT FILESYSTEM(file_system_name)
DRAIN | FORCE | IMMEDIATE | NORMAL | REMOUNT(RDWR | READ) | RESET
```

One of the parameters of this command is REMOUNT. Using this parameter it is possible to specify that this file system is to be remounted and that its mount mode is to be changed. REMOUNT takes an optional argument of RDRW or READ. If no argument is specified with the REMOUNT parameter, the mount mode is flip-flopped.

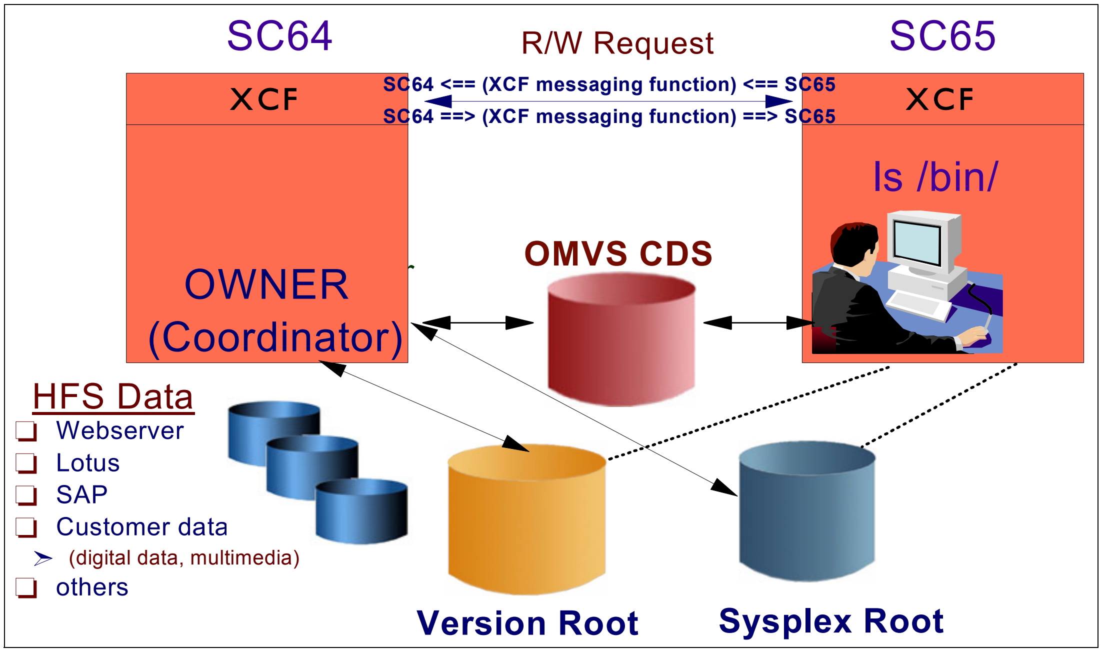
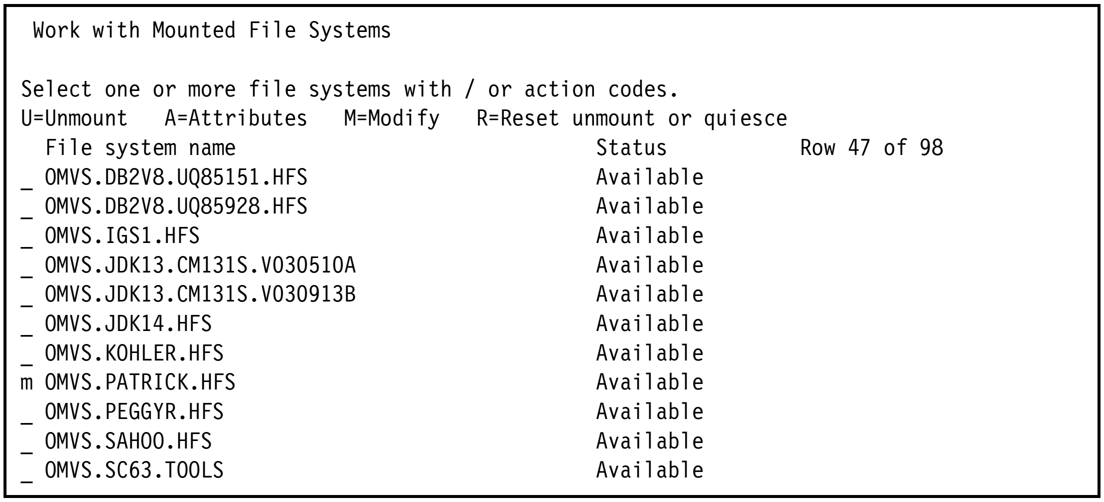


Figure 13-11 Accessing shared sysplex file systems

New in this release of z/OS V1R5 is that REMOUNT is now supported in a sysplex environment. To be able to use this support for your complete Shared HFS group, all systems must be running at V1R5 or higher, or on V1R4 with APAR OA02584.

If one of the members is down level, `errno EINVAL` and `errnoJr JrNotSupInSysplex` will be returned when remount is requested. There are several ways to use the remount function:

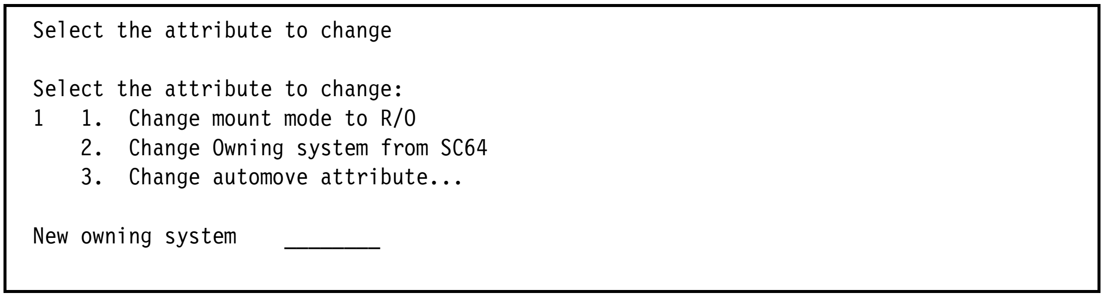
- From the BPX1UMT callable service using the **`MtmRemount`** flag
- From TSO using: `unmount filesystem(fsname) remount(rdwr | read)`
- From the shell using new options on the **`chmount`** command:
  - **`-r`** to switch mounted file system to R/O mode
  - **`-w`** to switch mounted file system to R/W mode
- From the ISHELL mount table display under file systems (see Figure 13-12)



```
  Work with Mounted File Systems

Select one or more file systems with / or action codes.
U=Unmount   A=Attributes   M=Modify   R=Reset unmount or quiesce
  File system name                              Status          Row 47 of 98
_ OMVS.DB2V8.UQ85151.HFS                        Available
_ OMVS.DB2V8.UQ85928.HFS                        Available
_ OMVS.IGS1.HFS                                 Available
_ OMVS.JDK13.CM131S.V030510A                    Available
_ OMVS.JDK13.CM131S.V030913B                    Available
_ OMVS.JDK14.HFS                                Available
_ OMVS.KOHLER.HFS                               Available
m OMVS.PATRICK.HFS                              Available
_ OMVS.PEGGYR.HFS                               Available
_ OMVS.SAHOO.HFS                                Available
_ OMVS.SC63.TOOLS                               Available
```

Figure 13-12 ISHELL BPXWP20 panel

Type an **m** in front of the HFS data set that you want to modify and press Enter. This will bring you into a panel where you can change the mount mode of your designated file system.



```
Select the attribute to change

Select the attribute to change:
1   1.  Change mount mode to R/O
    2.  Change Owning system from SC64
    3.  Change automove attribute...

New owning system     ________
```

Figure 13-13 ISHELL BPXWP60 panel

The main use of **remount** is to switch an R/O file system to R/W for maintenance. This way it is possible to apply services to the R/O file system without unmounting the file system and mounting it R/W. Applications reading from an R/O file system can continue to do so without disruption while the file system is remounted as R/W, and subsequently remounted to R/O.

Note: To collect information about the activity of mounted file systems, it is possible to use SMF record type 92. You must collect SMF record type 92 subtype 6 to get information on each time a file system is being remounted. This occurs on each system in the Shared HFS group.

zFS remount considerations

There are two considerations regarding the use of remount with zFS:

- If zFS HFS-compatibility aggregates are involved, and if both the primary file system and its clone are mounted, remount will fail because we cannot detach the aggregate. Sysplex remount will be rejected with `errno EINVAL` and `errnoJr JrAggregateErr`.
- If zFS multiple file systems aggregates are involved, and if the aggregate is attached R/O, then file systems in it can only be mounted R/O and remount to R/W will result in a zFS error.

13.5 Symlink symbolics

This support in z/OS V1R5 creates a way to specify mountpoints, using symlink symbolics, such that they resolve on systems based on system symbols, which may or may not be different between systems in a sysplex.

Support is added to allow use of static system symbols in symlinks with an identifier indicating substitution is necessary in the resolution of the symlink. This allows for unique pathname resolution based on the value of the system symbol on a particular system. Thus, it allows for mountpoints that you want to share with a subset of systems in the Shared HFS group.

The two identifiers that are created are:

- $SYSSYMA/ - Absolute symlink specification
- $SYSSYMR/ - Relative symlink specification

Note: Text must follow a $SYSSYMR/ or $SYSSYMA/ in order for it to be recognized as a valid identifier with text containing symbols to be resolved. However, only static system symbols should be used in order to avoid unexpected results.

Like $VERSION/ and $SYSNAME/, the identifiers need to be at the beginning of the link name. Only the first occurrence of $SYSSYMR/ or $SYSSYMA/ in the link name is recognized as an identifier for which the remaining text requires substitutions. Any other identifiers after the first one will remain as is in the resolved linkname.

13.5.1 Create system symbols and symlinks

To create the system symbols, use the IEASYSxx parmlib member on each system, as follows:

- IEASYMxx on SC65
  - SYMDEF(&DB2='DB2V7'
- IEASYMxx on SC70
  - SYMDEF(&DB2='DB2V8'

To define the symlinks, you can either use the ISHELL, or from the OMVS command line using the **ln** command, as follows:

- `ln -s '$SYSSYMA/pp/&DB2.' /usr/lpp/db2`
- `ln -s '$SYSSYMR/&DB2.' /pp/db2`

> **Note:** Quotes are needed because otherwise $SYSSYMA/ and $SYSSYMR/ would be interpreted as UNIX environment variables.

13.5.2 Examples using $SYSSYMA/ and $SYSSYMR/

Figure 13-15 on page 330 and Figure 13-17 on page 331 are examples of the $SYSSYMA/ and $SYSSYMR/ identifiers that indicate substitution is necessary in the resolution of the symlink. In the examples of the two systems in a shared HFS sysplex, there are systems SC65 and SC70.

> **Note:** For each symlink symbolics you create, choose either the $SYSSYMA/ or $SYSSYMR/ identifier, or for convenience purposes you can define both symlinks.

In this example it is desired to have the following file systems on SC65 and SC70:

- A file system for DB2 V7 is mounted at the directory DB2V7 on system SC65
- A file system for DB2 V8 is mounted at the directory DB2V8 on system SC70

> **Note:** No matter which identifier you choose, either the $SYSSYMA or $SYSSYMR, the result for the mountpoint will be the same.

When each system is IPLed, in the BPXPRMxx member, there is a MOUNT statement to mount the DB2 file system needed on that system, as shown in Figure 13-14 on page 330 and Figure 13-16 on page 330. The MOUNTPOINT in the MOUNT statement is the symlink. When the BPXPRMxx member is processed on each system, it results in a mount of the file system at the following mount points:

- Resolves to: /pp/DB2V7 - on system SC65
- Resolves to: /pp/DB2V8 - on system SC70

Example 1: $SYSSYMA/ symlink mount

For a user to access the file system for DB2, the user specifies the directory as follows:

```
/usr/lpp/db2
```

```
MOUNT FILESYSTEM('OMVS.&DB2..ZFS')
      MOUNTPOINT('/usr/lpp/db2')
      AUTOMOVE TYPE(ZFS) MODE(RDWR)
```

Figure 13-14 MOUNT statement in the BPXPRMxx parmlib member for system SC65

The file system that is mounted on system SC65 is OMVS.DB2V7.ZFS.

The file system that is mounted on system SC70 is OMVS.DB2V8.ZFS.

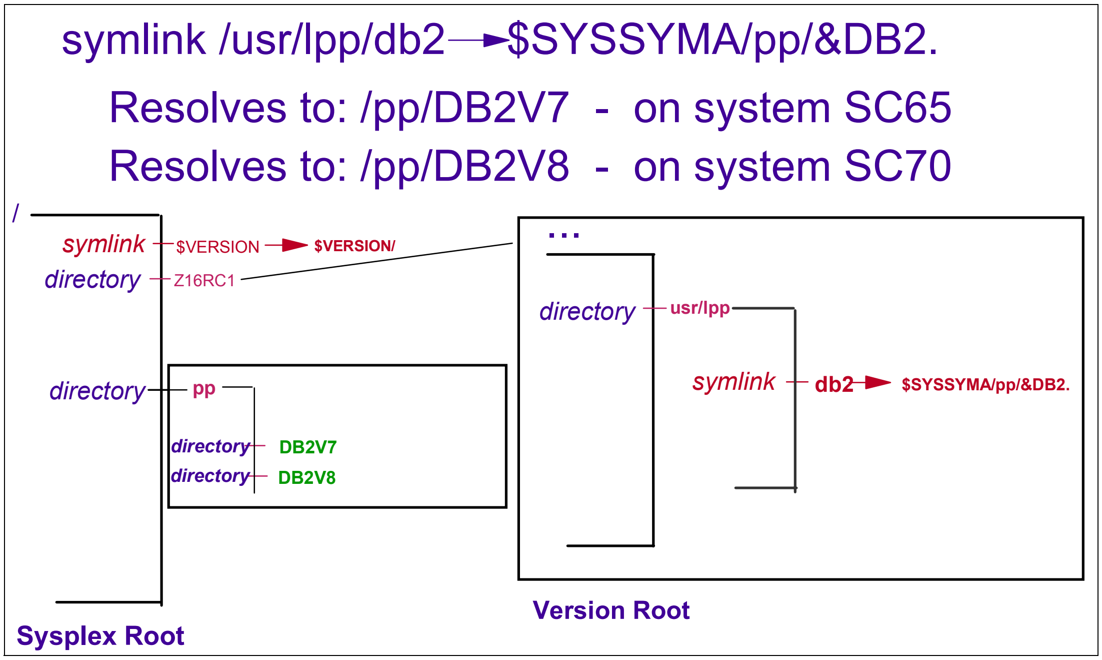


Figure 13-15 $SYSSYMA example and its symlink resolution

Example 2: $SYSSYMR/ symlink mount

For a user to access the file system for DB2, the user specifies the directory as follows:

```
/pp/db2
```

```
MOUNT FILESYSTEM('OMVS.&DB2..ZFS')
      MOUNTPOINT('/pp/db2')
      AUTOMOVE TYPE(ZFS) MODE(RDWR)
```

Figure 13-16 MOUNT statement in the BPXPRMxx parmlib member for system SC70

The file system that is mounted on system SC65 is OMVS.DB2V7.ZFS.

The file system that is mounted on system SC70 is OMVS.DB2V8.ZFS.

symlink /pp/db2 → $SYSSYMR/&DB2.

Resolves to: /pp/DB2V7 - on system SC65

Resolves to: /pp/DB2V8 - on system SC70

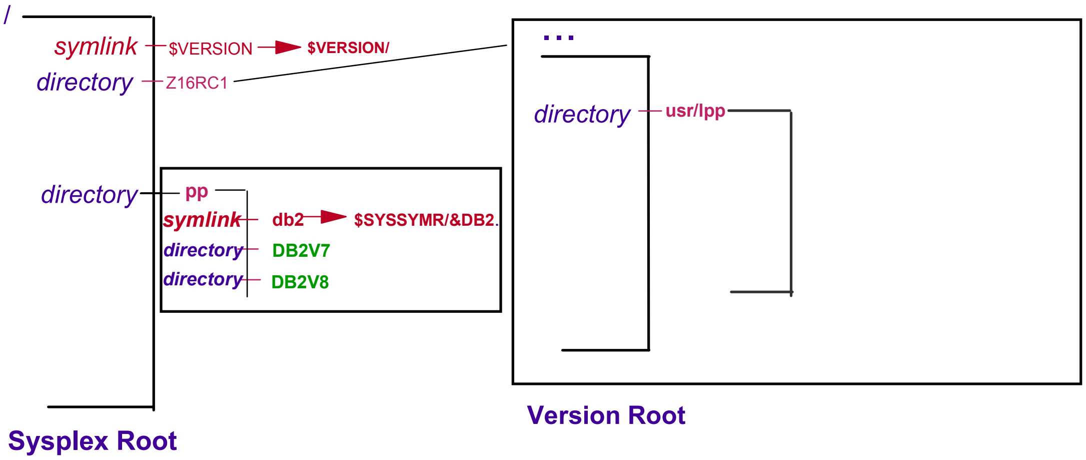


Figure 13-17 $SYSSYMR example and its symlink resolution

14

z/OS Security Server RACF

This chapter contains the following enhancements to the new version of the Security Server:

- Multilevel security
- SECLABELs and Mandatory Access Control (MAC)
- SECLABELs for z/OS UNIX processes and sockets
- SECLABELs for z/OS UNIX files and directories
- SECLABELs for z/OS UNIX Inter Process Communication (IPC)
- SECLABEL by system
- WRITE-DOWN by user privilege
- Name Hiding
- RACROUTE enhancements
- Miscellaneous enhancements
- New and changed messages
- Overview of the RACF templates
- Dynamic templates objectives
- Dynamic templates externals
- New messages for dynamic template support
- RACF support of DB2 V8
- RACF and LDAP server change logging
- RACF and password enveloping
- RACF and LDAP event notification
- Enterprise identity mapping (EIM)
- PKI Services architecture

14.1 Multilevel security

Multilevel security is a security policy that allows the classification of data and users based on a system of hierarchical security levels combined with a system of non-hierarchical security categories. A multilevel-secure security policy has two primary goals, as follows:

- First, the controls must prevent unauthorized individuals from accessing information at a higher classification than their authorization.
- Second, the controls must prevent individuals from declassifying information.

14.1.1 Previous multilevel security history

Between 1988 and 1990 IBM enhanced MVS, RACF, JES2, JES3, TSO, VTAM, DFP, and PSF to meet the B1 criteria. MVS/ESA Version 3 Release 1 Modification Level 3 passed the formal evaluation performed by the National Security Agency and obtained a B1 security designation. For several years, subsequent versions of RACF, MVS/ESA, and OS/390 were designed to continue to meet the B1 criteria, although no formal evaluations were done. But over time new functions such as UNIX System Services were added to MVS that could not be used on a system with a B1 security designation. At the same time, customer configurations evolved to require networking, which could not be used on a B1 system.

The security policy that can be implemented in a multilevel-secure environment is made of the following important features:

- Access controls
  - Mandatory access control (MAC)
  - Discretionary access control (DAC)
- Accountability
  - Identification and authentication
  - Auditing

OS/390 added new functions such as OS/390 UNIX, Extended MCS consoles (EMCS), and TCP/IP, which did not meet the B1 expectations. With z/OS V1R5, B1 support is extended to cover these functions.

14.2 Mandatory access control (MAC)

Mandatory access control is a method of limiting access to resources based on the sensitivity of the information that the resource contains and the authorization of the user to access information with that level of sensitivity. This type of access control is mandatory in the sense that subjects cannot control or bypass it.

The security administrator (the user with the RACF SPECIAL attribute) defines the sensitivity of each object by means of a security label (SECLABEL). The security administrator controls each subject's access to information by specifying which security labels the subject can use. A subject can access information in an object only when the subject's security label entitles the access. If the subject's security label does not have enough authority, the subject cannot access the information in the object.

14.2.1 Dominance and MAC

Mandatory access control is based on the theory of dominance, and is achieved through the use of security labels. The mandatory access checking occurs first, then the discretionary access checking (DAC) takes place. If the SECLABEL class is inactive, then only DAC processes the resource access control.

MAC checking

A mandatory access check compares the security labels of the subject and object and grants the subject access to the object as follows:

- A subject can read an object if the subject's security label dominates the object's security label.
- A subject can write to an object if the object's security label dominates the subject's security label. A subject cannot write to an object whose security label the subject's security label dominates, unless the security labels are equivalent - we say that the subject is not allowed to “write down.”
- A subject can both read and write an object only if the subject's and object's security labels are equivalent.

Dominance

One security label dominates a second security label when the following two conditions are true:

- The security level that defines the first security label is greater than or equal to the security level that defines the second security label.
- The set of security categories that define the first security label includes the set of security categories that defines the second security label.

Two security labels are said to be “disjoint” or “incompatible” if neither dominates the other because they have incompatible sets of categories. For example, if security label A has categories apples and pears, and security label B has categories pears and bananas, neither contains all the categories of the other and so neither dominates the other and they are disjoint.

14.3 Discretionary access control (DAC)

Once the user passes the mandatory access check, a discretionary check follows. The discretionary access check ensures that the user is identified as having a “need to know” for the requested resource. The discretionary access check uses other access control information, such as the access control list in the profile protecting a resource, or z/OS UNIX access control (permissions, the access control list, and the UNIXPRIV class).

Discretionary access control is the principle of restricting access to objects based on the identity of the subject (the user or the group to which the user belongs). Discretionary access control is implemented using access control lists. A resource profile contains an access control list that identifies the users who can access the resource and the authority (such as READ or UPDATE) the user is allowed in referencing the resource.

The security administrator defines a profile for each object (a resource or group of resources), and updates the access control list for the profile. This type of control is discretionary in the sense that subjects can manipulate it, because the owner of a resource, in addition to the security administrator, can identify who can access the resource and with what authority.

14.4 Security labels

Security labels can be associated with all users and resources in the system. The system uses these labels to determine if access to a resource is allowed under the mandatory access control (MAC) rules. Security labels are maintained in the RACF database, are usually defined by the security administrator, and can be only changed by that person.

The security label is composed of the following:

- A security level

  The security level indicates a level or hierarchical classification of the information.

- Zero or more security categories

  The security category defines the category or group to which the information belongs.

Users can access only the information in a resource to which their security labels entitle them. If the user's security label does not have enough authority, the user cannot access the information in the resource.

14.4.1 Activating the SECLABEL class

When the SECLABEL class is active, unless the security administrator has also set certain system options, a user needs a security label only if the resource the user wants to access has a security label, and resources are not required to have security labels. To increase security, the security administrator can use the **SETROPTS** command to set RACF system options that require that certain resources have security labels, and require that any user who tries to access these resources has a security label.

To activate the SECLABEL class and SECDATA class, the security administrator issues the following commands:

```
SETROPTS CLASSACT(SECLABEL) RACLIST(SECLABEL)
SETROPTS CLASSACT(SECDATA) RACLIST(SECDATA)
```

SECLABELs and MAC

MAC uses the SECLABEL to grant or deny access to the resource. The security label indicates the hierarchical level or classification of the information (such as top secret, secret, sensitive), and indicates to which non-hierarchical category the information belongs within that level (such as PROJECTA, PROJECTB).

As shown in Table 14-1 on page 337, a system might have three security levels (unclassified, sensitive, and secret), and three security categories (PROJECTA, PROJECTB, and PROJECTC). Then, as follows:

- PURPLE could be a SECLABEL name indicating SECLEVEL secret for categories PROJECTA, PROJECTB, and PROJECTC.
- GREEN could be a SECLABEL name indicating SECLEVEL sensitive for categories PROJECTA and PROJECTB.
- RED could be a SECLABEL name indicating SECLEVEL unclassified for category PROJECTC.

Table 14-1 Example of SECLABEL, SECLEVEL, and categories

SECLABEL	SECLEVEL	Category
PURPLE	SECRET	PROJECTA, PROJECTB, PROJECTC
GREEN	SENSITIVE	PROJECTA, PROJECTB
RED	UNCLASSIFIED	PROJECTC

SECLABEL dominance

Using the example in Table 14-1, security label dominance is as follows:

- PURPLE dominates GREEN because the hierarchical security level SECRET is greater than the security level SENSITIVE and PURPLE's non-hierarchical categories include all of GREEN's categories.
- PURPLE also dominates RED because the hierarchical security level SECRET is greater than the security level UNCLASSIFIED and PURPLE's non-hierarchical categories include all of RED's categories.
- GREEN does not dominate RED. It is true that GREEN's hierarchical security level is greater than the security level of RED. However, GREEN's non-hierarchical categories (PROJECTA, PROJECTB) do not include all categories for RED (PROJECTC).

Resource class definitions

The security administrator defines two profiles in the RACF SECDATA resource class that specify the security levels and security categories for the system described in Table 14-1. The profiles are defined as follows:

- The SECLEVEL profile contains a member for each hierarchical security level in the system. SECRET, SENSITIVE, and UNCLASSIFIED are examples of levels you could define. You might define SECRET to be a security level of 150, SENSITIVE to be a level of 75, and UNCLASSIFIED to be a level of 10. A security administrator can define up to 254 security levels.

  ```
  RDEFINE SECDATA SECLEVEL ADDMEM(UNCLASSIFIED/10 SENSITIVE/75 SECRET/150)
  ```

- The CATEGORY profile contains a member for each non-hierarchical category in the system.The security administrator can define zero or more categories that correspond to some grouping arrangement in the installation. PROJECTA, PROJECTB, and PROJECTC could all be categories defined.

  ```
  RDEFINE SECDATA CATEGORY ADDMEM(PROJECTA, PROJECTB, PROJECTC)
  ```

14.4.2 Defining SECLABELs

After defining the SECLEVEL and CATEGORY profiles, the security administrator defines a profile in the SECLABEL resource class for each security label, as follows:

- The security label is a name of up to eight uppercase alphanumeric or national characters. Each security label name must be unique.
- Each SECLABEL profile specifies the particular combination of a SECLEVEL member and zero or more members of the CATEGORY profile that applies to the security label.

You do not need to define a security label for every possible combination of level and category. Examples of SECLABEL profiles are:

```
RDEFINE SECLABEL PURPLE SECLEVEL(SECRET) ADDCATEGORY(PROJECTA PROJECTB
PROJECTC)
PERMIT PURPLE CLASS(SECLABEL) ACCESS(READ) ID(HARRY JANE TOM)
```

Note: When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a resource, RACF compares the list of security categories in the user's profile with the list of security categories in the resource profile. If RACF finds any security category in the resource profile that is not in the user's profile, RACF denies access to the resource. If the user's profile contains all the required security categories, RACF continues with other authorization checking.

14.5 Data protection in a multilevel-secure system

To ensure that a user does not declassify data, a subject can read from and write to (alter) only an object with an equivalent security label; however, a subject can copy information from an object having a security label that does not dominate the security label of the subject to an object having the subject's current security label. Or, for objects that support write-only processing, such as z/OS UNIX files, to an object with a security label that dominates the subject's security label.

The term “user” usually has the same meaning as the term “subject,” but sometimes implies a human subject. In this redbook, unless stated otherwise, the terms user and subject are used interchangeably.

A subject is an entity that requires access to system resources. Examples of subjects are:

- Human users
- Started procedures
- Batch jobs
- z/OS UNIX daemons

An object is a system resource to which access must be controlled. Examples of objects are:

- Data sets
- z/OS UNIX files and directories
- Commands
- Terminals
- Printers
- DASD volumes
- Tapes

There are two types of write-down access control:

- Write-down with MLS active

  To prevent users from accessing data that they are not authorized to see, a multilevel-secure system generally requires that in order to read data a user must have a security label that represents a level of security at least as high as the data's level of security. (This statement is a generalization of the concept of “dominance,” which is discussed in “Dominance and MAC” on page 334.)

- Write-down by user privilege

  The write-down by user privilege is a feature that controls the declassification of the data. To allow for controlled situations of write-down, z/OS allows the security administrator to assign a write-down by user privilege to individual users that allows those users to select the ability to write down.

14.5.1 Write-down with MLS active

To prevent users from declassifying data, a multilevel-secure system requires that in order to write data, the data must have a security label that represents a level of security at least as high as the user's level of security; we say that the user cannot “write down.” For example, if “Top Secret” is a higher level of security than “Secret,” a user whose security label gives authorization to Top Secret data cannot be allowed to write Top Secret data into a data set with a security label that classifies the data as Secret. For more information about this privilege, see “Write-down by user privilege” on page 340.

RACF MLS option

A subject is not allowed to write down if the RACF MLS option is active and the security administrator has not set up controlled write-down privilege (see “Write-down by user privilege” on page 340) and given the subject authorization to write down. This should be the case for most users in a multilevel-secure environment. Use the MLS option to prevent users from having the capability to write down. The MLS option helps prevent declassification of data.

```
SETROPTS CLASSACT(MLS)
```

The system does not allow a user to declassify data by writing down (that is, writing data to a lower classification than the classification at which it was read) except with explicit authorization to do so.

To ensure that a user does not declassify data the subject can do the following:

- A subject can read from and write to (alter) only an object with an equivalent security label; however:
  - A subject can copy information from an object having a security label that does not dominate the security label of the subject to an object having the subject's current security label.
  - A subject can copy information for objects that support write-only processing, such as z/OS UNIX files, to an object with a security label that dominates the subject's security label.

Data protection example

To illustrate what the subject can do, consider the following example. Based on the example shown in Table 14-1, assuming that the subject passes the discretionary access check, and considering the rules shown in “Access rules with MLS option active” on page 340, the following actions (summarized in Table 14-2 on page 340) can take place between subjects and objects with the assigned security labels:

- SECLABEL PURPLE objects
  - PURPLE can write to PURPLE.
  - PURPLE can read GREEN and copy it to PURPLE.
  - PURPLE can read RED and copy it to PURPLE.
- SECLABEL GREEN objects
  - GREEN can write to GREEN - For z/OS UNIX files and directories, which support write-only processing.
  - GREEN can also write to PURPLE.
- SECLABEL RED objects
  - RED can write to RED - For z/OS UNIX files and directories, which support write-only processing.
  - RED can also write to PURPLE.

Table 14-2 Data protection example

	PROJECT A	PROJECT B	PROJECT C
SECRET	SECLABEL = PURPLE		
SENSITIVE	SECLABEL = GREEN		No label defined
UNCLASSIFIED	No label defined	No label defined	SECLABEL=RED

Access rules with MLS option active

Access rules depend on the purpose for which a subject accesses an object and whether the subject is allowed to write down. A subject is not allowed to write down if the RACF MLS option is active and the security administrator has not set up controlled write-down and given the subject authorization to write down. This should be the case for most users in a multilevel-secure environment. In this case, the access rules, depending on the purpose for which the subject accesses an object, are:

Read only A subject can read an object when the subject's security label dominates the object's security label.

Write only A subject can write to an object when the object's security label dominates the subject's security label.

Read/Write A subject can read from and write to an object only if the security labels of the subject and object are equivalent.

14.5.2 Write-down by user privilege

The write-down by user privilege is a feature that controls the declassification of the data. There might be cases where you want to allow for controlled situations of write-down. The security administrator can assign a “write-down by user” privilege to individual users or groups of users that allows them to select the ability to write down. The security administrator activates and deactivates the privilege by creating the profile IRR.WRITEDOWN.BYUSER in the FACILITY class. A user can activate write-down mode if the profile exists, the user has at least READ access to it, and the FACILITY class is active and `SETROPTS` RACLISTed. If the user has UPDATE or higher access to the profile, write-down mode is active by default when the user enters the system. RACF provides the `RACPRIV` command, and z/OS UNIX provides the `writedown` command, that allow users who are authorized to the write-down privilege to reset and query the setting of their write-down mode.

RACPRIV command

Use the `RACPRIV` command to allow users, who are authorized to the profile IRR.WRITEDOWN.BYUSER in the FACILITY class, to set, reset, and query the setting of the write-down privilege that they are running within their address space. This command ends with an error message if write-down by user is not active on the system. To activate write-down by user, the profile IRR.WRITEDOWN.BYUSER must be defined in the FACILITY class, the FACILITY class must be active and RACLISTed, and the SETR MLS option must be active.

Note: The `RACPRIV` command must be issued from the TSO environment.

Access rules with write-down privilege

A subject is allowed to write down under the following circumstances:

- If the RACF MLS option is not active.

- If the RACF MLS option is active and the security administrator has set up controlled write-down and given the subject the write-down privilege, and the subject has activated write-down mode.

In this case, the access rules, depending on the purpose for which the subject accesses an object, are:

Read only A subject can read an object when the subject's security label dominates the object's security label.

Write only A subject can write to an object when the object's security label dominates the subject's security label, or when the subject's security label dominates the object's security label.

Read/Write A subject can read from and write to an object only if the subject's security label dominates the object's security label.

14.5.3 Reverse and equal mandatory access checking

So far we have discussed dominance checking for data sets, z/OS UNIX files and directories, and the majority of RACF general resource classes. However, for some RACF general resource classes dominance checking works differently, involving either a reversed check (reverse mandatory access checking) or a strict equality check (equal mandatory access checking.

Reverse mandatory access checking

With reverse mandatory access checking, access rules are the reverse of the access rules for mandatory access checking. We have to take in consideration two cases: when the subject is *not* allowed to write down, and when the subject *is* allowed to write down.

Reverse mandatory access checking applies to resources in the RACF classes as follows:

- APPCPORT
- CONSOLE
- WRITER

When the user is not allowed to write down

Read only A subject can read an object when the object's security label dominates the subject's security label.

Write only A subject can write to an object when the subject's security label dominates the object's security label.

Read/Write A subject can read from and write to an object only if the security labels of the subject and object are equal.

When the user is allowed to write down

Read only A subject can read an object when the object's security label dominates the subject's security label.

Write only A subject can write to an object if the security label of the subject dominates the security label of the object, or the security label of the object dominates the security label of the subject.

Read/Write A subject can read from and write to an object when the object's security label dominates the subject's security label.

Equal mandatory access checking

With equal mandatory access checking the security label of the user must be equivalent to the security label of the resource. Equivalence of security labels means that either the security labels have the same name, or they have different names but are defined with the same security level and identical security categories. The security label SYSMULTI is considered equivalent to any security label.

Equal mandatory access checking applies to resources in the following classes:

- APPL
- DSNR
- JESINPUT
- MQCONN
- SERVAUTH
- SERVER
- TERMINAL

Equal mandatory access checking is used for any class where two-way communication is expected.

The type of mandatory access check that is done for a resource depends on the definition of the class to which the resource belongs. The RACF class descriptor table contains the definitions of the resource classes for the system. The system programmer or security administrator can define additional classes using the class descriptor table macro, ICHERCDE. The RVRSMAC and EQUALMAC operands specify that reverse or equal mandatory access checking is done for resources in the class that the macro defines.

14.6 SECLABELs that the system creates

The system creates the following four labels automatically at initialization time.

- **SYSHIGH**: This label is equivalent to the highest security level defined by the security administrator, and all categories defined by the security administrator. It dominates all other security labels in the system. If another hierarchical security level is added to the system or if another non-hierarchical category is added, the system converts the SYSHIGH label to include the change. SYSHIGH should be restricted to special system-level address spaces such as consoles, and to system programmers, system operators, and system administrators.
- **SYSLOW**: This label is equivalent to the lowest security label defined by the security administrator, and no categories. It is dominated by all other security labels. If a resource is not in a class that requires reverse mandatory access checks or equal mandatory access checks, assigning a security label of SYSLOW to the resource allows all subjects to pass a mandatory access check for read access to the resource. Subjects still must pass the discretionary access check in order to access the resource. SYSLOW should be used only for resources that have no classified data content. It is appropriate to use SYSLOW for data sets that IBM supplies for which the following is true:
  - Most users only need to read them.
  - A limited number of users, such as system programmers, might need to update them. They should do so only when running at a very low classification, to prevent them from accidentally putting classified data into the data sets.
- **SYSNONE:** SYSNONE is treated as equivalent to any security label to which it is compared. SYSNONE, like SYSLOW, should be used only for resources that have no classified data content. It is different from SYSLOW in that all users might need to update resources with SYSNONE, even when running at a high classification. It is intended for use on resources that must be written to at different security labels when write-down is not allowed. It is used to ensure that a user is permitted read/write access to a data set such as a catalog. Use SYSNONE for a data set only when some other process (such as

catalog management, or a program via program access to data sets (PADS)) mediates the user's access to ensure that no classified data is written into the data set. Use SYSNONE for a z/OS UNIX file only when you limit discretionary access to the file to a specific UID, and the only access to the file is via a z/OS UNIX program with the setuid option that switches to that UID and ensures that no classified data is written into the data set. Do not use SYSNONE for users.

- **SYSMULTI:** This SECLABEL is new with z/OS V1R5 and is considered to be equivalent to any defined security label. It is intended for use by the following:
  - Server or daemon address spaces whose implementation and documentation explicitly support multilevel security, giving them the ability to perform and separate work for users running with different security labels
  - zFS directories that can contain data with different security classifications, such as the root directory of a file system

  SYSMULTI is not an appropriate security label for a data set or z/OS UNIX file, unless access to the data set is mediated via PADS or some other mechanism to ensure that either of the following is true:

  - Users can only write, and not read
  - Users can only read appropriate parts of the data

  SYSMULTI is not generally appropriate for users.

14.6.1 RACF options needed with SECLABEL

The following RACF options must be used when the SECLABEL class is activated:

- **MLACTIVE**: The MLACTIVE option requires security labels for most resources other than resources related to z/OS UNIX, and for all users entering the system. The MLACTIVE option has two suboptions, FAILURES and WARNING.
- **MLSTABLE:** The MLSTABLE option prevents authorized users from doing the following while the system is not quiesced:
  - Changing profiles in the SECLABEL class with the RALTER command
  - Changing the SE field in profiles
- **MLQUIET:** The MLQUIET option prevents users other than SPECIAL users, console operators, and started procedures from logging on, starting new jobs, or accessing resources. This option prevents these users from using the RACROUTE AUTH, DEFINE, and VERIFY requests. When the MLSTABLE option is active, authorized users cannot make changes to security labels or change the security labels associated with resources until the security administrator sets the MLQUIET option.
- **SECLABELCONTROL:** Specifies that users other than those with the RACF SPECIAL attribute cannot do the following:
  - Change a profile in the SECLABEL class with the `RALTER` command.
  - Change the SECLABEL field of a profile.
  - Issue an `ADDSD`, `ALTDSD,` or `DELDSD` command that causes the security label of a data set to change.

14.7 SECLABEL for z/OS UNIX processes and sockets

Before z/OS V1R5, when users enter the system through TSO/E they have the ability to select their current SECLABEL by specifying it on the logon panel, or they can use their default. The value they enter is saved in the TSO segment and used as the default the next time they log on. The SECLABEL of the terminal they are logging on to must dominate the user's SECLABEL.

With z/OS V1R5 this function has been modified to handle workstations (allowing for both reading and writing), and the usage and characteristics of the TERMINAL class have changed to require SECLABEL equivalence and to supply the SECLABEL if none is specified. The function has also been extended to support the z/OS UNIX environment where a user may enter the system from a remote IP address using an application such as rlogin.

The extension of SECLABELs to z/OS UNIX entails associating SECLABELs to IP addresses. Since the TERMINAL class cannot handle IP V6 addresses (due to their length), the usage and characteristics of the SERVAUTH class, which currently is used by the z/OS Communications Server (TCP/IP) to check server access authorization, have been changed so that IP V6 addresses can be accommodated.

14.7.1 SERVAUTH enhancements

The expanded usage of the SERVAUTH class requires a number of changes to its characteristics. Profiles in this class will require a SECLABEL if the MLACTIVE option is active. SECLABELs are checked for equivalence rather than following the normal MAC authorization rules. Because it is a port of entry, changes to the SERVAUTH class can cause information to be removed from VLF, just as they can for TERMINAL and APPCPORT.

When the existing MLACTIVE option is set additional checking will be done by RACROUTE REQUEST=VERIFY and INITACEE to ensure that server applications do not allow users running with different security levels in the same server address space. When anchoring an ACEE in a TCB, where the address space already has an ACEE in the ASXB, the SECLABELs associated with each ACEE will be checked for equivalence. If they are not equivalent, the request will be failed with message ICH408I. Other products anchoring ACEEs in TCBs for client users should also ensure equivalence.

Note: Assigning SYSMULTI to the server address space indicates that the server allows clients at multiple security levels since SYSMULTI is equivalent to any defined SECLABEL.

For applications that do not allow the specification of a SECLABEL, a SECLABEL for the user must be derived from one of the following:

- The user's port of entry
- A resource in the SERVAUTH or TERMINAL class

TCP/IP, z/OS UNIX, and RACF work together to determine the SECLABEL associated with the IP address when the SECLABEL class is active, and associate it with the user's security environment if the user is authorized to use it.

Program access to SERVAUTH resources

You can allow users to access IP addresses only when executing certain programs when you protect the names of network security zones (containing IP addresses) using SERVAUTH class resources. For example, when you control access to network security zones, you can permit network administrators to access certain zones only when using the **ping** and **traceroute** commands. For more information about using SERVAUTH resources to control access to network security zones, see *z/OS Communications Server: IP Configuration Guide* SC31-8775.

To set up program control for a SERVAUTH resource (representing a network security zone), create a profile in the SERVAUTH class specifying UACC(NONE), or specify ID(*) ACCESS(NONE) to ensure no access by general users. Then, permit certain users using WHEN(PROGRAM(program-name)) with the ID and ACCESS operands on the PERMIT command:

```
RDEFINE SERVAUTH resource-name UACC(NONE)
PERMIT resource-name CLASS(SERVAUTH) ID(user or group or *) ACCESS(READ)
WHEN(PROGRAM(program-name))
```

This example permits the specified users or groups to access network security zones protected by SERVAUTH resources only when executing the specified program or command.

Program access to SERVAUTH resources in ENHANCED program security mode operates much the same as it does in BASIC program security mode, with one exception. RACF allows program access to SERVAUTH resources to operate in ENHANCED program security mode only when one of the following is true:

- The program that established the current program environment has the MAIN attribute.
- The current program or the first program executed in the current or a parent MVS task has the BASIC attribute.

Note: For checking MAIN programs, the environment is considered “established” by the initial program executed in the job step, or the initial program executed by TSOEXEC or the IKJEFTSR service, or the initial UNIX program **exec()**ed or **spawn()**ed (non-local case only).

As with program access to data sets, you must maintain a clean environment to control program access to SERVAUTH resources. Unlike program access to data sets, the PADCHK/NOPADCHK operands have no meaning and are ignored.

Maintaining a clean environment

Several functions require a clean or controlled program environment:

- Use of PADS
- Use of program access to SERVAUTH resources
- Use of EXECUTE access for programs or libraries
- Improved UNIX security through the definition of FACILITY profile BPX.DAEMON

A “program environment” consists of a job step in a batch job, a started procedure or job, a user's TSO session, or a UNIX address space. In TSO, you can also create a separate program environment by invoking a TSO command using the TSO/E **TSOEXEC** command or the underlying IKJEFTSR programming service.

A clean or controlled environment indicates that the user has run only programs defined as controlled programs. You define programs through the PROGRAM class in RACF, as shown earlier. You can also define programs that reside in the UNIX file system as controlled programs, by using the UNIX **extattr** command with the **+p** option. Refer to *z/OS UNIX System Services Planning*, GA22-7800 for more information.

Stack access control using SERVAUTH

Stack access control allows control of access to a TCP/IP stack using an SAF security server. It provides a way to generally allow or disallow users or groups of users access to a TCP/IP stack. The function controls the ability of a user to open an AF_INET or AF_INET6 socket, and to get the host ID or host name. The TCP/IP stack to be protected is represented by the resource name EZB.STACKACCESS.sysname.tcpname. Access to the stack is allowed if the user is permitted to the security profile in the SERVAUTH class covering this resource or if the security server indicates there is no profile covering this resource. There are no new TCP definitions required.

Note: Some security products do not distinguish between a resource profile not defined and a user not permitted to that resource. If your product does not make this distinction, then you must define the stack access resource profile and permit users to it whenever the SERVAUTH class is active.

INITACEE

The following was added to the INITACEE:

- SECLSRVM event qualifier for mismatch with server's security label.
- INTA_SERV_POENAME security label of server.
- INTA_SERV_POENAME SERVAUTH resource or profile name. These changes allow applications willing to change their code to allow the specification of a seclabel by the user.

BPX1POE UNIX callable service

The__poe() callable service specifies the port of entry information the system is to use in determining various levels of permission checking in a multilevel-secure system. The authorization that is required to invoke this service is one of the following:

- Read access to the BPX.POE FACILITY class profile
- A UID of 0 when the BPX.POE FACILITY class profile is not defined

14.8 SECLABELs for z/OS UNIX files and directories

z/OS UNIX files and directories can be protected by security labels. The security label for a file or directory is stored in the file security packet (FSP) for the file or directory. The `SETROPTS mlfsobj` command controls whether security labels are required for z/OS UNIX files and directories. The zSeries file system (zFS) and the hierarchical file system (HFS) are both z/OS UNIX file systems that can be used with security labels. However, there are restrictions for using security labels with HFS files in a multilevel-secure environment. The hierarchical file system (HFS) does not fully support security labels and multilevel security. In a multilevel-secure environment, use HFS file system only in readonly mode. If you want to use files from an HFS file system in read-write mode, and use security labels in the file system, you must copy or move those files to a zFS file system.

It is possible for an HFS file to have a security label. For example, if the SECLABEL class is active when you create an HFS file system, the system assigns the root of that file system a security label in the same way that it assigns a security label to a zFS file system aggregate, from the data set profile. Subsequently, files created within that HFS file system will adopt security labels under the same rules as for zFS files. However, the HFS physical file system (PFS) does not support some functions of multilevel security, such as the `chlabel` command and the name-hiding function. If you attempt to use an HFS file system in read-write mode, if it has security labels it will not always behave predictably. Furthermore, if the file system does not have security labels, and the MLFSOBJ RACF option is active, users who try to access the file system will receive failures.

Once a file or directory has been assigned a security label, there is no way to delete or change the security label assigned to it, other than copying the file or directory to another directory with a different security label. To accomplish this when the SECLABEL class is active, the user must have security label authority to both the old file and the directory for the new file, and must ensure that no declassification of data occurs. Alternatively, the copy could be done after deactivating the SECLABEL class.

14.8.1 MLFSOBJ option

If you have the SPECIAL attribute, and if the SECLABEL class is active, you can prevent users (except trusted and privileged started tasks) from accessing z/OS UNIX file system resources, such as files and directories, that do not have security labels. While the **SETROPTS MLFSOBJ** option is in effect, all z/OS UNIX file system resources must have security labels.

To do this, enter:

```
SETROPTS MLFSOBJ(ACTIVE)
```

Restriction: This option cannot be activated when the SECLABEL class is inactive.

To cancel the MLFSOBJ option, specify MLFSOBJ(INACTIVE) on the **SETROPTS** command.

Note: Do not specify **SETROPTS MLFSOBJ(ACTIVE)** if any system sharing the RACF database is not at the necessary software level for multilevel security support. Use of the **SETROPTS MLFSOBJ** command should not cause problems on these systems, but it does not provide full protection on these systems. For details, see z/OS *Planning for Multilevel Security*, GA22-7509.

14.8.2 SECLABELs for zFS aggregates

The zSeries file system (zFS) supports security labels. A zFS file system is contained in a VSAM linear data set. The security label of the root within each zFS file system data set is determined at the time the file system aggregate (container) is allocated, from the security label of the profile in the DATASET class that covers the aggregate. If the SECLABEL class is active when allocating the aggregate, all file systems subsequently created within that aggregate contain a root with the security label that is specified in the profile for that aggregate. If no profile exists for the aggregate, or if it exists but does not specify a security label, then if the MLFSOBJ option is not active the root for any file systems within that aggregate have no security label. If the MLFSOBJ option is active, requiring security labels on all file system objects, the user's security label is assigned to the root of the file system. If a zFS file system or aggregate has a security label, changing the security label specified in the data set profile does not change the security label of any zFS file systems contained within it. Once a file system is created with a security label it has that security label forever. (However, if a zFS file system does not have a security label and is mounted read-only, the security label that z/OS UNIX assumes for it can change, as explained in “Assumed security labels” in *z/OS Planning for Multilevel Security*, GA22-7509.

14.8.3 SECLABELs for zFS files and directories

If the SECLABEL class is active, when a new z/OS UNIX file or directory is created within a zFS file system, the system assigns it a security label as follows:

- If the parent directory has no security label, the new file or directory is not assigned a security label.
- If the security label of the parent directory is SYSMULTI, the security label of the new file or directory is set to the security label of the requesting address space (the user). If the

user has no security label (this could occur only if the MLACTIVE option is not active), the new file or directory is not assigned a security label.

- If the security label of the parent directory is not SYSMULTI, the security label of the new file or directory is set to the security label of its parent directory.

If a z/OS UNIX file, directory, or symbolic link was created in a zFS file system without being assigned a security label (for example, if the SECLABEL class was not active when the file, directory, or symbolic link was created), the security administrator can assign a security label to it using the **chlabel** shell command.

Symbolic links

Symbolic links are also protected by security labels. When the name-hiding function is active a user's security label must dominate the security label of the symbolic link to read the link.

Hard links

Hard links are also protected by security labels. An attempt to create a hard link fails if the security label of the directory containing the link is not equivalent to the security label of the file to which the link points. An attempt to create a link to a character special file fails if either the file or its directory has a security label.

14.8.4 SECLABELs for HFS file systems

The hierarchical file system (HFS) does not fully support security labels and multilevel security.

In a multilevel-secure environment, use HFS file systems only in read-only mode. If you want to use files from an HFS file system in read-write mode, and use security labels in the file system, you must copy or move those files to a zFS file system. It is possible for an HFS file to have a security label. For example, if the SECLABEL class is active when you create an HFS file system, the system assigns the root of that file system a security label in the same way that it assigns a security label to a zFS file system aggregate, from the data set profile. Subsequently, files created within that HFS file system will adopt security labels under the same rules as for zFS files. However, the HFS physical file system (PFS) does not support some functions of multilevel security, such as the **chlabel** command and the name-hiding function. If you attempt to use an HFS file system in read-write mode, if it has security labels it will not always behave predictably. Furthermore, if the file system does not have security labels, and the MLFSOBJ RACF option is active, users who try to access the file system will receive failures.

MFS mounts

If you mount an HFS file system that does not have a security label in read-only mode, the system can assume a security label for it, as described in *z/OS Planning for Multilevel Security,* GA22-7509. If an HFS file system has a security label, changing the security label specified in the data set profile does not change the security label of the HFS file system. Once a file system is created with a security label it has that security label forever. (However, if an HFS file system does not have security label, and is mounted read-only, the security label that z/OS UNIX assumes for it can change, as explained in *z/OS Planning for Multilevel Security,* GA22-7509.) Even with assumed security labels, an HFS file system does not support the name-hiding function.

Note: If the name-hiding option is active, do not use HFS file systems unless they contain files and directories whose names should be viewable by all users.

14.8.5 UNIX callable services enhancements

The following UNIX callable services were enhanced to support SECLABEL:

- c_access (IRRSKA00)
- ck_file_owner (IRRSKF00)
- ck_owner_two_files (IRRSC200)
- R_chaudit (IRRSCA00)
- R_chmod (IRRSCF00)
- R_chown (IRRSCO00)
- R_setfacl (IRRSCL00)
- make_FSP (IRRSMF00)
- make_root_FSP (IRRSMR00)

UNIX callable services R_SETFSECL

The R_setfsecl service changes the security label in the File Security Packet (FSP) to the value specified in the CRED, or if no value is specified, the security label of the address space level ACEE.This function is available only to supervisor state callers passing a system CRED, or, if no security label is currently assigned, a user running with SPECIAL authority. It runs in cross-memory mode. It can be used only by the physical file system (zFS) or z/OS UNIX System Services.

14.9 SECLABELs for z/OS UNIX IPC

To enable the use of the SECLABEL between UNIX processes in a multilevel-security environment, the RACF option MLIPCOBJ must be activated using:

```
SETROPTS MLIPCOBJ (ACTIVE)
```

14.9.1 MLIPCOBJ option

Use the MLIPCOBJ option to control whether security labels are required for interprocess communication. The MLIPCOBJ option has two suboptions, ACTIVE and INACTIVE:

- MLIPCOBJ(ACTIVE) specifies that when the SECLABEL class is active, all IPC objects must have a security label. Those that don't can only be accessed by trusted or privileged started tasks.
- MLIPCOBJ(INACTIVE) specifies that IPC objects do not require a security label.

If the SECLABEL class is active, security labels are assigned to IPC objects during object creation, and security labels are checked before access is allowed to an IPC object that has a security label. However, as long as the MLIPCOBJ option is not active, any IPC object that is running without a security label can be accessed. When you activate the MLIPCOBJ option, IPC objects running without a security label can no longer be accessed. Before you activate the MLIPCOBJ option, let your system run with the SECLABEL class active, to allow the system to assign security labels to IPC objects as they are created. Run until you are sure that all active IPC objects have been created by users who have a security label. Or, re-IPL to be certain that all IPC objects have security labels.

Before you activate MLIPCOBJ (ACTIVE), ensure that you have done the following:

- Defined security labels by defining profiles in the RACF SECLABEL class
- Authorized all users to use the security labels they will need
- Activated and RACLISTed the SECLABEL class

Ensure that all IPC objects have security labels, either by re-IPLing after you assigned security labels to all users and activated the SECLABEL class, or by running with the SECLABEL class active until you are sure that all IPC objects have security labels. Requirement: The SECLABEL class must be active before you can activate the MLIPCOBJ option.

> **Tip:** Run with MLIPCOBJ(ACTIVE) set.

14.9.2 Interprocess communication (IPC) objects

For communications using interprocess communication (IPC) objects, when RACF creates an IPC security packet (ISP), if the SECLABEL class is active RACF copies the security label of the process, if one exists, into the ISP. RACF rejects requests for subsequent connections if the connecting process does not have a security label equivalent to the security label in the ISP. Once a security label has been assigned to an IPC object, it cannot be changed.

To establish multilevel security for IPC objects, activate the SECLABEL class and activate the MLIPCOBJ RACF option. If the SECLABEL class is active, and the MLIPCOBJ RACF option is not active, the system assigns a security label to an IPC object only if the creating process had one. If the IPC object does not have a security label, the system does not require a security label for connecting processes. However, if the connecting process does have a security label, the connection fails. If the SECLABEL class is active, activating the MLIPCOBJ option causes the system to require a security label for all IPC objects and for all connecting processes.

UNIX callable services enhancements

The following UNIX callable services were enhanced to support SECLABEL:

- makeISP (IRRSMI00)
- ck_IPC_access (IRRSKI00)
- ck_IPC_ctl

14.10 SECLABEL by system

In a sysplex it can be useful to limit the use of certain security labels to certain members of the sysplex. This allows one member of the sysplex to run work at security label A, while another handles work at security label B, keeping work separated based on security classification while still sharing the RACF database. The **SETROPTS** option SECLBYSYSTEM allows you to use security labels on a per-system basis, as follows:

