

ABCs of IBM z/OS

System Programming

Volume 1

Get an introduction to z/OS and storage concepts

Learn about TSO/E, ISPF, JCL, and SDSF

Understand z/OS delivery and installation



Karan Singh
Paul Rogers

Redbooks



International Technical Support Organization

ABCs of IBM z/OS System Programming Volume 1

August 2014

Note: Before using this information and the product it supports, read the information in "Notices" on page ix.

Fourth Edition (August 2014)

This edition applies to version 2 release 1 modification 0 of IBM z/OS (product number 5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2003, 2014. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
Preface	xi
Authors	xii
Now you can become a published author, too!	xiii
Comments welcome	xiii
Stay connected to IBM Redbooks	xiii
Chapter 1. Introduction to IBM z/OS	1
1.1 z/OS services	2
1.2 IBM server and operating system evolution	4
1.3 z/OS operating systems and hardware	6
1.4 Hardware requirements	8
1.5 IBM System z Server: Basic mode and logically partitioned mode	9
1.6 z/OS V2R1 base elements	10
1.7 z/OS optional features	17
1.8 z/OS Base Control Program	20
1.9 z/OS Security Server: RACF component	21
1.10 Data Facility Storage Management Subsystem	23
1.11 DFSMS Advanced Copy Services	25
1.12 z/OS Communications Server: TCP/IP	27
1.13 System Modification Program Extended	29
1.14 Time Sharing Option/Extended	31
1.15 z/OS UNIX System Services	33
1.16 z/OS UNIX physical file systems	35
1.17 System Management Facility	37
1.18 Resource Management Facility	39
1.19 Workload Manager	41
1.20 Infoprint Server	43
1.21 IBM z/OS Management Facility	45
1.22 z/OSMF Configuration Assistant task	48
1.23 z/OSMF Capacity Provisioning task	49
1.24 z/OSMF System Status and Resource Monitoring task	50
1.25 z/OSMF Workload Management task	52
1.26 z/OSMF Incident Log task	54
1.27 z/OSMF ISPF task	56
1.28 z/OSMF Workflows task	57
1.29 IBM Health Checker for z/OS	59
1.30 z/OS release cycle	61
Chapter 2. TSO/E, ISPF, JCL, and SDSF	63
2.1 z/OS facilities for system programmers	64
2.2 TSO/E	65
2.3 TSO/E highlights	66
2.4 TSO/E customization	69
2.5 TSO/E: Terminal control address space start procedure	71
2.6 TSO/E logon procedure	72

2.7 TSO/E logon process in a VTAM environment	74
2.8 TSO/E full-screen logon panel	75
2.9 TSO/E line-mode.	77
2.10 Using TSO/E as batch job	78
2.11 TSO/E Profile command	79
2.12 TSO/E languages	81
2.13 Interactive System Productivity Facility	83
2.14 ISPF: Data set types supported	85
2.15 ISPF Option 2	87
2.16 ISPF: Other Sequential data set	89
2.17 ISPF components	91
2.18 Sample CLIST to allocate ISPF and SDSF data sets	93
2.19 ISPF primary option menu	95
2.20 ISPF panel areas	97
2.21 Action bars	98
2.22 Customizing a TSO/ISPF/PDF session	100
2.23 Customizing an ISPF session	101
2.24 Allocating data sets: Utility option	102
2.25 Utility Selection Panel	103
2.26 Data set Utility: Allocating a data set	104
2.27 Allocate New Data Set panel	106
2.28 Edit profile for a user	109
2.29 Edit function: Option 2	111
2.30 Edit Entry panel	112
2.31 Editing a data set	113
2.32 ISPF edit: Various line commands	115
2.33 ISPF edit: Repeating and deleting lines	117
2.34 ISPF edit: Copying lines	118
2.35 ISPF/PDF edit: Primary commands	120
2.36 ISPF/PDF edit: Saving new or updated files	121
2.37 ISPF Data Set List Utility option	122
2.38 Working with a data set list	124
2.39 Job control language	126
2.40 JCL introduction	127
2.41 JCL-related actions	129
2.42 Required control statements	130
2.43 JCL streams and jobs	131
2.44 JES control statements in JCL	133
2.45 Introduction to JCL: Creating a data set	134
2.46 JCL: JOB statement	136
2.47 JCL: EXEC statement	138
2.48 JCL: DD statement	140
2.49 Submitting a job	144
2.50 System Display and Search Facility	145
2.51 SDSF: Panels hierarchy	146
2.52 JES2 SDSF Primary Option Menu	147
2.53 SDSF: JES3 panel hierarchy	148
2.54 JES3 SDSF Primary Option Menu	149
2.55 SDSF: Options menu	150
2.56 SDSF: viewing the output files	151
2.57 SDSF filters for panel displays	154
2.58 SDSF: Display Active Users (DA command)	157
2.59 Issuing MVS and JES commands	159