```
SETROPTS SECLBYSYSTEM
```

The SECLABEL class must be active before you can activate the SECLBYSYSTEM option.

To define system-specific security labels, the security administrator specifies on which systems a security label is to be active by adding a member list to the SECLABEL resource class profile. The member names are system SMF IDs containing one to four characters. For example, to define the security label named SECRET as being active only on the systems with SMF system IDs SYSA and SYSB, the security administrator could define SECRET with a command like:

```
RDEFINE SECLABEL SECRET....ADDMEM(SYSA,SYSB)
```

If no member list of system IDs is added, the security label is considered to be active on all systems sharing the RACF database. The security labels SYSHIGH, SYSLOW, SYSNONE, and SYSMULTI are always considered to be active on all systems. If a member list is added to one of their profiles, it is ignored. To activate the use of system-specific security labels, activate the SECLBYSYSTEM option and refresh the SECLABEL class:

```
SETROPTS RACLIST(SECLABEL) REFRESH
```

Restriction: The following restrictions apply to system-specific security labels:

- JES3 does not support the use of system-specific security labels. Do not activate the SECLBYSYSTEM SETROPTS option if you are using JES3.
- JES2 does not support using system-specific SECLABEL labels for systems that perform NJE and OFFLOAD processing. These systems must have all security labels active. In addition, JES2 printers cannot process output unless the security label associated with the output is active on the system controlling the printer.
- If you define system-specific security labels, using a generic TSO system name at logon might not work because the user could be allocated to a system where the user's security label is not active.

If you use Application Restart Manager (ARM) to manage applications and you use system-specific SECLABELs, ensure that the systems that you've told ARM to use when restarting an application are systems that have the appropriate security label active. Otherwise, ARM might try to restart an application requiring a particular security label on a system where the security label is not active, and the application restart will fail.

14.10.1 The SECLBYSTEM and NOSECLBYSYSTEM options

Use these options to control activation of security labels on a system image basis in a sysplex. For more information on using system-specific security labels, see “Using system-specific security labels in a sysplex” in *z/OS Planning for Multilevel Security*, GA22-7509

The SECLBYSYSTEM option specifies that security labels are defined on a system basis. When SECLBYSYSTEM is active, the SMF ID values specified in the member list of the profiles in the SECLABEL class determine whether or not a security label is valid for a system. A security label that is not valid for a system is considered inactive and cannot be used on that system. Only a user with SPECIAL or AUDITOR authority can list the SECLABEL. Once the SECLBYSYSTEM option is activated, you must issue the following command to complete the activation of security labels by system:

```
SETROPTS RACLIST(SECLABEL) REFRESH
```

The NOSECLBYSYSTEM option specifies that security labels are not defined on a system basis. All security labels are valid on all systems that share the RACF database.

Tip: The SECLABEL class must be active before you can activate the SECLBYSYSTEM option.

Note: Run with the NOSECLBYSYSTEM option active unless you need to use system-specific security labels.

14.10.2 Activating SECLABELs by system image

If you have the SPECIAL attribute, and if the SECLABEL class is active, you can allow activation of security labels on a system image basis. Specify the SMF ID of each selected system in the member list of profiles in the SECLABEL class to indicate that a particular security label is active on that system. Security labels that are not active on a particular system cannot be used or listed by users without SPECIAL or AUDITOR on that system. If you define a security label with no member list, the security label is active on all systems.

If you specify a member list for the following security labels, it is ignored:

- SYSHIGH
- SYSLOW
- SYSNONE
- SYSMULTI

When SECLBYSYSTEM is in effect, a batch job submitted with no security label executes with the security label of the JESINPUT class profile, unless the JESINPUT class security label is SYSMULTI. After activating SECLBYSYSTEM and to complete the activation of security labels by system, you can activate this option, as follows:

```
SETROPTS SECLBYSYSTEM
SETROPTS RACLIST(SECLABEL) REFRESH
```

This option cannot be activated when the SECLABEL class is inactive. To cancel the SECLBYSYSTEM option, specify NOSECLBYSYSTEM on the **SETROPTS** command by issuing the following commands:

```
SETROPTS NOSECLBYSYSTEM
SETROPTS RACLIST(SECLABEL) REFRESH
```

Note: Do not specify **SETROPTS SECLBYSYSTEM** if any system sharing the RACF database is not at the necessary software level for multilevel security support. Use of the **SETROPTS SECLBYSYSTEM** option should not cause problems on these systems, but it does not provide full protection on these systems. For details, see *z/OS Planning for Multilevel Security*, GA22-7509.

14.11 Name hiding

The name hiding feature stops a user from seeing data sets for which he does not have access. The names of data sets, files, and directories might contain information that must be protected from some users.

14.11.1 The name-hiding function

The name-hiding function restricts the display of names to only those to which the user has authorization. The security administrator controls the name-hiding function by activating and deactivating the RACF MLNAMES option using the **SETROPTS** command. The name-hiding function restricts the display only of names that the user does not already know; that is, if the user's request includes the name, the system does not hide the name. For example, assume that a user is not authorized to data set x.y.z. If the user asks to see all the names of all the data sets in a catalog that includes x.y.z, x.y.z is not displayed in the list. The system does not let the user know that the data set x.y.z exists. But, if the user asks to specifically see data set x.y.z, the system responds that the user does not have access to x.y.z, but does not hide the fact that x.y.z exists. For files and directories, RACF does a mandatory access check to determine whether the user is authorized to see a name. For data sets in a multilevel-secure environment, RACF does both a discretionary access check and a mandatory access check to determine if a user is authorized to see a data set name.

14.11.2 DFSMS and name hiding

The following DFSMS components support multilevel security and name hiding:

- DFSMSdfp supports multilevel security, including the name-hiding function.
- DFSMSrmm uses resource profiles in the RACF FACILITY, DATASET, and TAPEVOL classes to authorize access to information in the DFSMSrmm control data set about volumes and data sets. DFSMSrmm also supports the name-hiding function.

DFSMSdfp support

When the name-hiding function is active (the MLNAMES option is active), DFSMSdfp does not display the name or any other information about a data set that a user requests using a generic name unless the user has authorization to the data set. For example, if a user issues a `LISTCAT` command with the LEVEL keyword, `LISTCAT` displays only the names of data sets to which the user has authorization. Requests for information about a specific data set name, such as a `LISTCAT` command with the ENTRY keyword, or specifying an exact data set name on an ISPF catalog or VTOC listing panel, are not affected by the name-hiding function. A user who can read the VTOC or VTOC index can read the data set names listed in them. When the name-hiding function is active DFSMS limits read access to the VTOC and VTOC index, to protect the names of data sets.

DFSMSrmm support

DFSMS protects the VTOC with resources in the FACILITY class named STGADMIN.IFG.READVTOC.*volser*. When the name-hiding function is active, a user who does not have FACILITY class authorization to a volume cannot read the VTOC or VTOC index for that volume directly. (The user can still read a VTOC indirectly using system services and functions such as the ISPF panels that allow listing VTOCs, but is restricted to retrieving information only for those data sets the user can access.) Ways in which a user might access the VTOC include:

- The IEHLIST utility
- ISMF in ISPF
- The DSLIST utility for printing or displaying lists of data set names in ISPF

If you need to allow some users to read the complete VTOC for a volume when the name-hiding function is active, bypassing name-hiding restrictions, create a profile in the FACILITY class protecting the volume. Specify UACC(NONE) to prevent users who aren't in the access control list from accessing the VTOC, and add users who are allowed to read the VTOC to the access control list.

Example: To give the user USER10 authorization to read the VTOC for the volume with volume serial 123456:

```
RDEFINE FACILITY STGADMIN.IFG.READVTOC.123456 UACC(NONE)
PERMIT STGADMIN.IFG.READVTOC.123456
CLASS(FACILITY)ID(USER10)ACCESS(READ)
```

Note: The name-hiding function can be activated in an environment that is not multilevel-secure. *z/OS DFSMS Migration,* GC26-7398 describes the name-hiding function in an environment that is not multilevel-secure.

If you do not have a need to protect the names of data sets, files, and directories, run with the NOMLNAMES option set. Because the MLNAMES option can adversely affect performance, do not run with it active unless you need the protection it provides.

14.12 RACROUTE enhancements

The following sections are updated to support multilevel security:

- RACROUTE REQUEST=AUTH
- RACROUT REQUEST=DIRAUTH
- RACROUT REQUEST=EXTRAC
- RACROUT REQUEST=FASTAUTH
- RACROUT REQUEST=VERIFY
- RACROUT REQUEST=VERIFYX

14.12.1 RACROUTE REQUEST=DIRAUTH

This service compares two security labels. The security labels may be passed directly, or as part of an ACEE or UTOKEN. The class name determines the type of comparison made between the security labels, unless the TYPE parameter is specified. Note that the security labels should be obtained from the same system since a SECLABEL name may not represent the same security classification on different systems. Components that need to compare security labels for equality may do a check for an exact match without issuing this macro.

The message transmission managers (that is, VTAM, TSO/E, and Session Manager) use the RACROUTE REQUEST=DIRAUTH macro with RTOKEN specified to ensure that the receiver of a message, represented by the ACEE of the current address space, meets security-label authorization requirements. That is, the security label of the receiver of the message must dominate (be equal to or higher than) the security label of the message. When invoked as these managers do, with just the RTOKEN or just the RTOKEN and LOG keywords specified, if the security label of the receiver does not dominate the security label of the message, DIRAUTH performs additional processing to determine if the receiver has access to any security label that could dominate the message. To use this service, you must specify RELEASE=1.9 or a later release number.

The caller of RACROUTE REQUEST=DIRAUTH must be authorized (APF-authorized, in system key 0-7, or in supervisor state). The caller cannot hold any locks when issuing RACROUTE REQUEST=DIRAUTH.

This request is SRB-mode compatible. When issuing RACROUTE REQUEST=DIRAUTH in SRB mode, you must ensure that the jobstep task pointed to by the ASCBXTCB field in the target address space is active when you schedule the SRB and that it remains active until the SRB completes.

> **Note:** In order to ensure that the SRB does not run after the ASCBXTCB task completes, you must enable purgeDQ to deal with that SRB. In particular:
>
> - If using SCHEDULE, SRBPASID must be set to the ASID, and SRBPTCB must be set to the contents of ASCBXTCB.
> - If using IEAMSCHD, PURGESTOKEN must be specified with the STOKEN of the space, and PTCBADDR must be specified with the contents of ASCBXTCB.

When the task terminates, the purgeDQ issued causes the SRB's resource manager termination routine (RMTR) to be driven if the SRB has not begun to run, or waits for the SRB to complete if the SRB has begun to run.

RACROUTE REQUEST=DIRAUTH can also be invoked from a cross-memory environment, when in task or SRB mode. The ACEE used for authorization checking must reside in the HOME address space (unless ACEEALET specifies a different address space). Callers in cross-memory mode must be in supervisor state. RACROUTE REQUEST=DIRAUTH interrogates the setting of SETR MLS during dominance checking for ACCESS=READWRITE and ACCESS=WRITE to determine if write-down is allowed on the system. It does not support WARNING mode for SETR MLS and will process write-down violations as failures. RACROUTE REQUEST=DIRAUTH does not check whether or not security labels are required, nor does it always bypass security label checking if the ACEE indicates trusted or privileged authority. If a full authorization check including the checking of security labels is needed, then RACROUTE REQUEST=AUTH, RACROUTE REQUEST=FASTAUTH, or the `ck_access` callable service should be used.

For more information about the enhancements of the RACROUTE REQUEST see *z/OS Security Server RACROUTE Macro Reference*, SA22-7692.

14.13 Miscellaneous enhancements

The ICHERCDE macro generates entries for the class descriptor table. The class descriptor table contains information that directs the processing of general resources. The macro has been updated to meet multilevel security requirements.

14.13.1 ICHERCDE

SIGNAL and EQUALMAC keywords have been added to the ICHERCDE.

SIGNAL keyword

This keyword sends an ENF signal to listeners when a SETROPTS RACLIST, SETROPTS NORACLIST, or SETROPTS RACLIST REFRESH is issued for the class, activating, deactivating, or updating the profiles used for authorization checking. When SIGNAL=YES is specified, RACF sends a type 62ENF signal to listeners, with a parameter list mapped by IRRPENFP. Qualifier byte 1 indicates a SETROPTS RACLIST, qualifier byte 2 indicates a SETROPTS RACLIST REFRESH, and qualifier byte 3 indicates a SETROPTS NORACLIST. The parameter list contains the class name. Listeners of this signal should follow the guidelines documented in *z/OS MVS Programming: Authorized Assembler Services Guide*, SA22-7608 on coding listener exit routines, particularly:

- Avoid such time-consuming processing as obtaining large amounts of storage through the GETMAIN macro, issuing WAITs or issuing SVCs that issue the WAIT macro, and performing I/O operations.
- Avoid requests for the local lock.
- Avoid using multiple listener user exits.

RACROUTE REQUEST=LIST,GLOBAL=YES does not cause an ENF signal to be issued. When classes that are GLOBAL=YES ONLY RACLISTed are refreshed with SETROPTS RACLIST REFRESH, an ENF signal is issued. If they are SETROPTS NORACLISTed, the ENF signal is issued only on a system that has the class GLOBAL=YES RACLISTed. Avoid using SETROPTS NORACLIST in case of a RACROUTE REQUEST=LIST,GLOBAL=YES class unless everyone has disconnected from the dataspace. At that point, it is unlikely that anyone is listening for an ENF signal. If RACLIST=DISALLOWED is specified for a class, no signal is sent even if SIGNAL=YES is specified. No signals are sent when an application issues RACROUTE REQUEST=LIST.

The following classes now have SIGNAL=YES specified, to support multilevel security:

- SECLABEL
- SERVAUTH
- TERMINAL

EQUALMAC keyword

The following classes now have EQUALMAC=YES specified, to support multilevel security:

- APPL
- DSNR
- JESINPUT
- SERVAUTH
- SERVER
- TERMINAL
- MQCONN

This keyword specifies whether equal mandatory access checking is required when users attempt to access resources protected by profiles in this class. If EQUALMAC=YES is specified, whenever RACF performs a mandatory access check the security label of the user and the security label of the resource must be equivalent to pass the mandatory access check. Security labels are equivalent when they have the same security level and category definitions. The SYSMULTI security label is equivalent to any other security label.

- Use EQUALMAC=YES for classes where two-way communication is expected.
- EQUALMAC=YES cannot be specified with RVRSMAC=YES.

14.14 New and changed messages

The following messages are added:

- ICH431I, ICH432I, ICH433I, ICH434I, ICH435I
- ICH14078I
- IRRW001I, IRRW002I, IRRW003I
- ICH408I

The following message is changed:

- ICH408I

14.15 Overview of the RACF enhancements

The following enhancements to RACF are available with z/OS V1R5:

- Dynamic templates

  A template maps how the profiles are written on the RACF database. The templates exist in three places:

  - In the most recent version shipped with RACF via a new release or a new PTF.
  - In the RACF database written using the utility IRRMIN00 with PARM=NEW or PARM=UPDATE.
  - In storage at RACF initialization.

  The templates are shipped in SYS1.MODGEN(IRRTEMP1). Run IRRMIN00 with PARM=UPDATE to write them down in the RACF database. Then you need to IPL.

> **Attention:** With IRRMIN00, be careful about the risk of using PARM=NEW instead of PARM=UPDATE to use the new templates. Using PARM=NEW instead of PARM=UPDATE would initialize the RACF database.

- Design changes to IRRMIN00, see “Database initialization utility IRRMIN00” on page 359.

- Scenarios for applying new templates, see “Applying new templates” on page 362.

14.16 Dynamic templates objectives

The objectives of this new support are to avoid any possibility of an IPL or any outage to update RACF templates, and to simplify the upgrade and maintenance procedures. This is achieved by doing the following:

- Have RACF Initialization build the in-storage templates automatically from the latest level, whether or not it is remembered to update the database templates with IRRMIN00 PARM=UPDATE
- Have IRRMIN00 automatically write the latest level of templates to the database.
- Do not allow IRRMIN00 PARM=UPDATE to down-level the templates on the database.
- In the case of a PTF that does not require an IPL, provide a means of dynamically activating new templates by replacing the in-storage templates with the new templates.
- Do not allow an existing, active database to be newly initialized (from the system on which the database is active). Have IRRMIN00 write the latest level of the templates.

14.16.1 Shipping of the templates

The templates are no longer shipped in source format like SYS1.MODGEN(IRRTEMP1). They are now shipped as a module in compiled format. The name of the module is IRRTEMP2. This module is linked into two load modules: the RACF initialization load module(ICHSEC00) and IRRMIN00. The templates also contain the APAR and the release level so that RACF can determine the level of the templates, as illustrated in Figure 14-1 on page 357.

```
   $/VERSION FMID/APAR# rrrrrrrr.aaaaaaaa
   $/VERSION HRF7708 00000010.00000000
   $/VERSION OA01234 00000010.00000010
   $/VERSION OA01567 00000010.00000020
   $/VERSION HRFxxxx 00000023.00000020
```

rrrrrrrr This is the release level

aaaaaaaa This is the APAR level

Figure 14-1 Data lines shipped with templates

Description of the $version

You receive a module in compiled format as IRRTEMP2. It is linked into two load modules: the RACF initialization load module and the IRRMIN00 utility load module. The first data lines, show in Figure 14-1, are as follows:

- The 00000010 specifies the first release level. Release and APAR levels in prior releases are assumed to be 00000000.00000000.

  Since PTFs or small programming enhancements (SPEs) are shipped with changes to the templates, the FMID/APAR field is updated as before with the APAR number, and the APAR level is incremented by 10.
- In Figure 14-1, two PTFs are shipped for release HRF7708; the APAR level for the first one is 00000010 and the second is 00000020.

Those APARs are rolled up to the next release so the APAR level in the next release is 00000020. The APAR level is carried forward to the next release; it does not go back to 00000000 at each subsequent release.

- The release level for the next release starts out as 00000020. If line items are added to the release that cause a template change, they are incremented by 1, so that by the time the development on the next release is done and ready to ship, the release value could be anywhere from 00000020 to 00000029, say 00000023 in this example.
- Subsequent releases start with increments of 20 so the next release would start with an rrrrrrrr value of 00000040, and so on into the future.
- Templates A are at a higher, later level than Templates B if the release level is higher, or if the release level is the same but the APAR level is higher. Also, if in the future a PTF that contains template changes is being shipped to two releases in the service stream, say Rn and Rn-1, the latest level of the templates is shipped to all lower levels. The Rn templates would be shipped to Rn and Rn-1. This is a change in philosophy from the current service process. The template extension information used to be hard-coded in the RACF manager. Moving it to IRRTEMP2 eliminates the need to update manager modules as new alias fields are added in the future, making the addition of new alias fields as dynamic as the addition of new segments or normal fields to the templates.

RACF command to display template level

The **@SET LIST** operator command displays information on RRSF usage, but also displays the in-storage template level and the dynamic parse level in effect on the system. Prior to this, the output consisted of the FMID or APAR number for both. The RACF command displays the in-storage level as follows:

```
RACF STATUS INFORMATION:
TEMPLATE VERSION - HRF7708 00000010.00000000
DYNAMIC PARSE VERSION     - HRF7708
```

The inventory control block (ICB) has been modified to hold the level of the templates. The following fields have been added:

ICBTMPRL Indicates the release level

ICBTMPAL Indicates the APAR level

14.16.2 RACF initialization

During an IPL, RACF initialization automatically builds the in-storage templates from the latest level of the templates. When processing IRRTEMP2, RACF issues error messages if a template is invalid. The error messages are as follows:

```
IRR8029i CONTENTS OF A TEMPLATE STATEMENT FROM IRTEMP2.
IRR8003I NON-NUMERIC CHARACTER IN NUMERIC FIELD OF LAST STATEMENT.
```

There are two cases during the RACF initialization, as follows:

- If the master primary database level is higher or the same as IRRTEMP2, it builds the in-storage templates from that database as it did prior to dynamic template support. The level of the templates on the database is kept in the ICB. The level of the templates in IRRTEMP2 is on the version card.
- If the master primary database level is not higher or not the same as IRRTEMP2, it then builds the in-storage templates from IRRTEMP2 itself and issues the following message:

```
ICH579E RACF TEMPLATES ON DATABASE ARE DOWNLEVEL. HRF7708 00000000.00000000;
USING TEMPLATES AT LEVEL HRF7708 00000010.00000000 FROM IRRTEMP2. RUN
IRRMINOO PARM=UPDATE.
```

Note: This scenario occurs if you forget to update the templates on the database prior to IPLing. The correct templates are in storage. The database templates need to be updated so the utilities work properly. No re-IPL is necessary.

14.17 Database initialization utility IRRMIN00

The changes to the IARRMIN00 utility are:

- IRRMIN00 no longer makes use of the SYSTEMP data set pointed to by SYS1.MODGEN(IRRTEMP1). Now it gets the templates from IRRTEMP2.
- In addition to the templates, IRRMIN00 now writes the alias-related template extension information to the database.
- If you specify PARM=NEW, this fails if the output database is active on the system where IRRMIN00 is invoked.
- IRRMIN00 now will not apply downlevel templates to a database.
- IRRMIN00 now makes templates active dynamically for the new PARM=ACTIVATE invocation when the templates on the active master primary database are a higher level than the in-storage templates.

There are changes to the IRRMIN00 utility for the following options:

- PARM=NEW
- PARM=UPDATE
- PARM=ACTIVATE

14.17.1 PARM=NEW option

The PARM=NEW option formats a non-VSAM DASD data set as a RACF database. It divides the database into 4K blocks, or records, and initializes them as follows:

Block 0 Contains the inventory control block (ICB).

Block 1 to block 9 Contains the templates.

Block 10 Contains the alias-field related template extensions.

Block 11 Contains the segment table block (becomes the in-storage templates).

Block 12 –nn- Contains the block availability mask (BAM) blocks.

Block nn+1 on These are empty blocks for later use as index blocks and profile blocks. It also contains the SYSPRINT data set and all the template data lines; normal messages and error messages are written here as well.

PARM=NEW used to write the content of IRRTEMP1 to SYSPRINT, which included a commentary standard prologue. With the change from a source to an assembler module, IRRTEMP2, only the data lines will be echoed, starting with the $/VERSION record and ending with the $/END record. PARM=NEW will now fail if invoked against an active database on the system where IRRMIN00 is invoked. The following error messages are issued if a problem occurs:

```
IRR8006 UNABLE TO OPEN DD (DDNAME).
IRR8011 RACF DATA BASE HEADER RECORD IS INVALID.
```

A sample JCL job to initialize the database is shown in Figure 14-2.

```
//INITRDS  JOB  ,'INITIALIZE NEW DS',
//             MSGLEVEL=(1,1),TYPRUN=HOLD
//INITALZE EXEC PGM=IRRMIN00,PARM=NEW
//STEPLIB  DD  DSN=SYS1.LINKLIB,DISP=SHR,
//             UNIT=YYYY,VOL=SER=YYYYYY
//SYSPRINT DD  SYSOUT=*
//SYSRACF  DD  DSN=SYS1.RACF,DISP=(NEW,CATLG),
//             UNIT=XXXX,VOL=SER=XXXXXX,
//             SPACE=(CYL,(XX),,CONTIG),
//             DCB=DSORG=PSU
```

Figure 14-2 Sample JCL to initialize the RACF database

All the IRRMIN00 JCL examples are shipped in SYS1.SAMPLIB(RACJCL). As previously, you can STEPLIB to the latest level of IRRMIN00 (and therefore the latest level of the templates in RRTEMP2). The SYSTEMP DD card used in previous releases to point to IRRTEMP1 is no longer used. If present in the JCL, it is ignored.

If IRRMIN00 PARM=NEW is invoked against an active database which is on the same system, the invoked IRRMIN00 issues the following error message:

```
IRR8024  PARM=NEW SPECIFIED FOR ACTIVE RACF DATABASE. PROCESSING STOPPED.
```

Note: You can still run IRRMIN00 PARM=NEW from another system on an active database, or when the data set is not active.

14.17.2 PARM=UPDATE

PARM=UPDATE now does the following:

- Writes the new templates and template extensions to the database.
- Updates the ICB with template information.
- Writes the new segment table, and does not touch the BAM blocks, index blocks, and profile blocks.
- Writes to SYSPRINT the template data lines and writes normal and error messages.
- Fails if the new templates are not at a higher level than the ones on the database. Any failure issues new or existing error messages. They are written to SYSPRINT. If the new templates are not at a higher level than the database templates, issues the following error message:

  ```
  IRR8025  PARM=UPDATE SPECIFIED, BUT TEMPLATE UPDATE NOT REQUIRED.
  ```

A sample JCL job to update the database is shown in Figure 14-3.

```
//UPGDRDS  JOB  ,'UPGRADE OLD DS',
//             MSGLEVEL=(1,1),TYPRUN=HOLD
//INITALZE EXEC PGM=IRRMIN00,PARM=UPDATE
//STEPLIB  DD  DSN=SYS1.LINKLIB,DISP=SHR,
//             UNIT=YYYY,VOL=SER=YYYYYY
//SYSPRINT DD  SYSOUT=*
//SYSRACF  DD  DSN=SYS1.RACF,DISP=SHR
```

Figure 14-3 Sample JCL for a PARM=UPDATE

Upgrading to a new release

When using IRRMIN00 PARM=UPDATE to upgrade to a new release, you should continue to do so from the old release. You can either use a STEPLIB to the new system's SYS1.LINKLIB, or copy IRRMIN00 from the new system's SYS1.LINKLIB, or copy IRRMIN00 from the new system's SYS1.LINKLIB to a different (APF-authorized) library. You no longer need to determine where the latest copy of the templates resides; the templates can be found automatically by IRRMIN00. If the SYSTEMP DD statement is specified, as was done in the past to identify the copy of the templates, it is ignored.

14.17.3 PARM=ACTIVATE

PARM=ACTIVATE is a new IRRMIN00 function in this release that does the following:

- Looks for the master primary data set and if active and the templates are a higher level than those already active, activates them by replacing the in-storage templates with those on the database.
- The SYSPRINT data set is used for normal and error messages.
- PARM=ACTIVATE does not process IRRTEMP2 so there are no data lines written to SYSPRINT.
- It processes the templates on the database, which are in internal format.
- IRRMIN00 PARM=ACTIVATE allows a customer to install a PTF that changes the templates and contains modules that reside only in LINKLIB to dynamically replace the in-storage templates without requiring a re-IPL.
- If the PTF shipped modules that reside in LPA, an IPL would be necessary in order to access the changed modules.

The following messages are issued in case of a successful run of PARM=ACTIVATE:

```
IRR8026I PARM=ACTIVATE SPECIFIED; IRRMIN00 IS PREPARING TO ACTIVATE THE
TEMPLATES FMID OR APAR# RRRRRRRR.AAAAAAAA.
IRR8027I IRRMIN00 HAS FINISHED ACTIVATING THE TEMPLATES.
```

A sample JCL job to dynamically activate the templates is shown in Figure 14-4.

```
//ACTRDS   JOB  ,'ACTIVATE TEMPLATES, UPDATE DYN PARSE',
//             MSGLEVEL=(1,1),TYPRUN=HOLD
//STEP1    EXEC PGM=IRRMIN00,PARM=ACTIVATE
//SYSPRINT DD  SYSOUT=*
// IF STEP1.RC = 0 THEN
//STEP2    EXEC PGM=IKJEFT01,REGION=1M,PARM='IRRDPI00 UPDATE'
//SYSTSPRT DD SYSOUT=*,HOLD=YES
//SYSUDUMP DD SYSOUT=*,HOLD=YES
//SYSUT1   DD DSN=SYS1.SAMPLIB(IRRDPSDS),DISP=SHR
//SYSTSIN  DD DUMMY
// ENDIF
```

Figure 14-4 Sample JCL to dynamically activate the templates

If the IRRMIN00 job runs successfully, then the IRRDPI00 **UPDATE** command is entered to rebuild the dynamic parse data set IRRDPSDS. Generally, if the templates are changed, the dynamic parse data set is changed also. Updating the dynamic parse data set allows the use of the new fields by the RACF command processors.

IRRMIN00 return codes

The IRRMIN00 program sets the return codes identified in Table 14-3.

Table 14-3 IRRMIN00 return codes

Hex	Decimal	Meaning
0	0	Successful completion.
4	4	Attention – the RACF database is usable, but the target of SYSRACF may be wrong, or the template level on the RACF database or the level of IRRMIN00 executed may not be the one expected.
C	12	The program encountered a terminating error. The RACF database was not formatted or reformatted for PARM=NEW or PARM=UPDATE. The templates from the database were not activated for PARM=ACTIVATE.
10	16	The output database could not be opened. The RACF database was not formatted.
14	20	The program was entered at an incorrect entry point.

14.18 Applying new templates

With the use of the dynamic templates, applying new templates is more difficult. There are now several scenarios, as follows:

- Installing a new release or a PTF with an IPL.
- Installing a PTF without an IPL.

14.18.1 Installing a new release or a PTF with an IPL

When installing a new release or a PTF with an IPL there are two cases:

- IRRMIN00 PARM=UPDATE is run and system is IPLed.

  The in-storage and database templates are both at the latest level, the level in the version of IRRTEMP2 that you just installed with the new release or PTF.

- IRRMIN00 PARM=UPDATE is *not* run and system is IPLed.

  RACF Initialization determines that the templates in IRRTEMP2 are a higher level than those on the database, so it builds the in-storage templates from the ones in IRRTEMP2. Message ICH579E is issued to remind you that the database templates are not the latest and that you need to run IRRMIN00 PARM=UPDATE. You run IRRMIN00 PARM=UPDATE so the utilities that use the templates on the database rather than the in-storage templates have the latest information. The in-storage and database templates are both the latest level, the level in the version of IRRTEMP2 that you just installed with the new release or PTF. You *do not* have to re-IPL.

14.18.2 Installing a PTF without an IPL

When installing a PTF without an IPL there are two cases:

- IRRMIN00 PARM=UPDATE and IRRMIN00 PARM=ACTIVATE are run.

  The in-storage and database templates are both the latest level, the level in the version of IRRTEMP2 that you just installed with PTF. You *do not* have to re-IPL.

- IRRMIN00 PARM=UPDATE *is not* run, IRRMIN00 PARM=ACTIVATE is.

The PARM=ACTIVATE fails with the error message IRR8032. Since you forgot to run IRRMIN00 PARM=UPDATE the templates on the database are the same level as the in-storage templates. To fix this, customer has to run IRRMIN00 PARM=UPDATE and then run IRRMIN00 PARM=ACTIVATE. Customer *does not* have to re-IPL.

Note: In an RRSF environment, if IRRMIN00 PARM=ACTIVATE is run on a node, it must be run on the other node.

Note: No matter which scenario you use, IRRMIN00 PARM=UPDATE *must* be run when upgrading a RACF release or when applying PTFs that upgrade the templates. Otherwise there is an impact on the utilities IRRDBU00, IRRUT200, BLKUPD, and OEM products.

14.18.3 New messages

The following new messages pertain to using dynamic templates:

```
IRR8024   PARM=NEW specified for active RACF database.  Processing stopped.
IRR8025   PARM=UPDATE specified, but template update not required.
IRR8026   PARM=ACTIVATE specified; IRRMIN00 is preparing to activate the
          templates FMID or APAR rrrrrrrr.aaaaaaaa.
IRR8027   IRRMIN00 has finished activating the templates.
IRR8028   IRRMIN00 cannot process PARM=ACTIVATE due to system error.
IRR8029I  <contents of a template statement from IRRTEMP2>
IRR8030   PARM=ACTIVATE not supported.
IRR8031   PARM=ACTIVATE specified, but there is no master primary database
          active.
IRR8032   PARM=ACTIVATE specified, but the level of the database templates:
          FMID or APAR rrrrrrrr.aaaaaaaa, is not higher than the level of
          templates on the system: FMID or APAR rrrrrrrr.aaaaaaaa.
IRR8033   Unable to establish ESTAE environment. Return code from ESTAE is
          return-code.
IRR52206I Unable to establish ESTAE environment. Return code from ESTAE is
          return-code.
ICH579E   RACF TEMPLATES ON DATABASE ARE DOWNLEVEL: FMID or APAR
          rrrrrrrr.aaaaaaaa; USING TEMPLATES AT LEVEL FMID or APAR
          rrrrrrrr.aaaaaaaa FROM IRRTEMP2. RUN IRRMIN00 PARM=UPDATE.
```

14.19 RACF support for DB2 V8

The features introduced in DB2 Universal Database™ Version 8 for z/OS now affect the way in which security checks are performed by RACF. Prior to DB2 Version 5 and OS/390 V1R4, only DB2's “native” security mechanisms (GRANT and REVOKE) could be used to control access to DB2 objects such as tables, views, and databases.

DB2 V5 defined an exit point (DSNX@XAC) which was called whenever access control decisions needed to be made. In OS/390 V1R4, RACF shipped a plug-in to this exit point (IRR@XACS) which allowed RACF to be used to control access to DB2 objects.

Prior to DB2 V8, RACF shipped its version of DSNX@XAC in SYS1.SAMPLIB(IRR@XACS). That code supported, and continues to support, DB2 V5, V6, and V7. This simplifies and synchronizes the availability of the RACF DSNX@XAC code and DB2.

Further information on DB2 V8 can be found at:

http://www.ibm.com/software/data/db2/os390/db2zosv8.html

The following changes have been made to RACF in support of DB2 V8:

- RACF-supplied DB2 External Security Module IRR@XACS, see “Security module IRR@XACS” on page 364.
- Long name support, see “Long name support” on page 366.
- Sequence object support, see “Sequence object support” on page 367.
- Multilevel security support (MLS), see “Multilevel security support (MLS)” on page 369.
- Refresh privilege on materialized table queue (MQT), see “Refresh privilege on MQT” on page 371.
- Ability for DBADM to create views for others, see“Ability for DBADM to create views for others” on page 371.

14.19.1 Advantages of using RACF with DB2

There are several reasons to use RACF over native DB2 access controls, as follows:

- RACF generic profiles and member/grouping profiles allow an installation to protect multiple DB2 resources with a single RACF profile.
- The RACF group authorization mechanism is simpler to use than the DB2 secondary auth ID.
- Consolidating security into RACF reduces the DB2 security skills that are required.
- There are some advantages to a single security log (SMF).

14.19.2 Security module IRR@XACS

This routine is invoked by DB2 as the DB2/RACF authorization exit and acts as the interface to DB2. As a result, it must have a CSECT name of DSNX@XAC. This routine will either process the request or invoke a subroutine based on the requested function. Authorization requests, XAPLFUNC=2, are handled by this routine. Initialization and termination requests, XAPLFUNC = 1or 3, are handled by IRR@XAC1. The details of initialization and termination are defined in the module prolog for IRR@XAC1. For authorization requests, IRR@XACS is responsible for implementing the authority checking. This routine uses two possible methods to determine access control:

- Implicit privileges of ownership
- RACF profiles for DB2 resource

When an action occurs that needs authorization, DB2 builds a control block (DSNDXAPL) that contains security information and passes it to a common security module (DSNX@XAC).This is illustrated in Figure 14-5.

When the security module was initialized, a RACLIST GLOBAL=YES was done for each active class so the RACF profiles would be placed in storage (a dataspace) to improve performance.

The security module uses the information in the XAPL and in rules tables (defined in the module) to construct a series of RACROUTE FASTAUTH requests. The security module will pass a return code to DB2 (rc=0 if allowed; rc=8 if failed; rc=4 if deferred). If RACF defers to DB2 (no profile found for a resource), DB2 makes the final decision by looking in its catalogs.

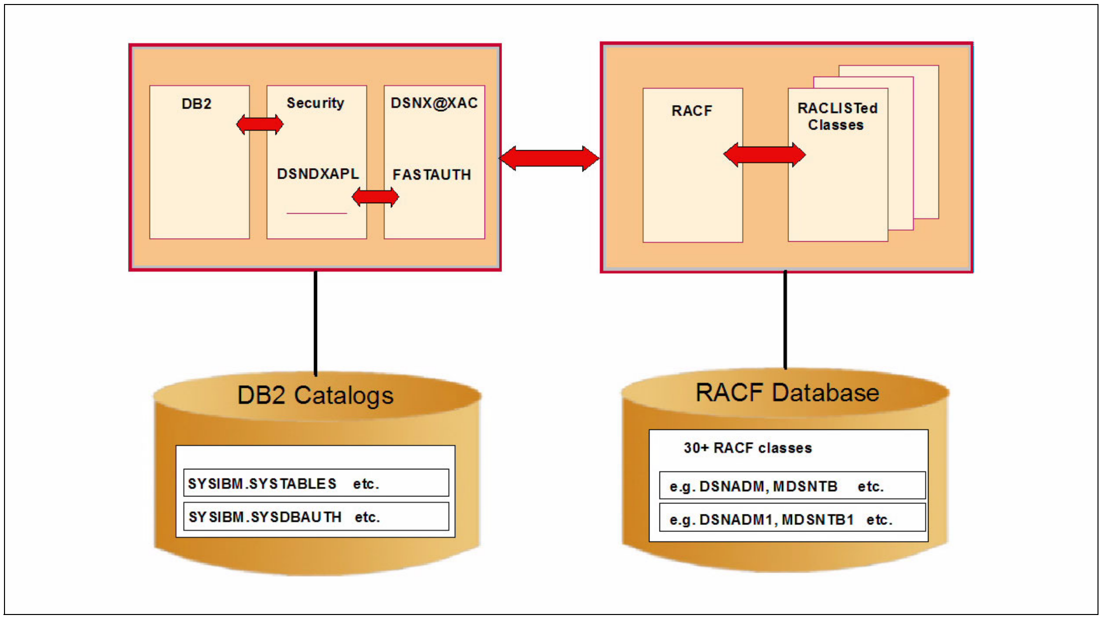


Figure 14-5 RACF and DB2 security checking prior to DB2 V8

14.19.3 Security with DB2 V8

The RACF plug-in DSNXRXAC is shipped with the DB2 V8, FMID HDRE810. The code is located in DB2.SDSNSAMP. This routine is invoked by DB2 as the DB2/RACF authorization exit and acts as the interface to DB2 V8. As a result, it must have a CSECT name of DSNX@XAC. This routine will either process the request or invoke a subroutine based on the requested function. Authorization requests, XAPLFUNC=2, are handled by this routine. Initialization and termination requests, XAPLFUNC = 1|3, will be handled by IRR@XAC1. The details of initialization and termination are defined in module prolog for IRR@XAC1.

Figure 14-6 shows that when an action occurs that needs authorization, DB2 builds a control block (DSNDXAPL) that contains security information and passes it to a common security module (DSNXRXAC).

When the security module was initialized, a RACLIST GLOBAL=YES was done for each active class so the RACF profiles would be placed in storage (a dataspace) to improve performance.

The security module uses the information in the XAPL and in rules tables (defined in the module) to construct a series of RACROUTE FASTAUTH requests.The security module passes a return code to DB2 (rc=0 if allowed; rc=8 if failed; rc=4 if deferred). If RACF defers to DB2, (no profile found for resource), DB2 makes the final decision by looking in its catalogs.

RACF continues to ship IRR@XACS, which is the DB2 V6 and DB2 V7 version of DSNX@XAC. DSNXRXAC is documented in *DB2 Universal Database for z/OS: RACF External Security Module Guide and Reference*, SA22-7938.

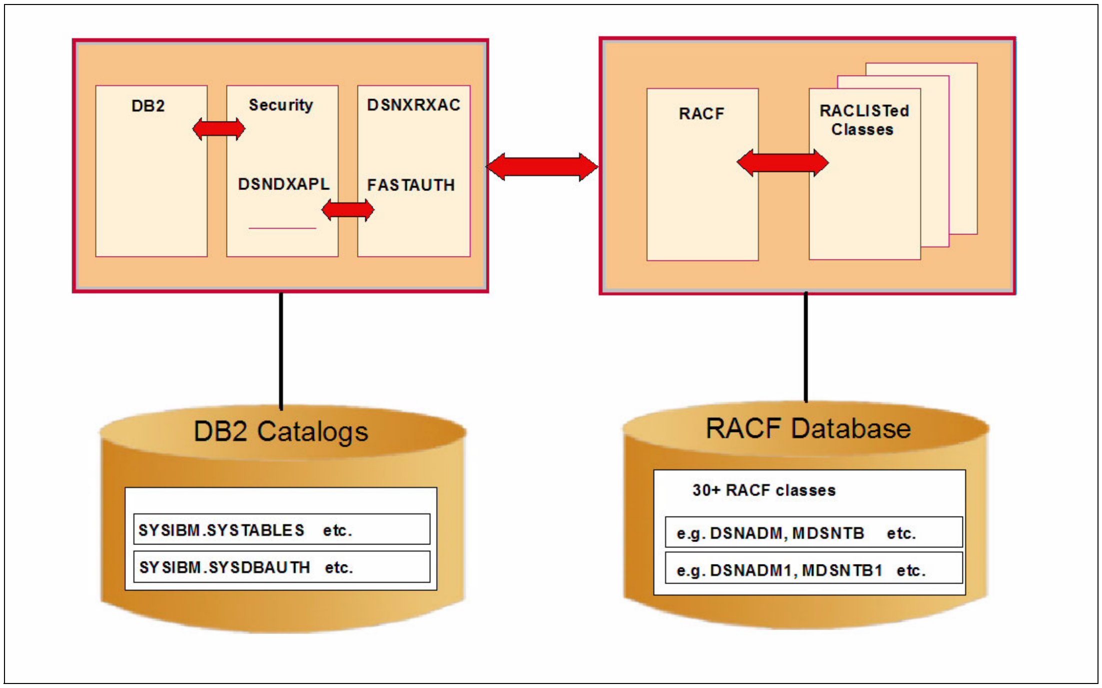


Figure 14-6 Security checking with RACF and DB2 V8

> **Note:** Prior to DB2 V8, DB2 was unable to pass RACF an ACEE for DB2 commands. IRR@X had always to defer to DB2. With DB2 V8, DB2 passes an ACEE to RACF. RACF then makes an access control decision.

14.19.4 Long name support

As DB2 extended the lengths for many of its constructs, this resulted in longer resource names in RACF. Longer DB2 resource names mean that the resource names that DSNXRXAC creates are longer as well. Several of the RACF general resource classes have had their DB2 resource name's maximum length extended from 100 characters to 246 characters (which is the maximum length for a RACF general resource name). The schema names are truncated at 100 characters when they build a RACF resource name. The RACF classes are as follows:

- MDSNTB
- DSNADM
- MDSMCL
- MDSNSG
- MDSNUT
- MDSNUF
- MDSNSC
- MDSNSP
- MDSNJR

DB2 V8 constructs

DB2 has raised the limits on many of its constructs. Since these names are used to perform authorization requests, the lengths of the resource names in these authorization checks are going to be longer, as shown in Table 14-4.

Table 14-4 DB2 V8 extended length of constructs

Object	Old max length	New max length
Collection ID	18	128
Column name	18	30
Distinct type	18	128
JAR	18	128
Package owner	8	128
Procedure	18	128
Procedure owner	8	128
Schema	8	128
Storage group	8	128
Table	18	128
Trigger	8	128
UDT	18	128
View	18	128

14.19.5 Sequence object support

DB2 V8 introduces the new object SEQUENCE, the new member class MDSNSQ, and the new grouping class GDSNSQ. A user must have one of the following privileges to use a sequence:

- Ownership of the SEQUENCE
- USAGE privilege on the SEQUENCE
- SYSADM

The USAGE privilege on the sequence is checked by verifying the user’s authority to the DB2-subsystem.schema-name.sequence-name USAGE resource in the MDSNSQ class.

Comparing identity columns and sequences

While there are similarities between IDENTITY columns and sequences, there are also differences. The characteristics of each can be used when designing your database and applications.

Identity columns

An identity column has the following characteristics:

- An identity column can be defined as part of a table only when the table is created. Once a table is created, you cannot alter it to add an identity column. (However, existing identity column characteristics may be altered.)
- An identity column automatically generates values for a single table. When an identity column is defined as GENERATED ALWAYS, the values used are always generated by the database manager. Applications are not allowed to provide their own values during the modification of the contents of the table.

Sequences

A sequence object has the following characteristics:

- A sequence object is a database object that is not tied to any one table.
- A sequence object generates sequential values that can be used in any SQL statement.
- Since a sequence object can be used by any application, there are two expressions used to control the retrieval of the next value in the specified sequence and the value generated previous to the statement being executed. The PREVVAL expression returns the most recently generated value for the specified sequence for a previous statement within the current session. The NEXTVAL expression returns the next value for the specified sequence. The use of these expressions allows the same value to be used across several SQL statements within several tables.

Creating a sequence

A sequence is a database object that allows the automatic generation of values. Sequences are ideally suited to the task of generating unique key values. Applications can use sequences to avoid possible concurrency and performance problems resulting from the generation of a unique counter outside the database.

A sequence can be created, or altered, so that it generates values in one of these ways:

- Increment or decrement monotonically without bound
- Increment or decrement monotonically to a user-defined limit and stop
- Increment or decrement monotonically to a user-defined limit and cycle back to the beginning and start again

The following is an example of creating a sequence object:

```
CREATE SEQUENCE order_seq
   START WITH 1
   INCREMENT BY 1
   NOMAXVALUE
   NOCYCLE
   CACHE 24
```

In this example, the sequence is called *order_seq*. It will start at 1 and increase by 1 with no upper limit. There is no reason to cycle back to the beginning and restart from 1 because there is no assigned upper limit. The number associated with the CACHE parameter specifies the maximum number of sequence values that the database manager pre allocates and keeps in memory.

The sequence numbers generated have the following properties:

- Values can be any exact numeric data type with a scale of zero. Such data types include: SMALLINT, BIGINT, INTEGER, and DECIMAL.
- Consecutive values can differ by any specified integer increment. The default increment value is 1.
- Counter value is recoverable. The counter value is reconstructed from logs when recovery is required.
- Values can be cached to improve performance. Pre allocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence. In the event of a system failure, all cached values that have not been committed are never used and considered lost. The value specified for CACHE is the maximum number of sequence values that could be lost.

Restriction: A sequence is not tied to a particular table column nor is it bound to a unique table column and only accessible through that table column. If a database that contains one or more sequences is recovered to a prior point in time, this could cause the generation of duplicate values for some sequences. To avoid possible duplicate values, a database with sequences should not be recovered to a prior point in time.

Nextval and prevval expressions

There are two expressions used with a sequence, as follows:

- Prevval expression

  The PREVVAL expression returns the most recently generated value for the specified sequence for a previous statement within the current application process.The same sequence number can be used as a unique key value in two separate tables by referencing the sequence number with a PREVVAL expression for any additional rows. Here is an example of adding a row:

  ```
  INSERT INTO line_item (orderno, partno, quantity)
  VALUES (PREVVAL FOR order_seq, 987654, 1)
  ```

- Nextval expression

  The NEXTVAL expression returns the next value for the specified sequence. A new sequence number is generated when a NEXTVAL expression specifies the name of the sequence. However, if there are multiple instances of a NEXTVAL expression specifying the same sequence name within a query, the counter for the sequence is incremented only once for each row of the result, and all instances of NEXTVAL return the same value for a row of the result. The same sequence number can be used as a unique key value in two separate tables by referencing the sequence number with a NEXTVAL expression for the first row. Here is an example of adding a row:

  ```
  INSERT INTO order (orderno, custno)
  VALUES (NEXTVAL FOR order_seq, 123456);
  ```

While these examples are not all of the characteristics of these two items, these characteristics will assist you in determining which to use depending on your database design and the applications using the database.

14.19.6 Multilevel security support (MLS)

DB2 V8 introduces the use of multilevel security support. It allows the automatic assignment of SECLABELs to rows in a table. To enable the use of SECLABELs at a row level, the user can either add a column and name it AS SECURITY LABEL or add the option AS SECURITY LABEL to a column which already exists. The column must be a single byte data type character (char(8)). The only technique to disable the security of a table is to drop the table, the table space, and the database. There can be only one security label in a table. The audit record IFCID 0142 is created when creating a table with the SECLABEL option.

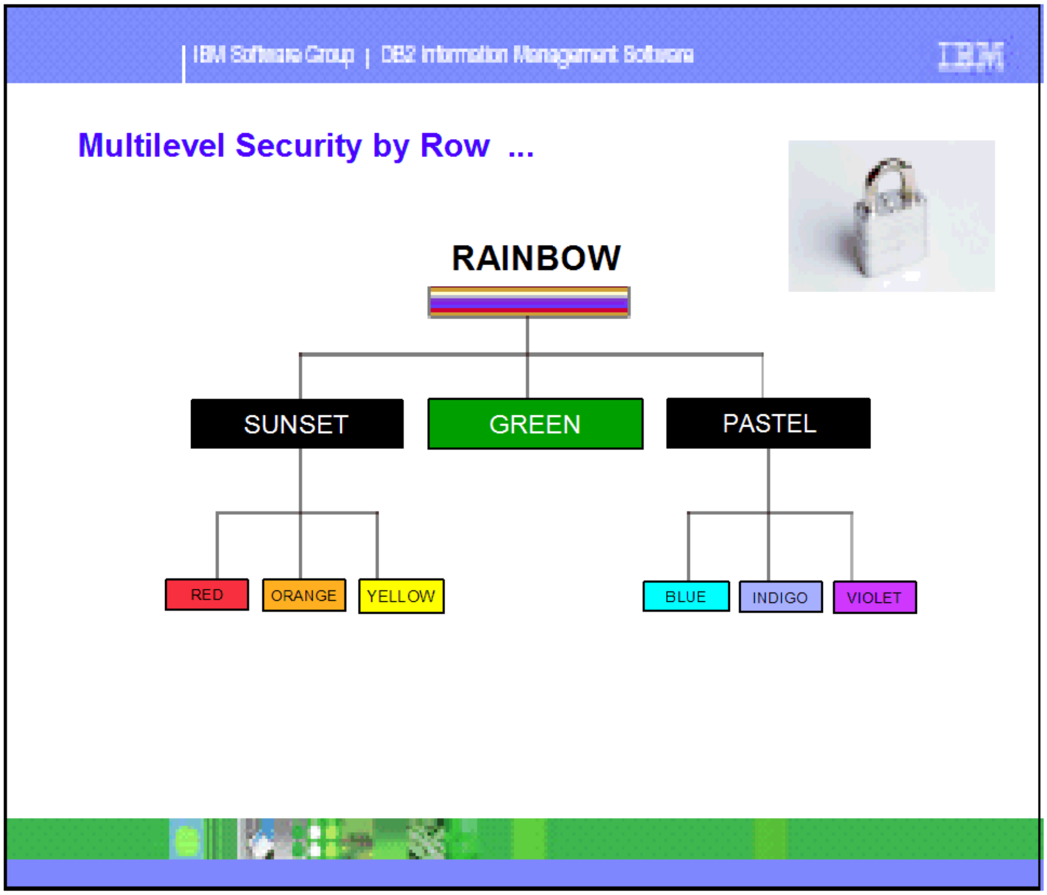
The multilevel security support also ensures that a proper relationship exists between the SECLABEL of the user and the SECLABEL of the row. The security rule for select is that your current security label must dominate the security label of all the rows read. If a user SECLABEL does not dominate the SECLABEL of the data row, then that row is not returned. The user's SECLABEL is compared to the data SECLABEL of the row to be selected. If the user SECLABEL dominates the data SECLABEL, then the row is returned. If user SECLABEL does not dominate the data SECLABEL, then the row is not included in the data returned, but no error is reported. A user must be identified to the Security Server with a valid SECLABEL. If not, an authorization error and an audit record is produced (IFCID 0140). The access rules for INSERT are similar, but the user's current SECLABEL is saved as a row is inserted. If a user does not have the write-down privilege, then the SECLABEL of inserted rows will be exactly the current SECLABEL. If the user does have the write-down privilege, then he or she can set the value of the SECLABEL column to any value.

Note: For a description of the write-down privilege, see “Write-down by user privilege” on page 340.

Understanding SECLABEL hierarchy

With the hierarchy established in the security server, the system would understand that users with authority to access SECLABEL RAINBOW, shown in Figure 14-7 on page 370, can access anything. Someone with authority to access PASTEL information can access any row associated with BLUE, INDIGO, VIOLET, or PASTEL. Someone with SUNSET can access SUNSET, RED, ORANGE, YELLOW. This is a lot more powerful than just having an exact match on security label (user's SECLABEL must exactly match the data's SECLABEL), since it has the notion of groups that make security administration easier to manage. With this additional capability, it is possible to implement that type of security scheme without requiring the application to access the data using special views or predicates. DSNXRXAC also verifies the SECLABEL on required resource checks that it performs.

Note: For a description of SECLABEL security, see “Data protection in a multilevel-secure system” on page 338.



IBM Software Group | DB2 Information Management Software

Multilevel Security by Row ...

Figure 14-7 SECLABEL hierarchy

DB2 V8 SECLABEL processing

DB2's “row-level” SECLABEL processing can be used with or without the RACF-supplied SECLABEL support in DSNXRXAC.

Additional information on DB2's row-level SECLABEL processing can be found at:

```
ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/db2security.pdf
```

An informative Web-audio presentation can be found at:

```
http://www-3.ibm.com/software/os/zseries/zserieswebcasts/030130/
```

DSNXRXAC also verifies SECLABELs on required resource checks that it performs. SLBLREQ=YES is added to selected DB2 classes. The SLBLREQ value is set to YES for the following DB2 classes:

DSNR	Connection to DB2
MDSNTB	Tables, Views, Indices
DSNADM	Administrative authorities
MDSNPN	Plans
MDSNSM	System privileges
MDSNBP	Buffer pool privileges
MDSNCL	Collections
MDSNTS	Tables spaces
MDSNSG	Storage group
MDSNUF	User defined functions
MDSNSP	Stored procedures
MDSNSC	Schema
MDSNJR	JAR (Java ARchive)

14.19.7 Refresh privilege on MQT

DB2 V8 introduces the new privilege REFRESH on tables. To refresh a table the user must have one of the following privileges:

- Ownership of the table
- DBCTRL on the database that contains the table
- DBADM on the database that contains the table
- SYSCTRL authority
- SYSADM authority

The modifications are made to check for the ownership of resources as follows:

- DSNXRXAC to check for the ownership of the table
- DBCTRL to check for the ownership of the database
- DBADM to check for the ownership of the database, SYSCTRL, and SYSADM

14.19.8 Ability for DBADM to create views for others

DB2 V7 allows users who have the DBADM privilege on a database to create aliases for other users. This option is set during the DB2 installation on panel DSNTIPP, where the XAPLCRVW is set to ON.

DB2 V8 now supports a DBADM check for creating views for other users if the XAPLCRVW bit is on. To enable this feature, user must set to YES the DBADM create view on the DSNTIPP panel.

14.19.9 Warning mode support

IRR@XACS has MSGSUPP set to YES. The ICH408I message is not issued, though the WARNING option was set for this profile.

DSNXRXAC now states MSGSUPP is set to NO. In this case, message ICH408I is issued if the profile is in WARNING mode.

14.19.10 Software prerequisites

DB2 V8 requires z/OS R3 with the exception of Multilevel Security Service (MLS), which requires z/OS R5.

> **Note:** The changes to the RACF Class Descriptor Table that enable the longer name support are being rolled back to z/OS R3. The APAR and PTF numbers for this support are not yet available.

14.19.11 Toleration

The DSNX@XAC which is shipped with DB2 V8 verifies that it is called by DB2 V8. This prevents DB2 V7 from using DSNX@XAC.

The DSNX@XAC which is shipped by RACF for DB2 V6 and DB2 V7 will be updated to check and ensure that they are called by a DB2 V6 or DB2 V7. The support will be delivered via an APAR before the general availability of DB2 V8.

14.19.12 Support for DB2 V8 on z/OS R3, z/OS R4, and z/OS R5

The new general resource class and router table entries to support the SEQUENCE object are installed on z/OS R3 and z/OS R4 by APAR, which is not yet available.

14.20 RACF and LDAP SPEs for change logging

This enhancement is for installations that have a mixed LDAP environment containing NT (Active Directory), AIX® (IBM Directory Services), and RACF, and that want to use the IBM Directory Integrator (IDI) to propagate user changes, especially password changes, between those systems. The missing link was that IDI did not know when user information in RACF changed and could not pull the un-encrypted password from RACF.

With the SPEs for RACF, SAF, and an LDAP, RACF uses a new SAF interface to request LDAP to create a change log entry for any change to a user profile, including password changes. The change log entries can be searched by IDI to determine that a change to a RACF user profile (in particular, to the password) has occurred. IDI can then issue a search to the RACF interface in LDAP and request the password envelope created by RACF.

The following changes are made available with the SPEs:

- RACF creates an LDAP change log entry for a user password change (RACF SPE, SAF SPE, and LDAP SPE).
- The metadirectory product (IDI) polls LDAP to determine if a RACF password has changed (LDAP SPE).
- The metadirectory uses LDAP to retrieve an encrypted envelope containing a clear RACF password (RACF SPE and LDAP SPE).

Note: The RACF password and envelope are kept in RACF, not in LDAP, and RACF decides who can retrieve the envelope.

14.20.1 LDAP change logging SPE

The change log is just a set of LDAP entries. They can be searched, modified, and deleted using exactly the same operations as are used on any other LDAP entry. The existing APIs, command line utilities, and LDAP browsers on any client platform (AIX, NT, z/OS, and SUN) will work with change log entries. Access to change log entries is controlled through access control lists (ACLs), as with any other LDAP entry.

Note: Change log support already exists in IDS (on AIX, NT, and OS/400®). The z/OS LDAP change log externals are similar and introduce no incompatibilities with IDS.

The LDAP Change Logging SPE has no effect on the rest of LDAP without the exploitation code in the RACF and SAF SPEs. DB2 is used by LDAP to store the change log entries.

Change logging implementation

The LDAP change logging support consists of:

- A new interface that an application can use to request that LDAP create a change log entry. The application supplies all the information needed to create the attributes in the change log entry.
- A new backend, GDBM, controlling the directory of change log entries.

  This backend is configured by adding a GDBM section to the configuration file. The GDBM section contains config options that can control the size of the change log and start or stop change logging. Change logging also uses the SDBM backend, so that must be configured as well. Finally, since RACF communicates to the LDAP server via a program call interface, the LDAP PC callable support must be configured.

  GDBM uses DB2 to store change log records. The scripts for creating the change log database are the same ones as used to create TDBM databases. Every database must have a unique owner and name.
- An enhanced SDBM search capability to retrieve the password envelope for a RACF user. This encrypted envelope contains the un-encrypted RACF password.

  RACF must be configured to instruct it to contact LDAP to create change log entries. It also must be set up to determine for which users the password should be enveloped and who has the authority to retrieve a password envelope. This configuration is described in the RACF SPE documentation.

Starting the LDAP server

When the LDAP server is started with the change log configured, it issues a message indicating whether change logging startup succeeded. The server may still start even if change logging startup fails if other backends start successfully. The following failure status message is issued so that it can be decided if the LDAP server should be stopped because change logging is not working:

```
GLD0245A Change log configuration failed and change logging is not enabled.
```

The success message indicating that logging is started is shown in Figure 14-8. The change log limits are maximum age of 86400 seconds and maximum number of entries of 1000. There are currently no change log entries.

```
GLD0244I Change logging is enabled
        Logging started status (0 = off, 1 = on): 1
        Limit in seconds on age of change log entries (0 = no limit): 86400
        Limit on the number of change log entries (0 = no limit): 1000
        Current number of change log entries: 0
        First change number in use: 0
        Last change number in use: 0
```

Figure 14-8 Successful start of the LDAP server messages

At this point, changes to a RACF user profile result in the creation of change log entries. When change logging is configured for the first time, the LDAP server creates the change log root entry. The initial ACL on this entry restricts access to the LDAP administrator (adminDN value in the configuration file).

Users authorized by the new ACL can now search, delete, and modify change log entries. Any LDAP client on any platform can be used to do this. A user cannot add a change log entry; only the LDAP server can do this.

The rootDSE entry contains information on the change log. Its location and the lowest and highest change numbers in use are present (each change log entry is identified by a unique change number). An application using the change log should use the rootDSE information to perform searches on the change log.

Using the change log

A change log entry created due to a RACF password change includes the change's attribute containing the specified string. This indicates to the application using the change log that the password has changed. The application can retrieve an encrypted envelope from RACF containing the new password by issuing an LDAP search request to the SDBM backend. SDBM does not keep the RACF password envelope and only a user authorized to read the envelope is able to obtain it via SDBM.

In the current RACF usage of the LDAP change log, the change's attribute is not filled in for changes to the non-password values in the RACF user profile. In this case, there is nothing in the change log entry to indicate which values have changed. The application using the change log must determine how it wants to proceed. It can use SDBM to retrieve the entire user profile or just the values that are important to it, and then process these values. Note that at this time there is no convenient way to retrieve the RACF password envelope along with all the other user profile values.

The application using the change log should regularly remove change log entries after it has processed them. This keeps the change log within the configured limits and avoids the possibility that the LDAP server will have to delete entries due to the limits set in the configuration file.

Change log entries are identified by an always increasing change log number, starting at 1. Numbers are assigned in the order in which LDAP is contacted to create change log entries (which depends on RACF to be in the same order as the changes occurred). Change log numbers can be skipped; you might get 1, 2, 4, 5.

GDBM is a new backend for change logging. Although it is very similar to the TDBM backend, GDBM can only be used to store change log information, while TDBM can be used to store any sort of information. Like TDBM, GDBM uses DB2 to store its entries. GDBM is configured by adding a GDBM section to the configuration file. There are several new configuration options that allow the user to control when change log entries get removed and whether to allow change logging.

Change log SPEs

This enhancement requires DB2 V5 or later, along with the following SPEs:

- LDAP Change Logging SPE: APAR OA03857.
- Currently RACF is the only exploiter. RACF Event Notification and Password Envelope SPE: APAR OA03853.
- SAF SPE: APAR OA03854.

Migration considerations

The hard-coded suffix used by the change log is cn=changelog. The LDAP server does not start if change logging is configured and any other suffix overlaps cn=changelog. To resolve this, before configuring change log in the configuration file, add another suffix to the TDBM backend in the configuration file. Then, start the LDAP server, and use the modRdn operation to move the overlapping suffix to the new suffix. Then the server can be stopped, the overlapping suffix removed from the configuration file, and change logging can be configured.

Although the TDBM minimum schema is updated in the LDAP change logging SPE, this should not cause any problem to an existing TDBM database. The next time the TDBM database is used, the LDAP server detects that there are additional attributes and object classes and adds them to the TDBM schema.

Sysplex migration considerations

All the servers sharing the change logging database must specify the same database options in their configuration files. They should also specify the same change logging size limits. If there is a server on which RACF changes should not be logged, change logging can be turned off in that server's configuration file. This server can still be used to search the change log. Change logging must be on in the server(s) where RACF changes to be logged are made.

Note: For additional information, see *Security Server LDAP Server Administration and Use*, SC24-5923. There is a Security Server LDAP Change Log document installed as /usr/lpp/ldap/doc/changelog.pdf.

14.21 Password enveloping

Password enveloping is the processing of a plaintext password to make it safe to store in the RACF database. The extraction of the password from the RACF database and subsequent enveloping of the password to be sent to the requestor is now possible with the changes made to the RACF address space to make it function properly when running code which exploits UNIX System Services.

To prevent an authorized application from retrieving the clear text of a password stored in the RACF database, the password is now encrypted. To encrypt it, a private key is used. This private key is part of a key ring which is associated with the RACF subsystem address space (RASP). The encrypted password is then stored in the user ID profile in the RACF database. The process of enveloping a password is made up of the following:

- The PASSWORD.ENVELOPE RACF profile
- Signing hash algorithm and encryption strength used to create the password envelope

- The IRR.PWENV.KEYRING key ring
- The IRR.RADMIN.EXTRACT.PWENV RACF profile
- The NOTIFY.LDAP.USER profile
- RACF sets up the password.envelope RACF profile

Note: When the password enveloping function is used, approximately 280 bytes of storage is used in the USER profile to contain the enveloped password.

Note: For the most part, any new password is enveloped for an eligible user, with the following exceptions:

- Initial **ADDUSER** passwords.
- When the new password is the same as the current password.
- When the **ALTUSER** or **PASSWORD** command is used to change the password, and the new password is equal to the user ID's default group name .
- When an application uses RACROUTE or ICHEINTY (as opposed to a RACF command) to set the password, and the password contains characters which would not be accepted by the RACF commands. RACF commands only accept the characters A to Z, 0 to 9, and the variant characters X'5B' (typically $), X'7B' (typically #), and X'7C' (typically @).
- When an application uses RACROUTE or ICHEINTY to set the password and specifies ENCRYPT=NO.

14.21.1 The PASSWORD.ENVELOPE profile

The PASSWORD.ENVELOPE profile in the RACFEVNT class controls whether new passwords are enveloped for a given user. If the user whose password is being changed has at least READ access to this resource, then the new password is enveloped.

Note: Generic characters may not be used in this profile name.

The enveloped password is neither displayed by **LISTUSER** nor unloaded by IRRDBU00, although IRRDBU00 indicates the presence of a password envelope for a given user with a YES/NO field.

There are no SMF records created as a result of failed access checks to this resource. Audit options in the profile can be used to log successes, and thus maintain a history of whose passwords have been enveloped.

If the user cannot be verified (RACROUTE REQUEST=VERIFY), then a new password is not enveloped. For example, if a revoked user ID password is changed by an administrator, the new password is not enveloped. If you want the new password to be enveloped in this scenario, specify RESUME on the same **ALTUSER** command in which the PASSWORD keyword is used to specify the new password.

14.21.2 Signing hash algorithm and encryption strength

Both the signing hash algorithm and encryption strength are configurable attributes. They are used to create the password envelope.The APPLDATA of the PASSWORD.ENVELOPE profile is used to specify the signing hash algorithm used to sign the PCKS#7 password envelope and the encryption strength used when encrypting the password envelope. The syntax of the APPLDATA string consists of a character string indicating the signing hash algorithm, followed by a forward slash (/), followed by a string indicating the encryption strength. The following values are allowed for the signing hash algorithm:

- MD5
- SHA1
- Default=MD5

The values allowed for the encryption strength are shown in Table 14-5 on page 377. The default is STRONG. It is recommended that you use the strongest encryption possible. If you are forced to use weaker encryption—for example, due to export regulations—then protect yourself against offline attacks by carefully controlling access to the RACF database, and any other repository in which password envelopes may be stored after being retrieved from RACF.

Table 14-5 Data encryption values

Value	Data encryption method
STRONG	Triple DES; a 168-bit encryption key.
MEDIUM	DES; a 56-bit encryption key.
WEAK	RC2; a 40-bit encryption key.

If the APPLDATA is specified incorrectly, an error message is issued to the console the next time a user who is eligible for password enveloping changes his password, or the next time an application requests the retrieval of a password envelope, and the defaults are used. The APPLDATA can be changed at any time.

Note: Strong encryption may not be available on all user systems depending on government export regulations.

14.21.3 The IRR.PWENV.KEYRING key ring

IRR.PWENV.KEYRING is the name of a key ring. It is associated with the identity of the RACF subsystem address space (RASP). It contains a certificate with private key for the RASP itself. This certificate is used to encrypt a user's new password. It is also used to decrypt the stored password when a PKCS#7 envelope is retrieved by an authorized application. The contents of the returned envelope are signed using this certificate. IRR.PWENV.KEYRING also contains certificates of all the principals who are intended to retrieve a user ID's changed password from RACF. A changed password is encrypted using the public keys contained within these certificates. RACF will encrypt passwords for up to 20 certificates on this key ring.

Note: The name of the key ring is case sensitive.

14.21.4 The IRR.RADMIN.EXTRACT.PWENV profile

IRR.RADMIN.EXTRACT.PWENV is a FACILITY class resource which authorizes the retrieval of the enveloped password from RACF using a new extension to the r_admin callable service (IRRSEQ00). Audit options in the profile can be used to log successes, and thus maintain a history of whose passwords have been retrieved, and by whom. Failures can also be logged. The logstring identifies the user whose password has been retrieved.

14.21.5 The NOTIFY.LDAP.USER profile

If the NOTIFY.LDAP.USER profile is defined, then an LDAP change log entry is created when a user ID's password is enveloped. In the case of a password change, a change log entry is only created if the user's password was enveloped. This change log entry contains the change's attribute identifying the password field as the changed field. The change's attribute does not contain the actual password value, but contains a value of ComeAndGetIt, denoting that there is an encrypted password envelope which can be subsequently retrieved. If other RACF profile fields are changed in the same request which updates the password (for example, ALTUSER SOMEUSER PASSWORD(NEWPASS) RESUME), then two change log entries are created: one to describe the password update, and another to describe the non-password update).

14.21.6 RACF setup

The steps to set up RACF to envelope passwords are the following:

- Set up password enveloping.
- Prepare RACF.
- Define a local certificate authority certificate using RACF as the certificate authority.
- Generate an X.509V3 certificate for RACF user IDs to use for user password enveloping.
- Activating password enveloping.
- Enabling RACF for heterogeneous password synchronization.

Setting up password enveloping

When a user changes his password, if there are errors during the password enveloping process RACF detects them. Error messages are reported to the console, not to the end user who is initiating a password change.

Preparing RACF

The user must run IRRMIN00 with PARM=UPDATE. If using z/OS V1R5, re-IPLing is not required, as described in "PARM=UPDATE" on page 360. An OMVS segment must be added to the RACF address space (RASP) user ID and to its default group. This is mandatory to set the LE run-time environment required to use the System SSL functions unless you have set up a default UNIX identity (by implementing the BPX.DEFAULT.USER profile in the FACILITY class). Do not set up the default UNIX identity just for the password enveloping function. You can specify any UID value you choose. For example, if the RACF subsystem is running under a user ID of RACFSUB, whose default group is STCGRP, use the following command:

```
ALTUSER RACFSUB OMVS(AUTOUID HOME(/) PROGRAM(/bin/sh)) ALTGROUP STCGRP
OMVS(AUTOGID)
```

Defining a local certificate authority certificate

If you plan to use RACF as your certification authority (CA), you have to create a certificate authority certificate. The following example creates a certificate called RACFCA used to create certificates. One of them could be the one that RACF uses to envelope the passwords. The following command creates a certificate:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('RACFCA') O('ibm') C('us'))
WITHLABEL('RACFCA') NOTAFTER(DATE(2020-12-31)) ICSF
```

Note: RACF signs the password envelope so that a recipient can verify that the envelope signature was created using RACF's certificate (which is created in the next step). If the recipient also wants to check the veracity of RACF's certificate, it needs this CA certificate to do so. In this case, the CA certificate must be exported to a key ring known to the recipient application and marked as trusted.

Generating an X.509V3 certificate

Use the following steps to generate an X.509V3 certificate:

1. A digital certificate containing a private key must be generated for RACF's use. This certificate and its associated private key are used in the RACF password enveloping process. In this example, **`RACDCERT`** commands are used to generate a certificate for the RACF address space whose RACF user ID is RACFSUB. This certificate is signed by RACF, which is the certificate authority in the context of this example. The local CA certificate is identified by the label RACFCA. The following command generates a certificate:

```
RACDCERT ID(RACFSUB) GENCERT SUBJECTSDN(CN('RACF AddrsSpace System1')O('ibm')
C('us') WITHLABEL('RASP1')SIGNWITH(CERTAUTH LABEL('RACFCA')) KEYUSAGE(HANDSHAKE
DATAENCRYPT DOCSIGN) NOTAFTER(DATE(2020-12-31)) ICSF
```

Note: If you change the user ID under which the RACF subsystem runs, then you will need to create and populate the key ring for this new identity as described in the following steps.

It is recommended that you use ICSF to store private keys, and this example reflects that recommendation. If you do not use ICSF, then omit this keyword from the command.

After performing the setup, avoid changing RACF' s private key. If you do, then RACF is not able to build PKCS#7 envelopes for existing passwords (since the passwords were encrypted under the old public key, they cannot be decrypted under the new private key). Normal operation resumes as users subsequently change their passwords.

2. Create a RACF key ring named IRR.PWENV.KEYRING. Note that the name of the key ring is case sensitive. The following command creates the keyring:

```
RACDCERT ID(RACFSUB) ADDRING(IRR.PWENV.KEYRING)
```

3. Connect RACF's certificate to the key ring as the default certificate using the following command:

```
RACDCERT ID(RACFSUB) CONNECT(LABEL(‘RASP1’) RING(IRR.PWENV.KEYRING) DEFAULT
USAGE(PERSONAL)).
```

4. RACF's certificate must be defined as the default certificate. These examples create a certificate which is TRUSTED. Use the **`RACDCERT LIST`** command to verify that. This also applies to the certificates which are created in the following steps. Use the following command to mark it trusted:

```
RACDCERT ID(RACFSUB) ALTER (LABEL(‘RASP1’)) TRUST
```

5. At the time a user's password is changed, if the user is eligible for enveloping then the user's new password is encrypted under the public key of the default certificate only, and stored back in the user's USER profile. A user is eligible for enveloping if the certificate that identifies it is loaded into the key ring IRR.PWENV.KEYRING which is associated with the user ID assigned to the RACF subsystem address space.

6. During the RACF password enveloping process, RACF encrypts data which can only be recovered by the intended recipient of that data. An intended recipient, such as the identity of the IBM Directory Integrator process which may be running off of the z/OS platform, is

identified by an X.509V3 certificate. Note that these certificates are only used to encrypt password information. Generally speaking, you should only permit trusted application identities (not humans) to recover user passwords. Certificates for intended recipients may be created by RACF, and exported to off-platform processes for instance. The creation of the certificates may be accomplished using the following **RACDCERT** commands, which generate certificates for IDI1 and APP2; in this example IDI1 and APP2 are the identities of processes that are authorized to retrieve RACF password envelopes. These certificates are signed with the local CA (RACF) certificate that is identified by the label RACFCA.

```
RACDCERT ID(IDI1) GENCERT SUBJECTSDN(CN('IBM Directory Integrator Server 1')
O('ibm') C('us')) WITHLABEL('IDI1') SIGNWITH(CERTAUTH LABEL('RACFCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(APP2) GENCERT SUBJECTSDN(CN('Application Server 2') O('ibm')
C('us')) WITHLABEL('APP2') SIGNWITH(CERTAUTH LABEL('RACFCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

The KEYUSAGE attributes HANDSHAKE, DATAENCRYPT, and DOCSIGN are mandatory for the certificates. You may also create certificates directly on the recipient platform and import them into RACF. Any key management system can be used to create the recipient key pair and certificate, as long as it can export certificates in an industry standard format understood by the RACF **RACDCERT** command.

Activating password enveloping

Use the following steps to activate password enveloping:

1. Define in the RACFEVNT class the profile PASSWORD.ENVELOPE which controls password enveloping. The APPLDATA of this profile is used to specify the signing hash algorithm used to sign the PCKS#7 password envelope, and the encryption strength used when encrypting the password envelope. Use the following command to specify the strongest signing and encryption:

   ```
   RDEFINE RACFEVNT PASSWORD.ENVELOPE UACC(NONE) APPLDATA('MD5/STRONG')
   ```

2. Enable password enveloping for users whose passwords are to be encrypted for the intended recipients whose digital certificates were set up previously. As an example, this is done by executing the following command:

   ```
   PERMIT PASSWORD.ENVELOPE CLASS(RACFEVNT) ID(USER1 USER2 GROUPA GROUPB)
   ACCESS(READ).
   ```

3. The profile may be given UACC(READ) to activate system-wide password propagation, or you can permit specific users and groups. However, assigning a UACC(READ) to the PASSWORD.ENVELOPE profile will also enable the enveloping of passwords for highly privileged users (that is, system SPECIAL users, or users which are defined for emergency recovery purposes). This approach is not recommended, unless sensitive user IDs are specifically permitted to this resource with ACCESS(NONE). (This is not necessary for PROTECTED user IDs.) Ultimately, of course, this is up to the policy of the individual installation. If a non-global approach is taken, keep in mind that over time, one must consider how newly added users and groups fit into the access scheme, so that users' passwords are properly enveloped or not. In particular, you may want to have intended group connections in place for new users before their initial logons when they must change their passwords. Also, keep in mind that if a user is connected to several groups, his effective authority is the highest allowed by any of his groups (assuming he is not specifically permitted by user ID, in which case this authority overrides that granted by any of his groups). For example, if list-of-groups processing (**SETROPTS GRPLIST**) is active, and user BOB is connected to groups GROUP1 and GROUP2, and GROUP1 is permitted with ACCESS(NONE), and GROUP2 is permitted with ACCESS(READ), and BOB is not

explicitly permitted, then BOB's effective access to PASSWORD.ENVELOPE is READ, and BOB's password will be enveloped.

Note: If you use the UACC(READ) approach, be aware that when you have users with the RESTRICTED attribute that they must be explicitly permitted to the PASSWORD.ENVELOPE profile, by user ID or by group name, if you want their passwords to be enveloped.

4. Optionally, an LDAP change log entry can be created whenever a user's new password is enveloped. This will be a required step if you use an application like IBM Directory Integrator to implement a heterogeneous password synchronization solution.
5. Activate the function by activating the RACFEVNT class using as it can be RACLISTed to improve performance, but this is not required:

   ```
   SETROPTS CLASSACT(RACFEVNT) RACLIST(RACFEVNT).
   ```
6. You must stop and restart the RACF subsystem address space after defining the PASSWORD.ENVELOPE profile and activating the RACFEVNT class. If the RACF subsystem address space is already up and running when you configure password enveloping and you do not stop and restart the address space, it will not have the proper environment set up to perform the function, and will fail any requests to envelope passwords.

14.21.7 Password retrieve

The encrypted password can be retrieved using R_admin (IRRSEQ00) callable service. When someone calls R_ADMIN (XTR_PWD_ENVELOPE), an IRRLOG00 command is built and sent into the RASP. The data in the PWDENV field in the RACF database is read. Using the private key (and cert) from the default certificate in the IRR.PWENV.KEYRING owned by the RASP user ID, the ensconced enveloped data message is opened which was created at the time of password change. Then, a PKCS#7 signed data message is created using the password payload. The signed data message is the default certificate in IRR.PWENV.KEYRING. The signed data message is then encoded/encrypted into another enveloped data message. All certificates (except the default which is used as signer) in the IRR.PWENV.KEYRING are intended recipients of this enveloped data message. In order to read the message, the recipient must have the private key corresponding to one of the recipient certificates. The initial password payload is never decoded from its DER-encoded form. The original ensconced copy of the password payload remains intact in the RACF database. The new enveloped data message (with all of the recipients) is not stored by RACF in any form. To authorize the use of R_admin API, issue the following commands:

```
RDEFINE FACILITY IRR.RADMIN.EXTRACT.PWENV UACC(NONE)
PERMIT IRR.RADMIN.EXTRACT.PWENV CLASS(FACILITY) ID(IDI1 APP2) ACCESS(READ)
```

14.22 LDAP event notification

You can configure RACF so that it creates LDAP change log entries in response to changes to RACF user profiles. An LDAP client can read the LDAP change log, detect updates to RACF users, and retrieve RACF user entries using only LDAP interfaces. Event notification, through the creation of an LDAP change log entry, is controlled by the NOTIFY.LDAP.USER resource in the new RACFEVNT class. If RACFEVNT is active, and the NOTIFY.LDAP.USER resource is protected (by either a discrete or generic profile), then LDAP change log entries created for the following types of user updates:

- Password changes, regardless of the interface used, as long as the new password is enveloped
- Updates to a user's revoke status (that is, changes to the FLAG4 field in the USER profile), regardless of the interface used
- User additions made using the **ADDUSER** command
- User modifications made using **ALTUSER** and **PASSWORD** commands
- User deletions made using the **DELUSER** command for the NOTIFY.LDAP.USER profile acting on a system-wide basis

The LDAP change log entry contains information such as the change initiator, the affected user, the type of update (add, modify, or delete), and the time and date of the change. It does not contain a list of fields which were changed, nor does it contain the new values for these fields. In the case of a password change, a change log entry will only be created if the user's password was enveloped. This change log entry contains the change's attribute identifying the password field as the changed field. The change's attribute does not contain the actual password value, but contains a value of ComeAndGetIt, denoting that there is an encrypted password envelope which can be subsequently retrieved.

Consider an example where other RACF profile fields are changed in the same request which updates the password, for example:

```
ALTUSER SOMEUSER PASSWORD(NEWPASS) RESUME)
```

In this case, two change log entries are created: one to describe the password update, the other to describe the non-password update. See APAR OA03857 for more information on the change log.

14.22.1 LDAP event notification when LDAP server is down

If the LDAP server is unavailable at the time the RACF change occurs, then that change log entry is lost. There is currently no queuing mechanism whereby a change notification can be retried at a later point in time. This does not affect the RACF database itself; LDAP notification is attempted only after the RACF database has been updated.

14.22.2 LDAP event notification and RRSF

Applications which exploit LDAP change log entries for registry synchronization must take network topology into account when propagating locally initiated RACF changes to other z/OS/RACF systems in the network. In particular, if RACF is configured in an RRSF network, and user or password updates are being kept in sync across RRSF nodes, then application deployment must include consideration of which propagation mechanism is used for specific types of changes to specific systems. Neglecting the interaction of the various propagation mechanisms could result in an unending cascade of updates for the same RACF change. For example, for an RRSF network which fully mirrors user profile and password updates, an LDAP-based propagation mechanism should only communicate with a single RRSF node, and let that node propagate the change to other RACF nodes. Further, this RACF node should be the only node configured to perform LDAP event notification for user updates.

14.22.3 R_proxyserv (IRRSPY00)

R_proxyserv (IRRSPY00) callable service has been updated to provide an interface for authorized applications to create their own LDAP change log entries for updates made using interfaces other than the RACF commands. This is done using the new function code 3 (PRXY_CHANGELOG).

14.22.4 LDAP event notification activation

Do the following to activate the LDAP event notification:

1. Define the RACFEVNT class profile named NOTIFY.LDAP.USER:

   ```
   RDEFINE RACFEVNT NOTIFY.LDAP.USER
   ```

   A generic profile could also be used.

2. Activate the RACFEVNT class:

   ```
   SETROPTS CLASSACT(RACFEVNT)
   ```

14.22.5 RACF changes

The RACF code is changed to exploit UNIX System Services (USS). The **IRRLOG00** command calls IRRPWC00 via CEEPIPI. IRRPWC00 code uses the POSIX(ON) services. Then RASP is considered as a UNIX process. Now that RASP calls USS services (and gets dubbed by USS), the entire RASP is considered a USS process and is shut down if someone does a **F OMVS,SHUTDOWN** command from the console.

14.22.6 New messages

The following new messages were added:

```
IRRB023I SYSTEM SERVICE BPX1xxx FAILED WITH RETURN CODE A REASON CODE B.

IRRC141I THE PASSWORD ENVELOPING FUNCTION CANNOT BE PERFORMED FOR USER user.  A
PROPER UNIX SYSTEM SERVICES ENVIRONMENT DOES NOT EXIST FOR THE RACF SUBSYSTEM
RUNNING UNDER USER ID raspuser
IRRC130I SYSTEM SSL FUNCTION x RETURNED ERROR CODE nnn DURING OPERATION NUMBER
opcode WHILE PROCESSING THE PASSWORD ENVELOPE FOR USER name.
```

14.22.7 Debug aid

System SSL trace is the built-in tracing system for the System SSL calls that are made. It is activated using **@SET trace(SYSTEMSSL)** from the console. The trace files appear in /tmp with names like gskssl.racf.pid.trc. Use the gsktrace tool to format the trace.

Note: The old versions of the gsktrace formatting tool do not work. You must copy the trace file to R15 or higher and run it there.

14.23 Enterprise Identity Mapping (EIM)

In today’s heterogeneous networks with partitioned servers and multiple platforms, administrators, users, and application developers all have to cope with the complexities that multiple user identities for individual users create within an enterprise. Users have to remember each user ID and password for each system they use. Administrators must perform password resets, attempt to synchronize user IDs and passwords, and remember every system in the network to which each individual has access. Application developers are often forced to use nonsecure techniques to solve this problem or to invest large amounts of money in writing applications that implement their own user registries and associated security semantics.

These problems quickly become a large administrative problem for all parties involved. One approach to handle a single sign-on environment is to create side-files containing all the users passwords and user IDs. This approach has several flaws. The passwords still need to be managed on these systems and the registries require that the user/password lists are synchronized. The passwords are commonly transmitted in clear-text, and also stored in clear-text or decryptable files directly accessible by the administrator.

Enterprise Identity Mapping (EIM) can solve these problems for administrators, programmers, and users. This section describes EIM and how to implement it in your environment.

The two key elements of EIM are the following:

- **User registry**: This registry operates like a directory and contains a list of valid user identities for a particular system or application. Other examples of common user registries are a Kerberos key distribution center (KDC) and the OS/400 user profiles registry.
- **EIM identifier**: Represents a person or entity in an enterprise. A typical network consists of various hardware platforms and applications and their associated user registries. Most platforms and many applications use platform-specific or application-specific user registries. These user registries contain all of the user identification information for users who work with those servers or applications. When you create an EIM identifier and associate it with the various user identities for a person or entity, as shown in Figure 14-9, it becomes easier to build heterogeneous, multiple-tier applications, for example, a single sign-on environment. When you create an EIM identifier and associations, it also becomes easier to build and use tools that simplify the administration involved with managing every user identity that a person or entity has within the enterprise.

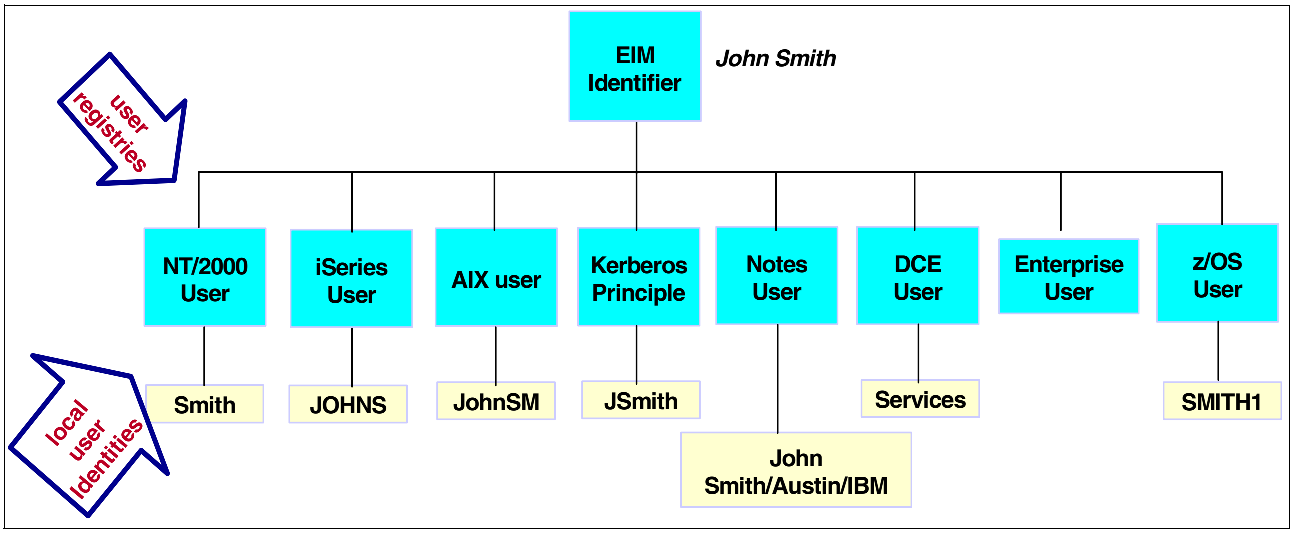


Figure 14-9 User registry with user identities

14.23.1 EIM server view

The EIM domain controller is a Lightweight Directory Access Protocol (LDAP) server that is configured to manage at least one EIM domain, as shown in Figure 14-10 on page 385. An EIM domain is an LDAP directory that consists of all the EIM identifiers, EIM associations, and user registries that are defined in that domain. Systems (EIM clients) participate in the EIM domain by using the domain data for EIM lookup operations. A minimum of one EIM domain controller must exist in the enterprise. On the EIM domain controller, the EIM applications can perform either EIM associations and EIM lookup.

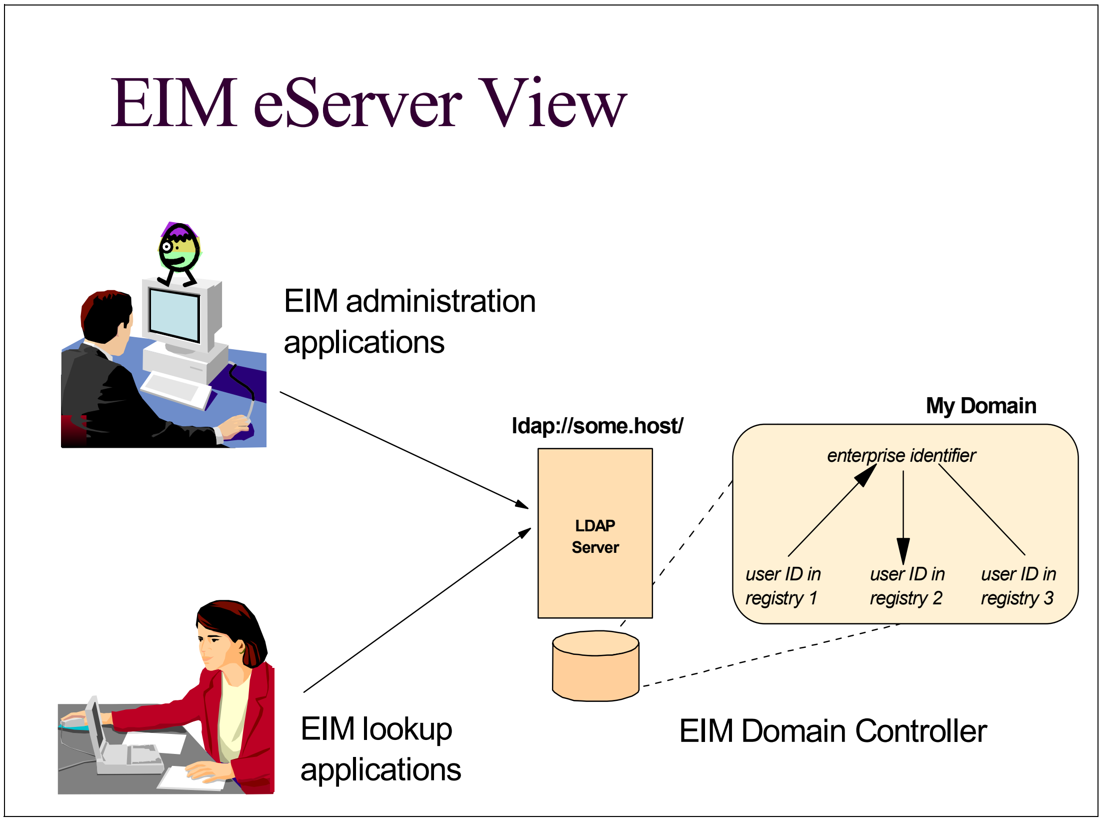


Figure 14-10 EIM eServer view

EIM associations

An EIM association is a relationship between an EIM identifier that represents a specific person and a single user identity in a user registry that also represents that person. When you create associations between an EIM identifier and all of a person's or entity's user identities, you provide a single, complete understanding of how that person or entity uses the resources in an enterprise. EIM provides APIs that allow applications to find an unknown user identity in a specific (target) user registry by providing a known user identity in some other (source) user registry. This process is called identity mapping. Before you can create an association, you first must create the appropriate EIM identifier and the appropriate EIM registry definition for the user registry that contains the associated user identity. An association defines a relationship between an EIM identifier and a user identity by using the following information:

- EIM identifier name
- User identity name
- EIM registry definition name
- Association type

An administrator can create different types of associations between an EIM identifier and a user identity based on how the user identity is used. User identities can be used for authentication, authorization, or both.

Authentication ensures that an entity or person who provides a user identity has the right to assume that identity. Verification is often accomplished by forcing the person who submits the user identity to provide secret or private information associated with the user identity, such as a password.

Authorization ensures that a properly authenticated user identity can only perform functions or access resources for which the identity has been given privileges. By using EIM lookup operations, applications now can use user identities in one user registry for authentication while using associated user identities in a different user registry for authorization.

In EIM, the three types of associations that an administrator can define between an EIM identifier and a user identity are as follows:

- **Source association:** When a user identity is used for authentication, that user identity should have a source association with an EIM identifier. A source association allows the user identity to be used as the source in an EIM lookup operation to find a different user identity that is associated with the same EIM identifier. If a user identity with only a source association is used as the target identity in an EIM lookup operation, no associated user identities are returned.
- **Target administration**: When a user identity is used for authorization rather than for authentication, that user identity should have a target association with an EIM identifier. A target association allows the user identity to be returned as the result of an EIM lookup operation. If a user identity with only a target association is used as the source identity in an EIM lookup operation, no associated user identities are returned. It might be necessary to create both a target and a source association for a single user identity. This is required when an individual uses a single system as both a client and a server, or for individuals who act as administrators. For example, a user normally authenticates to a Windows platform and runs applications that access an AIX server. Because of the user's job responsibilities, the user must occasionally also log directly into an AIX server. In this situation you would create both source and target associations between the AIX user identity and the person's EIM identifier. User identities that represent end users normally need a target association only.
- **Administrative association**: An administrative association for an EIM identifier is typically used to show that the person or entity represented by the EIM identifier owns a user identity that requires special considerations for a specified system. This type of association can be used, for example, with highly sensitive user registries. Due to the nature of what an administrative association represents, an EIM lookup operation that supplies a source user identity with an administrative association returns no results. Similarly, a user identity with an administrative association is never returned as the result of an EIM lookup operation.

14.23.2 EIM configuration

There are three basic configurations for an EIM domain. They are illustrated in Figure 14-11 on page 387, and defined as follows:

- A single domain controller is used by all EIM client applications. (This is shown in the top of the figure.)
- The LDAP server is configured to route requests to an EIM domain to another LDAP server using LDAP referrals. (This is pictured in the lower left of the figure.)
- A domain controller uses LDAP master/replica functions to provide a local copy of a domain for fast reference and backup in case the master server is unavailable. The configuration is supported when the master and replica are on the same z/OS system. (This is shown in the bottom right corner of Figure 14-11.)

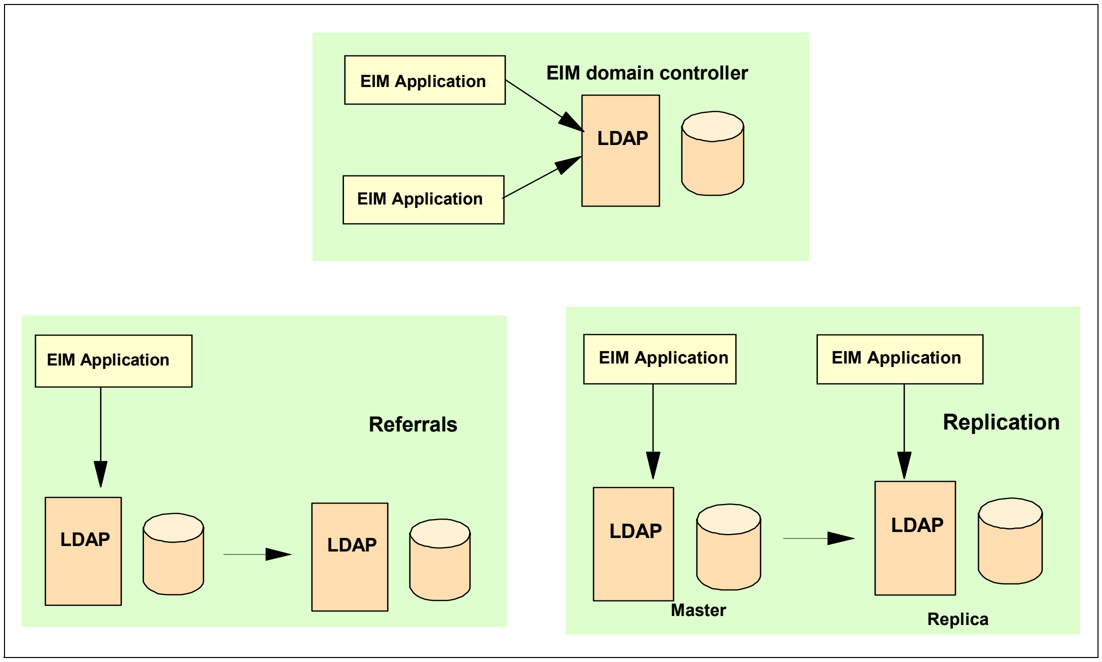


Figure 14-11 Basic EIM configurations

14.23.3 EIM authentication

The EIM domain controller supports the following for authentication:

- Simple
- Simple with CRAM-MD5 password protection
- External (digital certificates)
- GSSAPI (Kerberos)
- Secure socket layer (SSL)

Regarding security, five authentication methods are supported:

- Simple authentication
- Certificate authentication
- Kerberos credentials authentication
- CRAM-MD5 authentication
- DIGEST-MD5 authentication

Simple authentication

With simple authentication, a user ID and password are sent (in the clear) from the client to the server in order to establish who is contacting the LDAP server for information.

Secure Socket Layer (SSL) or Transport Layer Security (TLS) can be used to secure the socket connection between the client and the server by encrypting the data transferred over the connection. TLS is based upon SSL V3. Through a protocol handshake between the client and server, the choice of TLS or SSL is decided, with TLS being the preferred protocol. In the case of a simple bind, the encryption protects the password.

Certificate authentication

With certificate authentication, the identity from the client certificate sent to the LDAP server on an SSL/TLS socket connection is used to establish who is contacting the LDAP server for information. Certificate authentication is also referred to as SASL external bind and is provided by the ldap_sasl_bind API.

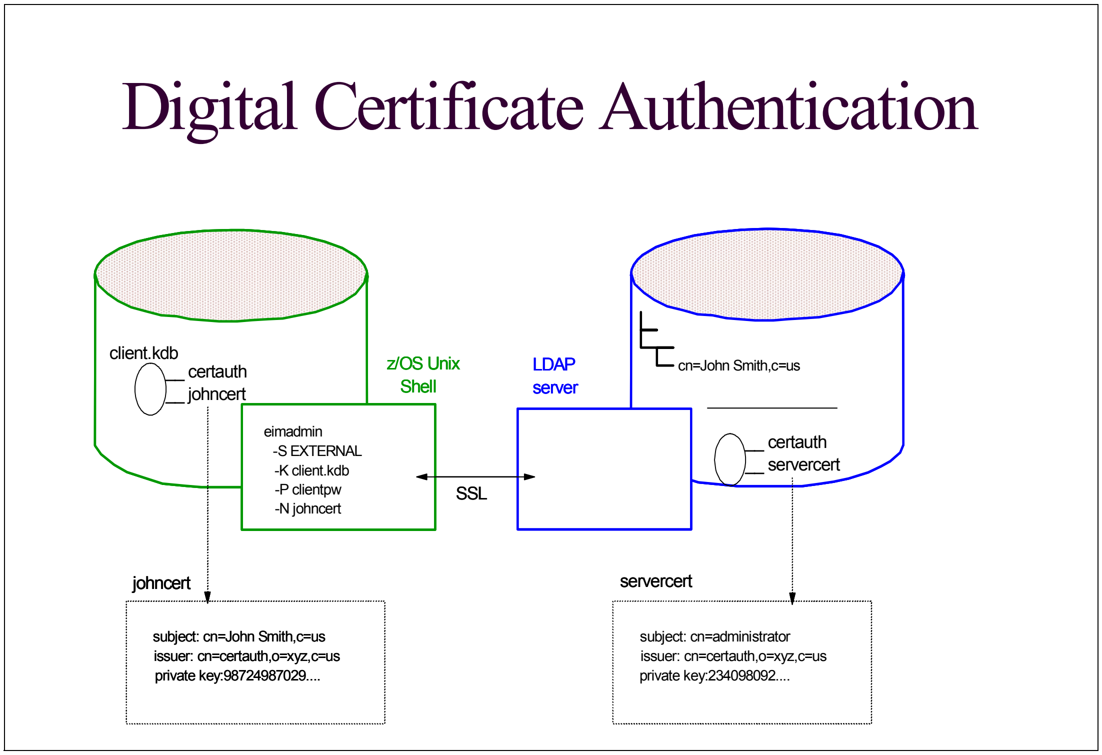
Kerberos credentials authentication

With Kerberos credentials authentication, a client application and an LDAP server accepting Kerberos authentication mutually authenticate each other using a Key Distribution Center (KDC). The identity is determined by algorithms on the server. Kerberos authentication is also referred to as SASL GSS API bind and is provided by the ldap_sasl_bind API.

CRAM-MD5 and DIGEST-MD5 authentication

With CRAM-MD5 and DIGEST-MD5 authentication, authentication is accomplished in a series of challenges and responses between the client application and server. The response from the client application to the server has a hashed password that is calculated by using an algorithm that is known by both the client application and server. The server checks to make certain that the authentication is correct by calculating its own password hash and comparing it to the client-calculated password hash. CRAM-MD5 and DIGEST-MD5 authentication is provided by the ldap_sasl_bind API.

Figure 14-12 is an example of a digital certificate authentication.



Digital Certificate Authentication