2.60 SDSF: Input queue panel	161
2.61 SDSF: Output queue panel	163
2.62 SDSF: Held Output queue panel	165
2.63 SDSF: Status panel	167
2.64 SDSF HELP panel	169
Chapter 3. The z/OS system programmer	171
3.1 z/OS Operating System: The role of a system programmer	172
3.2 z/OS system programmer management overview	174
3.3 The system programmer and z/OS operations	176
3.4 Requirements for z/OS installation	179
3.5 z/OS system with SMP/E	181
3.6 z/OS delivery options	185
3.7 Choosing an installation package	187
3.8 ServerPac Internet delivery	190
3.9 System and installation requirements	192
3.10 Reviewing your current system	193
3.11 Driving system and target system	195
3.12 z/OS installation using ServerPac	197
3.13 The RIM tape samples	199
3.14 Starting the CustomPac dialogs	200
3.15 Types of data sets and their volumes	201
3.16 Receiving the ServerPac order	203
3.17 Order Receive panel	204
3.18 Receive an order from tape	206
3.19 Edit JOB Statement panel	207
3.20 Panel with the job statement	208
3.21 Edit RECEIVE Job panel	209
3.22 CustomPac Order Management Menu panel	211
3.23 Selecting an order to install	212
3.24 Installation dialog	213
3.25 Choosing the installation type	214
3.26 JES and SDSF element selection	215
3.27 CREATE Configuration panel	217
3.28 Installation Variables panel	218
3.29 Define ZONE Information panel	220
3.30 Modify System Layout Options panel	222
3.31 SUMMARY of Features/Elements panel	224
3.32 Summary of data sets	225
3.33 Creating a Recommended System Layout	226
3.34 Summary of Physical Volumes panel	228
3.35 Current Volume Configuration panel	230
3.36 Display and change volume attributes panel	232
3.37 Define alias-to-catalog relationships	233
3.38 Define system-specific alias (SSA)	236
3.39 Define SSA and CATALOG Data panel	238
3.40 Job Selection List panel	240
3.41 GENERATE File Tailored Installation Jobs panel	242
3.42 List of installation jobs	244
3.43 Displaying the processing log	245
3.44 SAVE Configuration panel	246

Chapter 4. z/OS storage concepts	247
4.1 Processor storage overview	248
4.2 Virtual storage concepts	250
4.3 Frames, slots, and pages	252
4.4 z/Architecture address space	254
4.5 The address space concept	256
4.6 Addressing mode and residence mode	258
4.7 Storage managers	260
4.8 Virtual storage manager	262
4.9 Real storage manager	264
4.10 Auxiliary storage manager	265
4.11 Paging and swapping	266
4.12 Auxiliary page data sets	268
4.13 System z Flash Express	270
4.14 Storage-class memory on System z Flash Express	271
4.15 31-bit address space map	272
4.16 The common virtual storage area	274
4.17 z/OS nucleus	276
4.18 System queue area	278
4.19 Common service area	280
4.20 Link pack area	282
4.21 31-bit address space private area	284
4.22 Data spaces and IBM Hiperspace	286
4.23 64-bit address space map	288
4.24 Size and number notation in bytes	291
4.25 Segment tables and page tables in 31-bit addressing	292
4.26 Program status word format	294
4.27 31-bit virtual address	298
4.28 64-bit virtual address translation	300
4.29 Translating a 64-bit virtual address	302
4.30 z/OS 1 MB and 2 GB Large Pages	303
4.31 System initialization (IPL process)	304
4.32 z/OS address spaces	306
4.33 Subsystem definitions	308
4.34 Multiprogramming and multiprocessing	310
4.35 Program compile, link-edit, and execution	312
4.36 Library lookaside	313
4.37 Virtual lookaside facility	315
4.38 Memory hierarchy	316
Chapter 5. ShopzSeries	319
5.1 IBM software ShopzSeries	320
5.2 ShopzSeries order process	321
5.3 Specify order basics step 1 of 8	322
5.4 Select hardware systems step 2 of 8	323
5.5 Report installed software step 3 of 8	324
5.6 Shop for products step 4 of 8	325
5.7 Shop catalog	326
5.8 Specify order contents step 5 of 8	327
5.9 Select new licenses step 6 of 8	328
5.10 Specify delivery options step 7 of 8	329
5.11 Review and submit order step 8 of 8	331
5.12 In process orders	332