```
eimadmin
  -S EXTERNAL
  -K client.kdb
  -P clientpw
  -N johncert
```

johncert

subject: cn=John Smith,c=us
issuer: cn=certauth,o=xyz,c=us
private key:98724987029....

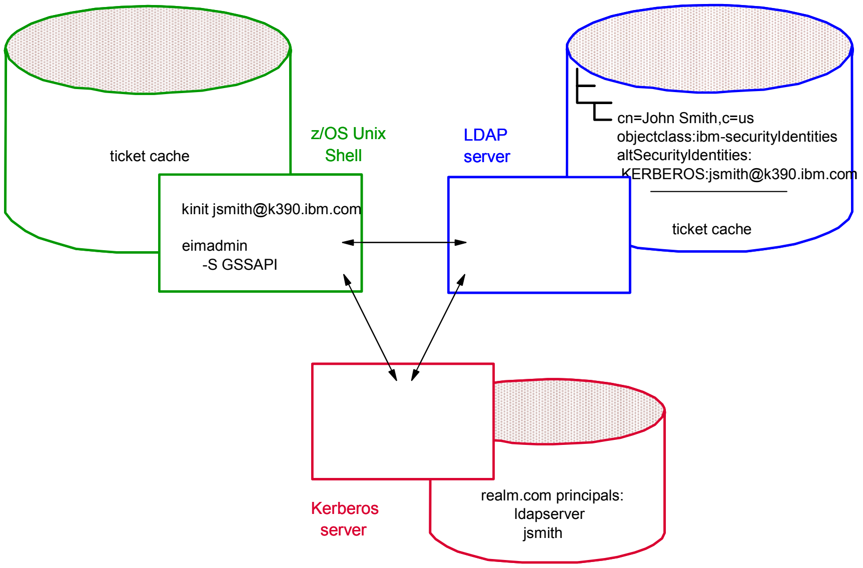
servercert

subject: cn=administrator
issuer: cn=certauth,o=xyz,c=us
private key:234098092....

Figure 14-12 Digital certificate authentication

Figure 14-13 on page 389 is an example of a Kerberos authentication.

Kerberos Authentication



ticket cache

z/OS Unix
Shell

kinit jsmith@k390.ibm.com

eimadmin
-S GSSAPI

LDAP
server

cn=John Smith,c=us
objectclass:ibm-securityIdentities
altSecurityIdentities:
KERBEROS:jsmith@k390.ibm.com

ticket cache

Kerberos
server

realm.com principals:
ldapserver
jsmith

Figure 14-13 Kerberos authentication

14.23.4 Updated APIs

The following APIs are updated due to the EIM enhancements in z/OS v1R5:

- eimChangeDomain
- eimConnect
- eimConnectToMaster
- eimCreateDomain
- eimDeleteDomain
- eimListDomains

14.23.5 EIM admin utility updated

The following updates were made to the utility:

-S connectType This is the method of authentication to the LDAP server. It must be one of the following values:

- SIMPLE (bind DN and password)
- CRAM-MD5 (bind DN and protected password)
- XTERNAL (digital certificate)
- GSSAPI (Kerberos)

-K keyFile This is the name of the SSL key database file, including the full path name. If the file cannot be found, it is assumed to be the name of a RACF key ring which contains authentication certificates. This value is required for SSL communications with a secure LDAP host (prefixed ldaps://).

-P keyFilePassword This is the password required to access the encrypted information in the key database file. Alternatively, you can specify an SSL password stash file for this option by prefixing the stash file name with "file://".

-N certificateLabel This identifies which certificate to use from the key database file or RACF key ring. If this option is not specified, the certificate marked as the default in the file or ring is used.

14.23.6 Migration and coexistence

The EIM data is unchanged. You should update the client applications only to exploit the added authentication options.

14.24 PKI Services architecture

The basic public key infrastructure (PKI) components are shown in Table 14-6.

Table 14-6 Components and functions of PKI

Components of the PKI	PKI component function
Administration Web application	Assists authorized administrators to review requests for certificates, approve or reject requests, renew certificates, or revoke certificates through their own Web browsers. The application consists of sample screens that you can easily customize to display your organization's logo. It also supports the following tasks: ► Reviewing pending certificate requests. ► Querying pending requests to process those that meet certain criteria. ► Displaying detailed information about a certificate or request. ► Monitoring certificate information, such as validity period. ► Annotating the reason for an administrative action.
End-user Web application	Guides your users to request, obtain, and renew certificates through their Web browsers. The application consists of sample screens that you can easily customize to meet your organization's needs for certificate content and standards for appearance. It offers several certificate templates that you can use to create requests for a variety of certificate types, based on the certificate's intended purpose and validity period, and supports certificate requests that are automatically approved.
Exit	Provides advanced customization for additional authorization checking, validating, and changing parameters on calls to the R_PKIServ callable service (IRRSPX00), and capturing certificates for further processing. You can call this exit from the PKIServ CGIs and use its IRRSPX00 pre-processing and post-processing functions. A code sample in C language code is included.
ICSF(optional)	Securely stores the PKI Services certificate authority's private signing key.
LDAP	The directory that maintains information about the valid and revoked certificates that PKI Services issues in an LDAP-compliant format. You can use an LDAP server such as z/OS Security Server LDAP.

Components of the PKI	PKI component function
PKI services daemon	The server daemon that acts as your certificate authority confirming the identities of users and servers, verifying that they are entitled to certificates with the requested attributes, and approving and rejecting requests to issue and renew certificates. It includes support for: ► An issued certificate list (ICL) to track issued certificates ► Certificate revocation lists (CRLs) to track revoked certificates
RACF (or equivalent)	Controls who can use the functions of the R_PKIServ callable service and protects the components of your PKI Services system. RACF creates your certificate authority's certificate, key ring, and private key. You can also use it to store the private key, if ICSF is not available.
z/OS HTTP Server	PKI Services uses the Web server to encrypt messages, authenticate requests, and transfer certificates to intended recipients.
R_PKIServ (IRRSPX00)	The application programming interface (API) that allows authorized applications, such as servers, to programmatically request the functions of PKI Services to generate, retrieve, and administer certificates.

14.24.1 PKI applications

The public key infrastructure provides applications with a framework for performing the following types of security-related activities:

- Authenticate all parties that engage in electronic transactions
- Authorize access to sensitive systems and repositories
- Verify the author of each message through its digital signature
- Encrypt the content of all communications

Figure 14-14 on page 392 is an example of the z/OS PKI services architecture.

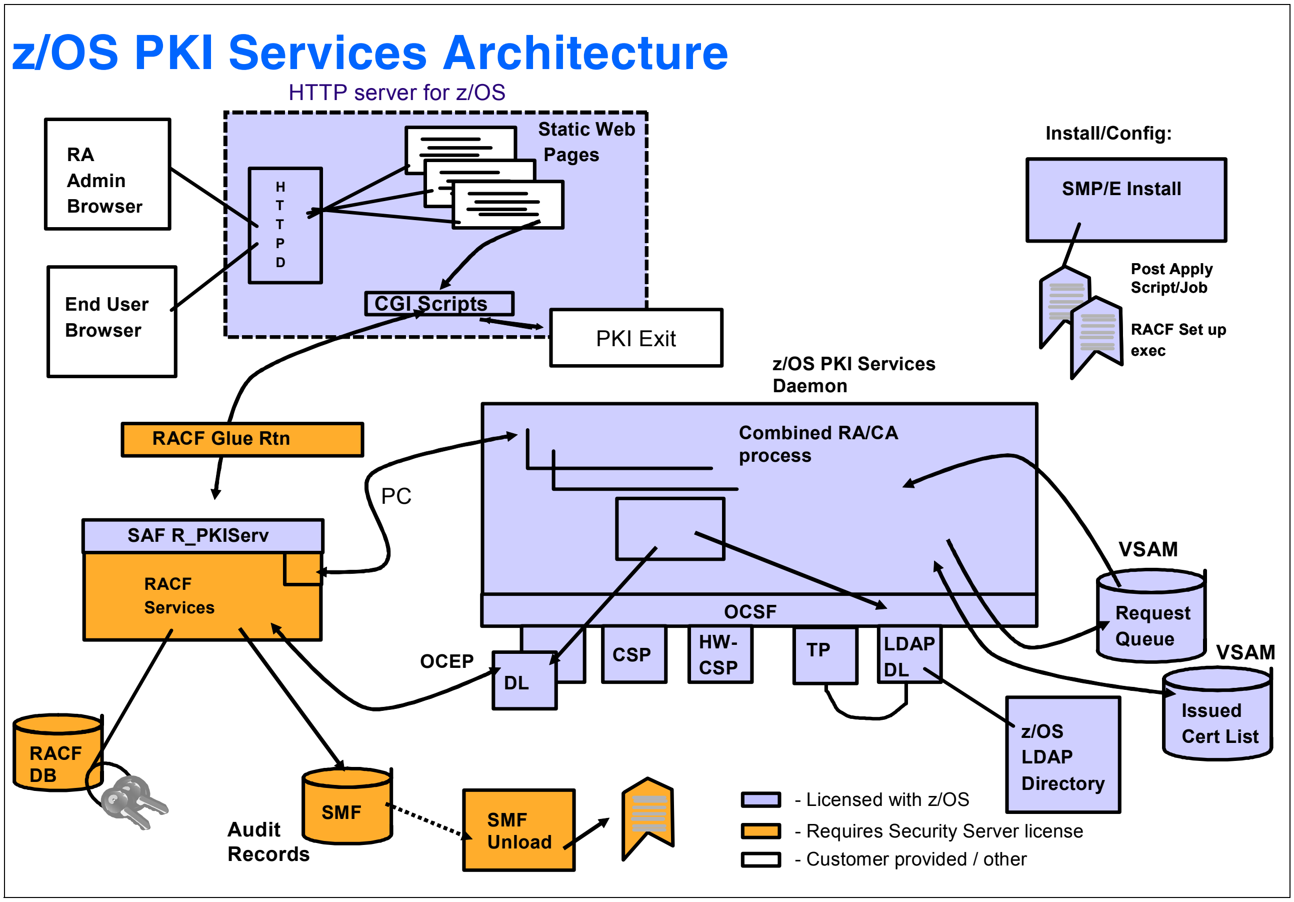
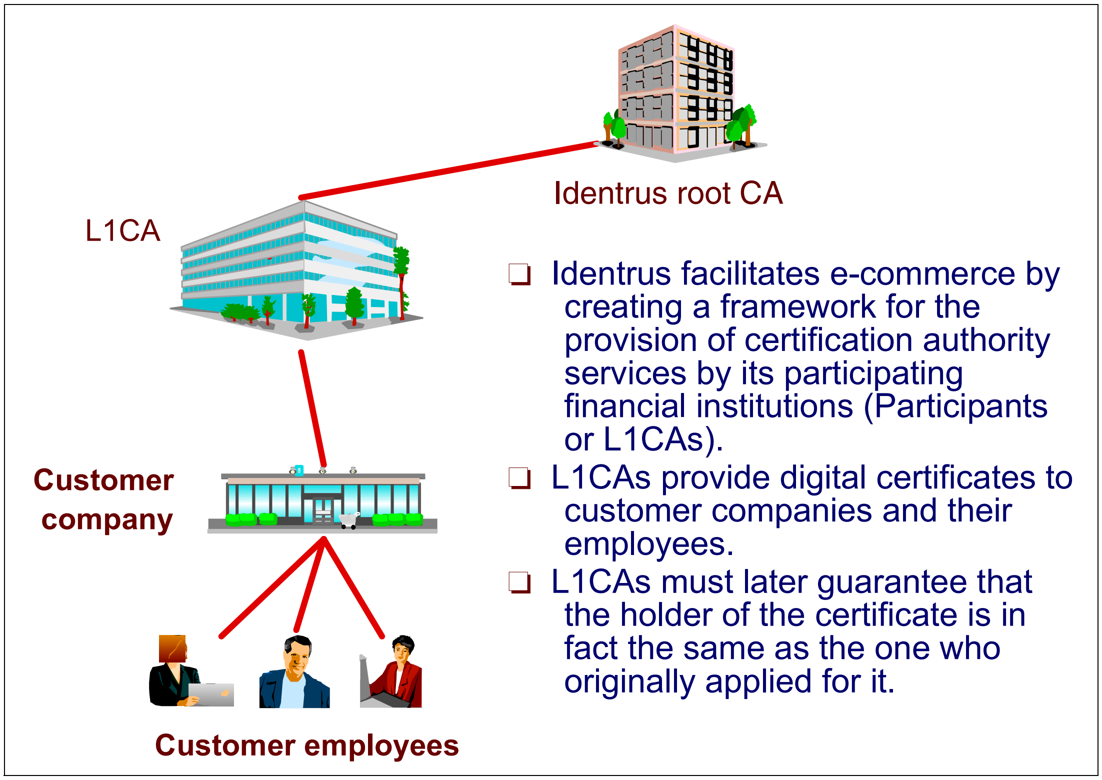


Figure 14-14 z/OS PKI services architecture

14.24.2 Identrus PKI

Identrus is a consortium of regulated financial institutions that have developed a common set of operating rules to facilitate electronic commerce. The Identrus model follows a standard hierarchical PKI deployment. In it, Identrus acts as the Root Certificate Authority, issuing certificates to it's Level 1 participants (L1CAs). The L1CAs (which are also called member banks) in turn issue certificates to their customers.

Figure 14-15 on page 393 shows an Identrus architecture.



- Identrus facilitates e-commerce by creating a framework for the provision of certification authority services by its participating financial institutions (Participants or L1CAs).
- L1CAs provide digital certificates to customer companies and their employees.
- L1CAs must later guarantee that the holder of the certificate is in fact the same as the one who originally applied for it.

Figure 14-15 Identrus PKI providing credentials

14.24.3 PKI Services supported certificate types

The types of certificates that you can request, based on the certificate templates that are included with PKI Services, and that are new with z/OS V1R5 are the following:

- One-year SAF browser certificate

  End-user client authentication where the security product (RACF, not PKI Services) is the certificate provider.

- One-year SAF server certificate

  Web server SSL certification where the security product (RACF, not PKI Services) is the certificate provider.

- Two-year Identrus authenticode - code signing certificate

  This template allows end users to request a server certificate for use within the Identrus infrastructure. It is used to sign software that will be downloaded across an untrusted medium.

- Four-year Identrus end-entity identity certificate

  This template allows end users to request a browser certificate for use within the Identrus infrastructure. It is used to sign communications between a subscribing customer (SC) and a relying customer (RC), and for S/MIME digital signatures.

- Four-year Identrus end-entity utility certificate

  This template allows end users to request a browser certificate for use within the Identrus infrastructure. It is used to sign communications between a subscribing customer (SC) and a relying customer (RC), between an RC and a relying participant (RP), and for S/MIME digital signatures.

- Four-year Identrus end-entity server encipherment (SSL) certificate

  This template allows end users to request a server certificate for use within the Identrus infrastructure. It is used to establish SSL communications between a subscribing customer (SC) and a relying customer (RC), and for S/MIME encryption.
- Four-year Identrus end-entity server signing certificate

  This template allows end users to request a server certificate for use within the Identrus infrastructure. It is used to sign communications between a subscribing customer (SC) and a relying customer (RC), an RC and a relying participant (RP), and for S/MIME digital signatures.

14.24.4 Four-year Identrus model

Certificate templates are samples of the most commonly requested certificate types. You can add, modify, and remove certificate templates to customize the variety of certificate types you offer to your users.

Four-year model example

After certificate issuance, a typical transaction using the four-year model includes these steps:

- The subscribing customer uses a private key and corresponding certificate, stored on their *smartcard*, to sign a transaction request that it sends to the relying customer.

> **Note:** PKI Services now supports key lengths up to 2048 bits for the CA signing private keys.

- The relying customer verifies that the signed data came from the subscribing customer and was not modified in transit.
- The relying customer requests services from its relying participant; for example, it can ask its relying participant to perform a status check on the subscribing customer’s certificate to verify that it is still valid and has not expired or been revoked.
- The relying participant requests services of the issuing participant to fulfill the relying customer’s service request. Based on the response from the relying participant the relying customer fulfills or rejects the request from the subscribing customer.

Figure 14-16 on page 395 shows how the Identrus four corner model works.

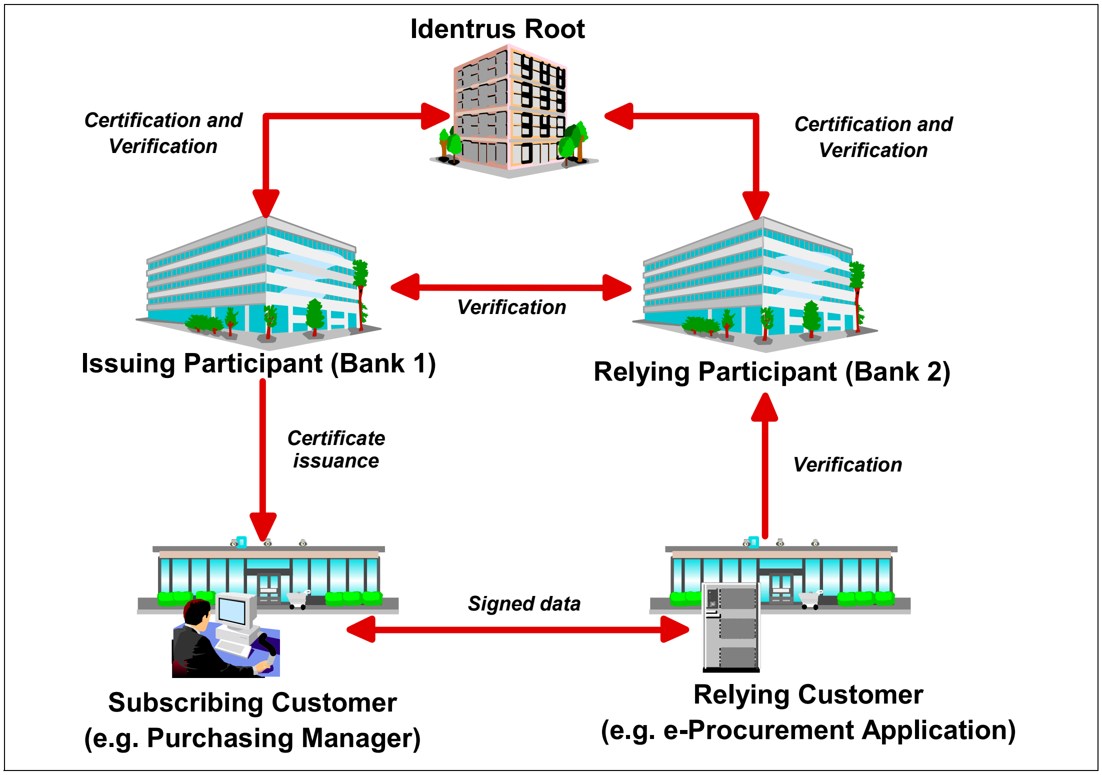

Identrus Root
Certification and Verification
Certification and Verification
Verification
Issuing Participant (Bank 1)
Relying Participant (Bank 2)
Certificate issuance
Verification
Signed data
Subscribing Customer (e.g. Purchasing Manager)
Relying Customer (e.g. e-Procurement Application)

Figure 14-16 The four-year Identrus model

14.24.5 The Identrus PKI

The Identrus PKI infrastructure is composed of elements that are central to the certificate authority. Identrus divides the function of the CA into three subcomponents:

Registration authority (RA) Determines who gets a certificate and what information it should contain (usually manually driven.)

Certificate authority (CA) Process that creates and manages certificates and revocation information.

Validation authority (VA) Process that can be invoked to verify the validity of a certificate using the Online Certificate Status Protocol (OCSP).

Figure 14-17 on page 396 illustrates the Identrus PKI infrastructure.

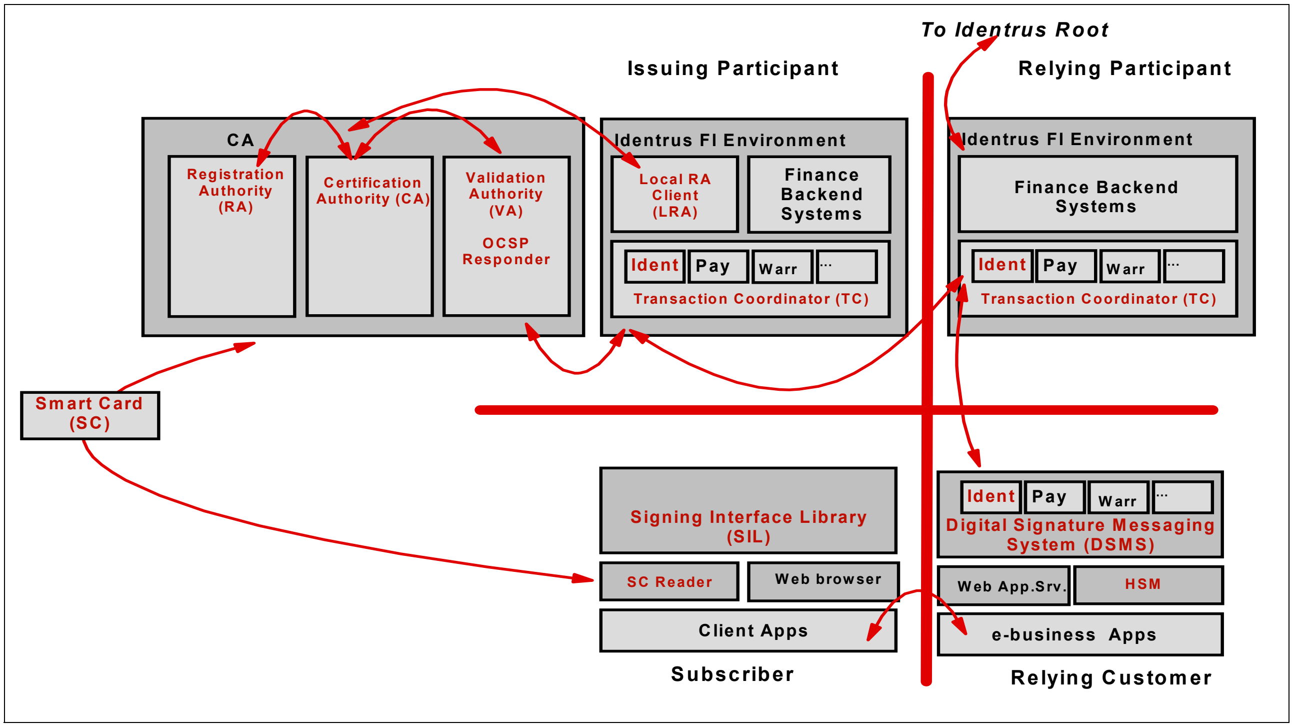


Figure 14-17 Identrus PKI

14.24.6 PKI Services

In PKI Services the registration authority (RA) and the certification authority (CA) are combined in one component. All other components must be supplied by other vendors, including the validation authority (VA). PKI Services publishes its certificate revocation information to LDAP to make it available to the VA vendor. Figure 14-18 on page 397 illustrates this and how it fits together with the Identrus PKI flow shown in Figure 14-17.

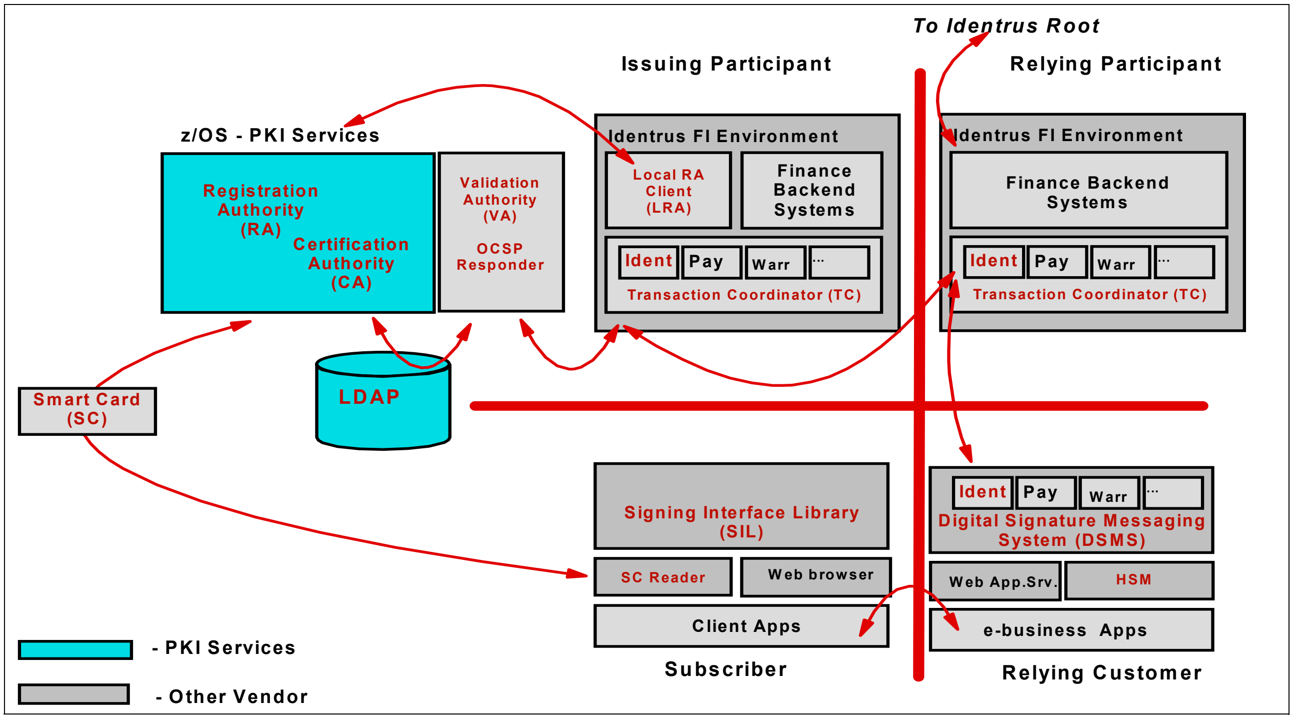


Figure 14-18 Where PKI Services fits with Identrus PKI

14.24.7 PKI certificate extensions

To be Identrus-compliant, PKI services needed to add additional certificate extension support as per the Identrus specs. This included more granularity for the KeyUsage extension and support for a new extension, as follows:

- ExtKeyUsage and AuthorityInfoAccess (AIA).

  The AIA extension is used to hold the URL of the (vendor-supplied) OCSP responder. Identrus also mandates that certain extensions must be marked critical. Thus, in this release, support now includes a new %%Critical%% directive. These directives are normally hard-coded in the certificate templates, or, in the case of KeyUsage and ExtKeyUsage, may be selectable by the end-user.

Certificates can contain a CertificatePolicies extension. This extension contains policy information, such as the way in which your CA operates and the intended purpose of the issued certificates. For more information about this extension, refer to RFC2459 on the Internet Engineering Task Force (IETF) Web site at:

```
http://www.ietf.org
```

By default, PKI Services does not include this extension in the certificates it creates. However, you can define your own CertificatePolicies extension by modifying fields in the CertPolicy section of the pkiserv.conf configuration file. When requesting a certificate the new KeyUsage and ExtKeyUsage parameters can be provided. To use KeyUsage for CA certificates, the Public Key Infrastructure for X.509 version 3 (PKIX) CertSign flag bit must be on.

Note: The CertificatePolicies extension also required additional support. Prior to z/OS V1R5, PKI Services could only be configured to have a global set of polices, which meant all certificates issued had the same set of policies. Now, specific policies can be defined in the certificate templates. The %%CertPolicies%% template file directive points to the actual policy declaration in the configuration file.

14.24.8 Summary of new fields with z/OS V1R5

When you request certificates, you provide information for the fields in certificate request forms. The following tables describe the fields in the end-user Web pages for the new parameters KeyUsage and ExtKeyUsage.

Key usage (KeyUsage)

The intended purpose of the certificate with the field KeyUsage is shown in Table 14-7. The intended purpose and possible KeyUsage value in the certificate. and its possible PKIX bits are shown.

Table 14-7 New KeyUsage values and their purpose

KeyUsage value	Intended purpose	PKIX bits
certsign	Certificate and CRL signing	KeyCertSign and cRLSign
crlsign	CRL signing	cRLSign
dataencrypt, dataenciph(erment)	Data encryption	dataEncipherment
digitalsig or digitalsignature	Authentication	digitalSignature
docsign or nonrepudiation	Document signing	nonRepudiation
handshake	Protocol handshaking (eg SSL)	digitalSignature and keyEncipherment
keyagree or keyagreement	Key agreement	keyAgreement
keycertsign	Certificate signing	keyCertSign
keyencrypt, keyenciph(erment)	Key transport key	Encipherment

Table 14-8 indicates the intended purpose of the of the certificate and its possible values.

Table 14-8 Extended key usage new fields

clientauth	Client side authentication
codesigning	Code signing
ocspsigning	OCSP response signing
serverauth	Server side authentication
timestamping	Digital timestamping
emailprotection	Email protection

Note: Certain values can be marked critical in the issued certificate's extension. This name-value pair can be repeated for each extension to be marked critical. The following extensions can be marked critical:

- BasicContraints (ignored because this extension is always marked critical)
- KeyUsage (ignored because this extension is always marked critical)
- ExtKeyUsage
- SubjectAltName, AltEmail, AltIPAddr
- AltDomain, AltURI
- HostIdMappings, HostIdMap
- CertificatePolicies, CertPolicies

Example: %%Critical=ExtKeyUsage%%

See *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693 for additional information of the use of these new fields.

14.24.9 Predefined Identrus Certificate types

The z/OS V1R5 certificate templates file comes with these certificate templates already defined. The Identrus predefined certificates which are available in the template file are listed in Table 14-9.

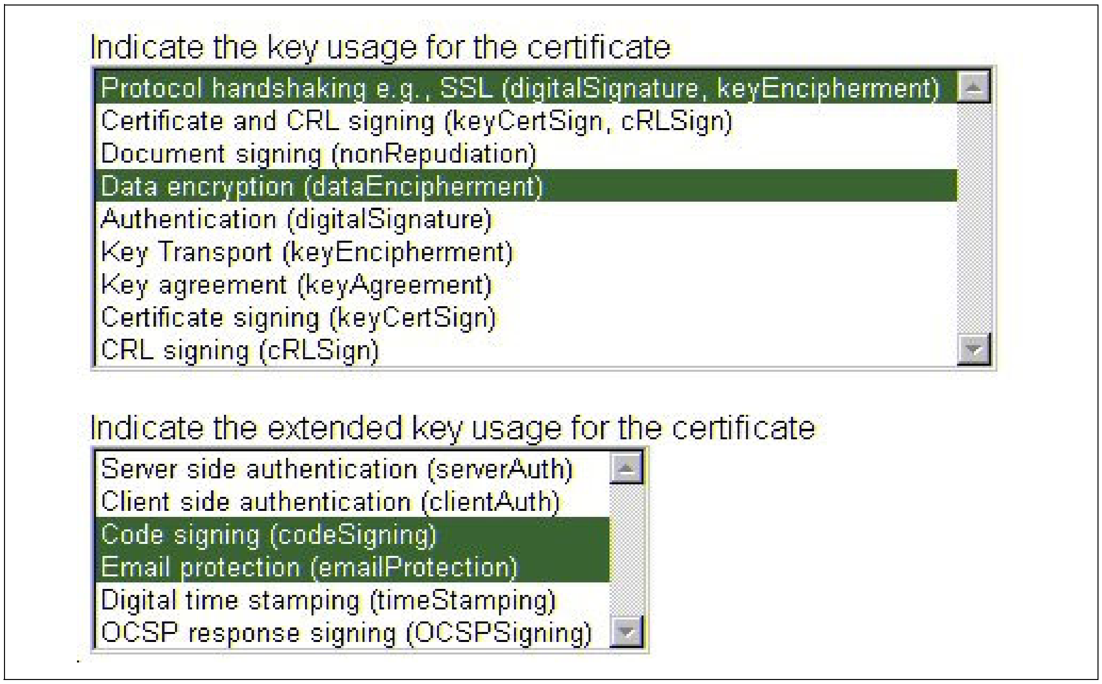
Table 14-9 Predefined Identrus certificate templates

Identrus certificate	Identrus certificate function
End-Entity Identity	Used to sign legally binding transactions and requires the following parameters: ► KeyUsage - digitalSignature, nonRepudiation ► CertificatePolicies - the Identrus "signing" policy ► AuthInfoAccess - OCSP information ► Standard OCSP protocol ► Identrus enhanced OCSP
End-Entity Utility	Used for SSL client auth and S/MIME data encryption and requires the following parameters: ► KeyUsage - digitalSignature, keyEncipherment, dataEncipherment, keyAgreement ► ExtKeyUsage - Client authentication, Email protection ► CertificatePolicies - the Identrus "utility" policy ► AuthInfoAccess - same as Identity certificate ► SubjectAltName – e-mail address
End-Entity Server Signing	Used to sign communications and requires the following parameters: ► AuthInfoAccess - Identrus enhanced OCSP ► Other info - same as Identity certificate
End-Entity Server Encipherment	Used for SSL server auth and requires the following parameters: ► KeyUsage - digitalSignature, keyEncipherment ► ExtKeyUsage - Server authentication ► CertificatePolicies - customer-defined policy with Identrus wording ► AuthInfoAccess - same as Server Signing certificate ► SubjectAltName - e-mail address
Authenticode	Used for code signing and requires the following parameters: ► KeyUsage - digitalSignature, keyEncipherment ► ExtKeyUsage - code signing ► CertificatePolicies - the Identrus "utility" policy ► SubjectAltName – e-mail address

14.24.10 Web pages INSERTs for KeyUsage/ExtKeyUsage

The Web pages are changed for this support in z/OS V1R5. Figure 14-19 shows how the new INSERTs (dialogs) for KeyUsage and ExtKeyUsage look on the administrators' page for approving requests with modifications. The selection dialogs are primed with the request's current values (in this case, handshaking, dataEncipherment, codeSigning and emailProtection).

This is also how they look to the end-user (less the preselected highlighting) when the INSERTs are used for the certificate request Web page, when defining custom certificate templates that allow the end-user to choose the usage values.



Indicate the key usage for the certificate

[✓ Protocol handshaking e.g., SSL (digitalSignature, keyEncipherment)]
Certificate and CRL signing (keyCertSign, cRLSign)
Document signing (nonRepudiation)
[✓ Data encryption (dataEncipherment)]
Authentication (digitalSignature)
Key Transport (keyEncipherment)
Key agreement (keyAgreement)
Certificate signing (keyCertSign)
CRL signing (cRLSign)

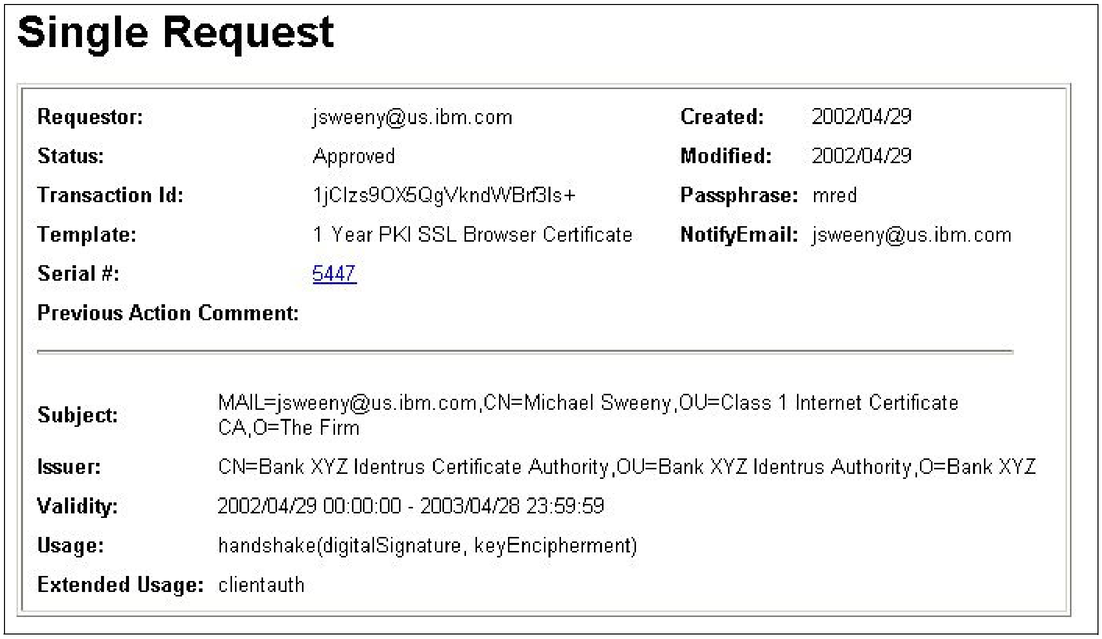
Indicate the extended key usage for the certificate

Server side authentication (serverAuth)
Client side authentication (clientAuth)
[✓ Code signing (codeSigning)]
[✓ Email protection (emailProtection)]
Digital time stamping (timeStamping)
OCSP response signing (OCSPSigning)

Figure 14-19 Modify and Approve Request Web page

14.24.11 PKI Web page changes

Since the PKI Web pages now have additional information to display (KeyUsage, ExtKeyUsage), there is a redesign to make them easier to read, especially the information for Usage and Extended Usage. Figure 14-20 on page 401 shows the new parameters KeyUsage and ExtKeyUsage.



Single Request

Requestor:	jsweeny@us.ibm.com	**Created:**	2002/04/29
Status:	Approved	**Modified:**	2002/04/29
Transaction Id:	1jClzs9OX5QgVkndWBrf3Is+	**Passphrase:**	mred
Template:	1 Year PKI SSL Browser Certificate	**NotifyEmail:**	jsweeny@us.ibm.com
Serial #:	5447		
Previous Action Comment:			

Subject:	MAIL=jsweeny@us.ibm.com,CN=Michael Sweeny,OU=Class 1 Internet Certificate CA,O=The Firm
Issuer:	CN=Bank XYZ Identrus Certificate Authority,OU=Bank XYZ Identrus Authority,O=Bank XYZ
Validity:	2002/04/29 00:00:00 - 2003/04/28 23:59:59
Usage:	handshake(digitalSignature, keyEncipherment)
Extended Usage:	clientauth

Figure 14-20 Single certificate request or an issued certificate

14.24.12 Certificate suspension

The Identrus changes now support the capability to revoke a certificate temporarily (suspend the use of a certificate).

Revoking or suspending a certificate means that you cannot continue to use the certificate. Some reasons you might want to do this are the following:

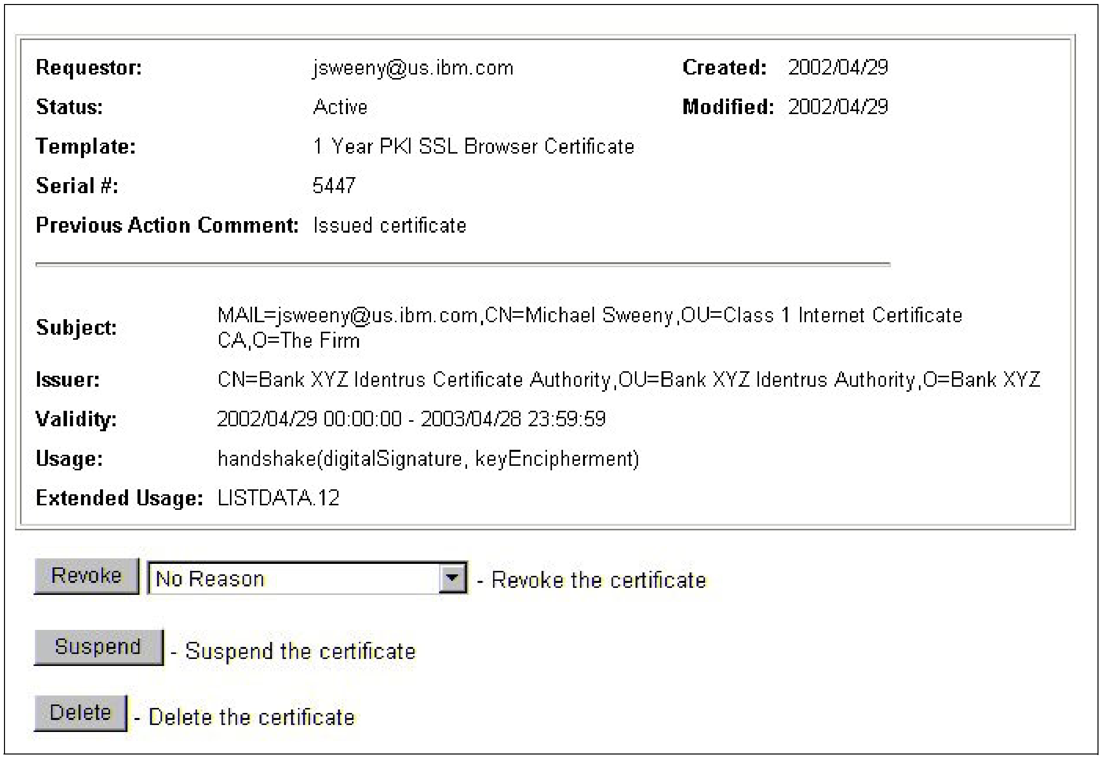
- To permanently revoke your certificate if you suspect your private key has been compromised.
- To suspend (temporarily revoke) your certificate if you want to discontinue using it for a period of time (known as the suspension grace period).

  Integral to this is the suspension grace period. The grace period is enabled by setting the %%MaxSuspendDuration%% directive in the configuration file. If a certificate stays suspended for longer than the grace period, it is revoked the next time the certificate revocation list (CRL) is refreshed.

If you suspend your certificate, the PKI administrator may resume (reactivate) the certificate, or permanently revoke it, if the certificate has not yet expired and the grace period has not elapsed. If the grace period has elapsed, the certificate is permanently revoked the next time the certificate revocation lists (CRLs) are issued.

Web page to suspend a certificate

Figure 14-21 on page 402 shows the Web page displayed when the administrator works with certificate details. It shows the new administrator action “Suspend” that is possible for certificates with “Active” status. All administrator pages that support “Revoke” now also support “Suspend.”



Requestor:	jsweeny@us.ibm.com	**Created:**	2002/04/29
Status:	Active	**Modified:**	2002/04/29
Template:	1 Year PKI SSL Browser Certificate		
Serial #:	5447		
Previous Action Comment:	Issued certificate		

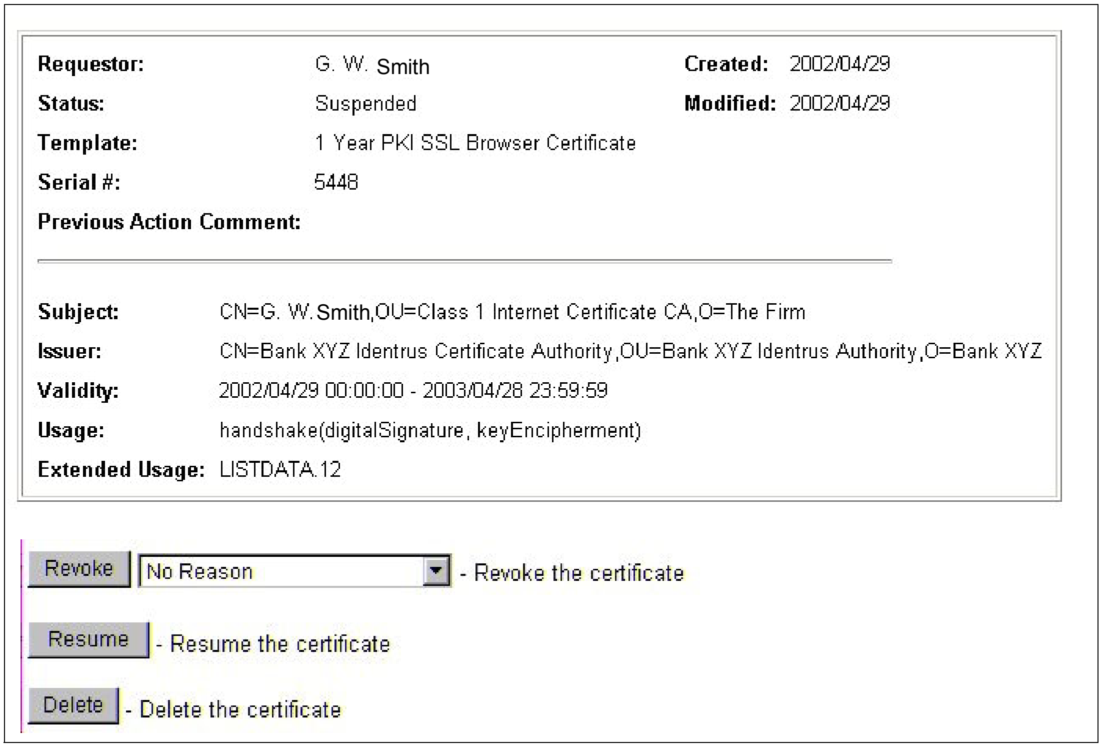
Subject:	MAIL=jsweeny@us.ibm.com,CN=Michael Sweeny,OU=Class 1 Internet Certificate CA,O=The Firm
Issuer:	CN=Bank XYZ Identrus Certificate Authority,OU=Bank XYZ Identrus Authority,O=Bank XYZ
Validity:	2002/04/29 00:00:00 - 2003/04/28 23:59:59
Usage:	handshake(digitalSignature, keyEncipherment)
Extended Usage:	LISTDATA.12

Figure 14-21 Web page to suspend a certificate

> **Note:** The new buttons Suspend and Revoke appear only when your search matches at least one certificate whose status is “Active.” The Resume button appears only when your search matches at least one certificate whose status is “Suspended.”

14.24.13 Resume a certificate

Figure 14-22 on page 403 shows the new administrator action “Resume” that is possible for certificates with “Suspended” status. Additionally, certificates that are “Suspended” may be permanently revoked by the administrator.



Requestor: G. W. Smith **Created:** 2002/04/29

Status: Suspended **Modified:** 2002/04/29

Template: 1 Year PKI SSL Browser Certificate

Serial #: 5448

Previous Action Comment:

Subject: CN=G. W.Smith,OU=Class 1 Internet Certificate CA,O=The Firm

Issuer: CN=Bank XYZ Identrus Certificate Authority,OU=Bank XYZ Identrus Authority,O=Bank XYZ

Validity: 2002/04/29 00:00:00 - 2003/04/28 23:59:59

Usage: handshake(digitalSignature, keyEncipherment)

Extended Usage: LISTDATA.12

Revoke No Reason ▼ - Revoke the certificate

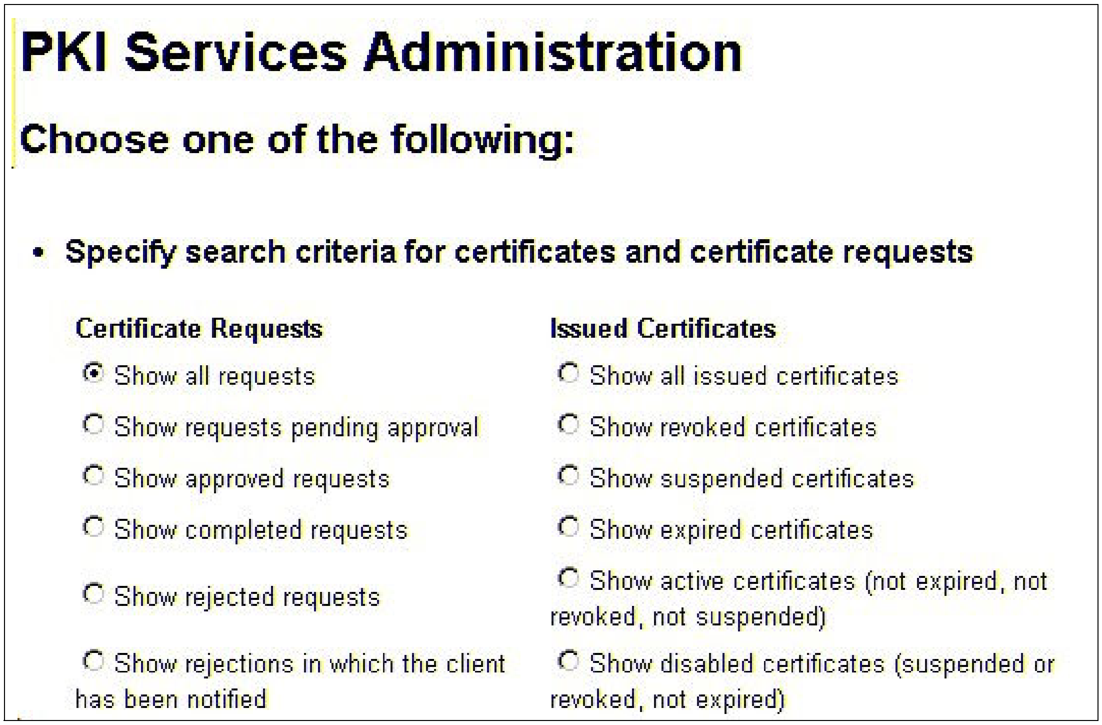
Resume - Resume the certificate

Delete - Delete the certificate

Figure 14-22 Web page to resume a suspended certificate

14.24.14 PKI Services home page

Figure 14-23 shows the PKI Services Administration home page. With z/OS V1R5, a new query was added to show all suspended certificates.



PKI Services Administration

Choose one of the following:

- **Specify search criteria for certificates and certificate requests**

Certificate Requests

● Show all requests

○ Show requests pending approval

○ Show approved requests

○ Show completed requests

○ Show rejected requests

○ Show rejections in which the client has been notified

Issued Certificates

○ Show all issued certificates

○ Show revoked certificates

○ Show suspended certificates

○ Show expired certificates

○ Show active certificates (not expired, not revoked, not suspended)

○ Show disabled certificates (suspended or revoked, not expired)

Figure 14-23 New certificate query

14.24.15 Miscellaneous changes

With the previous changes made to Identrus, the auditing of revocation information is required by Identrus. Also, additional changes have been made due to some requirements which had impacts on the following:

- Application domain
- Certificate revocation list (CRL)
- RACF support
- Performance

Application domains

All CGIs reside under the common URL, as follows:

```
https://<webserver-domain-name>/PKIServ
```

Thus, all users, including PKI administrators, have the same PKI Services home page, Web content, and supported certificate templates. In other words, by default, there is a single application domain: PKISERV.

The sample PKI Services template file (pkiserv.tmpl) contains two application sections: PKISERV and CUSTOMERS. The PKISERV application includes all templates and functions. The CUSTOMERS application contains all templates and functions but it does not contain the button at the bottom of the home page to link to the Administration home page. Therefore, using these two application sections, your users can be easily divided between two subsets: customers and administrators. You will probably want to separate your administration users and your end users. You may also need to further segment your end user population by adding different application domains for different groups of end users. Both of these objectives can be accomplished by using multiple applications. The PKI administrators can be directed to use the existing application domain:

```
http(s)://<webserver-domain-name>/PKIServ
```

Each subset of end users can be given a new, unique application domain:

```
http(s)://<webserver-domain-name>/<appl-domain-name>
```

See *z/OS Cryptographic Services PKI Services Guide and Reference,* SA22-7693 for details on creating multiple application domains and executing the following tasks:

- Updating the PKI Services template file
- Updating the Web server configuration files

The certificate revocation list

Certificate revocation lists (CRLs) are used to publish the serial numbers of the certificates that have been revoked. Applications that wish to validate a certificate would check for revocation by retrieving and examining the appropriate CRL. Prior to z/OS V1R5, PKI Services published one global CRL to the CA's entry in LDAP. On a system where a large number of certificates have been issued, the number of certificates that are revoked at any given time could be large, resulting in a very large global CRL. To alleviate this, the partitioning of the global CRL was created.

CRL distribution points

The partitioning of the Global CRL in PKI Services is based on certificate serial numbers. New directives in the PKI Services configuration file indicate whether CRL Distribution Points (DPs) should be created and what the LDAP entry names should be. CRL DPs are locations where partial CRLs are stored. They are stored in separate LDAP entries directly below the CA's entry. Applications that wish to validate a certificate need only retrieve the appropriate

DP CRL, (the one specified by the CRLDistributionPoint extension). If you anticipate that you will average more than 500 revoked non-expired certificates at any given time, consider using CRL distribution points. You begin using CRL distribution points when you accept the defaults settings contained in the PKI Services configuration file (pkiserv.conf). You can customize those settings by specifying the number of certificates per DP and the DP name using the parameters in the CertPolicy section of the pkiserv.conf. The certification revocation list is now audited and published to LDAP. See *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693 for more information about CRL. Figure 14-24 shows what a CRL distribution point looks like.

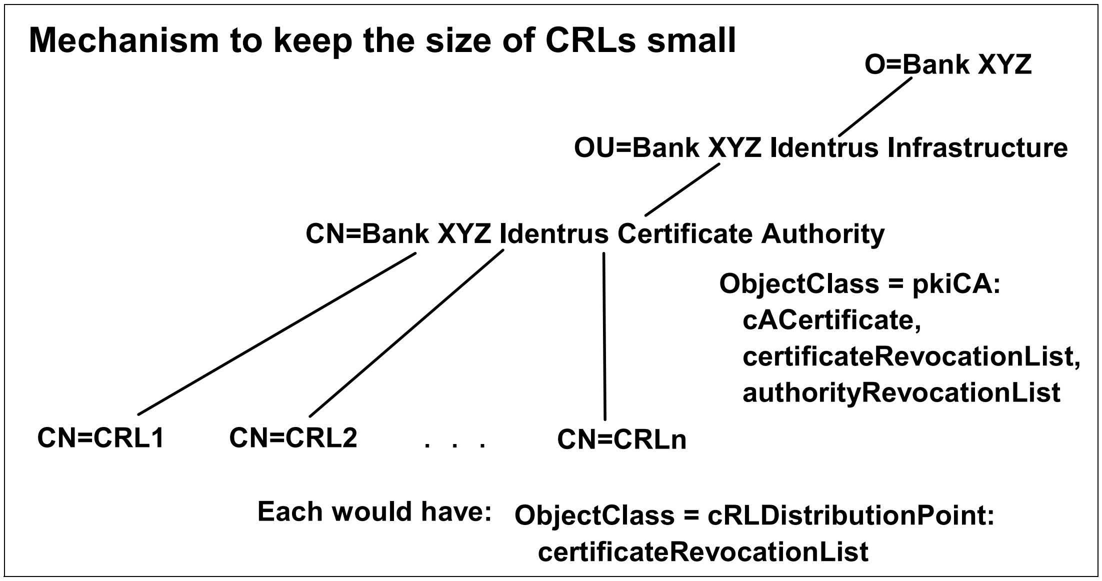

Mechanism to keep the size of CRLs small
O=Bank XYZ
OU=Bank XYZ Identrus Infrastructure
CN=Bank XYZ Identrus Certificate Authority
ObjectClass = pkiCA:
cACertificate,
certificateRevocationList,
authorityRevocationList
CN=CRL1
CN=CRL2
. . .
CN=CRLn
Each would have:
ObjectClass = cRLDistributionPoint:
certificateRevocationList

Figure 14-24 CRL Distribution Points (DPs)

CRL auditing

Due to Identrus compliance, CRL must be audited. The SMF80 type record will be issued during public key infrastructure (PKI) services processing. PKI Services writes a record for each CRL that is successfully published to LDAP. RACF and PKI Services write one record for each event. The new event code 79 (PKIDPUBR) for the SMF80 record type has been created for the audit.

14.24.16 RACF

With z/OS V1R5, RACF has increased its private key size limit to 2048 bits, but the 1024 bit limit is still in effect for software keys. RACDCERT is the command to generate keys and certificates in RACF. Use “SIZE(2048) PCI cryptographic coprocessor” (PCICC) to get a 2048 bit private key. Since RACF is normally used to generate the PKI Services certificate signing key, the PKI Services setup exec was modified to support 2048 bits keys. RACF adds a base set of certificate authority certificates to the RACF database at IPL. The list has been extended to include the following certificates:

- GTE CyberTrust Root CA
- Identrus Interoperability CA
- Entrust.net Secure Server CA

To list the certificates, issue:

```
RACDCERT CERTAUTH LIST
```

To use them, issue:

```
RACDCERT CERTAUTH ALTER(LABEL('…')) TRUST
```

Note: Keys are only allowed this large if generated by the hardware.

Performance improvement with VSAM

Depending on your environment, your VSAM performance may be improved by providing buffer space for the VSAM data sets as part of the IKYSPROC (alias PKISERVD) started procedure. PKI Services makes use of two VSAM clusters. The ObjectStore is where active requests are maintained. The issued certificate list (ICL) holds a copy of each certificate issued by PKI Services. The VSAM I/O support is rewritten to make use of new alternate indexes, two per cluster. Existing PKI Services customers must manually create these new alternate indexes using sample JCL provided. They must also update their configuration file to add the new directives specifying the data set names for the alternate index PATH data sets. The PKI Services daemon will fail to initialize if the indexes are missing and will issue message IKYD001I. Adding VSAM buffer space to the PKISERVD started task vastly reduces I/O. This is done by adding DD statements with the AMP keyword for each of the VSAM data sets. The configuration file also needs to be updated to have the DD names instead of the data set names.

The new configuration file directives added for the alternate indexes are the following:

- ObjectStatusDSN='pkisrvd.vsam.ost.status'
- ObjectRequestorDSN='pkisrvd.vsam.ost.requestr'
- ICLStatusDSN='pkisrvd.vsam.icl.status'
- ICLRequestorDSN='pkisrvd.vsam.icl.requestr'

PKISERVD is the sample procedure to start the PKI Services daemon. Installations have to run the JCL DD statements, shown in Figure 14-25, to add VSAM buffer space to the PKISERVD started task. Thus, it drastically reduces the number of I/Os. You need to specify the DD statements in the PKISERVD procedure with AMP values. Edit the JCL in 'SYS1.PROCLIB(PKISERVD)' to add the following DD statements to the bottom of the PKISERVD procedure.

```
//OST      DD  DSN=PKISRVD.VSAM.OST,DISP=SHR,
//  AMP=('BUFNI=8,BUFND=4')
//TID      DD  DSN=PKISRVD.VSAM.OST.PATH,DISP=SHR,
//  AMP=('BUFNI=8,BUFND=4')
//OSTAT    DD  DSN=PKISRVD.VSAM.OST.STATUS,DISP=SHR,
//  AMP=('BUFNI=1,BUFND=4')
//OREQ     DD  DSN=PKISRVD.VSAM.OST.REQUESTR,DISP=SHR,
//  AMP=('BUFNI=1,BUFND=4')
//ICL      DD  DSN=PKISRVD.VSAM.ICL,DISP=SHR,
//  AMP=('BUFNI=8,BUFND=4')
//ISTAT    DD  DSN=PKISRVD.VSAM.ICL.STATUS,DISP=SHR,
//  AMP=('BUFNI=1,BUFND=4')
//IREQ     DD  DSN=PKISRVD.VSAM.ICL.REQUESTR,DISP=SHR,
//  AMP=('BUFNI=1,BUFND=4')
```

Figure 14-25 DD statements used in the PKISERVD procedure

The customer must use the configuration file directives with the DD statements identified in the Figure 14-10 on page 407.

Table 14-10 Configuration file directives and their DD statement association

Configuration file directive	DD statement associated
ObjectDSN	OST
ObjectTidDSN	TID
ObjectStatusDSN	OSTAT
ObjectRequestorDSN	OREQ
ICLDSN	ICL
ICLStatusDSN	ISTAT
ICLRequestorDSN	IREQ

SSL versus OCSF

The OCSF architecture consists of a set of layered security services and associated programming interfaces designed to furnish an integrated set of information and communication security capabilities. Each layer builds on the more fundamental services of the layer directly below it. These layers start with fundamental components such as cryptographic algorithms, random numbers, and unique identification information in the lower layers, and build up to digital certificates, key management and recovery mechanisms, and secure transaction protocols in higher layers. The OCSF architecture is intended to be the multiplatform security architecture that is both horizontally broad and vertically robust. The Application Domains layer implements the application domain services, such as Secure Electronic Transaction (SET) and E-Wallet, E-mail services, or file archival services. The System Security Services layer is between the Application Domains layer and the OCSF Framework layer, as shown in Figure 14-26 on page 408. It implements security protocols that are used by the Application Domains layer. Software at this layer may implement cryptographic system security services such as Secure Sockets Layer (SSL), Internet Protocol Security (IPSEC), Secure/Multipurpose Internet Mail Extensions (S/MIME) and Electronic Data Interchange (EDI). The System Security Services layer also includes tools and utilities for installing, configuring, and maintaining the OCSF Framework and service provider modules. Figure 14-26 shows a simplified view of the layered architecture of an OCSF-based system. There are four major layers in the OCSF architecture:

- Application Domains
- System Security Services
- OCSF Framework
- Service Providers

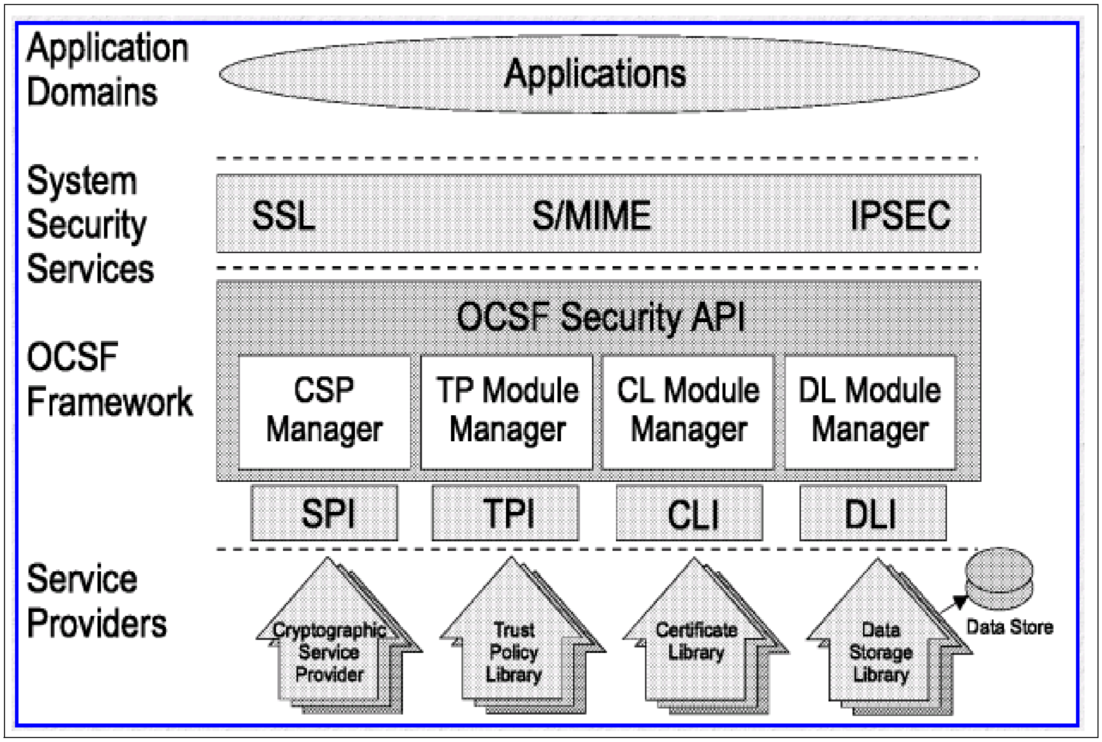


Figure 14-26 OCSF architecture

Today, SSL supports certificate revocation lists (CRLs) stored in an LDAP Server. Each time a certificate needs to be validated, a request is made to the LDAP Server to get the list of CRLs. This enhancement caches CRL lists locally instead of fetching them from the LDAP Server each time they are needed. This improves performance and alleviates the need to contact the LDAP Server each time certificate validation needs to be performed. You can specify a list of LDAP servers to be used for storing certificate revocation lists, and SSL will try to connect to each server in the list until a connection is obtained. You specify the list of LDAP servers on either the gsk_initialize or gsk_attribute_set_buffer APIs. When certificate validation is being performed, this list will be used to determine which LDAP server to connect to for the CRL information. This enhancement also provides greater availability by not being dependent on a single LDAP server.

Note: To provide better performance, open cryptographic services facility (OCSF) cryptography is replaced with SSL; OCSF is still used to validate the certificates in PKI Services daemon.

ICL cleanup

Certificates that have been created from requests are maintained permanently in an issued certificate database. Another name for this is the issued certificate list (ICL). Issued certificates are also published in an LDAP directory. A certificate can have only one of the following states (statuses) at a time:

- Active: The certificate has not yet expired, has not been revoked, and is not currently suspended.
- Expired: The certificate's validity period expired while it was active.
- Revoke: The certificate has not expired but it has been revoked. Such certificates are published on the next certificate revocation list (CRL).

- Revoked, Expired: The certificate was either revoked or suspended, and time has elapsed so that it is now also expired. Such certificates are not published on the next CRL.
- Suspended: The certificate has not expired but it is currently suspended. Such certificates are published on the next certificate revocation list (CRL).

The ICL cleanup is made by using the directive RemoveExpiredCerts= located in the configuration file pkiserv.conf.

14.24.17 Sysplex considerations

If your installation plans to use sysplex support (running multiple independent instances of PKI Services (one per image) that work in unison), you require the following:

- All systems in the sysplex that run PKI Services must be at z/OS V1R4 or higher.
- All instances of PKI Services must share the same VSAM data sets. To do so, they use VSAM record-level sharing (RLS). This requires setting up a coupling facility for data sharing (lock and cache).

See *z/OS Cryptographic Services PKI Services Guide and Reference,* SA22-7693 for more information about record-level sharing (RLS).

Note: Now R4 and R5 of PKI Services systems can coexist in the Sysplex. If you do so, the R5 features should be avoided. The features which should not be used are:

- CRL Distribution Points
- Template-specific Certificate Policies
- New extension support (for example, ExtKeyUsage)
- New Identrus certificate types

15

PSF 3.4.0 for z/OS

Print Services Facility™ (PSF) for z/OS is an IBM-licensed printer driver program that manages and controls data transmitted to Advanced Function Presentation™ (AFP) printers that are channel-attached, SNA-attached, or TCP/IP-attached. PSF 3.4.0 for z/OS is a replacement for PSF 3.3.0 for OS/390 and has productivity enhancements and new application support. This chapter describes PSF version 3.4.0.

This chapter contains the following topics:

- Overview
- PSF 3.4.0 IPDS™ support for AFP1 printers
- PSF 3.4.0 components
  - User environment
  - Application environment
  - Supported printers
- PSF 3.4.0 functional enhancements
  - TrueType/OpenType font support
  - Resource object capture (TrueType/OpenType fonts only)
  - HFS support (TrueType/OpenType fonts only)
  - Pre-Rip support
  - ASAP report enhancements
  - Trace entry documentation
- Migration and coexistence

15.1 Overview

Print Services Facility (PSF) 3.4.0 for z/OS provides host programming support under the OS/390 or z/OS operating system for Advanced Function Printing™ (AFP) printers. The PSF 3.4.0 for z/OS support is an enhancement of previous PSF releases.

PSF 3.4.0 for z/OS attached printers is supported in a page environment, and accepts four types of data: line data (both traditional line data and record format line data), composed page data (MO:DCA-P data), mixed data (line data mixed with MO:DCA-P data), and XML data.

PSF 3.4.0 for z/OS accepts most of the MO:DCA-P enhanced data stream which is a superset of the support provided in previous PSF releases. PSF translates these formatting control records into Intelligent Printer Data Stream™ (IPDS) commands. As in previous PSF releases, the interface between the application and PSF can be direct attachment (for channel-attached printers only) or PSF can be connected to JES and accept data from either a JES2 or JES3 spool.

15.2 PSF 3.4.0 IPDS support for AFP1 printers

IPDS is an IBM printer data stream designed to manage and control printer processes. It is distinguished from other data streams for printers because it provides all-points addressability, error recovery, and 2-way communications between the printer and Print Service Facility (PSF) licensed programs. Also, IPDS provides data stream compatibility across the IPDS product line independent of speed, physical attachment or rendering technology. The following IPDS command sets are supported in PSF 3.4.0 for z/OS:

- Device control - required
- Text with PTOCA PT/1, PT/2, and PT/3 - (either PT/1 or PT/2 or PT/3 required)
- IM image
- Overlays
- Page segments
- IO image with IOCA FS10, FS40, FS42, or FS45
- Graphics with GOCA DR/2 V0
- Bar codes
- Loaded font (LF1, LF3, Resident symbol sets)
- Object containers
  - Anacomp COM setup files
  - Color mapping table files
  - Encapsulated PostScript files
  - Encapsulated PostScript with transparency
  - Single-page PDF objects
  - Single-page PDF with transparency
  - IOCA tile resource objects
  - PDF resource objects
  - Resident color profile resource objects
  - TrueType/OpenType font
  - TrueType/OpenType collection

PSF does not perform any transform of data from a higher level command set to a lower level command set. For example, if a printer supports only the PT/1 level but a PT/2 level control sequence is received by PSF, PSF does not transform the PT/2 control sequence into a valid PT/1 level control sequence, but instead sends it to the printer and handles the subsequent error that the printer generates.

15.3 PSF 3.4.0 components

PSF 3.4.0 for z/OS is composed of two components:

- The PSF component, which converts input data sets to IPDS commands.
- The PPCC (Page Printer Communications Component), which sends the IPDS command stream to an SNA attached printer through ACF/VTAM.

PSF resides in a z/OS address space or Functional Subsystem (FSS). There can be up to 2000 PSF address spaces in a z/OS system. Each FSS can contain up to 128 Functional Subsystem Applications (FSAs). Each FSA, also known as a JES output writer, supports a single PSF printer. The printers are either directly controlled by the FSA or controlled by VTAM. PSF receives print data either from a JES spool or directly from the application program via the Direct Printer Services Subsystem (DPSS) of MVS.

Figure 15-1 illustrates the PSF 3.4.0 for z/OS attachment environment.

The legends used in the figure are:

PSF	Print Service Facility
JES	Job Entry Subsystem
FSS	Functional Subsystem
FSA	Functional Subsystem Application
PPCC	Page Printer Communication Component
TCP/IP	Transmission Control Protocol/Internet Protocol

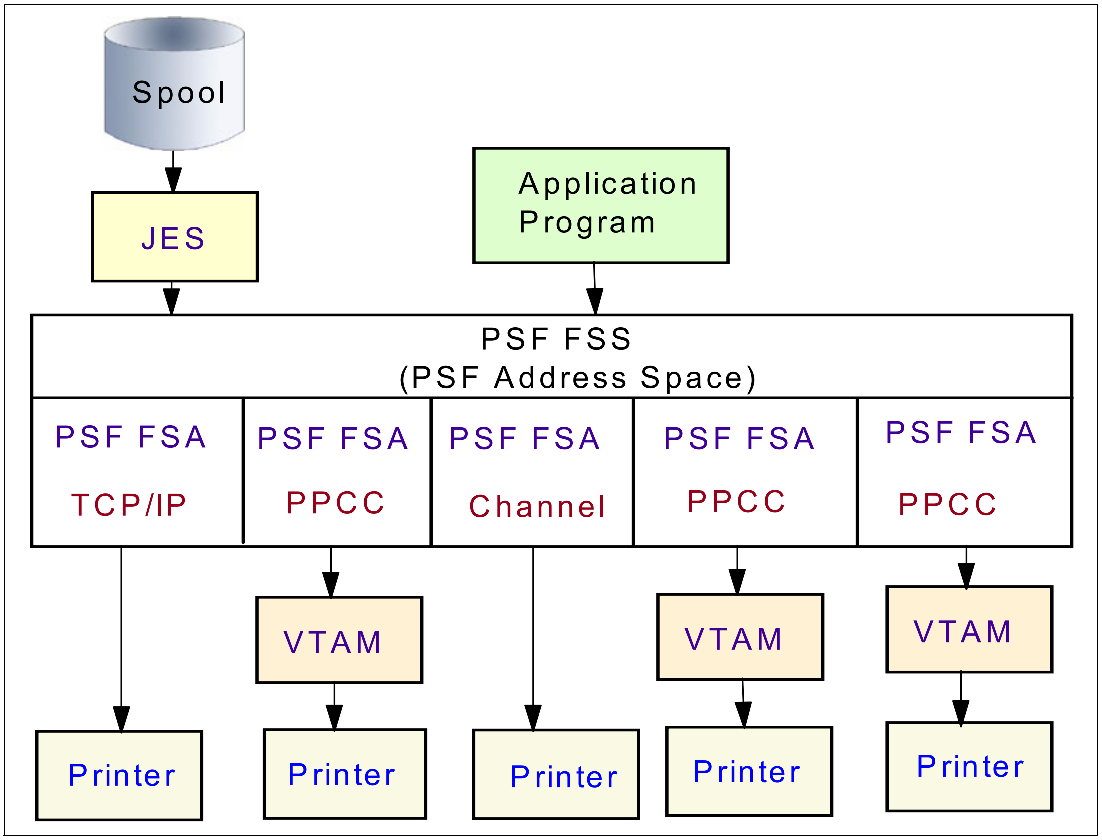


Figure 15-1 PSF 3.4.0 attachment environment

15.3.1 User environment

The PSF AFP1 user environment consists of:

- JES operators
- Printer operators
- Host operators
- Network operators (when printers are network connected)
- Operators of System Display and Search Facility (SDSF) or SDSF comparable products (when SDSF is used to control printers)
- User application programs

15.3.2 Application environment

PSF can support all types of Advanced Function Printers (AFP) concurrently, including channel- and communications-connected AFP printers and TCP/IP-connected AFP printers. Figure 15-2 illustrates the various printers this release supports.

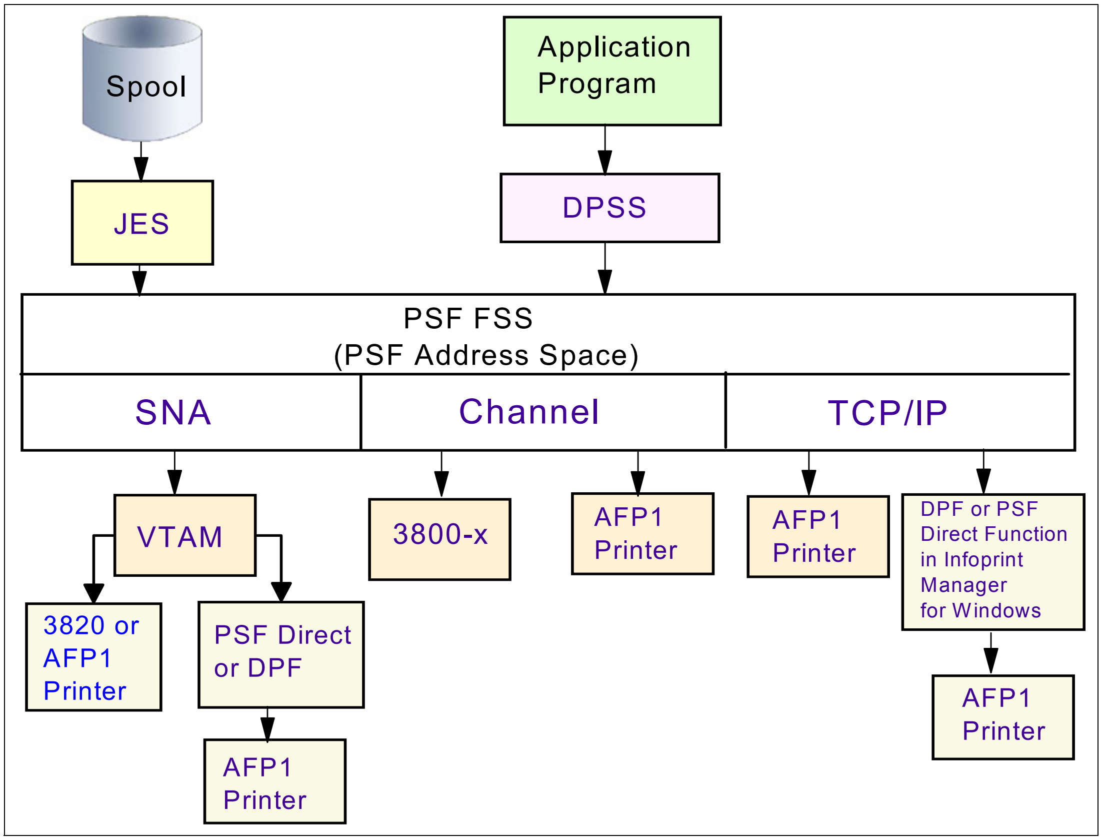


Figure 15-2 PSF 3.4.0 printer support

15.3.3 Supported printers

PSF supports the following types of printers:

- SNA attached
- TCP/IP

SNA-attached printers

PSF can send data to an SNA-attached printer in the same or a different SNA network domain as that where PSF resides. The network configuration for SNA-attached printers is contained entirely within VTAM. PSF has no knowledge of the network configuration beyond the Logical Unit name of the printer. Differences in processing based on the network configuration are handled by VTAM. PSF supports both LU 6.2 and LU1 printers.

TCP/IP-attached printers

PSF views TCP/IP-attached printers in the same way as it views SNA-attached printers. PSF performs error recovery, handles intervention required situations, and displays messages as it does for SNA-attached printers.

TCP/IP software provides universal communication services (interfaces) between physical networks and applications. The communications services reside at the network layer and are independent of the topology of the underlying physical network.

The routing of information is determined on the basis of the IP address, and is performed by IP gateways. After the network is configured correctly, the MVS host appears to be communicating directly with the TCP/IP-attached printer.

15.4 PSF 3.4.0 functional enhancements

The PSF 3.4.0 for z/OS new function includes the following enhancements:

- TrueType/OpenType font support
- Resource object capture (TrueType/OpenType fonts only)
- HFS support (TrueType/OpenType fonts only)
- Pre-rip support
- ASAP report enhancements
- Trace entry documentation

15.4.1 TrueType/OpenType font support

TrueType and OpenType fonts are commonly used in the Windows and Mac environments, but are not directly supported within the AFP environments. The AFP environment has been enriched and made more flexible by supporting these non-FOCA (Font Object Content Architecture) fonts. In PSF 3.4.0, these non-FOCA resources (TrueType and OpenType fonts) are supported as Data Object Font (DOF) resources or objects. To understand TrueType/OpenType fonts, we provide the following definitions:

Non-FOCA font These are fonts that are not defined by the IBM FOCA architecture. In this specific use of the term, it identifies the non-FOCA fonts that are supported by PSF. The non-FOCA fonts supported by PSF are TrueType fonts and OpenType fonts.

TrueType font An outline font format jointly developed by Microsoft and Apple that consists of a single file containing several tables and sub tables.

OpenType font OpenType is an open standard that extends the TrueType font technology. OpenType was jointly developed by Adobe and Microsoft. In particular, OpenType defines tables that can be used to carry the formatting information needed to fully support Unicode.

Data object font Data Object Font (DOF) is the MO:DCA and IPDS term for the architecture that makes it possible for PSF and the printer to converse about TrueType/OpenType fonts. The terms TrueType/OpenType fonts and DOF objects are used interchangeably in this section.

Resource access table A resource access table (RAT) is a table of TrueType/OpenType font format names, paths, filenames, and font attributes. The RAT resides in the same path as the TrueType/OpenType font. Every directory with installed TrueType/OpenType fonts has a RAT.

Font installer Introduces TrueType/OpenType Fonts to the system (installs and builds the Resource Access Table (RAT)).

RAT access utility The RAT access utility (RAU) is used to access the RAT for installed fonts. It also maps TrueType/OpenType Fonts from their formal name to a path and file name for PSF and AFP enablers using the Resource Access Table (RAT).

Text fidelity A directive to the printer that tells the printer how to handle certain errors when processing PTOCA controls.

PSF 3.4.0 supports non-FOCA fonts via the new Data Object Fonts (DOF) constructs in the MO:DCA and IPDS architecture. Documents printed by PSF are allowed to reference TrueType/OpenType fonts. All TrueType/OpenType font objects reside in an HFS (Hierarchical File System) or a zFS (z/OS File System) in their stored format. That is to say, these objects cannot reside in traditional PSF PDS/PDSE libraries.

DOF objects

A Data Object Font (DOF) object can be presented to PSF in two forms:

Wrapped (MO:DCA) As a MO:DCA object. This object is a TrueType/OpenType font inside of a MO:DCA Data Object Font construct. A wrapped DOF object is only used to place the DOF object inline in a MO:DCA data stream. Since PSF does not know how to interpret the TrueType/OpenType font data stream, the MO:DCA wrappers allow PSF to find the bounds of the DOF object and can introduce other DOF attributes to PSF such as font descriptors.

Unwrapped As a TrueType/OpenType font without the MO:DCA wrapper constructs. Therefore, this object cannot contain any MO:DCA specific information. All TrueType/OpenType fonts that exist as files in directories are processed as unwrapped DOF objects. They cannot be wrapped.

DOF organization

A Data Object Font (DOF) object can be stored as a TrueType/OpenType font, a TrueType/OpenType font collection, or a TrueType/OpenType linked font. These three forms (or packages) of TrueType/OpenType fonts are supported by PSF:

Font This is the basic TrueType/OpenType font element. It is a grouping of characters all of the same type face and style. It is stored as a single object or file.

Collection This TrueType/OpenType font form is a group of TrueType/OpenType fonts collected together and stored as a single object or file. Font collections are created by the font vendor.

Linked fonts With this TrueType/OpenType font form, a group of TrueType/OpenType fonts are associated by linking. There is one base font object or file. The base font object is linked to the other TrueType/OpenType fonts known as linked fonts. Each linked font is a TrueType/OpenType font object or file. Linking of fonts is performed at font installation time with the Font Installer.

DOF repositories

PSF supports references to non-FOCA fonts (DOF objects) that reside in various resource repositories from the print file itself to file system resource libraries and the printer resident set of DOF objects. For traditional MO:DCA resources the term *library* means the data set (PDS or PDSE) or group of data sets which contain resources. For non-FOCA fonts the term library means the path or set of paths that contain the TrueType/OpenType font resources. From here on, we refer to the non-FOCA font library as a *path library* as opposed to a PDS library used to store AFP fonts.

Note: There are no plans for non-FOCA fonts (DOF fonts) to reside in MVS PDS libraries and to be supported by PSF.

DOF Objects can be found in any of the following places (repositories):

- Resident in the printer
- Inline in the print file
- User path library
- System path library

This PSF support allows the installation's system programmer to identify the non-FOCA font path library to PSF, provide a mechanism for the print file creator to specify a private or user level TrueType/OpenType font path library, and process the fonts required for printing a document. In the following sections, the DOF object repositories are discussed in more detail.

Printer-resident TrueType/OpenType fonts

These TrueType/OpenType fonts can reside in the printer. The resource could have been shipped with the printer or captured by the printer during some previous use. If an object identifier (OID) for the TrueType/OpenType font is specified (in the RAT), PSF attempts to activate this DOF object in the printer.

Inline TrueType/OpenType fonts

These TrueType/OpenType font resources can be contained in the print file in a MO:DCA resource group structure. PSF looks for the specified DOF object inline after first trying to activate the printer-resident version of the object.

User path library

This font resource can reside in a path library identified by the print file producing application. The print application owner can identify a private path for TrueType/OpenType fonts to PSF using the new USERPATH parameter on the OUTPUT JCL statement. This TrueType/OpenType Font repository must be a zFS or HFS.

PSF attempts to locate all TrueType/OpenType fonts identified in the data stream in this user path library after it has looked for the specified font inline. If the font cannot be located in the user path library, PSF looks for the specified TrueType/OpenType font in the system path library.

System path library

The non-FOCA font can be contained in a path library specified to PSF in the PSF startup procedure. The printer administrator or system programmer must identify the TrueType/OpenType font repository to PSF using the new FONTPATH parameter on the PRINTDEV JCL statement in the PSF startup procedure. This TrueType/OpenType Font repository must be a zFS or HFS.

PSF looks for specified TrueType/OpenType fonts in the system font path after exhausting other possibilities for locating the font elsewhere. PSF searches the system path libraries for the TrueType/OpenType font in the order the paths are specified in the indicated system path DD statement.

Table 15-1 shows where PSF looks for a DOF object and what form (wrapped/unwrapped) it can or must have to process it.

Table 15-1 DOF object supported by PSF

Description	Printer	Inline	User path	System path
MO:DCA	Not applicable	YES	Not supported	Not supported
Unwrapped	YES	Not supported	YES	YES

Architecturally, the TrueType/OpenType fonts can exist in the following versions:

- Inline in a resource group with MO:DCA Data Object Font (DOF) wrappers
- In a library with DOF wrappers (not supported in this PSF release)
- In a library with a Resource Access Table (RAT) entry and no DOF wrappers
- In a library without a RAT entry and no DOF wrappers (not supported in this PSF release)

Note: Only TrueType/OpenType fonts that are printer resident, inline, or installed in a library with a RAT entry are supported in PSF.

Text fidelity

The user can control how the printer handles exceptions encountered while processing presentation text object content architecture (PTOCA).

With the PPFA FORMDEF parameter TEXTERROR, the user can specify how the printer is to handle IPDS exception X'0200..01' in the PTOCA controls. The user can specify that the printer should:

- Ignore the exception
- Report the exception, and continue processing
- Stop processing for the exception

PSF transforms the TEXTERROR parameter in the FORMDEF into IPDS text fidelity controls for the printer.

Installing the software

It is necessary to add and/or upgrade certain IBM products in your AFP printing environment to use TrueType/OpenType fonts. The software collection for using DOF objects includes, but is not limited to:

- Font Installer
- PSF 3.4.0
- RAT Access Utility (RAU)

While other pieces of software (such as ACIF and PPFA) assist in the handling of TrueType/OpenType fonts, for this installation discussion only the three listed here are important.

Font Installer

The Font Installer is a new IBM tool for the installation and maintenance of TrueType/OpenType fonts. It is installed and run in the Windows environment.

PSF 3.4.0

PSF 3.4.0 is packaged, orderable, and installable as it has been in all previous releases. There is nothing new here.

RAT Access Utility (RAU)

Since the RAU is a dependency of PSF 3.4.0, it is packaged with PSF. The executable is a member of the PSF 3.4.0 feature. It is distributed and installed with the PSF product. As a result of installing PSF 3.4.0 for z/OS, the RAU gets installed. The RAU is used by other AFP features in addition to PSF.

Using DOF objects with AFP

The following sections contain a high level explanation of the relationships that PSF has with the font installer, RAU, path libraries, customer data, and the printer. The discussion is broken into the following subjects:

Installing the DOF objects	Adding the TrueType/OpenType font objects to your AFP printing environment
Identifying DOF object path libraries	Specifying where DOF objects live
Referencing DOF objects	Identifying the DOF objects used by a document
Processing DOF objects	Locating and retrieving DOF objects for printing

Installing the DOF objects

IBM does not offer a TrueType/OpenType font package. However, there are several font vendors that provide these font packages. It is the customer's responsibility to select, purchase, and install a font package that can be used on all the platforms requiring these fonts.

A TrueType/OpenType font cannot be used by PSF until it has been installed into a path library (either system path library or user path library). The steps to install a TrueType/OpenType font object are as follows:

1. Perform the font vender font install process.

   After selecting a TrueType/OpenType font vender and ordering their font package, install the package according to the vendor’s instructions. Be sure that the font license agreement allows the fonts in the package to be copied to or mounted to multiple platforms (for example Windows, z/OS, UNIX). The vendor font install process may install the complete set of supplied TrueType/OpenType fonts or may allow the user to select which fonts to install. When complete, the user has a group of TrueType/OpenType fonts in a vendor-defined font path.

2. Perform the IBM Font Installer font install process.

   Run the IBM Font Installer process to enable the TrueType/OpenType fonts to be used in the AFP environment by PSF. By running the IBM Font Installer you can:

   – Enable the non-FOCA fonts for use by PSF

   – Enable the non-FOCA fonts for capture (optional)

- Link the non-FOCA fonts (optional)

Now you have the desired TrueType/OpenType fonts enabled for the AFP environment. As part of this install process the fonts can also be:

- **Enabled for capture:** Marked so that the printer on which the font is used retains a copy of the font. This copy can then be activated for later uses of the font so the font does not need to be downloaded every time it is needed. Capturing a font is a printer-specific function. Not all printers support this capability.
- **Link fonts**: Tie multiple TrueType/OpenType fonts together in a group of fonts so that a single font reference causes the entire set of fonts to be downloaded to the printer. The first font is the base font and all subsequent fonts are the linked fonts. Referencing the base font gets all the linked fonts downloaded to the printer as well.

3. Enable the font path library to the z/OS environment.

   With the TrueType/OpenType vendor fonts installed and IBM Font Installer enabled, it is time to enable the path library to the z/OS environment. The font path library must be File System (FS) mounted to the z/OS environment as the font path library. To use the z/OS font path library as a system path library with PSF, it must be identified in the PSF startup procedure using the FONTPATH parameter on the PRINTDEV JCL statement.

Figure 15-3 shows the DOF installation process.

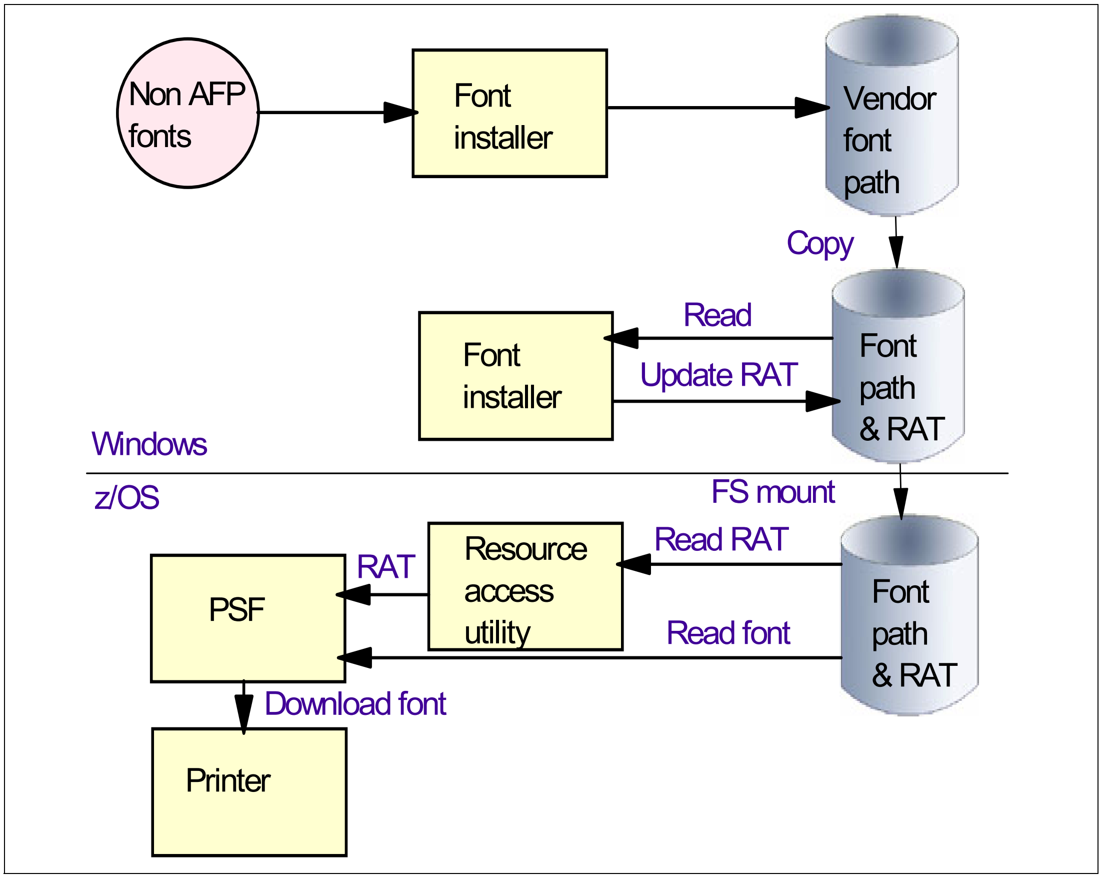


Figure 15-3 DOF installation

Identifying the DOF object path libraries

The DOF objects must be installed in a path library. They can be installed in the default level system path library or in the print file specified user path library.

The system path library can be identified to PSF. The system path library is where PSF tries to find a DOF object if PSF cannot find that object anywhere else (that is, in the printer, inline in the print file, or in the user path library).

- Add DD statements to the PSF startup procedure for the desired path libraries. Since non-FOCA fonts reside in file systems, these DD statements have PATH parameters and not DSN parameters.
- The customer must add the FONTPATH parameter to the PRINTDEV statement in the PSF startup procedure. The FONTPATH parameter references the proper path library DD statement (also in the PSF startup procedure).

The user path library can be specified by the application for printing (or generating) the print file. The user path library is used to locate the DOF object after PSF has tried locating the DOF object in the printer and inline in the print file. If PSF cannot locate the DOF object in the printer, inline, and in the user path library, PSF looks for the DOF object in the system path library.

- Add a USERPATH parameter to the OUTPUT statement in the print file generating application.
- The USERPATH parameter on the OUTPUT statement is optional.

Referencing DOF objects

The TrueType/OpenType fonts can be referenced through the use of MO:DCA constructs. These MO:DCA constructs are generated by the document formatter. A TrueType/OpenType font reference is achieved with the Map Data Resource (MDR) structured field. An MDR structured field that references a TrueType/OpenType font is said to be referencing a Data Object Font (DOF) resource.

The MDR for DOF object is specified in the MO:DCA environment group construct. There are three environment groups which typically contain MDR structured fields: the Resource Environment Group (REG) at the beginning of a print file, the Active Environment Group (AEG) at the beginning of a page, and the Object Environment Group (OEG) for GOCA and BCOCA™ objects.

Within the MDR for a DOF object the TrueType/OpenType font name (or DOF object name) is specified by the full font name (for example, Arial Narrow Bold Italic).

Furthermore, the MDR for a DOF object can contain the name of an AFP code page to be used with the DOF object. PSF downloads and/or activates the specified code page and binds it to the DOF object for processing this print file.

For a DOF object to be processed correctly by PSF, the DOF object referenced in a print file must:

- Reside in the printer (be printer resident).
- Reside inline in the print file.
- Reside in the user path library and be properly installed.
- Reside in the system path library and be properly installed.

For line data print files the MDR for the DOF comes from the PAGEDEF. The PAGEDEF contains data maps for page formats that can have MDR structured fields that reference TrueType/OpenType fonts.

Processing DOF objects

For every DOF object reference encountered by PSF, PSF does one of the following (whichever comes first):

- Expose the new DOF object to the Installation Exit 7 if it is active. Like the Data Object Resource (DOR), with Exit 7 processing, the proper font name can be changed by the installation.
- Determine that the DOF object has already been processed and is active in the printer.
- Activate the DOF object from the printer-resident set of DOF objects.
- Download the DOF object from the inline set of DOF objects.
- Download the DOF object from the user path library set of DOF objects.
- Download (and optionally capture) the DOF object from the system path library set of DOF objects.

If the referenced DOF object is part of a TrueType/OpenType font collection, the entire font collection is activated/downloaded to the printer.

When the referenced DOF object is a base font of a linked TrueType/OpenType font, the base font and all the linked fonts are activated/downloaded to the printer.

PSF activates/downloads and ties together any AFP code page referenced in the MDR along with the DOF object.

For any DOF object reference encountered by PSF that results in the DOF object being downloaded to the printer, PSF enables the DOF object for printer capture if the font object is installed in the system path library and enabled for capture.

The DOF object stays active in the printer based on the criteria shown in Table 15-2.

Table 15-2 DOF longevity

Origin of DOF	Activation longevity	Stored longevity (in printer)
Printer resident (shipped)	Min: Current page Max: Current printfile or longer	Forever
Printer resident (captured)	Min: Current page Max: Current printfile or longer	Until space is required for another captured resource
Inline	Min: Current page Max: Current printfile	Never captured
User path	Min: Current page Max: Current printfile	Never captured
System path (Captured)	Min: Current page Max: Current printfile or longer	Until space is required for another captured resource
System path (not captured)	Min: Current page Max: Current printfile or longer	Never captured

In PSF, the activation longevity of a DOF object is controlled by the RRLV for the font objects. The RRLV is installation controllable, giving the installation some ability to determine the amount of font data kept in the printer between print jobs.

What is supported

If all the dependencies listed are met, all of the following statements are true:

- The user is able to print using TrueType/OpenType fonts:
  - Line data
  - MO:DCA data
- TrueType/OpenType font objects can reside:
  - Resident (or captured) in the printer
  - Inline with MO:DCA wrappers
  - In user path libraries (installed)
  - In system path libraries (installed)
- TrueType/OpenType fonts can be mapped:
  - For a complete printfile (REG)
  - For a page (AEG)
  - For GOCA and BCOCA objects (OEG)
- TrueType/OpenType fonts cannot reside in a traditional PDS/PDSE resource library.
- TrueType/OpenType fonts are organized:
  - In single font objects
  - In collections of font objects
  - In linked lists with a base font and multiple linked fonts
- TrueType/OpenType fonts:
  - Can be enabled for capture by the printer
  - Can be activated in the printer from resident or captured font objects
  - Can use an AFP code page
  - Are referenced by the long proper font name
  - Can be referenced via an OID in the MDR or RAT
- A print file may contain references to both:
  - AFP fonts
  - TrueType/OpenType fonts

References

The following MO:DCA and IPDS architecture change requests (ACR) are essential for understanding the PSF 3.4.0 support of TrueType/OpenType Fonts:

- ACR #105 to MO:DCA REL. 5.0, Base Unicode Support with TrueType Fonts
- ACR #109 to MO:DCA REL. 5.0, Encoding Scheme ID (X'50') Triplet Extensions
- ACR #113 to MO:DCA REL. 5.0, Base Unicode Support Additions
- ACR #115 to MO:DCA REL. 5.1, Resource Access Table (RAT)
- IPDS ACR #384, Data-Object Fonts (TrueType/OpenType)
- IPDS ACR #392, Text Fidelity Control
- Registry of PSD-Owned Object Identifier (OID) Branches in the ISO Naming Tree, Reinhard Hohensee.

Dependencies

TrueType font support in PSF 3.4.0 has the dependencies shown in Table 15-3.

Table 15-3 Dependency

Dependency	Explanation
Font installer	Installs a TrueType font into the path library by generating an OID and creating a RAT entry for the font.
RAT access utility	Looks for the specified TrueType Font via the full font name using the RAT and returns a path library, file name, and other font attributes to PSF.
PPFA	For PSF to be able to print line data requiring DOF objects, PPFA changes are needed to generate PAGEDEF objects with MDR structures that reference the DOF objects.
ACIF	Wraps TrueType Fonts with MO:DCA Object Containers.
Scheduler JCL changes	Defines the JCL for specifying system path libraries and user path libraries. See HFS Support in this document.
Printer support	Must have printers that support TrueType Fonts for testing.

15.4.2 Resource object capture (TrueType/OpenType fonts only)

In this release, PSF supports resource object capture for the TrueType/OpenType font object. This provides TrueType/OpenType font support equivalent functionality to AFP font support.

The TrueType/OpenType font object is also referred to as a Data Object Font or DOF object when it is referenced in the MO:DCA data stream. These DOF object references to TrueType/OpenType font objects can designate the font for capture by the printer and therefore the font can become a printer-resident TrueType/OpenType font object. A captured TrueType/OpenType font object can exist in the printer's storage across many printer IMLs or PSF sessions and can subsequently be activated without downloading the font object to the printer again.

There can be two classes of printer-resident TrueType/OpenType font objects in the printer. Printer-resident fonts can be shipped or captured:

Shipped These are the manufacturer's shipped TrueType/OpenType fonts. These fonts are with the printer when it comes from the manufacturer. These fonts also stay in the printer's storage forever. They are permanent.

Captured A captured TrueType/OpenType font is acquired by the printer through dynamic processing or printing. When a TrueType/OpenType font is downloaded to the printer, it can be nominated for retaining across printer sessions (it can be nominated for capture). Unlike shipped fonts, captured fonts can be deleted from the printer storage any time the printer has a higher priority object or function to process. They are temporary.

This discussion is about the *captured* printer-resident fonts. However, the shipped printer resident fonts may be mentioned to draw parallels or make distinctions. As long as a captured printer-resident font remains in the printer's storage, it behaves like a shipped printer-resident font.

What can be captured?

Any of the following TrueType/OpenType font objects can be processed by PSF as captured printer-resident fonts:

- A single TrueType/OpenType font object.
- An entire TrueType/OpenType font collection.

The steps for making a TrueType/OpenType font object a captured printer-resident font are the following:

1. Font is enable for printer capture.

   As part of the font installation process for TrueType/OpenType fonts, the font can be enabled for printer capture. This process assigns an Object ID (OID) to the font. This enabling for printer capture is performed by the Font Installer.

2. Font is captured by the printer.

   When a TrueType/OpenType font object has been enabled for capture, PSF can suggest the printer retain the font object (capture it) when it is downloaded to the printer. PSF only nominates a TrueType/OpenType font object for capture if the font object was found in the system font path library. When allowing the capture of a TrueType/OpenType font object, PSF uses the OID from the enabling process. This OID is then the mechanism for recalling the font later.

   A captured TrueType/OpenType font object is a printer-resident font object. It remains in the printer storage across many print sessions. The printer can, however, delete the captured TrueType/OpenType font object if storage is needed for some other higher priority object or function. If a TrueType/OpenType font object gets deleted from printer storage, it may become captured again on subsequent DOF object references in a print file.

3. Font is activated in the printer.

   Once a TrueType/OpenType font object has been captured by the printer (or if the font object was shipped with the printer), that font object can be activated using the assigned OID for the font object. At any point in the TrueType/OpenType font processing, if PSF encounters an OID for the DOF object reference, PSF asks the printer to recall the font object from printer storage rather than download the font. This is called activation.

   If activation of the font fails, PSF downloads it. In this case the PSF rules for TrueType/OpenType font object capture prevail.

15.4.3 RHFS support (TrueType/OpenType fonts only)

Beginning with PSF 3.4.0 for z/OS, TrueType/OpenType fonts are supported. These non-FOCA fonts must reside in a z/OS UNIX System Service file system (zFS or HFS) to be used. PSF obtains the TrueType/OpenType font from the HFS resource path library instead of from the conventional PDS resource library.

The following definitions are used in the section:

PDS A single PDS or PDSE data set containing AFP resource members.

PDS library A set of one or more PDSs or PDSEs in which PSF searches for a particular resource member.

Path A single file system path containing non-FOCA resource files.

Path library A set of one or more file system paths in which PSF searches for a particular non-FOCA resource file.

All AFP resources still reside only in the traditional PDS libraries and cannot reside in the path library. The non-FOCA resources, TrueType/OpenType fonts, reside in the new path library and cannot reside in the traditional PDS libraries.

PSF is supporting TrueType/OpenType fonts as they reside in a path (or font) library instead of in the PDS library because:

- TrueType/OpenType fonts are commonly used in the Windows and Mac environments. Therefore, the file system organization is the traditional home of TrueType/OpenType fonts, which are files in the file system.
- These objects could have very long and descriptive file names. These long names are not supportable because a PDS supports only 8-byte member name.
- The file system is more flexible because it is readily shareable across multiple platforms. Thus, the path library for TrueType/OpenType fonts is easily shared between the PSFs and other print servers in all environments.
- The alternative is to require the enterprise to migrate the TrueType/OpenType fonts to the traditional PDS library. This implies complicated reblocking, renaming, and/or name mapping algorithms.

In addition to residing in the new path library, TrueType/OpenType fonts must be installed into the path library. This is accomplished using the Font Installer program.

To access the individual resources (TrueType/OpenType fonts) in the path library, PSF uses UNIX System Services assembler callable services. These services provide OPEN, READ, and CLOSE functionality for this path library support.

There are two different path libraries that can be specified for TrueType/OpenType fonts:

- User path library
- System path library

User path library

The print application can optionally identify a private path or set of paths (directory or set of directories) for TrueType/OpenType Fonts using the OUTPUT JCL statement. Use the new USERPATH parameter on the OUTPUT JCL statement to do this. This TrueType/OpenType font repository must be a file system accessible using UNIX System Services.

In the OUTPUT statement, you can think of the USERPATH parameter for non-FOCA resources as being similar to the USERLIB parameter for AFP resources. In both cases, these parameters tell PSF where to look for resources that are specific to the print file being processed.

PSF attempts to locate TrueType/OpenType fonts in this user path library (if one is specified) before searching for it in the system path library. If the TrueType/OpenType font cannot be found in the user path library, PSF looks in the system path library. The USERPATH parameter on the OUTPUT JCL statement specifies path names (library).

Figure 15-4 shows an example of a JCL fragment with USERPATH parameter. Here, TrueType/OpenType user paths '//jdoe/fonts/truetype' and '//jdoe/fonts/truetype/myfonts/' are searched for TrueType/OpenType fonts before looking in the system path library. In this example AFP fonts and TrueType/OpenType fonts are mixed in the printfile.

```
.
.
//OUT01    OUTPUT ...,
//             USERLIB=('MY.PRIVATE.RESOURCE',
//             'MY.PRIVATE.FONTS'),
//             USERPATH=('//jdoe/fonts/truetype',
//               '//jdoe/fonts/truetype/myfonts/')
//             ...
//PRTFILE  DD  SYSOUT=*,OUTPUT=*.OUT01
..
```

Figure 15-4 Example of USERPATH parameter

> **Note:** Each path name can be up to 255 characters long including the “/” characters. Up to 8 paths can be specified on the USERPATH parameter of the OUTPUT JCL statement. PSF searches these user paths in the order they are specified.

System path library

The printer administrator or system programmer must identify the TrueType/OpenType font system path library (directory or set of directories) to PSF. This TrueType/OpenType font repository must be a path library and must be identified to PSF with a DD statement and PATH parameter in the PSF startup procedure.

In the PRINTDEV, you can think of the FONTPATH parameter for TrueType/OpenType resources as being similar to the FONTDD parameter for AFP resources. In both cases, these parameters tell PSF where to look for fonts that are generally used for printing.

PSF looks for specified TrueType/OpenType fonts in the system font path after looking for the font in the user path library (if one is specified). PSF searches the system path library for the TrueType/OpenType font in the order the paths are specified in the indicated system path DD statement.

The DD statement used to identify the TrueType/OpenType Font system path library may have any DD name. The DD name for the TrueType/OpenType Font system path library must be specified in the PRINTDEV JCL statement for any printer that needs these fonts. The TrueType/OpenType font system path library DD name is identified to PSF with the FONTPATH parameter in the PRINTDEV statement.

Figure 15-5 shows a PSF startup procedure example JCL fragment. In this example, TrueType/OpenType font system paths '//u/fonts/truetype' and '//u/fonts/truetype/local' are searched for any requested TrueType/OpenType fonts.