5.13 Finished order	333
5.14 Download order information	334
5.15 Download instructions.....	335
Chapter 6. z/OS maintenance concepts	337
6.1 Aspects of software management.....	338
6.2 Software management tasks.....	340
6.3 The z/OS software management cycle	342
6.4 How current your should software be	343
Related publications	345
IBM Redbooks	345
Other publications	345
Online resources	346
How to get IBM Redbooks	346
Help from IBM	346

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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

AIX®	IMS™	S/390®
BladeCenter®	IMS/ESA®	ServicePac®
BookManager®	Language Environment®	System p®
CICS®	Lotus®	System Storage®
DB2®	MVS™	System x®
Domino®	NetView®	System z®
DS8000®	OS/390®	System z10®
Enterprise Storage Server®	Parallel Sysplex®	System z9®
eServer™	PR/SM™	Tivoli®
FFST™	Print Services Facility™	VTAM®
FICON®	PrintWay™	WebSphere®
First Failure Support Technology™	ProductPac®	z/Architecture®
FlashCopy®	RACF®	z/OS®
GDPS®	Redbooks®	z/VM®
Geographically Dispersed Parallel Sysplex™	Redbooks (logo)  ®	z10™
Hiperspace™	Resource Measurement Facility™	z9®
IBM®	RETAIN®	zEnterprise®
	RMF™	

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds 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.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Other company, product, or service names may be trademarks or service marks of others.

Preface

The ABCs of IBM® z/OS® System Programming is a 13-volume collection that provides an introduction to the z/OS operating system and the hardware architecture. Whether you are a beginner or an experienced system programmer, the ABCs collection provides the information that you need to start your research into z/OS and related subjects.

Whether you want to become more familiar with z/OS in your current environment, or you are evaluating platforms to consolidate your online business applications, the ABCs collection will serve as a powerful technical tool.

Volume 1 provides an updated understanding of the software and IBM zSeries architecture, and explains how it is used together with the z/OS operating system. This includes the main components of z/OS needed to customize and install the z/OS operating system. This edition has been significantly updated and revised.

The other volumes contain the following content:

- ▶ Volume 2: z/OS implementation and daily maintenance, defining subsystems, IBM Job Entry Subsystem 2 (JES2) and JES3, link pack area (LPA), LNLST, authorized libraries, System Modification Program/Extended (SMP/E), IBM Language Environment®
- ▶ Volume 3: Introduction to Data Facility Storage Management Subsystem (DFSMS), data set basics, storage management hardware and software, catalogs, and DFSMS Transactional Virtual Storage Access Method (VSAM), or DFSMStvs
- ▶ Volume 4: z/OS Communications Server, Transmission Control Protocol/Internet Protocol (TCP/IP), and IBM Virtual Telecommunications Access Method (IBM VTAM®)
- ▶ Volume 5: Base and IBM Parallel Sysplex®, z/OS System Logger, Resource Recovery Services (RRS), Global Resource Serialization (GRS), z/OS system operations, z/OS Automatic Restart Manager (ARM), IBM Geographically Dispersed Parallel Sysplex™ (IBM GDPS®)
- ▶ Volume 6: Introduction to security, IBM Resource Access Control Facility (IBM RACF®), Digital certificates and public key infrastructure (PKI), Kerberos, cryptography and IBM eServer™ z990 integrated cryptography, zSeries firewall technologies, Lightweight Directory Access Protocol (LDAP), and Enterprise Identity Mapping (EIM)
- ▶ Volume 7: Printing in a z/OS environment, Infoprint Server, and Infoprint Central
- ▶ Volume 8: An introduction to z/OS problem diagnosis
- ▶ Volume 9: z/OS UNIX System Services
- ▶ Volume 10: Introduction to IBM z/Architecture®, zSeries processor design, zSeries connectivity, LPAR concepts, HCD, and IBM DS8000®
- ▶ Volume 11: Capacity planning, IBM Performance Management, z/OS Workload Manager (WLM), IBM Resource Management Facility (IBM RMF™), and IBM System Management Facility (SMF)
- ▶ Volume 12: WLM
- ▶ Volume 13: JES2 and JES3 System Display and Search Facility (SDSF)

Authors

This book was produced by a team of specialists from around the world working at the IBM International Technical Support Organization (ITSO), Poughkeepsie Center.

Karan Singh is a Project Leader at the ITSO, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of z/OS.