```
..
//FONT01   DD  DSN=SYS1.OLNFONTS,...
//         DD  DSN=SYS1.RASTFONT,...
//TTFONT02 DD  PATH='//U/FONTS/TRUETYPE',...
//         DD  PATH='//U/FONTS/TRUETYPE/LOCAL',...
//PRT601   PRINTDEV  ...,
//             FONTDD=*.FONT01,        AFP FONTS
//             FONTPATH=*.TTFONT02,    OTHER FONTS
//             ...
..
```

Figure 15-5 Example of FONTPATH parameter

- The length and syntax of the value specified by the PATH parameter for the system path library is dictated by the z/OS system JCL PATH parameter.
- The value specified on the FONTPATH parameter of the PRINTDEV JCL statement is a DD name and is therefore limited to 8 characters and other coding rules for JCL DD names.

If PSF cannot access a specified path (either user path or system path), message APS513I is generated and the print file is not printed. When PSF cannot find the TrueType/OpenType font file requested, message APS512I is generated and the printing for this print file is canceled.

15.4.4 Pre-rip support

The Resource Environment Group (REG) support added to PSF 3.2.0 for OS/390 via APAR OW44362 allows preloading of complex resources before printing of the first page is started. This can avoid device underruns that might occur if the resource downloading took place between pages.

With PSF 3.4.0 for z/OS, resource preprocessing support has been added which is also known as pre-rip support. Resource preprocessing is an extension of the resource preloading concept. It can be used with objects that have a long rasterization time, and causes this rasterization to be done after the resource is preloaded, but before printing of the first page is started. This can avoid device underruns that might occur if such rasterization took place between pages. Examples of resource objects that might benefit from this preprocessing are large IOCA FS45 image objects (that need to be rotated, scaled, or trimmed) and complex EPS and PDF objects.

For example, with the PSF 3.2.0 for OS/390 REG support, the customer can preload an IOCA FS45 image but it still needs processing on a page basis every time it is included via an Include Object (IOB) and needs rotating, scaling, and/or trimming. With printers such as the Infoprint Color 130 Plus, this can be very undesirable since it takes a long time to stop this printer and a lot of blank paper is generated.

With pre-rip support, PSF honors the Preprocess Presentation Object structured field in a REG. This PPO structured field causes PSF to send the Rasterize Presentation Object (RPO) IPDS command to the printer. The preloaded object specified by the RPO command is processed by the printer as if it had been included directly in the page or overlay via an IDO (or an IO command in a page). The result of this pre-ripping is cached and not printed yet. Later, when the IO/IDO is received, the printer actually places this object on a page.

15.4.5 ASAP report enhancements

In PSF 3.3.0 the AFP Statistics option was added to enable a user to generate a report which provided detailed and summary information about the resources used to print a document (for example, the library from which PSF loaded a resource).

For PSF 3.4.0 the following enhancements are added to the AFPSTATS Report option:

- Resource reload EVENT record
- GRID font support
- Font mapped EVENT record
- Resource substituted EVENT record
- Multiple print files per job step
- Unused inline resource EVENT record
- Transmission count

Resource reload EVENT record

During printing of a data set installation exit 07 may get called at resource-access time when PSF determines a resource is needed to continue printing and the exit may request that the resource be reloaded. The exit is called once for each of the following resources: map medium overlay, map page segment, map coded font, map page overlay, include page segment, and include page overlay.

The user may not know if, which, or when the resource reload occurred and whether the reload was successful. For example, if the resource is an inline resource, the resource cannot be reloaded.

Using the current AFP Statistics function the user can't determine this from a report. To aid in this determination, support has been added to the AFPSTATS function so a resource reload EVENT record is written when a reload is requested, whether or not the reload is successful.

The following types of resource reload EVENT records are created:

- Resource reload successful.
- Resource reload unsuccessful.
- Resource reload ignored - inline resource.
- Resource reload not permitted.

The resource name and the page number on which the reload was requested is included in the record. The Resource reload EVENT record is included in the Processing Detail and Processing Summary sections of the report.

GRID font support

When PSF encounters a GRID in an MCF structured field, it attempts to map the GRID to a character set name and a code page name. If successful, information for the character set name and the code page name is currently provided in the report.

If the mapping is not successful, PSF assumes it is a resident coded font and activates it using the GRID. Information is provided in the report for a coded font identified with a member name; however, information is not provided for a coded font identified by a GRID. A GRID is not equivalent to a member name. It is an eight-byte ID (four 2-byte fields) that identifies the coded font so it's not obvious to a user what coded font was used.

Support has been added to the AFPSTATS function where information on a GRID font is collected and provided in the report. Therefore, in the report where a coded font name is provided the GRID is also provided, but is broken into the four two byte fields:

GCSGID	Graphic character set global identifier
CPGID	Code page global identifier
FGID	Font global identifier
FW	Font width

Font mapped EVENT record

There are many ways to map a font. The different mapping functions performed by PSF are done at different times during the processing of the font, and the type of mapping performed depends on how the font was specified in the job. Plus a font can be mapped several times.

Mappings, supported by PSF, are:

- Name to GRID

- GRID to name
- Outline to Raster
- Raster to Outline

These mappings may make it difficult for a user to correlate the font originally specified in the print file to the final font after all the mappings have been done. Because of this, support has been added to the AFPSTATS function where every time a font is mapped a Font mapped EVENT record is written. This record is included in the Processing Detail and Processing Summary sections of the report.

The Font mapped EVENT record contains:

- Why the font was mapped; for example, an outline font was specified, but the printer only supports raster fonts
- Name of the font being mapped and the name of the font after mapping
- The page number where the font was referenced

Resource substituted EVENT record

During printing of a data set, installation exit 07 may get called at resource-access time when PSF determines a resource is needed and the exit may substitute for the current resource with another resource. The exit is called once for each of the following resource types: map medium overlay, map page segment, map coded font, map page overlay, include page segment, include page overlay, object container, PAGEDEF, and FORMDEF. PSF supports substitution of all types of resources with some restrictions.

Font substitution and font mapping are not the same. *Font substitution* is the exit substituting one font for another. The *mapping* process is used by PSF to verify the specified font can be used in the current environment or to find an equivalent font that can be used.

The exit may request a resource substitution, but for some reason the substitution failed. The user may not know the resource substitution failed and because this information isn't provided in the AFPSTATS report it may be difficult for a user to determine it.

Support has been added to the AFPSTATS function where every time a resource is substituted, a Resource substituted EVENT record is written. The record includes the original resource name and the name of the substituted resource and the number of the page being processed. This record is included in the Processing Detail and Processing Summary sections of the report.

Multiple print files per job step

In a job step, multiple print files can be specified by coding a DD statement for each of the print files in the job step. An AFPSTATS report can be generated for each of the print files; however, there is no information in the report that can be used to determine which report belongs to which print file.

Each DD statement must have a unique name. Support has been added to the AFPSTATS function where the name of the DD statement associated with a print file is provided in the Print File Information section of the report. This enables the user to correlate the DD statement with it's associated print file.

Unused inline resource EVENT record

The user can specify inline resources in line data or MO:DCA-P data that should be used by PSF when printing the data set, but there is no guarantee that they are used. For example, if the name of an inline FORMDEF matches the FORMDEF name specified in the OUTPUT statement FORMDEF parameter, the inline FORMDEF is used. If they don't match, the inline FORMDEF is not used and PSF loads the FORMDEF from a library. There are also several other rules PSF uses when determining if an inline resource should be used. The supported inline resources are:

- FORMDEFs
- PAGEDEFs
- Page segments
- Overlays
- Fonts (code pages, font character sets, coded fonts)
- Color map tables
- EPS resources
- BCOCA, IOCA, GOCA objects
- Microfilm setups
- TrueType/OpenType fonts

If an inline resource is used, information on the resource is currently provided in the report. If not used, the user may be able to determine this from a report assuming they know all the inline resources in the print file, but it's not obvious.

Changes have been added to the AFPSTATS function so information for every inline resource encountered in a print file is collected whether or not the resource is used. If an inline resource is encountered and not used, an Unused inline resource EVENT record is written. The record includes the name of the inline resource, page number where it was encountered, and why it was not used. The record is included in the Processing Detail and Processing Summary sections of the report.

Transmission count

In the Summary of Pages portion in the Processing Summary section of the report, the number of transmissions of the print file is provided.

15.4.6 Trace entry documentation

The trace mapping structures for PSF internal and external traces are defined in these control blocks:

APSGITM Internal trace mapping structure.

APSGXTM External trace mapping structure.

An IBM Support Center representative can determine the layouts (or descriptions) of internal trace entries in APSGITM or external trace entries in APSGXTM. The prologues of APSGITM and APSGXTM describe how to find the trace entry layouts.

15.5 Migration and coexistence

With the new function support, PSF 3.4.0 for z/OS has the following migration and coexistence considerations:

TrueType font support

If you run a job with TTFs on a version of PSF prior to 3.4.0, you will get an PS369I message which says an MDR structured field is attempting to include an unsupported object class and PSF stops processing the current page.

Pre-rip support

If you run a job with a program printout in the resource environment group on a version of PSF prior to 3.4.0, you will get an APS118I message which says PSF doesn't know what this structured field is and the structured field will be ignored.

16

Communication Server for z/OS V1R5

Communication Server is the z/OS base element that supports secure TCP/IP, SNA, and UNIX networking on Enterprise systems, connecting different types of communication subsystems and applications to each other and supporting usage of various communication devices (such as terminals and printers) listed in the system's hardware configuration. Major components of Communication server are IP Services and SNA Services.

This chapter explains enhancements of the following IP and SNA services in z/OS V1R5 Communication Server:

- Enhanced TCP/IP asynchronous I/O support
- Full VLAN support
- Improved diagnostics for DLC dumps
- INOP dump enhancements
- IPv6 and firewall support for enterprise extender
- IPv6 support for SMF recording
- IPv6 support for SNA display of IP addresses
- IPv6 support for XCF, SameHost and ESCON®
- OSA performance enhancements
- Increased maximum number of allowed sockets
- IPv6 support and upgrade for Sendmail
- IPv6 support for CICS sockets API
- IPv6 support for SNMP applications
- IPv6 support for SNTP, syslogd, TFTPD and DCAS
- IPv6 support for TSO rexec and rsh and associated MVS daemons
- msys FTP customization support
- SNA DISPLAY command and system definition enhancements
- SNA DLC performance enhancements
- SNA storage enhancements
- SNA dump and trace enhancements
- SNA EE enhancements
- SNA resource definition and usability enhancements
- SNA session setup and problem determination enhancements

16.1 IPv6 support overview

IBM Communication Server for the z/OS V1R4 operating system introduces a new version of standard Internet Protocol stack IPv6. IPv6 is the next generation of the Internet protocol designed to replace the current version, Internet Protocol Version 4 (IPv4). The most significant IPv4 characteristic is 32-bit addressing space. That allows theoretically over four billion nodes. In practice, the interaction between routing and addressing makes it impossible to utilize more than a small portion of available nodes. Continued growth of the Internet could lead to the exhaustion of IPv4 addresses early in the 21st century.

IPv6 uses a 128-bit address space. That, according to RFC2374, provides practically limitless global addressability. It also adds many improvements to IPv4 in areas such as routing and network autoconfiguration. IPv6 is expected to gradually replace IPv4. During the transitional period two internet protocols will coexist for a number of years.

Not all IPv6 features are supported on z/OS V1R5. Communication server provides the following functionality:

- IPv6 functions supported in z/OS V1R5 Communication Server:
  - Applications, APIs
  - DLCs
  - Network management
  - Policy
  - Routing
  - Security
  - Enterprise extender
- Non-IPv6 functions supported in z/OS V1R5 Communication Server:
  - ITN3270 server enhancements
  - FTP enhancements
  - Sysplex enhancements
  - Network management
  - Policy
  - Security
  - Msys
  - SNA enhancements
  - Enhancements for new hardware
  - Routing enhancements
  - Mail

16.2 IPv6 support in z/OS V1R5 Communication Server

z/OS V1R5 Communication Server is a continuation of the effort started in z/OS V1R4 to provide IPv6 functionality. Figure 16-1 on page 435 shows an overview of the changes to the various components of Communication Server that have changed in this release in support of IPv6.

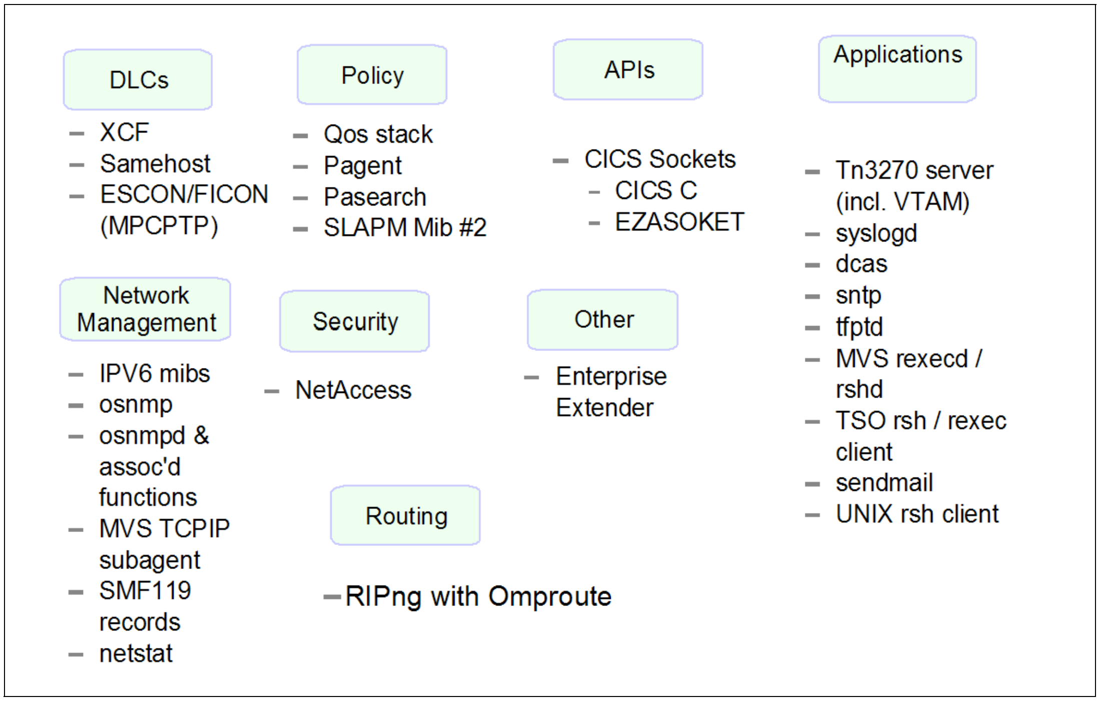


Figure 16-1 z/OS CS V1R5 IPv6 contents overview

Functions not IPv6-enabled in z/OS V1R5 Communication Server are as follows:

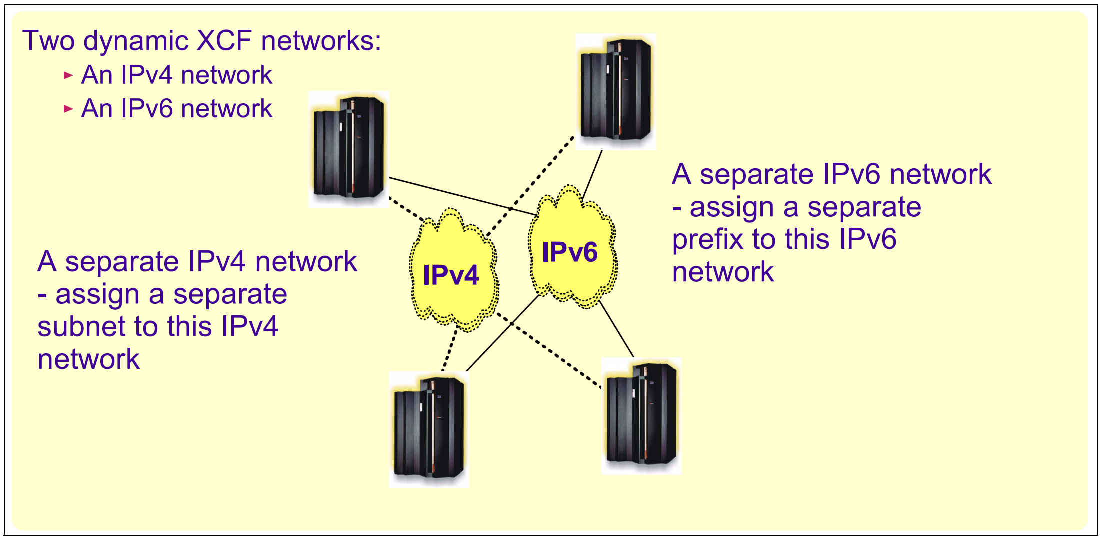
- Sysplex distributor
- Dynamic VIPA and Dynamic VIPA takeover
- Sysplex-wide security associations
- Intrusion detection services
- IPsec
- Hipersockets

16.2.1 IPv6 multipath channel point-to-point interface support

Multipath channel point-to-point (MPCPTP) Data Link Control (DLC) is updated to support IPv6 protocol. With the new support in z/OS V1R5 Communication Server (CS), a new interface type (MPCPTP6) can be used to carry IPv6 traffic. Multipath channel point-to-point interface support extends connectivity options for IPv6, using:

- **IUTSAMEH function:** Support in VTAM that simulates virtual channel-to-channel (CTC) connectivity between z/OS CS images. IPv6 traffic over ESCON channels, over XCF links in a sysplex, or between z/OS CS images using the simulated device provided by the IUTSAMEH function in VTAM is configured using an INTERFACE statement of type MPCPTP6.
- **XCF to other stacks in sysplex:** Both static and dynamic XCF are supported in the same sysplex, but not at the same time.
- **ESCON or FICON channel:** Can connect to another z/OS image directly, but not over the channel-attached router.

The MPCPTP6 interface may point to the same TRLE as an MPCPTP DEVICE, thus allowing IPv4 and IPv6 traffic to share the same physical resources, as shown in Figure 16-2 on page 436. Prior to this enhancement, the only network attachment supported for IPv6 traffic was the OSA-Express. The Netstat DEVLINKS/-d displays are changed to describe the MPCPTP6-type interface.



Two dynamic XCF networks:

- An IPv4 network
- An IPv6 network

A separate IPv4 network - assign a separate subnet to this IPv4 network

A separate IPv6 network - assign a separate prefix to this IPv6 network

Figure 16-2 IPv6 new interface support

16.2.2 IPv6 enabled TN3270 Server

In z/OS V1R5 Communications Server, the Telnet Server supports IPv6 format IP addresses, depending on the support level of the TCP/IP stack. If the stack is running IPv6, Telnet completely supports IPv6. If the stack is running IPv4, Telnet is IPv4 capable. There is no external parameter needed in Telnet to turn on IPv6 support. Telnet is always IPv6 capable if the stack supports IPv6.

Note: If the TCP/IP stack is running in IPv4 mode, no IPv6 function is available in Telnet.

An IPv6, (AF_INET6 socket), enabled TN3270 Server provides connectivity to System Network Architecture (SNA) applications from remote TN3270 clients in the following ways:

- Supports clients with IPv6 addresses.
- Supports IPv6 addresses in messages, displays, command responses, and USS messages.
- Supports IPv6 addresses as client identifiers for all mapping statements in TN3270 server configuration that allows an IP address client identifier.
- Upgrades TN3270 server Secure Sockets Layer (SSL) to use transport-layer security (SSL/TLS) support.
- Changes have been made to VTAM to support TN3270 visibility when clients are IPv6 clients:
  - Passing IPv6 addresses to VTAM
  - Enhance VTAM displays that include IP addresses to accommodate IPv6 addresses
  - Logon exit
  - Session Management Exit (SME)
  - CMIP (common management information protocol) topology managers

16.2.3 IPv6 (AF_INET6 socket) enabled applications

Changes applied to applications are similar and provide IPv6 address support to partners. Changes are made to socket API calls and structures as well as to configuration files. The applications are as follows:

- tftpd (trivial file transfer protocol server)
- syslogd
- dcas (digital certificate access server)
- sntpd (simple network time protocol server)
- sendmail 8.12.1 (new port of sendmail picks up IPv6 enablement too)
- MVS rshd and rexecd servers
- z/OS UNIX rexecd and rshd servers were IPv6-enabled in z/OS V1R4
- TSO/E rsh and rexec clients provide updated versions that can be used in all z/OS environments (batch, TSO, and REXX for example)
- A new z/OS UNIX rsh client that is now IPv6-enabled

16.2.4 IPv6 (AF_INET6 socket) enabled CICS sockets

z/OS V1R5 Communications Server enables IP CICS sockets to support IPv6. This includes the following changes:

- IPv6 enabled Listener in a CICS address space.
  - Includes a new interface to the Listener security exit. The Security/Transaction Exit program allows the user to examine and change certain pieces of data that are passed to the child server program by the Listener.
  - If the Listener gives the accepted socket to the child server program, the child server program must be able to take that socket. If the Listener is defined as an INET6 Listener, the EBADF errno is issued if the child server's TAKESOCKET is AF_INET. If the Listener is defined as an INET Listener, the EBADF errno is issued if the child server's TAKESOCKET is AF_INET6.
  - Can be configured to accept IPv4 or IPv6 clients.
- IPv6 enabled the CICS sockets infrastructure.
- IPv6 enabled CICS sockets libraries to support RFC2553 draft (Basic Socket Extensions for IPv6).
  - CICS C sockets

    To compile a C Socket program that contains calls to CICS TCP/IP, you must change the standard procedure for C Socket compilation provided with CICS. The CICS sample compile procedures can be found in SDFHSAMP. You should also tailor them to the version of CICS and C Compiler you have installed on your system. See *z/OS Communications Server IP CICS Sockets Guide*, SC31-8807 for details.
  - CICS EZASOKET - This API is invoked by calling the EZASOKET or EZACICSO program and performs the same functions as the C language calls. The parameters look different because of the differences in the programming languages. Changes in z/OS V1R5 CS add new support for COBOL language call format, Assembler language call format, and PL/1 language call format.

Not supported for IPv6

The following are not supported with IPv6:

- EZACICAL APIs (Version 2.2.1 - COBOL, PL/I, Assembler Language) will not be enhanced to support IPv6. IBM recommends that socket programs using this older API be migrated to the Sockets Extended API.
- DNS/WLM for IPv6 clients.
- DNS caching; therefore, IBM recommends using DNS Bind 9 Caching.

16.2.5 IPv6 sockets-related API AF_INET6 enablement overview

Figure 16-3 displays a list of all of the programming interfaces that can be used to interface with the TCP/IP services protocol stack provided by z/OS V1R5 Communications Server. All of the APIs, with the exception of the PASCAL API, interface with the LFS layer. The Pascal API is not enhanced for IPv6 support.

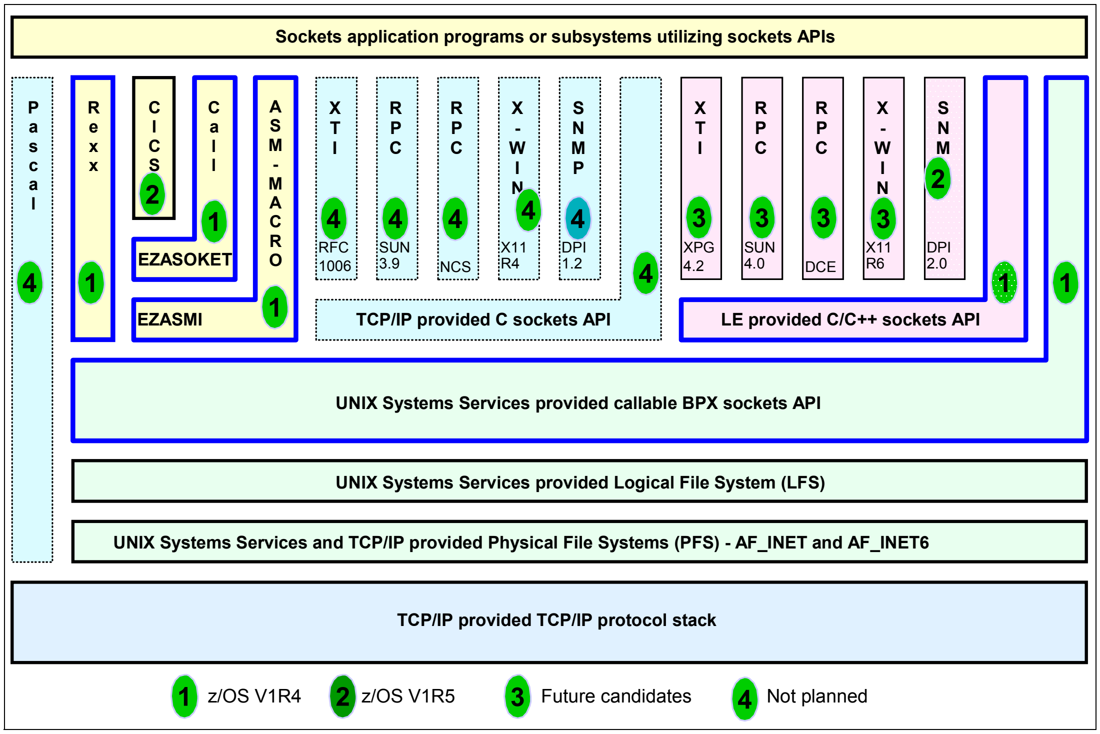

Sockets application programs or subsystems utilizing sockets APIs
Pascal 4
Rexx 1
CICS 2
Call 1
EZASOKET
ASM-MACRO 1
EZASMI
XTI 4 RFC 1006
RPC 4 SUN 3.9
RPC 4 NCS
X-WIN 4 X11 R4
SNMP 4 DPI 1.2
4
TCP/IP provided C sockets API
XTI 3 XPG 4.2
RPC 3 SUN 4.0
RPC 3 DCE
X-WIN 3 X11 R6
SNM 2 DPI 2.0
1
LE provided C/C++ sockets API
1
UNIX Systems Services provided callable BPX sockets API
UNIX Systems Services provided Logical File System (LFS)
UNIX Systems Services and TCP/IP provided Physical File Systems (PFS) - AF_INET and AF_INET6
TCP/IP provided TCP/IP protocol stack
1 z/OS V1R4
2 z/OS V1R5
3 Future candidates
4 Not planned

Figure 16-3 Sockets application programs or subsystems utilizing sockets APIs

16.2.6 IPv6 dynamic routing using OMPROUTE

OMPROUTE implements the IPv6 Routing IP protocol on z/OS V1R5 Communication Server. RIPng is defined by IETF in RFC 2080.

RIPng provides dynamic routing needed for Dynamic VIPA (Virtual IP Addresses) functions on z/OS Communication server. It is an alternative to the static TCP/IP gateway definitions.

IPv6 RIP is an Interior Gateway Protocol (IGP) designed to manage a relatively small network. IPv6 RIP is based on the Bellman-Ford or the distance-vector algorithm. IPv6 RIP has limitations and is not suited for every TCP/IP environment. Before using the IPv6 RIP function in OMPROUTE, read RFC 2080 to decide if RIP can be used to manage the IPv6 routing tables of your network. For more information about RFC 2080, refer to *z/OS Communications Server: IP Configuration Reference,* SC31-8775.

The IP layer routing mechanism is the same for static and dynamic routing and for both versions of IP being used. The routing daemon that manages the IP route table is common for IPv4 (RIP and OSPF) and IPv6 (RIPng). OSPFv3 (Open shortest path first) for IPv6 is targeted for a future release.

16.2.7 Quality of Service (QoS) Policy Agent support for IPv6

Quality of Service (QoS) is a term for the overall service provided to applications and users. Network service providers that need to provide QoS express their goals in terms such as *throughput* and *delay* in Service Level Agreements (SLAs).

The design of the LDAP object tree should be carefully thought out. The Policy Agent uses a variety of mechanisms to search for and retrieve objects from an LDAP server.

Updated commands and applications

Some commands and applications have been updated, as follows:

- **QoS policy schema:** This policy schema is updated. File pagent_at.conf contains a set of LDAP directory attributes for QoS policy objects defined with the LDAP server. It must be included in the LDAP server initial configuration file. There are several schema definition files that must be installed in the proper order. All of these files are located in the /usr/lpp/tcpip/samples directory, although not all files must be installed. The new files are: pagent_qosschema.ldif, pagent_r5idsschema.ldif, and pagent_schema_r5updates.ldif.
- **Network SLAPM2 subagent:** The z/OS CS Network SLAPM2 subagent (nslapm2) allows network administrators to retrieve data and determine if the current set of Network SLAPM2 policy definitions are performing as needed or if adjustments need to be made. The Network SLAPM2 subagent supports the Network Service Level Agreement Performance Monitor (NETWORK-SLAPM2) MIB. Refer to /usr/lpp/tcpip/samples/slapm2.mi2 for more information about the Network SLAPM2 MIB.
- **`pasearch` command**: This command displays detailed information for policy definitions that are managed by the Policy Agent. When migrating, you need to change automation for the **`pasearch`** command display output if necessary, or modify any automation tools or programs that operate on the **`pasearch`** command output. The display output is changed.
- **Policy Agent (PAGENT):** The Policy Agent retrieves policy rules and actions from a policy configuration file and/or from an LDAP server and installs them in the z/OS Communications Server stack, and must be used to download updated policy to the LDAP server.

Policies and services not IPv6 supported

One service that is not IPv6-enabled for QoS policy agent is:

- **Intrusion Detection Services:** You must modify Policy Agent policy rules as needed to include IPv6 addresses. However, IPv6 is not supported for Intrusion Detection Services rules and actions in z/OS V1R5 Communications Server.

16.2.8 SNMP support for IPv6

SNMP (Simple network management protocol) is a set of protocols that describes management data and the protocols for exchanging that data between heterogeneous systems. Protocols include both the description of the management data, defined in the Management Information Base (MIB), and the operations for exchanging or changing that information. The SNMP agent, `osnmp` command, Trap Forwarder daemon, and Distributed Protocol Interface for SNMP subagents are all enhanced to operate over IPv6 networks and handle IPv6-related management data.

- **SNMP agent:** When the SNMP agent is started, it retrieves a local host IP address for itself. If this is an IPv6 address, then SNMPv1 traps are sent with 0.0.0.0 encoded as the source address. This can be prevented by using the **`-A`** option on the **`osnmpd`** command to force the agent to get an IPv4 address when it initializes. Refer to *z/OS Communications Server: IP Configuration Reference*, SC31-8775 for details. The **`pwtokey`** and **`pwchange`** commands are enhanced to accept IPv6 addresses as input.

  Update the SNMPD.CONF or update PW.SRC and SNMPTRAP.DEST to use IPv6 addresses.

- **The DPI® version 2.0:** Distributed Protocol Interface is enabled to use IPv6 connectivity if available. The DPI is an application interface used by the SNMP agent to communicate with subagents.
- **`osnmp` command:** This command is enhanced in the way that it displays MIB objects of type Counter64. They will now be displayed as decimal equivalents of the 64-bit field.
- **Trap Forwarder daemon**: This daemon is enabled for AF_INET6 API.
- **TN3270 Telnet subagent**: Starting in z/OS V1R5 Communications Server, a new SNMP TN3270 Telnet subagent is provided. This subagent provides Telnet transaction data for monitored Telnet connections via the SNMP protocol.
- **TCP/IP subagent**: This enhancement is included in z/OS V1R5 Communication Server. The IPv6 MIB data is supported in version-neutral MIB objects. Version-neutral MIB objects can support both IPv4 and IPv6 processing. The TCP/IP subagent now supports some version-neutral standard MIB data from the following Internet drafts:
  - IP-MIB from draft-ietf-ipv6-rfc2011-update-01.txt
  - TCP-MIB from draft-ietf-ipv6-rfc2012-update-01.txt
  - IP-FORWARD-MIB from draft-ietf-ipv6-rfc2096-update-02.txt
  - The ifAdminStatus MIB object from the IF-MIB (RFC 2233) now reflects the desired state of an interface. This is non version-neutral.

16.2.9 IPv6 support for SMF record type 119

The formats for several of the type 119 SMF records are changed to define additional subsections reflecting new information. Among changed 119 SMF record formats are records collected for Telnet servers and clients, FTP servers and clients, and API activity and stack usage information. New sections in the SMF 119 record format to capture additional IPv6-related data such as interface records and statistics records are:

- **ICMPv6:** Internet Control Message Protocol for the Internet Protocol Version 6 (IPv6) statistics section is added to the TCP/IP statistics record.
- **Stack's IPv6**: Enablement status statistical data is added to the stack initialization and termination event record.
- **IPv6 interface:** Statistics are reported to interface statistics records (including support for multiple IP addresses on an interface).

Non IPv6-specific SMF 119 record enhancements

- **FTP server and client:** Security information is added to the various SMF records for the purpose of identifying the security mechanism and levels for FTP transfers.
- **TCP connection:** The termination record is modified to report Telnet-specific information, such as LU name, application name, and protocol.

The EZASMF77 macro in data set SYS1.MACLIB produces assembler level DSECTs that can be used to map the various new record formats for SMF 119 records.

16.2.10 Netstat IPv6 enhancements

In z/OS V1R5 Communications Server, Netstat is changed in the following way for IPv6 support:

- IPv6-enabled Netstat reports are enhanced to support LONG report format in preparation for future IPv6 support where applicable. The existing stack-wide output-format option (FORMATSHORT/LONG) configured on the IPCONFIG profile statement, or Netstat FORMAT/-M option, can be used to instruct these Netstat reports to produce output according to either the old or new format for the following reports:
  - **`D TCPIP,,NETSTAT,ACCESS,NETWORK`**
  - Netstat CACHINFO/-C
  - IDS/-k
  - VCRT/-V
  - VDPT/-O
  - VIPADCFG/-F
  - VIPADYN/-v
- Non IPv6-specific support adds new Netstat reports for:
  - Support for a hostname filter for connection-type reports
  - Existing IP address filter to BYETINFO report
  - Interface/link filter to DEVLINKS reports
  - Interface statistics sections to DEVLINKS report

16.2.11 IPv6 Enterprise Extender enhancements

The Enterprise Extender (EE) network connection is a simple set of extensions to the existing open high-performance routing (HPR) technology, as shown in Figure 16-4 on page 442. It performs an efficient integration of the HPR frames using UDP/IP packets. To the HPR network, the IP backbone is a logical link. To the IP network, the SNA traffic is UDP datagrams that are routed without any hardware or software changes to the IP backbone. There is no overhead because of protocol transformation or additional transport functions. Integration is performed at the routing layers.

z/OS V1R5 Communications Server enhances Enterprise Extender (EE) in the following areas:

- IPv6 and firewall support

  Support is added to enable the use of IPv6 addressing for Enterprise Extender connections. The capability to specify a hostname, instead of an IP address for Enterprise Extender definitions, is now provided. The remote endpoint receives the partner's hostname and performs name-to-address resolution on the hostname to obtain the correct IP address for Enterprise Extender connection establishment in a network where firewalls and network address translation are used. The ability to exchange hostnames, instead of explicit IP addresses, allows Enterprise Extender nodes to co-exist in a network where network address translation is used.

  > **Important:** Connection network groups are only IPv4 or IPv6, not mixed. An XCA major node can support both IPV4 and IPV6, but each would have a separate virtual routing node (VRN) defined - an IPv4 VRN and an IPv6 VRN.

16.2.12 Enterprise Extender enhancements in z/OS v1R5

z/OS V1R5 Communications Server enhances Enterprise Extender (EE) in the following areas:

- EE dial usability enhancement for Dynamic PUs

  A new type of model PU can be defined for use as the model for dynamic non-connection-network PUs used for Enterprise Extender on the host receiving the dial. This allows the user flexibility in coding certain operands for this type of dynamic PU, which previously used default characteristics.

- Connection network diagnostics enhancement

  A new message has been introduced to inform the operator when a dial fails or a connection INOPs over a virtual routing node (VRN). Prior to this enhancement, retries for a dial failure or an INOP for an existing connection that was routed over a connection network link would repeatedly fail without the operator knowing what virtual routing node was being used.

- Protocols that have IP addresses in datagram flow will encounter problems when NAT is active in a network.

  Architectural changes to the network are necessary since HPR passes IP addresses in the protocol data part of datagrams and could be supported on multiple platforms. Flow control vectors sent in EE protocol will carry the hostname of the EE VIPA to provide connectivity, thus replacing IP addresses. Therefore, resolution of the hostname must be done.

- Multiple VRNs for Enterprise Extender

  z/OS V1R2 CS provided an enhancement to allow Enterprise Extender connection networks to span multiple APPN subnetworks and/or NETID. In z/OS V1R2 CS, you were limited to one local and one global connection network for Enterprise Extender connectivity. z/OS V1R5 Communications Server lifts this restriction.

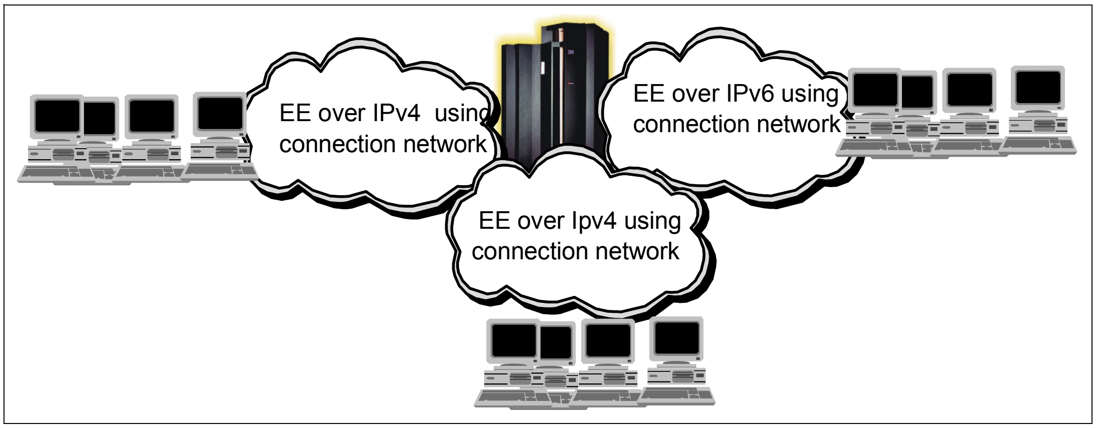
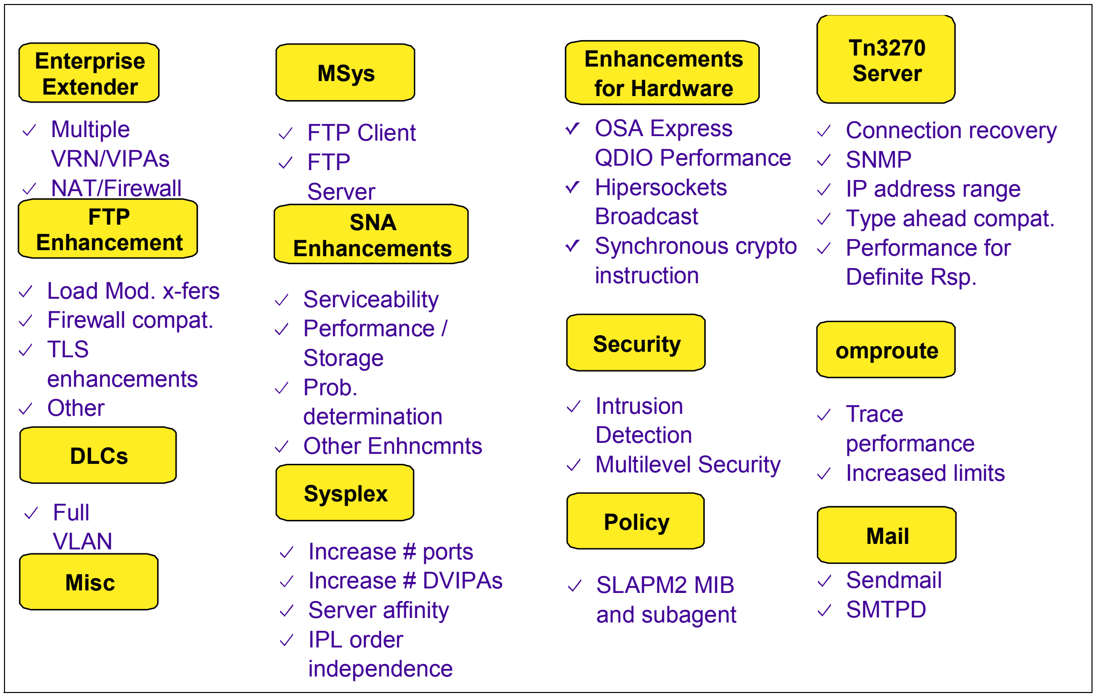

EE over IPv4 using connection network
EE over IPv6 using connection network
EE over Ipv4 using connection network

Figure 16-4 Enterprise Extender (EE) network connections

16.2.13 Non-IPv6 Support in z/OS V1R5 CS

z/OS V1R5 Communication server is a continuation of the effort started in z/OS V1R4 to enable functionality for new IPv6 protocol. However, at the present time some new Communication Server functions, shown in Figure 16-5 on page 443, are developed based on non-IPv6 support.



Enterprise Extender

✓ Multiple VRN/VIPAs
✓ NAT/Firewall

FTP Enhancement

✓ Load Mod. x-fers
✓ Firewall compat.
✓ TLS enhancements
✓ Other

DLCs

✓ Full VLAN

Misc

MSys

✓ FTP Client
✓ FTP Server

SNA Enhancements

✓ Serviceability
✓ Performance / Storage
✓ Prob. determination
✓ Other Enhncmnts

Sysplex

✓ Increase # ports
✓ Increase # DVIPAs
✓ Server affinity
✓ IPL order independence

Enhancements for Hardware

✓ OSA Express QDIO Performance
✓ Hipersockets Broadcast
✓ Synchronous crypto instruction

Security

✓ Intrusion Detection
✓ Multilevel Security

Policy

✓ SLAPM2 MIB and subagent

Tn3270 Server

✓ Connection recovery
✓ SNMP
✓ IP address range
✓ Type ahead compat.
✓ Performance for Definite Rsp.

omproute

✓ Trace performance
✓ Increased limits

Mail

✓ Sendmail
✓ SMTPD

Figure 16-5 z/OS CS V1R5 non-IPv6 contents overview

16.2.14 Functions supported by zSeries 990 hardware

HiperSockets™ was introduced in z/OS V1R2 Communications Server. In z/OS V1R5 Communications Server, the following HiperSockets enhancements, exclusive to the IBM eServer zSeries 990, are available:

- Spanned channels

  Spanning channels is the ability for Internal Coupling Channels and HiperSockets channels to be configured to multiple Channel SubSystems, and be transparently shared by any or all of the configured LPARs without regard to the Logical Channel SubSystem to which the LPAR is configured.

  With the introduction of a new Channel SubSystem, transparent sharing of Internal Coupling Channels (ICs) and HiperSockets is possible. The Multiple Image Facility (MIF) allows sharing of channel resources across LPARs. ICs and HiperSockets can be configured as MIF-spanning channels. This support is applicable to Internal Coupling Channels (ICP CHPID type) for Parallel Sysplex and to HiperSockets (IQD CHPID type).

- Increased number of HiperSockets CHPIDs (iQDIO Internal LANs)

  The number of Internal LANs that can be configured is increased from 4 to 16. When HiperSockets was introduced, up to 4 internal Local Area Networks (LANs) could be configured. An IQD CHPID represents one Internal LAN. That number is now increased to up to 16 internal LANs (IQD CHPIDs).

- Increased number of supported TCP/IP stacks

  The number of supported TCP/IP stacks is increased from 1024 to 4096. Because each TCP/IP stack requires one communication queue, this means 4096 TCP/IP stacks are now supported (instead of 1024 TCP/IP stacks). A HiperSockets channel must be spanned in order to communicate between LPARs in different LCSSs.

- OSA-Express QDIO performance improvements

  z/OS V1R5 Communications Server extends Virtual LAN support by allowing you to assign a Virtual LAN identifier (VLAN ID) to an OSA-Express link or interface. The VLAN ID is configured in TCPIP profile to an OSA-Express link (for IPv4) or interface (IPv6) configuration statement. This allows all packets using an OSA-Express to carry a VLAN ID, and thus segregate traffic into different Virtual LANs without needing multiple real LANs or creating new subnetworks.

  z/OS V1R5 Communications Server also introduces new configuration parameters to provide more granular control over fixed storage usage for OSA-Express QDIO and HiperSockets interfaces, as well as some control over the inbound performance of OSA-Express QDIO interfaces.

  z/OS V1R5 Communications Server provides improved performance for IPv4 by offloading checksum processing to an OSA-Express in QDIO mode.

- Broadcast support for Hipersockets Interface (IPv4)

  HiperSockets broadcast support (which is very similar to OSA-Express QDIO broadcast support) is provided for user-configured HiperSockets (iQDIO) MPCIPA devices.

- zSeries synchronous crypto instructions and new PCIX crypto coprocessor

  z/OS V1R5 Communications Server supports new IBM CP assist cryptographic instructions for IPSec. The IBM eServer zSeries 990 provides IBM CP assist that improves symmetric encryption/decryption performance, as well as SHA1 performance. These instructions are synchronous clearkey.

  Some zSeries 990s support a new cryptographic coprocessor, called the PCIX Cryptographic Coprocessor (PCIXCC). However, as discussed in the next section, IPSec attempts encryption/decryption first by using the new ICSF crypto assist instructions; and if crypto assist is not present, IPSec attempts encryption/decryption using PCIXCC. If PCIXCC is not available or fails, IPSec performs encryption/decryption by using software.

  New instructions for cryptography improves performance for DES, TDES, and SHA-1 services in IPSEC (IP Security). The support provides the z/OS Integrated Cryptographic Service Facility (ICSF). IPSEC and VTAM Session-level encryption are affected. VTAM Session-level encryption requires the coprocessor.

Table 16-1 PCIX crypto support of the z990 model levels

Encryption method	z990 GA1 (6/13/03)	z990 GA2 (10/31/03)
Crypto instructions	IPSEC uses if available	IPSEC uses if available
Coprocessor	Not available	Used if available
Software	Used if crypto instructions not available or fails	Used if crypto instructions or coprocessor not available or where application fails

16.2.15 z/OS V1R5 sysplex enhancements

In Figure 16-6, VIPABACKUP is a statement in the VIPADYNAMIC block to designate Dynamic VIPAs (DVIPAs) to provide automatic backup when the owning stack fails. It was introduced in Communications Server for OS/390 V2R8 for cases of outages of the routing TCP/IP so that existing connections to other TCP/IPs in the sysplex are not disrupted.

Note: A Dynamic VIPA (DVIPA) can move to other TCP/IP stack members in a sysplex or it can be activated by an application program or by a supplied utility. Dynamic VIPAs are used to implement Sysplex Distributor.

In z/OS V1R5 Communications Server, VIPABACKUP is enhanced so that a DVIPA can be activated on a backup TCP/IP before it is activated elsewhere in the sysplex with the VIPADEFINE statement, as shown in Figure 16-6. The VIPABACKUP statement has new statements that allow this to occur. A new MOVEABLE parameter, subnet mask definition, and an optional SERVICEMGR parameter can be coded on TCP/IP initialization or during `VARY OBEY` processing. The IPCS command to display the configuration is enhanced for VIPABACKUP to show the new parameters, if specified.

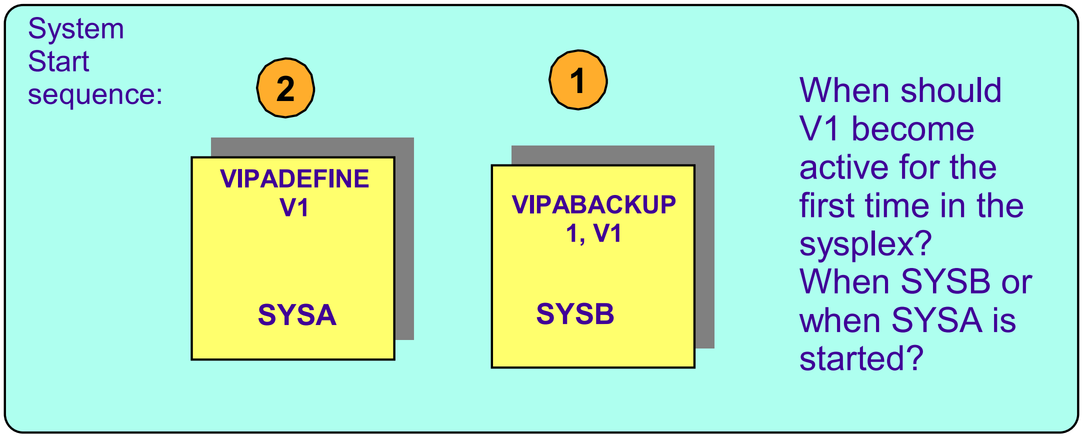

System Start sequence:
2
1
VIPADEFINE V1
SYSA
VIPABACKUP 1, V1
SYSB
When should V1 become active for the first time in the sysplex? When SYSB or when SYSA is started?

Figure 16-6 VIPABackup acts like VIPADEFINE until the VIPADEFINE is done

Sysplex Distributor round-robin distribution

z/OS V1R5 Communications Server introduces a new DISTMethod ROUNDROBIN parameter on the VIPADISTRIBUTE statement that may be used to assign incoming connections for the Distributed DVIPA among available server instances in a round-robin method.

> **Note:** This enables round-robin distribution of incoming connection requests for a Distributed DVIPA, regardless of the setting of IPCONFIG SYSPLEXROUTING and WLM or policy information.

The enhancements and changes to existing support made in z/OS V1R5 are as follows:

- VIPADefine or VIPABackup

  Either could be started first; the IPL order is not significant, as shown in Figure 16-6.

- Maximum port number defined for distributed Dynamic VIPA

  The VIPADISTRIBUTE statement increases the ports from 4 to 64. Applications that are candidates for workload distribution with Sysplex Distributor and that listen on more than 64 ports, are able in z/OS V1R5 Communications Server to use a single distributed Dynamic VIPA by exploiting the ability to dynamically assign Sysplex Distributor ports. New distributed DVIPAs configured without a PORT parameter on the VIPADISTRIBUTE statement determine where to distribute work based on where there are applications with listening sockets bound to the distributed DVIPA.

- DVIPA limit increase

  In z/OS V1R5 Communications Server, the limit for Dynamic Virtual IP Addresses (DVIPAs) is increased from 256 to 1024. As part of this change, some TCP/IP control blocks associated with DVIPAs were moved from common storage to TCP/IP private storage. Expanding the limit from 256 to 1024 allows you more flexibility in defining your network configuration.

- Allow VIPABACKUP before VIPADEFINE

  On the rare occasion that a disaster occurs, it might be necessary to IPL all of the systems in a sysplex. Assuming that many Dynamic VIPAs are in use and the VIPADEFINE statements are spread across the available TCP/IP stacks in the sysplex, most of the Dynamic VIPAs have a lengthy wait before the owning operating system, TCP/IP, and application are started and fully operational. The intent of Dynamic VIPAs defined with VIPADEFINE and VIPABACKUP is to move the Dynamic VIPAs under a functioning application as soon as possible. Therefore, optional parameters have been added to the VIPABACKUP statement to allow the Dynamic VIPA to be activated when the VIPABACKUP is processed at TCP/IP initialization or **VARY TCPIP,,OBEYFILE** processing. This makes the IPL order to be independent.

- Timed affinity feature of Sysplex Distributor

  The TIMEDAFFinity parameter on the VIPADISTRIBUTE statement indicates to Sysplex Distributor that connections to a particular distributed DVIPA need to take into account the client origin. Connections from the same client, as identified by IP address, need to be routed to the same server instance, even when multiple server instances are hosted by a single target stack.

- Significant performance improvement for short-lived connections

  The use of the SYSPLEXPORTS function introduced in z/OS V1R4 Communications Server caused performance degradation for short-lived connections. z/OS V1R5 Communications Server provides significant performance improvement for short-lived connections by having the stack obtain a group of ephemeral ports from the Coupling Facility (CF) and managing port allocation instead of calling the CF for each bind().

  This performance improvement only applies to applications that bind to port 0. Applications that bind to a specific (> 1023) port will work the same as they did in z/OS V1R4 Communications Server.

- Round-robin method may be used to assign incoming connections

  z/OS V1R5 Communications Server introduces a new DISTMethod ROUNDROBIN parameter on the VIPADISTRIBUTE statement that may be used to assign incoming connections for the Distributed DVIPA among available server instances in a round-robin method.

Note: The routing stack and all backup routing stacks for a Distributed DVIPA should be at z/OS V1R4 with the enabling PTF or later for this function to work properly in scenarios of takeover.

16.2.16 Client/Server affinity in a Sysplex Distributor environment

The Timed Affinity feature of Sysplex Distributor allows affinities to be established between a specific client (identified by its IP address) and a particular instance of a server application for which work is being balanced with Sysplex Distributor, using a Distributed Dynamic VIPA. This feature ensures that a client that establishes a relationship with a server will be directed to that particular server for subsequent connections.

Prior to z/OS V1R5 CS, Sysplex Distributor normally distributed each connection request, as it arrives to one of the candidate server instances, based on available capacity and policies in effect when the connection request arrives, as shown in Figure 16-7 on page 447. In this example, after a break in the connection, the attempt to reconnect allowed the reconnect to go to a different server.

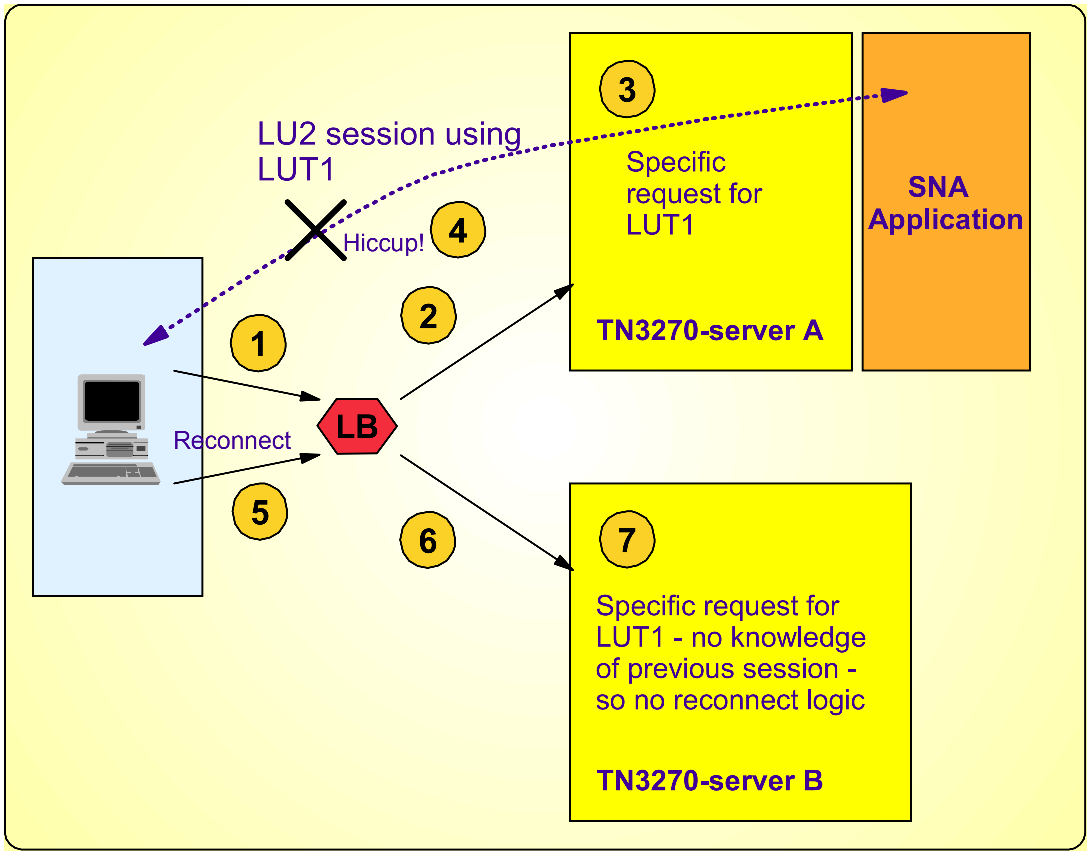

3
LU2 session using LUT1
Specific request for LUT1
SNA Application
Hiccup!
4
2
1
TN3270-server A
LB
Reconnect
5
6
7
Specific request for LUT1 - no knowledge of previous session - so no reconnect logic
TN3270-server B

Figure 16-7 On a reconnect, a different server is assigned

Timed affinities in z/OS V1R5 CS

For some applications, it is necessary to establish an affinity between a client and a particular server instance that needs to span multiple connections. TN3270 printer sessions, for example, as shown in Figure 16-8 on page 448, are based on a connection request from a TN3270 client, and that printer session connection request needs to be routed to the same TN3270 server that is serving the LU2 session. In Figure 16-8, before z/OS V1R5 CS, you can see that the connection is random and can fail.

Similarly, Web-based applications, such as shopping carts, might need to have all connections from a particular client come to the server instance that has the contents of the shopping cart stored as session state, and therefore the necessity for a change is made in z/OS V1R5 CS for a new TIMEDAFFinity parameter on the VIPADISTRIBUTE statement.

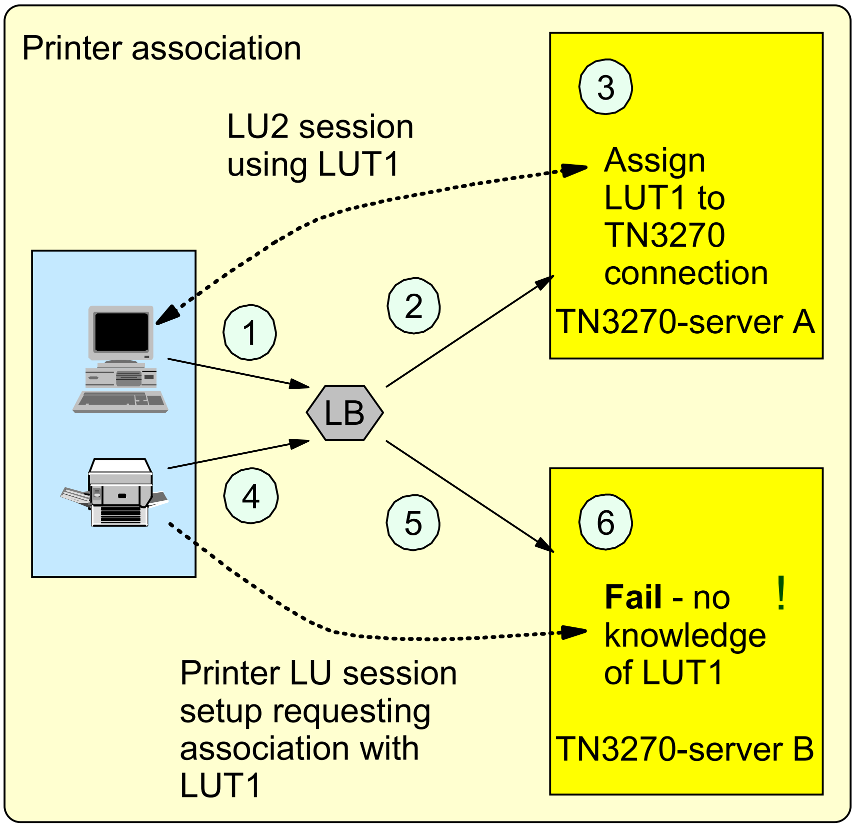

Printer association
LU2 session using LUT1
3
Assign LUT1 to TN3270 connection
TN3270-server A
2
1
LB
4
5
6
Fail - no knowledge of LUT1
!
TN3270-server B
Printer LU session setup requesting association with LUT1

Figure 16-8 Printer LU session requiring the same server, but fails

New parameter TIMEDAFFinity

The TIMEDAFFinity parameter on the VIPADISTRIBUTE statement indicates to Sysplex Distributor that connections to a particular distributed DVIPA need to take into account the client origin. Connections from the same client, as identified by IP address, need to be routed to the same server instance, even when multiple server instances are hosted by a single target stack.

The timed affinity is configurable on the VIPADISTRIBUTE statement in seconds, as follows:

```
TIMEDAFFINITY nnn
```

When specifying a value for TIMEDAFFinity on the VIPADISTRIBUTE statement, the first connection from a particular client is routed as normal to a target stack and listening application. At that time, both the Sysplex Distributor routing stack and the target stack establish an affinity to govern subsequent connection requests from the same client. This affinity maintains a connection count, initially one. As subsequent connection requests for the same distributed DVIPA and port come in, they are routed to the same server instance and the affinity connection count is incremented. As affinity-based connections, including the first one, are closed, the connection count is decremented.

16.2.17 TN3270 Server enhancements in z/OS V1R5 CS

In z/OS V1R5 Communications Server, monitoring of the total Telnet transaction can be done within the Telnet Server. Monitoring is requested by using the new TN3270 Server Profile statements, MONITORGROUP and MONITORMAP. The transaction data can then be retrieved by using either or the following:

- **D TCPIP,,TELNET,CONN,CONN=connid** detail command
- SNMP

  The SNMP support is provided by a new SNMP TN3270 Telnet subagent. The SNMP transaction data is defined in a new Enterprise-specific TN3270 MIB. A sample of this MIB is installed in the HFS as file /usr/lpp/tcpip/samples/mvstn3270.mi2. Activation of the SNMP TN3270 Telnet subagent is controlled by the new TNSACONFIG Profile statement. See *z/OS Communications Server: IP Configuration Reference*, SC31-8776 for more detailed information on all the new TN3270 Telnet Server Profile statements.

  Note: The SNMP Telnet transaction data will only be available on TCP/IP stacks where the TN3270 Telnet server is active and where there are Telnet connections being monitored for transaction data due to MONITORGROUP and MONITORMAP Profile statements.

- Improve TN3270/e connection recovery

  THe end user need not know LUNAME to recover a session.TN3270 IP address range configuration

In z/OS V1R5 Communications Server, IP ranges can be specified in the IPGROUP or DESTIPGROUP statements in addition to exact IP addresses and IP subnets. Only the right-most portion of the IP address can be part of the range. For IPv4 addresses, that is the last octet, and for IPv6 addresses that is the last two hexadecimal bytes.

UNLOCKKEYBOARD statement in PROFILE.TCPIP

The Telnet Server allows customized control of the keyboard unlock function in conjunction with SNA read commands received from the host application. This control is implemented by a new TCP/IP profile parameter called UNLOCKKEYBOARD. Use of this parameter enables the system programmer to dictate whether a 3270 unlock keyboard datastream sequence is sent to a TN3270 client before or after a read command is forwarded from the host application. This also provides control over whether or not a clear screen and unlock keyboard sequence are sent to TN3270 clients when Telnet receives the application BIND. This will more fully implement the TN3270E functional extensions to RFC2355 by employing use of the Keyboard Restore Indicator (KRI) in the TN3270E header. There are no external changes associated with this functional extension to RFC 2355.

IPV6_INTERFACE statement

When specifying the Delay_Acks=value parameter on the IPV6_INTERFACE statement, the value specified is added to the routing table for routes that take this interface. Specifying YES delays transmission of acknowledgments when a packet is received with the PUSH bit on in the TCP header. Specifying NO results in acknowledgments being returned immediately. Valid values are YES and NO. The default value is YES.

16.2.18 QoS Policy in z/OS V1R5 CS

In z/OS V1R5 CS, the Policy Agent is changed to include a new performance collection function. Performance collection allows policy performance data to be collected and maintained for retrieval by external performance monitor applications, and also provides optional logging of the performance data to a performance log file. A Policy API (PAPI) interface is added to allow external user applications to access policy data, as shown in Figure 16-9.

Performance collection in z/OS V1R5 CS is another aspect of policy performance. It provides more relevant QoS performance data than the SLA subagent, allows this data to be collected and monitored in near real time by user applications through the Policy API (PAPI), and provides optional logging of the data to a performance log file for offline monitoring.

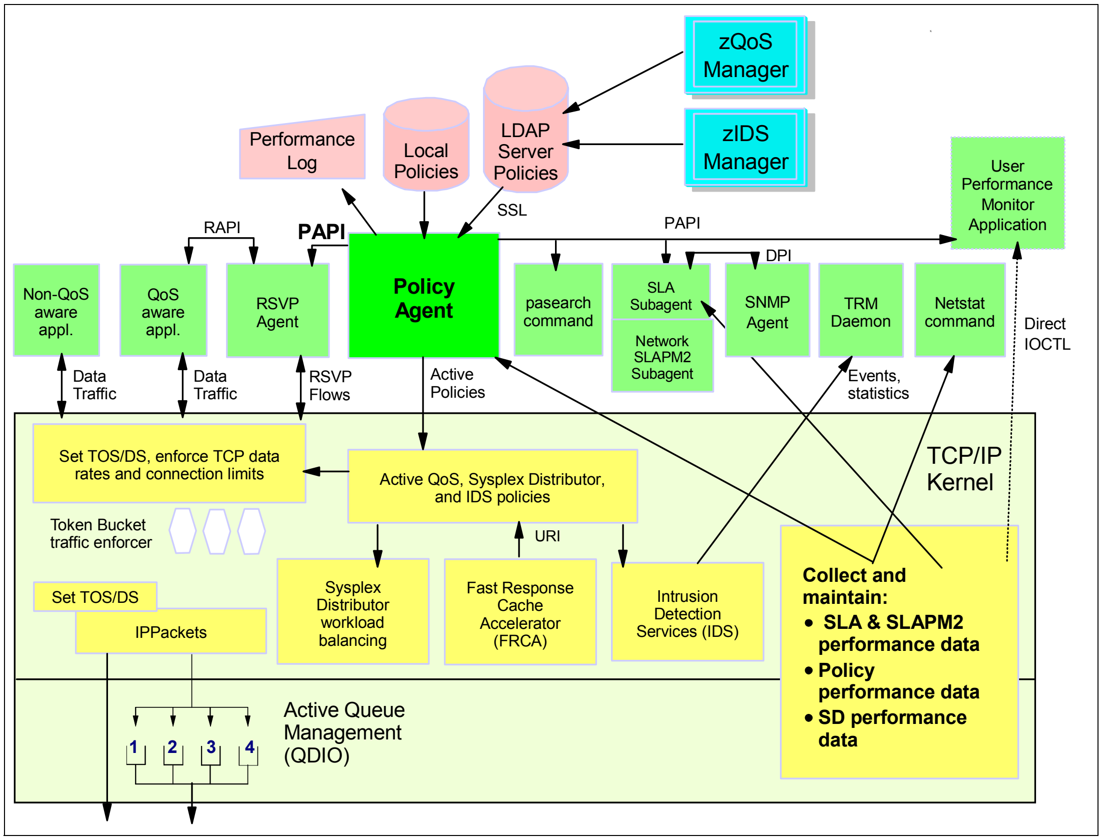


Figure 16-9 Changes made to Policy Agent in z/OS V1R5

Policy Agent enhancements

In z/OS V1R5, the following enhancements have been made to the Policy Agent for the performance collection function, as shown in Figure 16-9:

- QoS-aware (Integrated Services) applications and non-QoS-aware (Differentiated Services) applications can both utilize QoS support in the stack.
- QoS-aware applications use the RSVP API (RAPI) to communicate with the RSVP Agent.
- Non-QoS-aware applications can pass data classification information dynamically.
- RSVP Agent communicates with other RSVP Agents on routers/hosts.
- RSVP Agent is supported as an end system only, not as a router.
- Policy Agent reads policies from local files and/or an LDAP server and installs them into the Policy Table in the stack.
- The **pasearch** command displays active and inactive policies.
- The **Netstat** command displays active QoS policy statistics.
- Scope=DataTraffic QoS policies are used by the stack to provide Differentiated Services QoS to data traffic, including enforcing TCP data rate, setting TOS (DS) byte, enforcing connection limits, token bucket policing, and so forth.

- Scope=RSVP QoS policies are read by the RSVP Agent using the Policy API (PAPI), and applied to RSVP data flows.
- IDS policies are used by the stack to provide reporting options for attacks and scans, and to enforce Traffic Regulation.
- Policy Agent and stack interact with Sysplex Distributor function to provide workload balancing based on network performance.
- Interfaces use appropriate QoS technology: RSVP reservations can be made on ATM links; Queued Direct I/O (QDIO) can provide priority queuing based on TOS (DS) settings and VLAN priority tagging for directly connected LANs.
- Stack maintains performance statistics for data traffic and associated policies.
- SLA Performance Monitor MIB (SLAPM2-MIB) subagent provides policy monitoring, as shown in Figure 16-10 on page 452.
- SNMP traps can be generated for various performance “out-of-bounds” conditions and for significant events such as policy deletion.

16.2.19 QoS Policy enhancements - new SLAPM2 MIB

The existing SLAPM-MIB (Service Level Agreement Performance Monitoring) is replaced by SLAPM2-MIB and a new subagent, nslapm2. The original SLAPM-MIB introduced in CS390 V2R8 as experimental RFC2758 and the slapm subagent are still supported. The new MIB and subagent provide many improvements, as follows:

- SlapmSubcomponentTable deprecated (TCP Connection entries).
- Changes made to improve monitoring of policy rules.
- Monitor checks if it is meeting the desired level of service.

PAPI, the new Policy Agent API, was created to allow applications access to near real-time policy performance data.

Performance collection in z/OS V1R5 CS is another aspect of policy performance. It provides more relevant QoS performance data than the SLA subagent, allows this data to be collected and monitored in near real time by user applications through the PAPI, and provides optional logging of the data to a performance log file for offline monitoring.

The Integrated WLM/QoS Performance Monitor API is also being used by the new Network SLAPM2 Subagent (nslapm2). This subagent provides policy performance monitoring using the NETWORK-SLAPM2-MIB. This new subagent is the replacement for the SNMP SLA subagent.

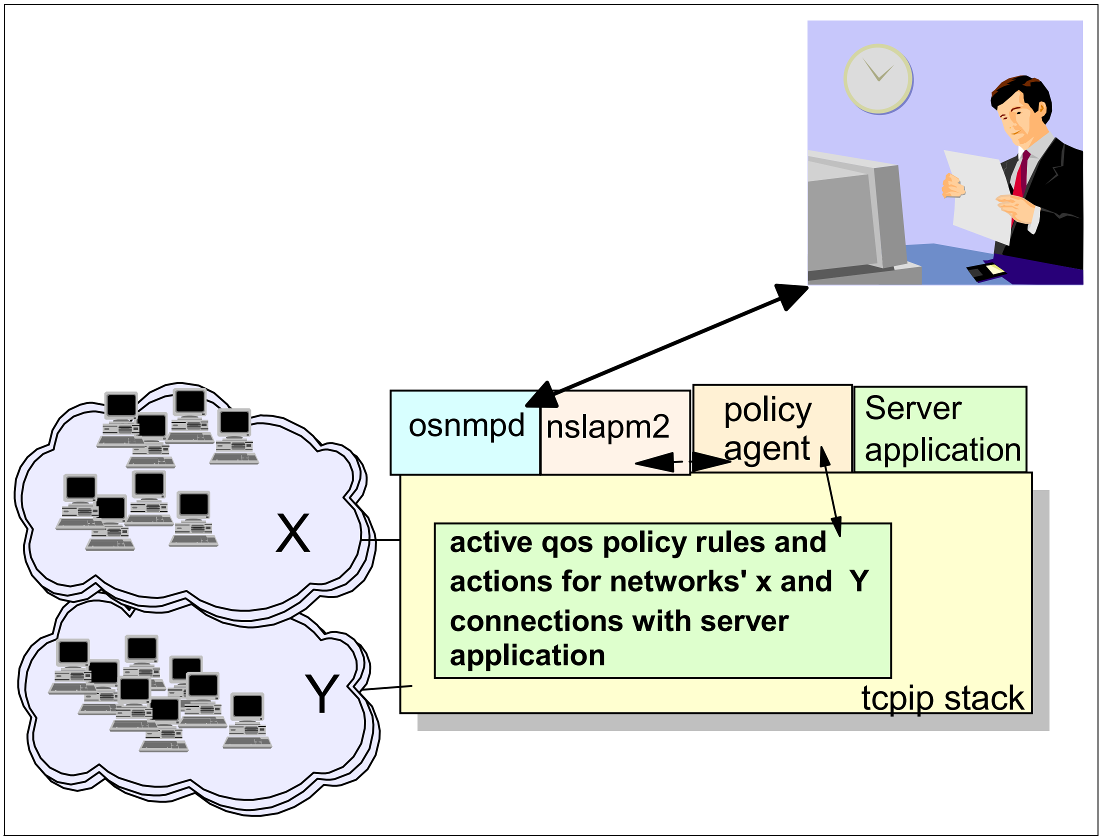

osnmpd
nslapm2
policy agent
Server application
X
Y
active qos policy rules and actions for networks' x and Y connections with server application
tcpip stack

Figure 16-10 QoS Policy enhancements with the new SLAPM2-MIB

16.2.20 FTP enhancements in z/OS V1R5 CS

The following enhancements for the FTP functions are new in z/OS V1R5 CS:

- Autoconfigure target library for FTP load module transfer
- Define FTP ephemeral port range for firewall compatibility
- FTP TLS support enhancements
- FTP serviceability improvements
- Enforce nonzero error return code in FTP
- Allow the FTP server load module to run above the 16M line
- Display status of FTPKEEPALIVE timer
- FTP SERVAUTH Port of Entry support

Autoconfigure target library for FTP load module transfer

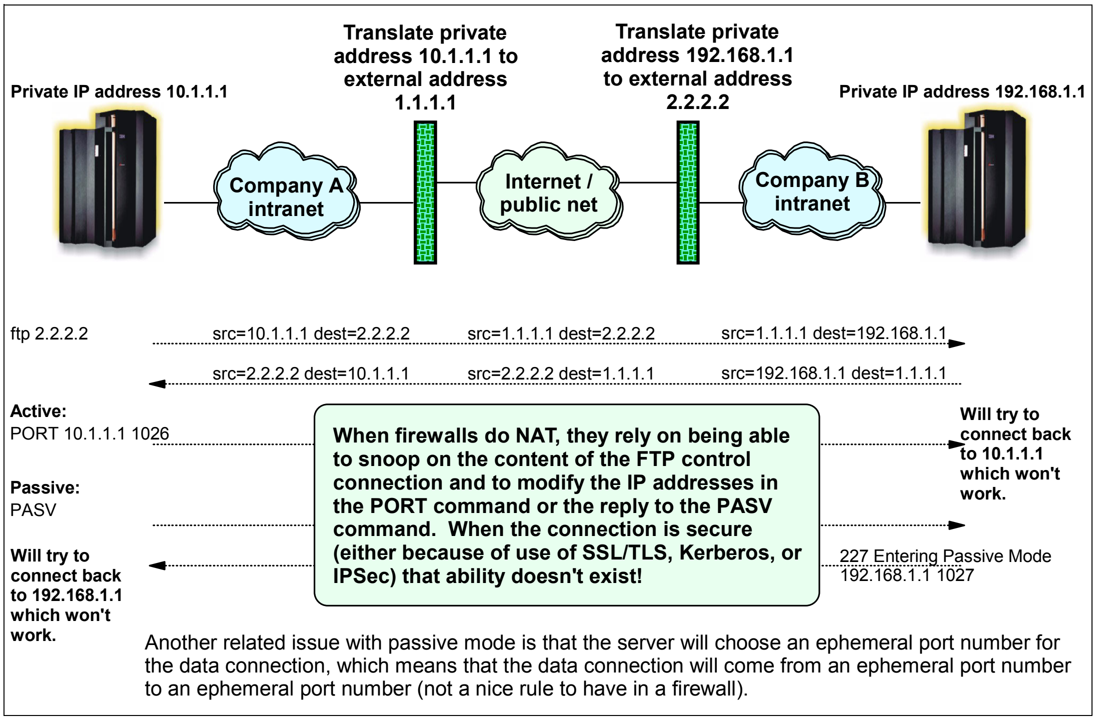
You can specify to FTP whether to create MVS directories as partitioned data sets (PDSs) or as partitioned data sets extended (PDSEs). Prior to this release, you could create an MVS PDS with FTP, but not a PDSE. When transferring load modules, you must allocate an MVS directory on the target host before the transfer. The target MVS directory characteristics must be compatible with the source MVS directory for the transfer to succeed.

Define FTP ephemeral port range for firewall compatibility

This enhancement adds a configuration option to direct the FTP client to use the EPSV command instead of PORT or PASV commands to establish the data connection for an IPv4

FTP session. FTP allows the operating system to select port numbers used for listening data sockets. Some firewall implementations can be configured to restrict the range of port numbers allowed to applications such as FTP. This enhancement allows you to configure the FTP server to select listening data ports from a specific range of values so you can coordinate the FTP server with your firewall configuration. You can configure this range with the PASSIVEDATAPORTS statement in FTP.DATA.

Figure 16-11 shows the environment before z/OS V1R5. Figure 16-12 on page 454 shows the new environment with the ability to define FTP ephemeral port ranges for firewall compatibility.


Private IP address 10.1.1.1
Translate private address 10.1.1.1 to external address 1.1.1.1
Translate private address 192.168.1.1 to external address 2.2.2.2
Private IP address 192.168.1.1
Company A intranet
Internet / public net
Company B intranet
ftp 2.2.2.2
src=10.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=192.168.1.1
src=2.2.2.2 dest=10.1.1.1
src=2.2.2.2 dest=1.1.1.1
src=192.168.1.1 dest=1.1.1.1
Active:
PORT 10.1.1.1 1026
Will try to connect back to 10.1.1.1 which won't work.
Passive:
PASV
Will try to connect back to 192.168.1.1 which won't work.
227 Entering Passive Mode 192.168.1.1 1027

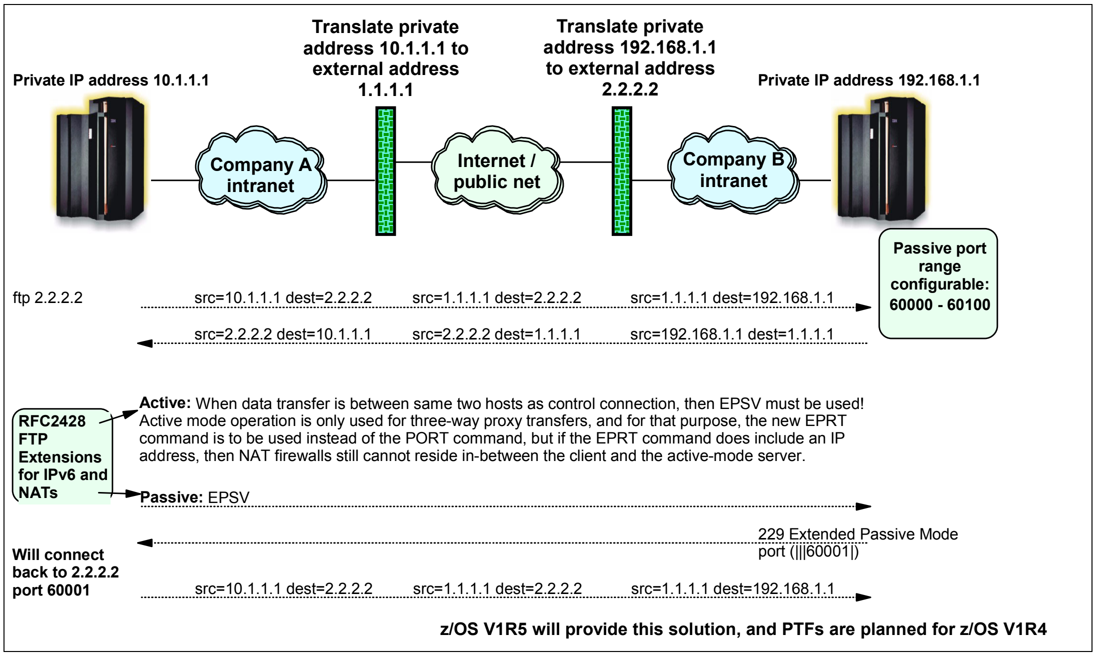
When firewalls do NAT, they rely on being able to snoop on the content of the FTP control connection and to modify the IP addresses in the PORT command or the reply to the PASV command. When the connection is secure (either because of use of SSL/TLS, Kerberos, or IPSec) that ability doesn't exist!

Another related issue with passive mode is that the server will choose an ephemeral port number for the data connection, which means that the data connection will come from an ephemeral port number to an ephemeral port number (not a nice rule to have in a firewall).

Figure 16-11 Secure FTP and Network Address Translation (NAT) firewalls today

In Figure 16-12, the FTP server must support the `EPSV` command for IPv4 sessions. The z/OS FTP server supports RFC2428 (that is, EPSV and EPRT commands) for both IPv4 and IPv6 sessions.

> **Note:** Some servers support the `EPSV` command, but do not reply in the format described in RFC2428. The z/OS FTP client does not support such servers. If the FTP server reply to EPSV does not conform to RFC2428, the client will react as if the server had rejected the EPSV command, and will stop using `EPSV` during the session.


Translate private address 10.1.1.1 to external address 1.1.1.1
Translate private address 192.168.1.1 to external address 2.2.2.2
Private IP address 10.1.1.1
Private IP address 192.168.1.1
Company A intranet
Internet / public net
Company B intranet
Passive port range configurable: 60000 - 60100
ftp 2.2.2.2
src=10.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=192.168.1.1
src=2.2.2.2 dest=10.1.1.1
src=2.2.2.2 dest=1.1.1.1
src=192.168.1.1 dest=1.1.1.1
RFC2428 FTP Extensions for IPv6 and NATs
Passive: EPSV
229 Extended Passive Mode port (|||60001|)
Will connect back to 2.2.2.2 port 60001
src=10.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=192.168.1.1

Active: When data transfer is between same two hosts as control connection, then EPSV must be used! Active mode operation is only used for three-way proxy transfers, and for that purpose, the new EPRT command is to be used instead of the PORT command, but if the EPRT command does include an IP address, then NAT firewalls still cannot reside in-between the client and the active-mode server.

z/OS V1R5 will provide this solution, and PTFs are planned for z/OS V1R4

Figure 16-12 Secure FTP, NAT and filtering firewalls with z/OS V1R5 CS

FTP TLS support enhancements

The FTP server can be configured to allow a user to log in without specifying a password. The server uses the TLS authenticated X.590 certificate provided by the FTP client to perform this login. This support allows you to take advantage of using a certificate instead of a password to complete the login procedure.

Improve FTP serviceability

This enhancement improves diagnosis of failures in the FTP server and client. The enhancements include the following areas:

- Client error logging to the system log is provided.
- Client error codes are extended to further describe failures in the client.
- Dynamic allocation failure reporting is enhanced to ensure all failures record needed information. This includes S99ERROR, S99INFO, and S99ERSN.
- All Language Environment (LE) and UNIX System Services (USS) failure reporting, including errnojr (referred to as errno2), is provided.
- Message EZA2589E text is changed to include the failing operation.

Enforce nonzero error return code in FTP

The new LOGCLIENTERR and updated CLIENTERRCODES statements direct the FTP client to provide enhanced diagnostic information when the client detects a failure, specifically, as follows:

- **LOGCLIENTERR:** This generates a message on the system log and the batch job log (or returns it to the user in an interactive environment) with complete information about the command code, reply code, and computed return code related to the failure.
- **CLIENTERRCODES**: A new option, EXTENDED, allows an append of the code of the failing subcommand to the client error code for the batch or interactive return code. Several client error codes are new and all the possible client error codes are set more consistently and reliably.

Allow the FTP server load module to run above the 16M line

The FTP server load module EZAFTPLS (alias ftpdns) is linkedited with RMODE=ANY so that it can be loaded above the 16M line. This is an enhancement, because below the line space is a limited resource. This enhancement does not require any action to implement. The linkedit occurs with RMODE=ANY. However, if you have previously coded your security exits to have the dependency of working below the line, you should examine the exits and modify them appropriately.

Display status of FTPKEEPALIVE timer

The FTP client `LOCSTAT` subcommand displayed all client timers except the FTPKEEPALIVE timer. The following enhancements are made:

- `LOCSTAT` was enhanced to also display the FTPKEEPALIVE timer.
- The `FTP STAT` subcommand displays the value of the server's FTPKEEPALIVE timer.
- FTP issues message EZYFT47I for every statement coded in FTP.DATA that it ignores. A new configuration statement, SUPPRESSIGNOREWARNINGS, can be coded in either the FTP client's or FTP server's FTP.DATA to suppress message EZYFT47I.

FTP SERVAUTH Port of Entry support

The FTP Daemon uses NETACCESS profiles in the SERVAUTH class for Port of Entry authorization for IPv6 clients. The FTP Daemon may optionally be migrated to use NETACCESS profiles instead of profiles in the TERMINAL class for Port of Entry authorization for IPv4 clients.

> **Note:** The FTP Port of Entry authorization support, using NETACCESS SERVAUTH profiles, requires a NETACCESS statement configured in the TCPIP PROFILE and security server support for the SERVAUTH= parameter on SAF macro RACROUTE REQUEST=VERIFY. The NETACCESS statement includes support for IPv6. RACF, in z/OS V1R5, provides support for the SERVAUTH parameter on the RACROUTE REQUEST=VERIFY macro.

16.2.21 Intrusion Detection Services enhancements in z/OS V1R5 CS

The Intrusion Detection Services (IDS) support is enhanced to include interface flood detection as part of its ATTACK FLOOD support. This support identifies a potential interface flood condition and an installation can take action in a timely manner.

Physical Interface Flood Detection detects packet floods consisting of ICMP, UDP and TCP, and unsupported protocols. This raises an IDS event when a specified percentage of traffic received on a physical interface is being discarded. A minimum discard count specifies the minimum number of discards that must occur on the interface in a one minute interval before a flood condition is raised. This is used to avoid false flood detections when the inbound traffic volume is low.

LDAP policy changes

New IDS actions, ibm-idsIfcFloodMinDiscard and ibm-idsIfcFloodPercentage, can be defined in the LDAP policy to allow you to modify the minimum number of discards and the percentage of discards that identify an interface flood. Refer to z*/OS Communications Server: IP Configuration Guide*, SC31-8775 and *z/OS Communications Server: IP Configuration Reference*, SC31-8776 for additional information.

IDS policy changes

Interface Flood Detection is activated if the IDS policy turns on Flood detection (which currently only detects SYN floods). The source MAC address identification using OSA Express in QDIO mode requires a microcode update and is only available on the z990 processor.

Messages and reports

New syslogd messages related to interface flood are added as a result of these enhancements.

The trmdstat flood summary and detail reports are updated to include the interface flood information. A new flood statistics report has been created to display the flood statistics data collected.

The Netstat IDS/-k report has been updated to include the interface flood information.

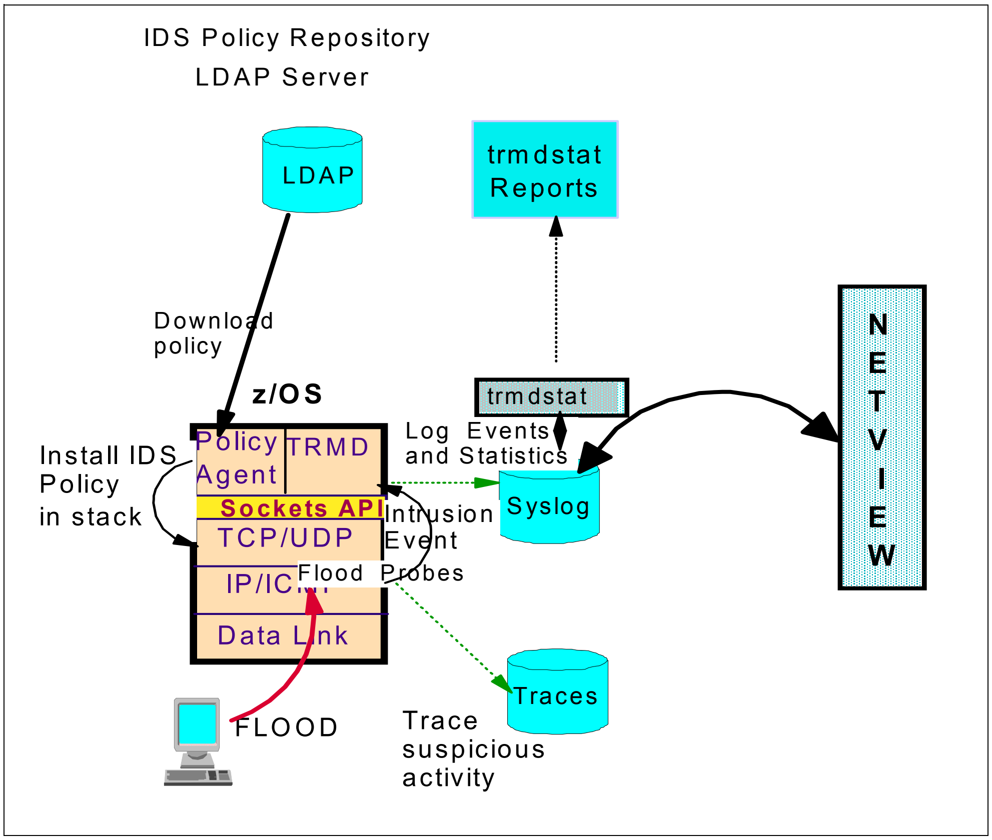


Figure 16-13 Overview of the IDS support and flood detection

16.2.22 Multilevel security overview

In z/OS V1R5 Communications Server, TCP/IP provides support for the z/OS multilevel security (MLS) environment. This environment is intended for government and commercial customers who require advanced Mandatory Access Control (MAC) security features based on security labels (SECLABELs).

This support, as illustrated in Figure 16-14 on page 457, prevents declassification of information. All users and data associated with a SECLABEL via RACF keeps information from being written to a user or resource with a lower SECLABEL. Also, it prevents information from being read from a user or resource with a higher SECLABEL.

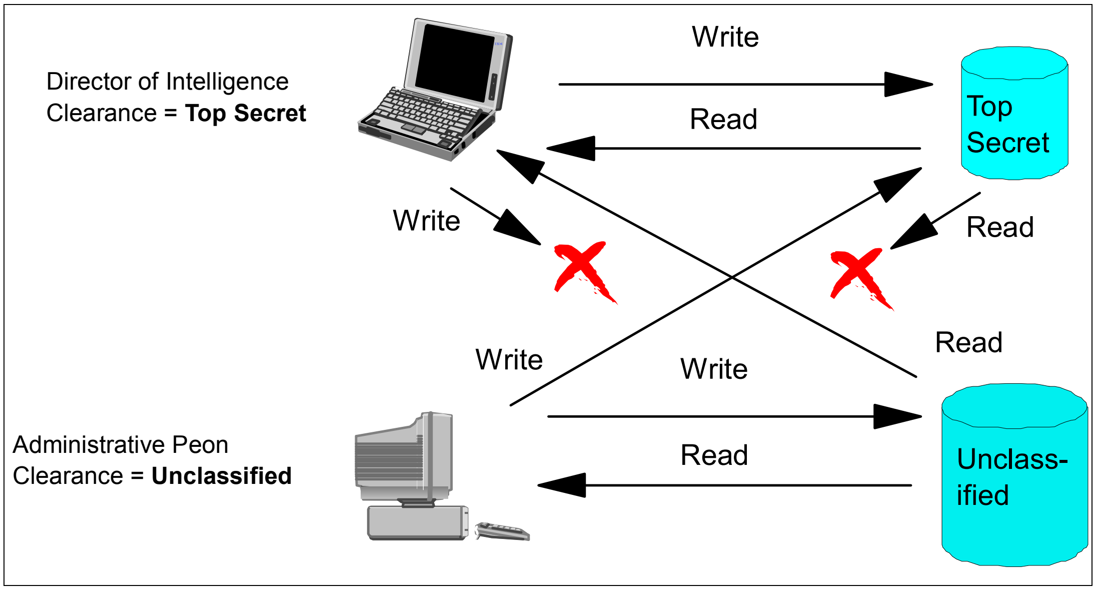


Figure 16-14 SECLABEL protection for reading and writing information

Figure 16-15 illustrates that TCP/IP resources are inherently both READ and WRITE. Therefore, TCP/IP communications in an MLS environment require the SECLABELs to be equivalent, so access is denied.

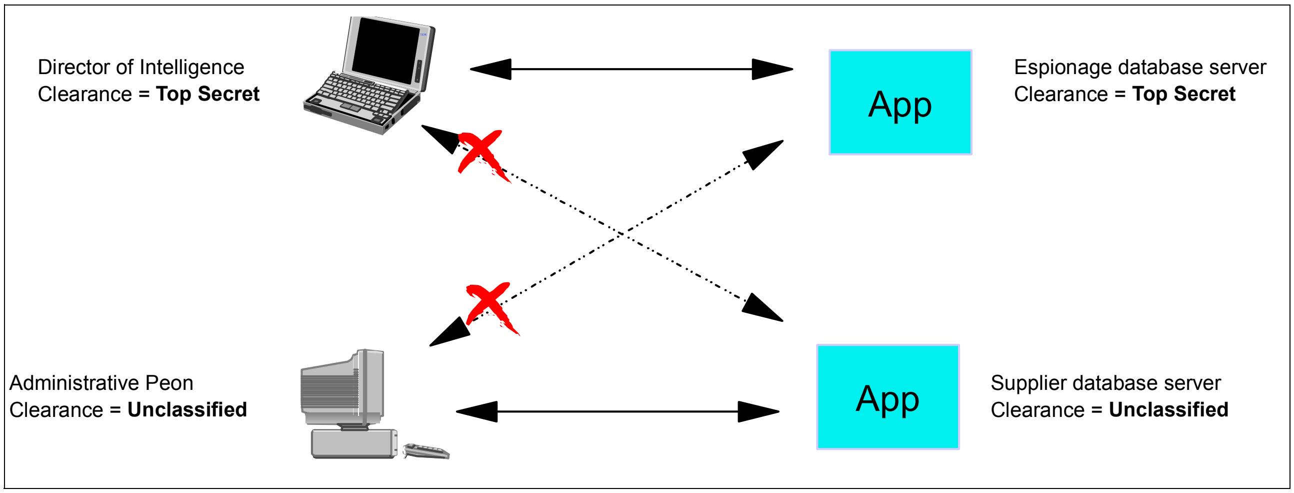


Figure 16-15 TCP/IP communications in a MLS environment

16.2.23 Multilevel security enhancements in z/OS V1R5 CS

Applications, users, and networks are classified and assigned a level of security. Classifications are done in RACF using SECLABELs. Multilevel security for Communication Server TCPIP stacks is based upon the existing NETACESS function.

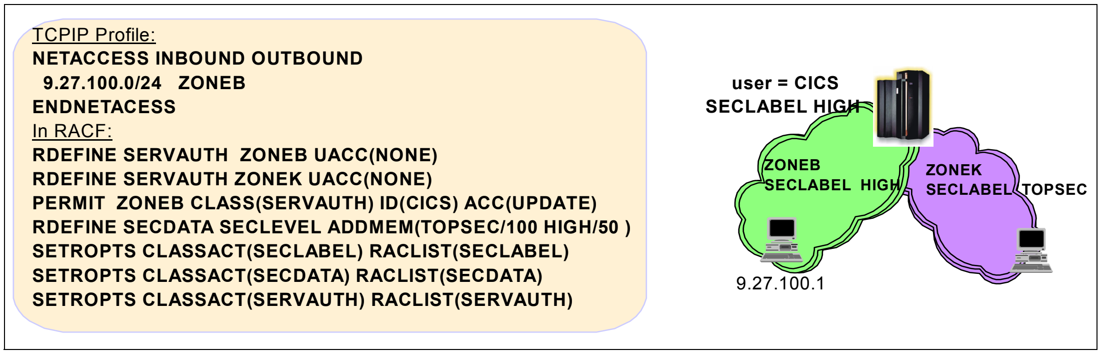
IP addresses, both INBOUND and OUTBOUND are defined to be in a security zone, as shown in Figure 16-16. The zones are defined as follows:

INBOUND Defines the IP addresses that data can be read from over a socket

OUTBOUND Defines the IP addresses that data can be sent to over a socket

The security zone is defined to RACF using a SERVAUTH class profile. A user may be RACF PERMITted to access the SERVAUTH profile, as shown in Figure 16-16.

In a multilevel security environment, the user and SERVAUTH profile (zone) are assigned a SECLABEL. For a user to communicate with a partner in a security zone, the SECLABELs must be equivalent.



```
TCPIP Profile:
NETACCESS INBOUND OUTBOUND
  9.27.100.0/24   ZONEB
ENDNETACESS
In RACF:
RDEFINE SERVAUTH  ZONEB UACC(NONE)
RDEFINE SERVAUTH ZONEK UACC(NONE)
PERMIT  ZONEB CLASS(SERVAUTH) ID(CICS) ACC(UPDATE)
RDEFINE SECDATA SECLEVEL ADDMEM(TOPSEC/100 HIGH/50 )
SETROPTS CLASSACT(SECLABEL) RACLIST(SECLABEL)
SETROPTS CLASSACT(SECDATA) RACLIST(SECDATA)
SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)
```

Figure 16-16 RACF profiles and NETACCESS definitions

16.2.24 Mail application enhancements

The following mail applications have been enhanced in z/OS V1R5 CS:

- z/OS UNIX sendmail agent/server
  - Support for mail filters
  - IPv6
  - Supports TLS
  - Many security controls provided for the operating environment
- SMTPD server
  - New IPMAILERNAME configuration parameter allows mail to be forwarded to the specified host name.

Sendmail is upgraded from version 8.8.7 to version 8.12.1

z/OS UNIX Sendmail is a mail program running in an UNIX System Services shell. The Sendmail program Version 8.12.1 is based on the Berkeley UNIX 4.1c code and it is enhanced in the following areas:

- Mail filter support

  The Sendmail Mail Filter API (Milter) is designed to allow third-party programs access to mail messages as they are being processed in order to filter meta-information and content.

- IPv6 support

  IP version 6 (IPv6) is a new version of the Internet Protocol, designed as the successor to IP version 4 (IPv4). The most significant change from IPv4 to IPv6 is the expanded addressing capabilities. Thus, it affects the IP resolving.

- TLS support is an extension to the SMTP service that allows an SMTP server and client to use transport-layer security to provide private, authenticated communication over the Internet. This gives SMTP agents the ability to protect some or all of their communications from eavesdroppers and attackers.

SMTP support for IP Mailer Name

An optional statement IPMAILERNAME is added to the SMTPPROC configuration data set. This statement enables Simple Mail Transfer Protocol (SMTP) to forward non-local mail to the specified IP mailer name.

Non-local mail is mail that must go through a Mail Transfer Agent (MTA) to get to another host. SMTP supports the following configuration statements to assist in forwarding non-local mail:

- **IPMAILERNAME:** This statement is for non-local mail destined for SMTP servers in the IP network using a hostname.
- **IPMAILERADDRESS:** This statement is for non-local mail destined for SMTP servers in the IP network using a static IP address.
- **MAILER:** This statement is for non-local mail destined for SMTP servers in the NJE network using the JES spool.

16.2.25 OMPROUT enhancements

z/OS V1R5 Communications Server enhances OMPROUTE in the following areas:

- OMPROUTE detailed trace can be diverted to the CTRACE facility rather than to a file. When the OMPROUTE CTRACE with option DEBUGTRC (or option ALL) is active, debug output is written to CTRACE and not to the destination specified by this variable.
- For IPv4, OMPROUTE can now handle an unlimited number of VIPA interfaces in addition to 255 real interfaces. For IPv6, the number of OMPROUTE interfaces is limitless.
- The limit on multipath dynamic routes increased from 4 to 16.
- OMPROUTE can be configured to ignore local interfaces that are not configured to it by specifying IGNORE_UNDEFINED_INTERFACES option.
- OMPROUTE allows display of generic interfaces (interfaces that are not running any routing protocol).
- OMPROUTE allows OSPF MD5 authentication keys to be specified in ways compatible with many vendor routers (Cisco).

16.2.26 TCP/IP enhancements

In z/OS V1R5 Communications Server, enhancements have been made in TCP/IP.

MVS system symbol resolution enhancements

Automatic resolution of MVS system symbols is supported for the Resolver setup file and for the TCPIP.DATA file. In previous releases, automatic resolution of MVS system symbols was not supported for the Resolver setup file or for the TCPIP.DATA file; it was necessary to use the EZACFSM1 utility program to resolve MVS system symbols for those files.

Increase maximum number of allowed sockets on z/OS V1R5 CS

The maximum number of sockets allowed has been increased to 65535 for the following Sockets APIs:

- Macro (EZASMI)

- Sockets Extended (EZASOKET)
- CICS Sockets (not including C Sockets under CICS)
- IMS Sockets
- REXX Sockets

TCP/IP asynchronous I/O support enhancements

Performance of asynchronous stream socket receive operations is improved when applications are changed to use common storage buffers. Authorized applications (executing in supervisor state or system key, APF-authorized, or superuser) must ensure that all I/O buffers are in common storage (such as ECSA or CSM-managed storage). Currently only USS Callable Services Sockets API and the LE C/C++ Socket API support this enhancement.

msys for Setup FTP customization support

Complete configuration of FTP servers and FTP clients is available using msys for Setup on z/OS V1R5 CS.

CSM buffer tracking in z/OS V1R5 Communication server

CSM monitor function is available to monitor CSM buffers between many components of z/OS CS used by IBM Software Support to help diagnose CSM storage problems.

16.3 SNA support in z/OS V1R5 Communication Server

SNA services are, like IP Services, a major component of z/OS V1R5 Communication server. z/OS V1R5 Communications Server is a network communication access method and an implementation of Systems Network Architecture (SNA) that includes advanced peer-to-peer networking (APPN) and high-performance routing (HPR). It provides the interface between application programs residing in a host processor, and resources residing in an SNA network. It also links peer users in the SNA network.

Enhancements have been made to the following SNA and IP services:

- SNA `DISPLAY` command and system definition
- SNA DLC performance
- SNA storage
- SNA Dump and Trace
- SNA EE
- SNA Resource Definition and Usability
- SNA Session Setup and Problem Determination
- INOP Dump
- SNA Performance/Storage in z/OS V1R5 Communication server

16.3.1 SNA serviceability improvements

z/OS V1R5 Communication server introduces usability improvements to the CSDUMP commands:

- The CSDUMP processing issues a new message that indicates the reason why a CSDUMP was triggered.

- MODIFY CSDUMP,DELETE command allows deletion of any or all of the sense code triggers set earlier.
- DISPLAY CSDUMP command is new. It displays current CSDUMP messages and setting of sense code triggers to take the dump.

APPN Trace enhancements

z/OS V1R5 Communications Server brings an enhancement to APPN trace of an APPN LU-LU session. New subtrace option TGVC provides TG Vectors in appropriate trace records for accurate problem diagnosing.

VTAM Trace improvements

New and changed VTAM Internal Trace (VIT) entries allow for high-speed interfaces in order to reduce record loss.

CSM Buffer tracking

z/OS V1R5 Communications Server introduces the CSM monitor function to monitor CSM buffers between components of z/OS CS. This trace helps in diagnosing CSM storage problems.

VTAM INOPCode commands

MODIFY INOPCODE and DISPLAY INOPCODE commands are new in z/OS V1R5 CS. They are used to display and to control the setting of VTAM inoperative condition (InOp) code attributes.

16.3.2 SNA performance and storage enhancements

The HPR resequencing optimization significantly improves inbound processing of out-of-order and segmented HPR packets, received over unreliable or multi-link transmission group connections.

Multipath Channel-to-Channel HPDT packing

High Performance Data Transfer (HPDT) Multi-Path Channel (MPC) protocol enables significant throughput optimization for small SNA or Enterprise Extender data packets. Packed sizes are examined, and for optimal outbound transmission, packing could be turned on or off at TRLE level by eliminating all of the alignment bytes in transmitted packets in the HPDT data segment.

For point-to-point connections using the HPDT Multi-Path Channel protocol, throughput of small SNA or Enterprise Extender data packets can be significantly improved by enabling HPDT packing. This solution provides for better utilization of the HPDT MPC data stream by eliminating all of the alignment bytes transmitted in the HPDT data segment.

A new PACKING operand is provided on the TRLE definition statement to allow for control of HPDT packing.

Storage management enhancements

A new VTAM modify command allows the IO Buffer pool expansion limit parameter to be modified without VTAM recycling. This enhancement includes a new VTAM modify command called BFRUSE with XPANLIM operand. Issue:

```
MODIFY procname,BFRUSE,BUF=IOBUF, XPANLIM=value
```

CRA buffer pool enhancements

Requests for Component Recovery Area (CRA) buffer pool increase are improved for conditions when CRA was too small to hold all of the data.

16.3.3 Enhanced SNA functionality in z/OS V1R5 CS

Enhancements have been made in this release to improve support of SNA applications.

Enterprise extender

Enterprise Extender (EE) supports SNA applications to include Internet Protocol (IP) networks and IP-attached clients with similar levels of reliability, scalability, and control as SNA users have. EE integration uses standard IP technology and does not require new hardware or software in the IP backbone.

- z/OS V1R5 Communication server enables definition of multiple VRN (Virtual Routing Node) for Enterprise Extender. One or more local or global VRNs could be defined in XCA (External Communication Adapter) major node of Enterprise Extender. Each local or global VRN represents an IPv6 or IPv4 network. In addition, a unique static local VIPA could be defined on the GROUP definition statement.
- A new message has been introduced if dial fails or a connection INOPs over a virtual routing node (VRN). Prior to this enhancement, retries for a dial failure or an INOP for an existing connection would fail without the operator being notified.
- A new type of model PU can be defined for PUs used for Enterprise Extender on the host receiving the dial. This allows the user flexibility in coding operands such as TG characteristics, disconnect timer delay, or whether to attempt redial when connection fails.
- IPv6 and Network address translation (NAT) firewall changes were described previously.

Allow non-sysplex network nodes (NNs) for generic resource (GR) ENs

This enhancement removes the restriction in previous versions of CS that generic resource function on an end node (EN) must be in the same sysplex.

Sift-down support for model major nodes

z/OS V1R5 Communications Server provides the ability to sift parameters for model LU definition statements in the model major node. This could reduce repetitive parameter definitions.

SWNORDER and DLRORDER enhancements

The SWNORDER and DLRORDER parameters that can be specified as start options have been enhanced to allow greater control over PU selection during connection processing. By specifying SWNORDER and DLRORDER on the XCA or NCP major nodes, the start option value can be overridden on a line-by-line basis.

Session setup and problem determination enhancements

With this enhancement a backup network node server does not need to be connected to the same sysplex as the served end nodes. SSCPORD search option can be specified as a VTAM start option or in the ADJSSCP table as an operand on the NETWORK and CDRM statements. It provides VTAM the ability to search ADJSSCP tables of adjacent SSCPs in a specified order.

Multiple VRN/VIPA support for Enterprise Extender

Enterprise Extender in z/OS V1R5 Communications Server provides the ability to specify multiple local and global virtual routing nodes (VRS) that represent networks, as shown in

Figure 16-17. In this figure, Node B defines two local VRNs (LVRNA and LVRNB) that are IPv4 networks and two global VRNs (GVRNB4 and GVRNB6) that are IPv6 and IPv4 networks.

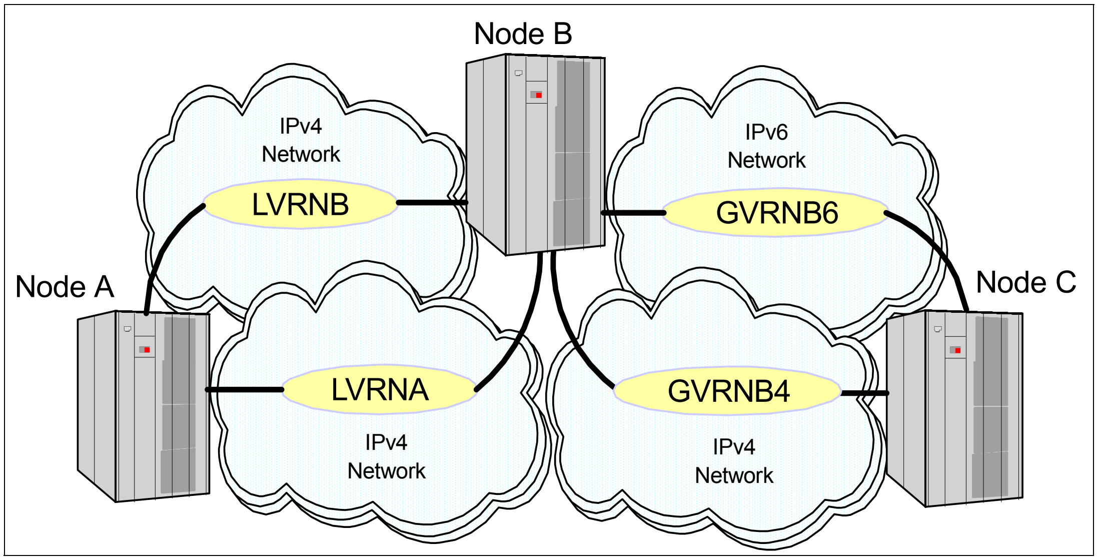


Figure 16-17 Specification of multiple local and/or multiple global EE connection networks

Enterprise extender is defined as XCA (External Communication Adapter) major node and allows multiple static VIPA defined in GROUP statement.

16.3.4 Enterprise Extender connection network

Name-to-address resolution for acquiring the source VIPA address of the local and remote EE endpoints is required for EE connection networks that utilize IPv6 addressing. EE connection networks that utilize IPv4 addressing can also use name-to-address resolution, provided that both endpoints are z/OS V1R5 or higher. Otherwise, the appropriate source VIPA address can be explicitly defined to VTAM.

Hostnames can be used for configurations that require network address translation (NAT) between the two EE connection endpoints.

A new message has been introduced to inform the operator when a dial fails or a connection INOPs over a virtual routing node (VRN).

Prior to this enhancement, retries for a dial failure or an INOP for an existing connection that was routed over a connection network link would repeatedly fail without the operator knowing what virtual routing node was being used.

16.3.5 SNA problem determination in z/OS V1R5 CS

Changes have been made to improve problem determination in SNA network environments.

RTP display enhancement

The D NET,RTPS command is enhanced. This enables an operator to filter the HPR pipe displays and control messages on the display. New filters were added to the command to allow for displaying RTP (Rapid Transport Protocol) messages by characteristics associated with the first hop:

- TG number

- CPNAME
- Adjacent Link Station name

DLUR message enhancements

The message enhancements are:

- DLUS accounting messages are issued when a DLUR (Dependent LU Requester) served physical unit begins or ends communication with its DLUS (Dependent LU Server) or, if an INOP of the SSCP to PU session occurs.
- DLUS serviceability aid is a group of messages issued to indicate negative response received for a request, such as an ACTLU, DACTLU, ACTPU, or DACTPU.

Support for concurrent APING commands

In z/OS V1R5 Communications Server more than one display `APING` command can be active concurrently.

MAXSLOW parameter for slowdown monitoring

Slowdown monitoring and operator awareness for XCA subchannels is enhanced by introducing a MAXSLOW parameter to allow a second time value. This second time value is the number of seconds an XCA subchannel is allowed to remain in a slowdown condition before the operator is notified. Default value for detecting an extended period of an XCA subchannel slowdown is 180 seconds.

Session setup and problem determination enhancements

The DSIRFMSG start option enhances the ability to receive message groups when the search to locate a session partner fails.

16.3.6 Migration and coexistence considerations

There are many migration and coexistence considerations when migrating to z/OS V1R5 Communication Server.

Migration changes for Netstat enhanced messaging

The `Netstat` command displays the status of a host. Changes in CS cause several messages to appear in “long” format. That could affect automation programs that run off Netstat or front-end programs to Netstat.

Migration consideration for QoS Policy Agent

If the LDAP_SchemaVersion parameter on the Policy Agent ReadFromDirectory configuration statement is not specified, verify the schema version of the policy objects defined on the LDAP server. If schema version 3 objects are defined, no action is necessary. Otherwise, specify the correct schema version on the ReadFromDirectory statement.

Migration of SMTP (Simple Mail Transport Protocol) daemon

PASCAL application programming interface (API) MonQuery function has a possible return code value of UNAUTHORIZEDuser in certain error conditions.

Enterprise extender

- Enterprise Extender message EZZ4313I could be used to automate VTAM definitions activation. In previous releases, TCP/IP issued message EZZ4313I for MPCPTP devices before the LINK became active. In z/OS V1R5 Communications Server changes in automation procedure are necessary because TCP/IP does not issue message EZZ4313I until the LINK is marked active.

- Several changes were made in the area of hostname resolution enhancements that might require migration actions. The default value of the IPRESOLV operand on the PATH statement is changed from 45 seconds to 0 seconds (no time-out). Also, a length limitation of 64 characters has been imposed on the HOSTNAME operand of the PATH statement in the Switched Major Node.

Migration consideration for SMF

Automated SMF processing tools needs to be updated in order to process SMF 119 records collected for Telnet servers and clients, FTP servers and clients, API activity, and stack usage statistics. Utilize the EZASMF77 macro (in SYS1.MACLIB) to get the correct mappings of the changes in SMF record 119.

Migration consideration for REXEC server

User written exits may fail for IPv6 addresses. Exits must be updated; or, when configuring z/OS TCP/IP stack as an AF_INET6 network, specify the IPV6=N start option in the RXSERVE procedure.

Migration of Sendmail

The definition files default directory has been changed.

Migration of TN3270

User written exits and supporting files need to be updated for IPv6 stack.

Dump datasets

During recovery from a failure, both VTAM and TCP/IP address spaces may be dumped. The VIT dataspace and TCP/IP CTRACE dataspace may also be included in the dump. This will provide a more useful dump, but it will lead to larger dump dataset sizes.

Migrating applications in restricted stack access environments

Applications may need to be enhanced to provide stack affinity in CINET environments if stack access is restricted to specific users.

Tracing

New CTRACE formatting filter enhancements allow additional TCP/IP CTRACE and VTAM internal trace to be performed in storage constrained conditions.

Sysplex functions

In a sysplex environment, consider the following:

- **Client/Server affinity**: Client/Server affinity is established between a client and a particular server instance for applications that needs to span multiple connections. TN3270 printer sessions, for example, are based on a connection request from a TN3270 client, and that printer session connection request needs to be routed to the same TN3270 server that is serving the LU2 session. Similarly, Web-based applications, such as shopping carts, might need to have all connections from a particular client come to the server instance that has the contents of the shopping cart stored as session state. Distributing and target stack must be on z/OS V1R5 or later.
- **Round-robin distribution:** This kind of distribution might be appropriate for specific server applications. You can select round-robin distribution among DVIPA/port targets for a particular application, rather than distribution based on capacity and policy information,

through a configuration option. The primary and backup distributing stacks must be on z/OS V1R4, V1R5 or later.

Migration consideration for session setup messages

Additional session setup messages will be issued which may affect automation programs.

Migration considerations for Enterprise Extender

Enterprise Extender communications over IPv6 requires that partner host is on z/OS V1R5 (or higher) at this time.

A

RMF Performance Monitor metrics

This appendix provides the contents of the RMF Website found on the internet at:

http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pm_metrics.htm

This wesbite contains a detailed description of the metrics used in the RMF Performance Monitor. As this information is not published in any other media, it is provided in this appendix for the readers convenience.

A.1 RMF PM - Metrics

RMF PM has two formats for presenting performance data:

- **Single-Value Metrics**, for example
  - **% utilization** (of a processor, of a channel, ...)
  - **i/o activity rate** (of a logical control unit, ...)
- **Value-List Metrics**, for example
  - **% utilization by job**
  - **# delayed jobs for i/o by mvs image**

    The unique indicator in the name of a Value-List Metric is the keyword **by.**

% channel path partition utilization

The channel path utilization percentage for an individual logical partition. RMF uses the values provided by CPMF (Channel Path Measurement Facility).

In LPAR mode, the calculation is:

```
% partition utilization = (CBT / CET) * 100
```

CBT

Cumulative channel path busy time

CET

Cumulative channel path elapsed time

In BASIC mode, blanks are shown.

% channel path total utilization

The channel path utilization percentage for the entire system during an interval. For shared channels in LPAR mode, or for all channels in BASIC mode with CPMF not available, the calculation is:

```
% total utilization = (SCB / N) * 100
```

SCB

Number of SRM observations of channel path busy

N

Number of SRM samples

For unshared channels in LPAR mode, the value for total utilization is the same as partition utilization. For all channels in BASIC mode with CPMF available, the calculation is:

% total utilization = (CBT / CET) * 100

CBT

Cumulative channel path busy time

CET

Cumulative channel path elapsed time

% enqueue delay

The percentage of time during the report interval that the system or job was waiting to use a serially reusable resource that another system or job was using.

% HSM delay

The percentage of time during the report interval that the system or job was waiting for services from the Hierarchical Storage Manager (HSM). A **high HSM delay** value might be caused by one or more of the following:

- HSM address spaces delayed (Check HSM address spaces on the Job report)
- Delay on HSM volumes (Check HSM device volumes on the DEVR report)
- HSM doing its housekeeping during prime time
- Not enough primary or level one space
- HSM dispatching priority too low.

% JES delay

The percentage of time during the report interval that the system or job was waiting for services from the Job Entry Subsystem (JES). A **high JES delay** value might be caused by one or more of the following:

- JES address spaces delayed (Check JES address spaces on the Job report)
- Delay on JES volumes (Check JES device volumes on the DEVR report)
- JES dispatching priority too low.

% operator delay

The percentage of time during the report interval that the system or job was waiting for the operator to reply to a message or mount a tape, or the address space was quiesced by the operator.

% processor delay

The percentage of time during the report interval that the system or job or enclave was waiting for a processor. A **high processor using** value might be caused by one or more of the following:

- looping user
- high dispatching priority for a processor-bound job (in compatibility mode) or high importance for the service class of a processor-bound job (in goal mode)
- small block size I/O
- excessive use of expensive supervisor service

A **high processor delay** value might be caused by one or more of the following:

- ineffective choice of dispatching priorities in either the SRM IPS (compatibility mode) or ineffective choice of importances in the active service policy (goal mode)
- high priority work using an excessive amount of CPU
- ineffective mean-time-to-wait usage

% storage delay

The percentage of time during the report interval that the system or job was waiting for a COMM, LOCL (both include shared pages), SWAP, or VIO page, was on the out/ready queue, or was a result of a cross-memory address space or standard hiperspace paging delay.

For enclaves, only COMM, cross-memory, and shared page delays apply. A **high storage delay** value can be associated with common storage paging (COMM), local storage paging (LOCL), swap-in delay (SWAP), swapped out and ready delay (OUTR), and other delays

(OTHR) which includes virtual I/O paging and paging delays from cross-memory address spaces and standard hiperspaces. A high storage delay associated with **common storage paging** might be caused by one or more of the following:

- insufficient page data sets
- not enough central storage
- poorly tuned paging configuration
- too many address spaces in storage
- too many "logical swap" address spaces in storage
- excessive storage isolation of address spaces
- too many extremely large address spaces resident
- paging data set on shared device
- high use of user I/O on paging volume
- "common I/O" contends with "swap I/O"
- common data set on wrong device

A high storage delay associated with **local storage paging** might be caused by one or more of the following:

- insufficient page data sets
- not enough central storage
- address space is under isolated (causing trim) or over isolated (causing others to page/swap)
- poorly tuned paging configuration
- too many address spaces in storage
- too few (artificially low) address spaces in storage
- too many "logical swap" address spaces in storage
- paging data set on shared device
- high use of user I/O on paging volume
- too much swapping
- page-ins are from trimming at swap-out
- "local I/O" contends with "swap I/O"
- program pages in each address space rather than in PLPA
- too many extremely large address spaces resident

A high storage delay associated with **virtual I/O** might be caused by one or more of the following:

- insufficient page data sets
- poorly tuned paging configuration
- paging data set on shared device
- high use of user I/O on paging volume
- virtual I/O contending with swap I/O

A high storage delay associated with **swap-in** activity might be caused by one or more of the following:

- too much swapping
- workload too heavy
- insufficient page/swap data sets
- misplaced page/swap data sets
- swap data sets on slow devices
- too few (artificially low) address spaces in storage
- paging data set on shared device
- high use of user I/O on paging volume
- swapped pages moved to backing store on cached device
- not enough central storage

A high delay value for address spaces that are **swapped out and ready** might be caused by one or more of the following:

- too few (artificially low) address spaces in storage
- workload too heavy
- unbalanced workload
- not enough central storage
- poorly tuned paging configuration
- insufficient page/swap data sets
- too many address spaces in storage
- too many or too few logical swap address spaces
- paging/swapping too slow
- exchange swap rate too high
- too many detected wait swaps
- improper use of storage isolation

Other storage delays might be caused by one or more of the following:

- paging delays from cross-memory address spaces
- paging delays from standard hiperspaces (but not ESO hiperspaces)

% subsystem delay

The percentage of time during the report interval that the system or job was waiting for services from

- Job Entry Subsystem (JES)
- Hierarchical Storage Manager (HSM)
- Cross-System Coupling Facility (XCF)

% XCF delay

The percentage of time during the report interval that the system or job was waiting for services from the Cross-System Coupling Facility (XCF). A **high XCF delay** value might be caused by one or more of the following:

- Path capacity exceeded.
- Other applications are tying up the path.

- XCF delays on the receiving system.
- Some data paths are unavailable or offline.

% total delay

The percentage of time during the report interval that the job was not using any resources and was delayed for at least one of the following resources:

- processor - the job had ready work on the dispatching queue.
- storage - the job was delayed by paging, swapping or virtual input/output (VIO) activity, or was on the out/ready queue.
- device - the job was waiting for a DASD or tape.
- Job Entry Subsystem (JES)
- Hierarchical Storage Manager (HSM)
- Cross-System Coupling Facility (XCF)
- **OPER** - the job was waiting for the operator to reply to a message or to mount a tape, or the address space was quiesced by the operator.
- **ENQ** - the job was waiting to use a serially reusable resource that another job was using.

Note: If a job with several tasks is simultaneously delayed for more than one resource, RMF counts this job only once as delayed when it calculates delay percentage.

% idle

The percentage of time during the report interval that the system or job was idle. RMF considers a job idle if it is in terminal wait, timer wait, or is waiting to be selected by JES, and it is not using or waiting for any resource that RMF monitors.

% using

The percentage of time during the report interval that the system or job was using one or more processors or devices.

Note: If a job with more than one task is simultaneously using and delayed for the same resource, RMF counts the job once as using and once as delayed (regardless of how many times it is found using and delayed). If a job is delayed for more than one resource, it is counted once for the overall delay and once for each resource causing a delay.

% workflow

Workflow percentage is the speed at which a job is moving through the system in relation to the maximum speed at which it could move through the system. A low workflow percentage indicates that the job has few of the resources it needs and is contending with other jobs for system resources. A high workflow percentage indicates that the job has the resources it needs to execute and is moving through the system at a relatively high speed. For example, a job that could execute in one minute if all the resources it needed were available, would have a workflow of 25 percent if it took four minutes to execute.

% unknown

RMF considers the system or jobs that are not delayed for a monitored resource, not using a monitored resource, or not in a monitored idle state to be in an unknown state.
The value represents the percentage of time during the report interval that the job was in the system, but not in any monitored state. Examples of address spaces in an unknown state include those waiting for devices other than DASD or tape and those that are waiting for work (idle) using a method that RMF does not recognize. Started tasks (STCs) are usually found in this category.

% connect time

The sum of the percentages of time during the report interval that devices used by the job were connected to channel path(s) to transfer data between the devices and central storage. Because a job can be connected to more than one device at a time, the value in connect time percentage can be greater than 100 %.

Note: This can include devices other than DASD and tape; for example, graphic displays.

% using

The percentage of time during the report interval that one job or all jobs in a group or in the system were using one or more devices. RMF considers a job to be using a device as soon as the job's I/O request is queued in the channel for the device. Therefore, the using percentage for a device includes both active time on the device and queuing delay in the channel.

i/o activity rate

The rate per second that I/O instructions (SSCH, RSCH, and HSCH) to a device completed successfully.

IOS queue time

The average number of milliseconds an I/O request must wait on an IOS queue before an SSCH instruction can be issued. A delay occurs when a previous request to the same subchannel is in progress.

response time

The average response time (in milliseconds) that the device required to complete an I/O request.

i/o intensity

The product of the number of users and the time waiting in average for a DASD device because of one of the following reasons:

- The path and device are busy
- The SIO is pending
- The device is busy
- The SIO is queued

There is no common name for I/O intensity in the literature. Other programs might use different names. The following terms are equivalent to I/O Intensity: DASD MPL, Response Time Volume.

% active time

The percentage of time during the report interval that the device was active.

```
active time = connect time + disconnect time + pending time
```

% connect time

The percentage of time during the report interval that the device was connected to a channel path.

% disconnect time

The percentage of time during the report interval that the device had an active channel program, but was not connected to the channel. Disconnect time includes seek time, normal rotational delay time, and extra rotational delay time because the channel was busy.

% pending time

The percentage of time during the range period that I/O requests were waiting in a channel queue before a path was available. Pending time includes the time spent waiting for a device, a control unit, a head of string, or a channel.

% I/O delay

The percentage of time during the report interval that the job is waiting for any DASD or tape, or has an I/O request queued in the channel for a device, but not transmitting data (for example, is being disconnected to seek). A **high device delay** value for a job usually means that another job has a high using value for the same device. Use the Device Delay report to determine what volume a job is waiting for; then use the Device Resource Delay report to determine how the job using that volume is spending its time.

General reasons for a **high device using** value might include:

- unnecessary I/O (such as using DASD instead of VIO for temporary data sets)
- data sets on a slow device

Using time for a volume will approximately equal connect time (time that the device was connected to a channel path). Using time does not include disconnect time (time that the device had an active channel program but was not connected to the channel) and pending time (time that I/O requests were waiting in a channel queue before a path was available).

A high **connect percentage (CON %)** might be caused by one or more of the following:

- programs not resident
- inappropriate application parameters
- inefficient use of device by application(s)
- not enough in-storage buffering
- heavy BLDL activity
- high VTOC activity

A high **disconnect percentage (DSC %)** might be caused by one or more of the following:

- small block size I/O
- multiple revolutions per I/O due to missing channel connects or reconnects
- long seeks because of data set placement or multiple extents on high use data sets
- heavy BLDL activity

- high miss ratio for cached device
- misplaced VTOC or CATALOG or both
- channel, control unit, or head of string contention

A high **pending percentage (PND %)** might be caused by one or more of the following:

- shared DASD contention
- device not responding
- channel, control unit, or head of string contention
- poorly balanced I/O
- PND time of 100 % usually means another system had the device reserved

% delay device busy

The percentage of time during the range period when there was an I/O request delay because the device was busy. Device busy might mean that another system is using the volume, another system reserved the volume, or a head of string busy condition caused the contention.

% control unit busy

The percentage of time during the range period when there is an I/O request delay because the control unit was busy. If the device is shared at the control unit level, a sharing system might be using the device. If the device is not shared at the control unit level, the contention is the result of other activity to different devices over an alternate path serviced by this control unit.

% director port busy

The percentage of time during the range period when there is an I/O request delay because the ES Connection Director port was busy.

% using

The percentage of time during the report interval that the job was using the volume.
RMF considers a job to be using a device as soon as the job's I/O request is queued in the channel for the device. Therefore, the using percentage for a device includes both active time on the device and queuing delay in the channel.

% all channel paths busy

The percentage of time during the measurement interval when all channel paths belonging to the LCU were busy at the same time. Only channel paths that are both online to the system and connected to a device are included in the calculation:

```
% all channel paths busy = CHPID0 * CHPID1 * CHPID2 * CHPID3
```

where CHPIDn = Percentage busy of each channel path involved

% control unit busy

This value shows for each channel path of the LCU the relationship between requests deferred due to control unit busy and total successful requests serviced by that path. Each CHPID of the LCU measures the distribution of control unit contention. The calculation is:

```
% control unit busy = ((CUB / (DPB + CUB + SUC)) * 100
```

- DPB = Number of deferred I/O requests due to director port busy

- CUB = Number of deferred I/O requests due to control unit busy
- SUC = Number of successful I/O requests on that path

% director port busy

This field indicates **director port contention** . It is the number of times an I/O request was deferred because the director port was busy during the measurement interval. The calculation is:

```
% director port busy = ((DPB / (DPB + CUB + SUC)) * 100
```

- DPB = Number of deferred I/O requests due to director port busy
- CUB = Number of deferred I/O requests due to control unit busy
- SUC = Number of successful I/O requests on that path

% CHPID taken

The rate at which I/O requests to devices of this LCU are satisfied by each CHPID during the interval. By reviewing the rate at which each channel path of the LCU satisfies I/O requests, you can see how evenly the work requests are distributed among the available paths and how effectively those paths are arranged for the LCU. The calculation is:

```
% CHPID taken = (TO / SI) * 100
```

TO

Total number of I/O operations accepted on that path

SI

Number of seconds in the interval

# delayed i/o requests

The average number of **delayed requests** on the control unit header (CU-HDR). Each time a request is enqueued from the CU-HDR, RMF counts the number of requests remaining on the queue and adds that number to the accumulator. The calculation is:

```
# delayed i/o requests = (AL - ER) / ER
```

AL

Accumulated queue length

ER

Total number of enqueued requests

### delayed i/o request rate

The rate per second at which the IOP places delayed I/O requests on the CU-HDR for this LCU. This is done when all paths to the subchannel are busy and at least one path to the control unit is busy. For devices with only one path, or for devices where multiple paths exist and the busy condition is immediately resolved, the IOP does not count the condition. The calculation is:

```
delayed i/o request rate = ER / SI
```

ER

Total number of enqueued requests

SI

Number of seconds in the interval

% delay by volume

The percentage of delay caused because the job was waiting to use the named volume.

% using

The percentage of time during the report interval that one job or all jobs in a group or in the system were using one or more processors.

% TCB+SRB

The percentage of total processor time used by the job during the report interval.

working set

The working set represents the (central or expanded) storage the user has when a job is actually running. Shared page counts are not included in the working set.

% delay for SWAP

The percentage that swap-in delays contributed to the delay of a job.

% delay for COMM

The percentage that common storage (common service area (CSA) or link pack area (LPA)), including shared pages, contributed to the delay of a job.

% delay for LOCL

The percentage that local (private) storage paging, including shared pages contributed to the delay of a job.

% delay for OTHR

The percentage that various types of delays contributed to the delay of a job.
This is the sum of:

- VIO (virtual I/O)
- Paging delays from cross-memory address spaces. For example, if the DB2 address space does not have sufficient central/expanded storage, CICS could be delayed by cross-memory page-in in the DB2 address space. This would show up as a cross-memory delay for CICS.
- Paging delays from standard hiperspaces (but not ESO hiperspaces). This delay could be caused by a job running DFSORT with hipersorting if the DFSORT hiperspace's pages were migrated from expanded to auxiliary storage.

% delay for OUTR

The percentage that swapped-out-and-ready delays contributed to the delay of a job.

% available

The percentage of common storage (CSA, ECSA, SQA, or ESQA) available for allocation at the end of the specified range period.

% not released

The percentage of allocated common storage (CSA, ECSA, SQA, or ESQA) that was not released when a job ended.

% utilization

The percentage of common storage (CSA, ECSA, SQA, or ESQA) used during the specified range period.

# frames not released

The amount of allocated common storage (CSA, ECSA, SQA, or ESQA) that was not released when a job ended.

# frames used

The amount of common storage (CSA, ECSA, SQA, or ESQA) used during the specified range period.

# frames defined

The amount of common storage (CSA, ECSA, SQA, or ESQA) defined to the system at IPL.

# frames idle

The average number of frames held by a job while it was idle.

# frames total

The sum of the active and idle frames.

> **Note:** The shared page counts are not included in this value.

# frames active

The average number of frames held by a job while it was active.

# frames fixed

The average number of fixed frames a job was using during the report interval including frames both above and below the 16 megabyte line. A fixed frame is a frame that cannot be paged out of central storage.

# frames DIV

The DIV frame count represents the number of Data-in-virtual frames in relation to the number of Data-in-virtual samples.

# slots

The total number of the auxiliary storage slots a job used, averaged over the report interval.

es rate per residency time

The value is the rate of page-moves from expanded storage to central storage per active second. This count is the total page-move count divided by the time the user was swapped-in. It includes single and blocked pages, but does not include shared, hiperspace or VIO pages.

pgin rate

The rate at which pages are being read into central storage. It is calculated by dividing the total page-in count (for the group) by the resident time. The address-space related shared storage page-ins are included in the value.

migration age

Migration age is the average number of seconds a page resides on expanded storage before it migrates to auxiliary storage.

unreferenced interval count

The average high unreferenced interval count (UIC) is an indicator of central storage contention. A high UIC count indicates that storage contention is low and you are not experiencing any storage problems.

% frames active

The percentage of storage allocated to jobs that are active.

% frames available

The percentage of available storage.

% frames idle

The percentage of storage allocated to jobs that are idle.

% frames CSA

The percentage of storage allocated to the common storage area (CSA).

% frames LPA

The percentage of storage allocated to the link pack area (LPA).

% frames NUC

The percentage of storage allocated to the nucleus (NUC).

% frames SQA

The percentage of storage allocated to the system queue area (SQA).

# delayed jobs for COMM

The average number of jobs in each group that are delayed for common storage (common service area (CSA) or link pack area (LPA)), including shared pages.

# delayed jobs

The average number of jobs in each group that are delayed for any of the storage reasons COMM, LOCL, SWAP, OUTR, or OTHR.

# delayed jobs for OTHR

The average number of jobs in each group that are delayed for various types of delays. This is the sum of:

- VIO (virtual I/O)
- Paging delays from cross-memory address spaces. For example, if the DB2 address space does not have sufficient central/expanded storage, CICS could be delayed by

cross-memory page-in in the DB2 address space. This would show up as a cross-memory delay for CICS.

– Paging delays from standard hiperspaces (but not ESO hiperspaces). This delay could be caused by a job running DFSORT with hipersorting if the DFSORT hiperspace's pages were migrated from expanded to auxiliary storage.

# delayed jobs for OUTR

The average number of jobs in each group with swapped-out-and-ready delays.

# delayed jobs for LOCL

The average number of jobs in each group that are delayed for local (private) storage paging, including shared pages.

# frames online

Central storage

Number of central storage frames, excluding read-only frames.

Nucleus frames are included in this Metric.

Expanded storage

Number of usable expanded storage frames.

# delayed jobs for SWAP

The average number of jobs in each group with swap-in delays.

pgin rate per residency time

The average number of page-ins per second for an address space. The calculation is the total number of non-swap page-ins (including VIO page-ins, hiperspace page-ins, page-ins caused by page faults, and shared storage page-ins) during the range period divided by the total time an address space was swapped-in (residency time).

execution velocity

The execution velocity of the MVS system, workload group, service class or service class period being reported on. This value is calculated independent of a specified goal.
The value for execution velocity is calculated as CPU using, divided by the sum of CPU using and total delays gathered by WLM. A high value indicates little workload contention while a low value indicates that the requests for system resources are delayed.

response time

The average response time (in seconds) for all transactions of a job class (*SYSTEM, *TSO, *BATCH, *STC, *ASCH or *OMVS), a WLM workload, or WLM service or report class that ended during the range period. The response time value is the sum of the queued time and the active time for an average ended transaction.

transaction rate

The number of transactions per second for a job class (*SYSTEM, *TSO, *BATCH, *STC, *ASCH or *OMVS), a WLM workload, or WLM service or report class during the range period.

% partition utilization

MVS view of CPU utilization. For example, if an MVS partition has 5% of the processor capacity and the physical CPU utilization reported by RMF for the partition is 5%, this indicates an MVS view of 100% CPU utilization. This Metric is available in LPAR mode only, because in **Basic mode** (non-LPAR mode) this value is shown in the **% total utilization** Metric.

% workflow

The average speed at which the jobs in the group are moving through the system in relation to the maximum speed at which they could move through the system. A low workflow percentage indicates that jobs in the group have few of the resources they need and are contending with other jobs for system resources. A high workflow percentage indicates that jobs in the group have the resources they need and are moving through the system at a relatively high speed. For example, jobs in a group that could process in four minutes if all the resources that they needed were available, would have a workflow of 25% if they took sixteen minutes to process.

% average CPU utilization

The average utilization percentage for all processors during the report interval.

# active users

The average number of active users in the system or in a group of address spaces. Active users include those using a monitored resource, those delayed for a monitored resource, and those doing activities that RMF does not measure. Each system user is either active, idle or unknown during a report interval.

% SRB

The average percentage of SRB time used by the system.

% TCB

The average percentage of TCB time used by the system.

% TCB+SRB

The average percentage of processor time used by all address spaces per processor.

# users

The average number of total users in the system or in a group of address spaces.

# using jobs

Average number of users using devices.

# using jobs

Average number of users using the processor.

# processor online

The number of processors online during the range period.

% workflow

Workflow percentage with respect to the processor is the speed at which one job or all jobs in a group or in the system are using the processor(s) in relation to the maximum speed at which they could do this. The calculation for this value is:

```
%workflow = (%using / (%using + %delay)) * 100
```

In this formula, the values of %using and %delay refer to the processor.

% workflow

Workflow percentage with respect to devices is the speed at which one job or all jobs in a group or in the system are using the devices in relation to the maximum speed at which they could do this. The calculation for this value is:

```
%workflow = (%using / (%using + %delay)) * 100
```

In this formula, the values of %using and %delay refer to devices.

# using jobs

The average number of jobs using either the processor or devices during the report interval.

# delayed jobs

The average number of jobs that are delayed during the report interval because of at least one of the following reasons:

- Waiting for a processor
- Waiting for a device
- Waiting for storage
- Waiting for a subsystem (JES, HSM, XCF)
- Waiting for the operator
- Waiting for serially reusable resource (enqueue)

# delayed jobs for enqueue

The average number of jobs for each group that are waiting to use a serially reusable resource that another system or job was using.

# delayed jobs for HSM

The average number of jobs for each group that are waiting for services from the Hierarchical Storage Manager (HSM).

A **high HSM delay** value might be caused by one or more of the following:

- HSM address spaces delayed (Check HSM address spaces on the Job report)
- Delay on HSM volumes (Check HSM device volumes on the DEVR report)
- HSM doing its housekeeping during prime time
- Not enough primary or level one space
- HSM dispatching priority too low.

# delayed jobs for JES

The average number of jobs for each group that are waiting for services from the Job Entry Subsystem (JES).

A **high JES delay** value might be caused by one or more of the following:

- JES address spaces delayed (Check JES address spaces on the Job report)
- Delay on JES volumes (Check JES device volumes on the DEVR report)
- JES dispatching priority too low.

# delayed jobs for operator

The average number of jobs for each group that are waiting for the operator to reply to a message or mount a tape, or the address space was quiesced by the operator.

# delayed jobs for subsystem

The average number of jobs for each group that are waiting for services from

- Job Entry Subsystem (JES)
- Hierarchical Storage Manager (HSM)
- Cross-System Coupling Facility (XCF)

# delayed jobs for XCF

The average number of jobs for each group that are waiting for services from the Cross-System Coupling Facility (XCF).

A **high XCF delay** value might be caused by one or more of the following:

- Path capacity exceeded.
- Other applications are tying up the path.
- XCF delays on the receiving system.
- Some data paths are unavailable or offline.

# delayed jobs for I/O

The average number of jobs for each group that are waiting for any DASD or tape, or has an I/O request queued in the channel for a device, but not transmitting data (for example, is being disconnected to seek).

A **high device delay** value for a job usually means that another job has a high using value for the same device. Use the Device Delay report to determine what volume a job is waiting for; then use the Device Resource Delay report to determine how the job using that volume is spending its time. General reasons for a **high device using** value might include:

- unnecessary I/O (such as using DASD instead of VIO for temporary data sets)
- data sets on a slow device

Using time for a volume will approximately equal connect time (time that the device was connected to a channel path). Using time does not include disconnect time (time that the device had an active channel program but was not connected to the channel) and pending time (time that I/O requests were waiting in a channel queue before a path was available). A high **connect percentage (CON %)** might be caused by one or more of the following:

- programs not resident
- inappropriate application parameters
- inefficient use of device by application(s)
- not enough in-storage buffering
- heavy BLDL activity
- high VTOC activity

A high **disconnect percentage (DSC %)** might be caused by one or more of the following:

- small block size I/O
- multiple revolutions per I/O due to missing channel connects or reconnects

- long seeks because of data set placement or multiple extents on high use data sets
- heavy BLDL activity
- high miss ratio for cached device
- misplaced VTOC or CATALOG or both
- channel, control unit, or head of string contention

A high **pending percentage (PND %)** might be caused by one or more of the following:

- shared DASD contention
- device not responding
- channel, control unit, or head of string contention
- poorly balanced I/O
- PND time of 100 % usually means another system had the device reserved

# delayed jobs for processor

The average number of jobs for each group that are waiting for a processor. A **high processor using** value might be caused by one or more of the following:

- looping user
- high dispatching priority for a processor-bound job (in compatibility mode) or high importance for the service class of a processor-bound job (in goal mode)
- small block size I/O
- excessive use of expensive supervisor service

A **high processor delay** value might be caused by one or more of the following:

- ineffective choice of dispatching priorities in either the SRM IPS (compatibility mode) or ineffective choice of importances in the active service policy (goal mode)
- high priority work using an excessive amount of CPU
- ineffective mean-time-to-wait usage

# delayed jobs for storage

The average number of jobs for each group that are waiting for a COMM, LOCL (both include shared pages), SWAP, or VIO page, was on the out/ready queue, or was a result of a cross-memory address space or standard hiperspace paging delay. For enclaves, only COMM, cross-memory, and shared page delays apply. A **high storage delay** value can be associated with common storage paging (COMM), local storage paging (LOCL), swap-in delay (SWAP), swapped out and ready delay (OUTR), and other delays (OTHR) which includes virtual I/O paging and paging delays from cross-memory address spaces and standard hiperspaces. A high storage delay associated with **common storage paging** might be caused by one or more of the following:

- insufficient page data sets
- not enough central storage
- poorly tuned paging configuration
- too many address spaces in storage
- too many "logical swap" address spaces in storage
- excessive storage isolation of address spaces
- too many extremely large address spaces resident

- paging data set on shared device
- high use of user I/O on paging volume
- "common I/O" contends with "swap I/O"
- common data set on wrong device

A high storage delay associated with **local storage paging** might be caused by one or more of the following:

- insufficient page data sets
- not enough central storage
- address space is under isolated (causing trim) or over isolated (causing others to page/swap)
- poorly tuned paging configuration
- too many address spaces in storage
- too few (artificially low) address spaces in storage
- too many "logical swap" address spaces in storage
- paging data set on shared device
- high use of user I/O on paging volume
- too much swapping
- page-ins are from trimming at swap-out
- "local I/O" contends with "swap I/O"
- program pages in each address space rather than in PLPA
- too many extremely large address spaces resident

A high storage delay associated with **virtual I/O** might be caused by one or more of the following:

- insufficient page data sets
- poorly tuned paging configuration
- paging data set on shared device
- high use of user I/O on paging volume
- virtual I/O contending with swap I/O

A high storage delay associated with **swap-in** activity might be caused by one or more of the following:

- too much swapping
- workload too heavy
- insufficient page/swap data sets
- misplaced page/swap data sets
- swap data sets on slow devices
- too few (artificially low) address spaces in storage
- paging data set on shared device
- high use of user I/O on paging volume
- swapped pages moved to backing store on cached device

- not enough central storage

A high delay value for address spaces that are **swapped out and ready** might be caused by one or more of the following:

- too few (artificially low) address spaces in storage
- workload too heavy
- unbalanced workload
- not enough central storage
- poorly tuned paging configuration
- insufficient page/swap data sets
- too many address spaces in storage
- too many or too few logical swap address spaces
- paging/swapping too slow
- exchange swap rate too high
- improper use of storage isolation

Other storage delays might be caused by one or more of the following:

- paging delays from cross-memory address spaces
- paging delays from standard hiperspaces (but not ESO hiperspaces)

Note: On the STOR and STORS reports, the **OTHR** column includes all other storage delays that are not shown in a separate column under % Delayed For (for example VIO).

execution velocity goal

The target execution velocity for ended transactions that has been in effect for the service class period during the reported range.

performance index

This index helps to compare goals. If, for example, several execution velocity goals with the same importance are not met, this index helps you decide which group was impacted the most. RMF calculates the performance index depending on the type of goal:

- **Execution velocity goal**:
  `perf index = goal% / actual%`
- **Average response time goal**:
  `perf index = actual(sec) / goal(sec)`
- **Response time goal with percentile**:
  `perf index = actual(sec) / goal(sec)`

In this context "actual" means the maximal response time that actually was reached for the percentage of the goal. To calculate this, perform the following 3 steps:

- Calculate the number of transactions N that correspond to the goal:
  `N = (sum of all transactions * goal% ) / 100`
- Add up all transactions until a bucket M is reached where the sum is greater than N.
- The "actual" response time in the formula for the performance index shown above is the response time value belonging to the bucket M.

Note: Due to this methodology, the maximal value of the performance index for this goal type is 4.

important service units (capacity) / transaction

Actual service rate (in unweighted CPU service units per second) as consumed per transaction in a resource group with a high importance (1 or 2).

percentile achieving response time goal

The percentage of transactions that actually ended within the time specified in the goal.

response time

Average response time for all transactions as reported by the CICS TOR or IMS CTL region. However, for subsystem data, it is possible that active time is greater than total time.

Note: All of these response times are for ended transactions only. Thus, if there is a problem where transactions are completely locked out, either while queued or running, the problem will not be seen until the locked-out transactions end.

queue time

Queue time is the difference between total and active time. For **CICS** , this may be the queue time for transactions within the TOR, AOR, and other regions, and also processing time within the TOR. For **IMS** , this may be the queue time for transactions within the MPR and also processing time within the CTL region. In all other cases, this is the average time that transactions spent waiting on a JES or APPC queue.

Note: Queue time may not always be meaningful, depending on how you schedule work. For example, jobs are submitted in hold status and left until they are ready to be run, all of the held time counts as queued time. This time may or may not represent a delay to the job.

transaction ended rate

The number of transactions ended per second.

active time

For **CICS** transactions, active time is the execution time in AOR, only for routed transactions.
For **IMS** transactions, active time is the execution time within the MPR.
For **Batch, TSO,** etc., active time is the average time that transactions spent in execution.

service units (capacity) / transaction

Actual service rate (in unweighted CPU service units per second) as consumed per transaction.

response time goal

The goal that has been in effect for the service class period during the reported range:

- The average target response time for all ended transactions

response time goal percentile

The goal that has been in effect for the service class period during the reported range:

- The percentage of transactions that should complete within the time specified in the goal.

service rate

The actual service rate, in unweighted CPU service units per second, as consumed by that resource group.

processor utilization

Average value of processor utilizations within the coupling facility.

In case of a stand-alone coupling facility, the utilization of the individual CPs should be approximately the same. In a PR/SM environment where this CP is shared with other partitions, the utilization is the logical utilization of the CP (that is, only the utilization by the coupling facility).

If the average utilization is high, you can take the following actions:

- In a PR/SM environment, you can dedicate the CP to the integrated coupling facility or assign additional CPs to the partition.
- Move structures to a coupling facility with lower utilization.
- Consider additional or larger coupling facilities.

# effective logical processors

Number of effective available logical processors in a shared environment. This value is only useful in a CFCC environment. CFCC measures the time of real command execution as well as the time waiting for work. The reported value shows the ratio between the LPAR dispatch time (CFCC execute and wait time) and the RMF Mintime length.

For example, if a CFCC CEC contains 6 LPs, and the measured CF LPAR has two logical processors and is limited at 5% the number of effective LPs is 0.3.

total request rate

The sum of synchronous and asynchronous requests completed against any structure within this coupling facility per second. This includes requests that changed from synchronous to asynchronous.

# frames installed

The total amount of storage in the coupling facility, including both allocated and available space.

# frames available

The amount of coupling facility space that is not allocated to any structure and not allocated as dump space

sync request rate (CF structure)

Number of hardware operations per second that started and completed synchronously to the coupling facility on behalf of connectors to the structure.

async request rate (CF structure)

Number of hardware operations per second that started and completed asynchronously to the coupling facility on behalf of connectors to the structure.

sync service time (CF structure)

Average time in microseconds required to satisfy a synchronous coupling facility request for this structure

async service time (CF structure)

Average time in microseconds required to satisfy an asynchronous coupling facility request for this structure. This value also includes operations that started synchronously but completed asynchronously.

% subchannel delay

The percentage of all coupling facility requests MVS had to delay because it found all coupling facility subchannels busy.

If this percentage is high, you should first ensure that sufficient subchannels are defined (see MAX field below).

If there are sufficient subchannels and this percentage is still high, it indicates either a coupling facility path constraint or internal coupling facility contention.

% path delay

The percentage of all coupling facility requests that were rejected because all paths to the coupling facility were busy.

A high percentage results in elongated service times which is a reduction of the capacity of the sending processor. If coupling facility channels are being shared among PR/SM partitions, the contention could be coming from a remote partition.

Identifying Path Contention: There can be path contention even when this count is low. In fact, in a non-PR/SM environment where the subchannels are properly configured, Subchannel Busy, not Path Busy, is the indicator for path contention. If Path Busy is low but Subchannel Busy is high, it means MVS is delaying the coupling facility requests and in effect gating the workload before it reaches the physical paths. Before concluding you have a capacity problem, however, be sure to check that the correct number of subchannels is defined in the I/O gen (see Subchannel Max).

PR/SM Environment: If coupling facility channels are being shared among PR/SM partitions, Path Busy behaves differently. Potentially, you have many MVS subchannels mapped to only a few coupling facility command buffers. You could have a case where the subchannels were properly configured (or even under-configured), Subchannel Busy is low, but Path Busy is high. This means the contention is due to activity from a remote partition.

Possible actions: Dedicate the coupling facility links on the sending processor or add additional links.

CF sync request rate (view from MVS image)

Number of hardware operations per second that started and completed synchronously to the coupling facility on behalf of connectors from this system.

CF async request rate (view from MVS image)

Number of hardware operations per second that started and completed asynchronously to the coupling facility on behalf of connectors from this system

CF sync service time (view from MVS image)

Average time in microseconds required to satisfy a synchronous coupling facility request.

CF async service time (view from MVS image)

Average time in microseconds required to satisfy an asynchronous coupling facility request. This value also includes operations that started synchronously but completed asynchronously.

% using for a dataset

Percentage of time when a job has had an I/O request accepted by the channel for the volume on which the data set resides, but the request is not yet complete.

% delay for a dataset

Percentage of time when a job was waiting to use the data set because of contention for the volume where the data set resides.

i/o rate

Rate of I/O requests. The i/o rate is measured at the hardware level and is the sum of the i/o activity of all systems attached to the volume or ssid.

% cache read hits

Percentage of I/Os that where processed within the cache (cache hits) based on the total number of I/Os.

- – % cache READ hits is the percentage for READ operations
- – % cache WRITE hits is the percentage for WRITE operations
- – % cache DFW hits is the percentage for DASD FAST WRITE operations
- – % cache CFW hits is the percentage for WRITE and READ-AFTER-WRITE operations.

% cache read misses

Percentage of I/Os that where NOT processed within the cache based on the total number of I/Os.

Definition: % cache read misses = 100 - % cache read hits
% cache READ misses is the percentage for READ operations
% cache WRITE misses is the percentage for WRITE operations

% of read operations

Percentage of READ requests based on all READ and WRITE requests.

non-cache dasd i/o rate

I/O rate of all requests that accessed DASD. This is the sum of Stage rates (normal or sequential I/O requests that accessed DASD) and other request rates (inhibit cache load, DFW BYPASS, CFW BYPASS, DFW INHIBIT).

CPC capacity (MSU/h)

Processor capacity available to the Central Processor Complex (CPC). The data is in Millions of unweighted CPU service units per hour.

image capacity (MSU/h)

Defined MSU capacity limit for the partition. No data are available, if the partition is not under control of the License Manager. The data is in Millions of unweighted CPU service units per hour.

% weigth of max

Average weighting factor in relation to the maximum defined weighting factor for this partition.

% WLM capping

Percentage of time when WLM capped the partition because the four-hours average MSU value exceeds the defined capacity limit.

four hour MSU average

The average CPU consumption of the partition over the last four hours measured in millions of unweighted CPU service units per hour (MSU/h).

four hour MSU maximum

The maximum CPU consumption of the partition over the last four hours measured in millions of unweighted CPU service units per hour (MSU/h). This value can be greater than the defined capacity

actual MSU

Actual MSU consumption of the image running in the specified partition. Data is in millions of unweighted CPU service units per hour.

average number of logical processors

The average number of logical processors assigned to this partition.

% effective logical dispatch time

Average effective dispatch time as percentage of the total online time.

% total logical dispatch time

Average total dispatch time as percentage of the total online time.

% effective physical utilization (CPC)

The effective utilization of the physical processors by all partitions running in the CPC.
This data is based on the total interval time of all physical processors and does not include LPAR management time. RMFPM gives you the ability to select between the sum of all CP partitions and the sum of all ICF integrated Coupling Facility) or IFL (Integrated Facility for Linux) partitions.

% effective physical utilization (partition)

The effective utilization of the physical processors by the partition.
This data is based on the total interval time of all physical processors and does not include LPAR management time. RMFPM gives you the ability to differentiate between CP partitions and ICF(integrated Coupling Facility) or IFL (Integrated Facility for Linux) partitions.

% total physical utilization (CPC)

The total utilization of the physical processors by all partitions running in the CPC. This data is based on the total interval time of all physical processors and includes LPAR management time. RMFPM gives you the ability to select between the sum of all CP partitions and the sum of all ICF integrated Coupling Facility) or IFL (Integrated Facility for Linux) partitions.

% total physical utilization (partition)

The total utilization of the physical processors by the partition.
This data is based on the total interval time of all physical processors and includes LPAR management time. RMFPM gives you the ability to differentiate between CP partitions and ICF(integrated Coupling Facility) or IFL (Integrated Facility for Linux) partitions.

% total LPAR management

The average LPAR management time percentage.

remaining time until capping in seconds (by partition)

The projected time until WLM soft capping will start. WLM soft capping takes place to prevent you from using more than the defined capacity over a long period of time. This is under the assumption you continue to use your system as you have done in the immediate past. The maximum number RMF reports is 14400 seconds or 4 hours. If RMF reports 14K, it means the remaining time until capping is at least 14K seconds.

B

IARV64 system trace entry

This appendix describes the system trace entry created in the system trace table for the high virtual storage service, IARV64. Trace entries are created for the following IARV64 services.

GETSTOR, DETACH, PAGEFIX, PAGEUNFIX, PAGEOUT, PAGEIN, DISCARDDATA, CHANGEGUARD, GETSHARED, SHAREMEMOBJ, CHANGEACCESS

The trace entry will be a system service entry (SSRV) with a new SSRV entry identifier of x'14B' to identify it as an IARV64 trace entry. An additional 1-byte field within the trace entry unique fields will identify the IARV64 service for which the entry was created.

B.1 The SSRV trace entry

Trace entry

The following information is provided in each trace entry:

- **WORD 1:**
  - Request type identifier - 1 byte
    - x'01' = GETSTOR
    - x'02' = GETSHARED
    - x'03' = DETACH
    - x'04' = PAGEFIX
    - x'05' = PAGEUNFIX
    - X'06' = PAGEOUT
    - X'07' = DISCARDDATA
    - X'08' = PAGEIN
    - x'0A' = SHAREMEMOBJ
    - X'0B' = CHANGEACCESS
    - X'0D' = CHANGEGUARD
  - Request Flags (1 byte) - Flags specific to the IARV64 request type (see description under Request flags).
  - Keys Used flag (1 byte) - Keys-used flag from the IARV64 parm list:
    - X'80' - KEY specified
    - X'40' - USERTKN specified
    - X'20' - TTOKEN specified
    - X'10' - CONVERTSTART specified
    - X'08' - GUARDSIZE64 specified
    - X'04' - CONVERTSIZE64 specified
  - Misc Byte (1 byte)
    - Storage key for GETSTOR and GETSHARED requests
    - Number of ranges in range list for requests that deal with a range list
    - 0 for all other requests

WORD 2:

- Return Code/Abend Code (4 bytes)

WORD 3:

- Reason Code (4 bytes)

WORD 4:

- ALET specified on the IARV64 request (4 bytes)

WORDS 5 - 10:

- Additional 'variable' fields depending on the IARV64 service (see description under Variables).

Request flags

The following is a summary of the flags traced in the Request Flags byte:

- GETSTOR / GETSHARED
  - X'80' - COND=YES request
  - X'40' - FPROT=NO request
  - X'20' - CONTROL=AUTH request (only applies to GETSTOR)
  - X'10' - SVCDUMPRGN=NO request (only applies to GETSTOR)
  - X'08' - CHANGEACCESS = GLOBAL request (only applies to GETSHARED)
  - X'04' - GUARDLOC=HIGH request (only applies to GETSTOR)
- DETACH
  - X'80'- COND=YES request
  - X'40 - MATCH=USERTOKEN request
  - X'20'- AFFINITY=SYSTEM request
  - X'10'- OWNER=NO request
- SHAREMEMOBJ
  - X'80' - COND=YES request
  - X'40' - SVCDUMPRGN=NO request
- CHANGEGUARD
  - X'80' - COND=YES request
  - X'40' - TOGUARD request
  - X'20' - FROMGUARD request
- PAGEFIX
  - X'80' - LONG=NO request
- DISCARDDATA
  - X'80' - CLEAR=NO request
- CHANGEACCESS
  - X'80' - READONLY request
  - X'40' - SHAREDWRITE request
  - X'20' - HIDDEN request

Variables

The following is a summary of the variable data that will be traced for each IARV64 service:

- GETSTOR / GETSHARED
  - Origin address of the memory object - 8 bytes
  - Size of the memory object in segments - 8 bytes
  - User token - 8 bytes
- DETACH
  - Memory object start address (for MATCH=SINGLE) requests (zeroes for MATCH=USERTOKEN requests) - 8 bytes
  - User token - 8 bytes

- PAGEFIX, PAGEUNFIX, PAGEOUT, PAGEIN, DISCARDDATA, CHANGEACCESS
  - Address of rangelist - 8 bytes
  - VSA from 1st range list entry
  - # of blocks from 1st range list entry
- CHANGEGUARD
  - Memory object start (if ConvertStart not specified), or convert start address (if ConvertStart was specified) - 8 bytes
  - Number of segments to be converted - 8 bytes
- SHAREMEMOBJ
  - Range list address - 8 bytes
  - VSA from 1st range list entry
  - User token - 8 bytes

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 500. 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

Other publications

These publications are also relevant as further information sources:

- *z/OS MVS Planning: Operations*, SA22-7601
- *ServerPac: Using The Installation Dialog*, SA22-7815
- *z/OS and z/OS.e Planning for Installation*, GA22-7504
- *z/OS MVS Programming: Workload Management Services*, SA22-7619
- *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- *z/OS JES2 Commands,* SA22-7526
- *z/OS JES2 Diagnosis*, SA22-7531
- *z/OS JES2 Initialization & Tuning Guide,* SA22-7532
- *z/OS JES2 Initialization & Tuning Reference*, SA22-7533
- *z/OS JES2 Installation Exits*, SA22-7534
- *z/OS JES2 Introduction*, SA22-7535
- *z/OS JES2 Macros*, SA22-7536
- *z/OS JES2 Messages*, SA22-7537
- *z/OS JES2 Migration*, GA22-7538
- *z/OS Planning for Multilevel Security*, GA22-7509
- *z/OS SDSF Operation and Customization*, SA22-7670
- *z/OS Resource Measurement Facility (RMF) Report Analysis*, SC33-7991
- *z/OS Resource Measurement Facility (RMF) User’s Guide*, SC33-7990
- *z/OS Resource Measurement Facility (RMF) Programmer’s Guide*, SC33-7994
- *z/OS MVS Assm Services Reference IAR-XCT*, SA22-7607
- *z/OS UNIX System Services Planning*, GA22-7800
- *z/OS Security Server RACF Security Administrator’s Guide,* SA22-7683
- *z/OS Planning for Multilevel Security,* GA22-7509

- *z/OS UNIX System Services Programming: Assembler Callable Services Reference*, SA22-7803
- *z/OS MVS Auth Assembler Services Guide,* SA22-7608
- *MVS Programming: Authorized Assembler Services Guide,* SA22-7608
- *z/OS ISPF Dialog Developer’s Guide and Reference*, SC34-4821
- *z/OS ISPF Dialog Tag Language Guide and Reference*, SC34-4824
- *z/OS DFSMS Migration*, GC26-7398
- *z/OS Integrated Security Services Enterprise Identity Mapping (EIM) Guide and Reference*, SA22-7875
- *z/OS Security Server LDAP Client Programming*, SC24-5924
- *z/OS ISPF Dialog Developer’s Guide and Reference*, SC34-4821
- *z/OS ISPF User’s Guide Volume II*, SC34-4823
- *z/OS ISPF Planning and Customizing*, SC34-4814
- *z/OS ISPF Software Configuration and Library Manager (SCLM) Reference*, SC34-4818
- *z/OS MVS Program Management: Advanced Facilities*, SA22-7644
- *Support for Unicode: Using Conversion Services,* SA22-7649
- *z/OS MVS Programming Workload Management Services*, SA22-7619
- *z/OS C/C++ Run-Time Library Reference*, SA22-7821
- *z/OS ISPF Services Guide*, SC34-4819
- *z/OS Cryptographic Services PKI Services Guide and Reference,* SA22-7693
- *z/OS OCSF Applications Programming*, SC24-5899
- *z/OS Introduction and Release Guide,* GA22-7502
- *z/OS Language Environment Vendor Interfaces*, SA22-7568
- *DB2 Universal Database for z/OS: RACF External Security Module Guide and Reference*, SA22-7938
- *Security Server LDAP Server Administration and Use*, SC24-5923
- *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693
- *z/OS Communications Server IP CICS Sockets Guide*, SC31-8807
- *z/OS Communications Server: IP Configuration Reference*, SC31-8775

Online resources

These Web sites and URLs are also relevant as further information sources:

- Document available on the web that describes what’s new in DFSORT for z/OS V1R5:

  `http://www.storage.ibm.com/software/sort/mvs/release_14/pdf/sortnew.pdf`

- To see debugging tools

  `http://www-3.ibm.com/software/awdtools/debugtool/`

- To access Infoprint Central from a workstation

  `http://wtsc65oe.itso.ibm.com/Infoprint/En_US/IPS.html`

- To access information on the Subcapacity Reporting Tool (SCRT), Sub-capacity eligible products and the associated pricing strategies refer to::

http://www-1.ibm.com/servers/eserver/zseries/swprice/

- For more information on eWLM as the emerging multi-tied workload strategy refer to *The Great Balancing Act*, available on the IBM thinkresearch Web site at:

  http://www.research.ibm.com/thinkresearch/pages/2002/20020529_ewlm.shtml

- The RMF Spreadsheet Reporter is available for download via the RMF homepage:

  http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/

- The RMF Spreadsheet Reporter Version 5.1 is available for download from the RMF tools website:

  http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/rmftools.htm#spr_win

- The most recent version of RMF PM can be downloaded from the RMF PM wesbite at:

  http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmweb.htm

- There is an excellent summary of metrics in RMF PM that are available to monitor your system and your sysplex resources at the following website:

  http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pm_metrics.htm

- There is an excellent document available on the web that describes what's new in DFSOR. Ity can be found at:

  http://www.storage.ibm.com/software/sort/mvs/release_14/pdf/sortnew.pdf

- To access the DB2 information center from a workstation

  http://publib.boulder.ibm.com/infocenter/db2help/index.jsp

- Information on DB2 V8 may be found at:

  http://www.ibm.com/software/data/db2/os390/db2zosv8.html

- Additional information on DB2's row-level SECLABEL processing may be found at:

  ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/db2security.pdf.

- An informative DB2 V8 web-audio presentation may be found at:

  http://www-3.ibm.com/software/os/zseries/zserieswebcasts/030130/

- The Unicode collation algorithm is described in detail in the Unicode Consortium's technical report #10. For the detail report, refer to URL:

  http://www.unicode.org/unicode/reports/tr10/

- Allkeys.txt Unicode file can be found at URL:

  http://www.unicode.org/unicode/reports/tr10/allkeys.txt

- For a detailed explanation of normalization, including specific information about the normalization forms, refer to The Technical Report #15 provided by the Unicode Consortium

  http://www.unicode.org/unicode/reports/tr15/

- Unicode consortium Collation Technical Report:

  http://www.unicode.org/reports/tr10/

- Internet Engineering Task Force (IETF) Web site for RFC2459:

  http://www.ietf.org

-

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

z/OS Version 1 Release 5 Implementation

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

z/OS Version 1 Release 5 Implementation

Redbooks

BCP, JES3, JES2, SDSF, RMF, Communications Server, Consoles

Infoprint Server, ISPF, WLM, PSF for z/OS

z/OS UNIX, RACF, SMP/E, ServerPac

z/OS Version 1 Release 5 offers a number of enhancements that improve availability, scalability and performance, application flexibility, and ease of use. In this IBM Redbook, we describe these functional enhancements and provide information to help you install, tailor, and configure this release.

After giving an overview of this release, we cover the enhancements made to the following components:

- ServerPac
- Base Control Program (BCP)
- JES3
- JES2
- SDSF
- Infoprint Server
- ISPF
- Workload Manager (WLM)
- Console restructure
- RMF
- SMP/E for z/OS and OS/390
- UNIX System Services (USS)
- z/OS Security Server RACF
- PSF 3.4.0 for z/OS
- Communication Server for z/OS V1R5

We also provide RMF Performance Monitor metrics and describe the system trace entry created in the system trace table for the high virtual storage service IARV64.

This redbook is intended for systems programmers and administrators responsible for customizing, installing, and migrating to these newest levels of z/OS.

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

SG24-6326-00 ISBN 0738491411