Paul Rogers, now retired, was a Consulting IT Specialist at the ITSO, Poughkeepsie Center. He wrote extensively and taught IBM classes worldwide on various aspects of z/OS, z/OS UNIX System Services, Infoprint Server, and JES3. Before joining the ITSO 23 years ago, Paul worked in the IBM Installation Support Center (ISC) in Greenford, England, providing IBM OS/390® and JES support for IBM Europe, Middle East, and Africa (EMEA) and the Washington Systems Center. He worked for IBM for 44 years.

Thanks also to the following contributors to this edition:

Richard Conway, Robert Haimowitz

ITSO, Poughkeepsie Center

Thanks to the authors of the second edition of this book:

Redelf Janssen is an IT Architect in IBM System z9® Technical Sales, Bremen, Germany. He holds a degree in computer science from the University of Bremen and joined IBM in 1988. He is responsible for supporting IBM System z9 clients in Germany. His areas of expertise include IBM System z9, z/OS, storage management, and availability management. Redelf has co-authored IBM Redbooks® publications about several OS/390 and z/OS releases, in addition to *ABCs of IBM z/OS System Programming*.

Alvaro Salla is an IBM retiree who worked for IBM for more than 30 years, specializing in large systems. He has co-authored many IBM Redbooks publications and spent many years teaching about large systems from S/360 to IBM S/390®. He has a degree in Chemical Engineering from the University of Sao Paulo, Brazil.

Thanks to the authors of the first edition of this book, *ABCs of IBM z/OS System Programming Volume 1*, published in December 2003:

Miriam Gelinski is a staff member of Maffei Consulting Group, where she is responsible for supporting customer planning and installing zSeries software. She holds a Bachelor's degree in information systems from the Universidade São Marcos. Before joining the Maffei Consulting Group, Miriam worked for the IBM zSeries brand for two years, where she was responsible for implementing new workloads on zSeries.

Joao Natalino Oliveira is a Certified I/T Consulting Specialist for IBM zSeries in Brazil, where he provides support for Brazil and Latin America. He has 28 years of experience in large systems, including IBM Multiple Virtual Storage (IBM MVS™), OS/390, and z/OS. His areas of expertise include performance and capacity planning, server consolidation, and system programming. Joao holds a Bachelor's degree in Mathematics And Data Processing from Fund. Santo Andre, Brazil.

Alvaro Salla (see previous biography).

Valeria Sokal is an MVS system programmer at an IBM client. She has 14 years of experience as a mainframe system programmer.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run 2 - 6 weeks in length, and you can participate either in person or as a remote resident working from your home base.

Obtain more information about the residency program, browse the residency index, and apply online at the following website:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review Redbooks form:
ibm.com/redbooks
- ▶ Send your comments in an email:
redbooks@us.ibm.com
- ▶ Mail your comments:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>



Introduction to IBM z/OS

IBM z/OS is an integrated enterprise server operating system (OS). It incorporates into one product a leading-edge and open communications server, distributed data and file services, IBM Parallel Sysplex system support, object-oriented programming, distributed computer environment, and an open application interface. As such, it is uniquely suited to integrate today's heterogeneous and multivendor environments.

Although the official product name of the OS is z/OS, the former product name, IBM Multiple Virtual Storage (MVS), is still used when describing any aspect of the OS. By incorporating the base OS, z/OS continues to build on the classic strengths of MVS: Reliability, continuous availability features, and security. This provides a scalable system that supports massive transaction volumes, large numbers of users with high performance, advanced system and network management, security, and continuous availability.

Businesses are relying on mainframe servers to power their transformation into on-demand enterprises. Together, IBM System z® and z/OS deliver industry-leading capabilities designed to help reduce IT complexity, increase business flexibility, and drive down costs. The IBM mainframe is ready to help you use, extend, and integrate core business applications. It is positioned as a leading platform to manage and integrate your IT operating environment.

With middleware and tools to complete the system, you can begin to make on-demand business a reality by using the most important asset you have: Your z/OS mainframe. From automation to advanced virtualization technologies, with open and industry standards, z/OS can help deliver competitive advantages for an on-demand business.

This chapter presents an overview of the z/OS operating system, and includes the following topics:

- ▶ z/OS operation system evolution
- ▶ Hardware required to install and run z/OS
- ▶ z/OS base and optional products, with various brief descriptions
- ▶ z/OS products requiring customization
- ▶ System programmer skills needed to install and maintain the z/OS operating system
- ▶ Requirements to install z/OS
- ▶ Installation package delivery options

1.1 z/OS services

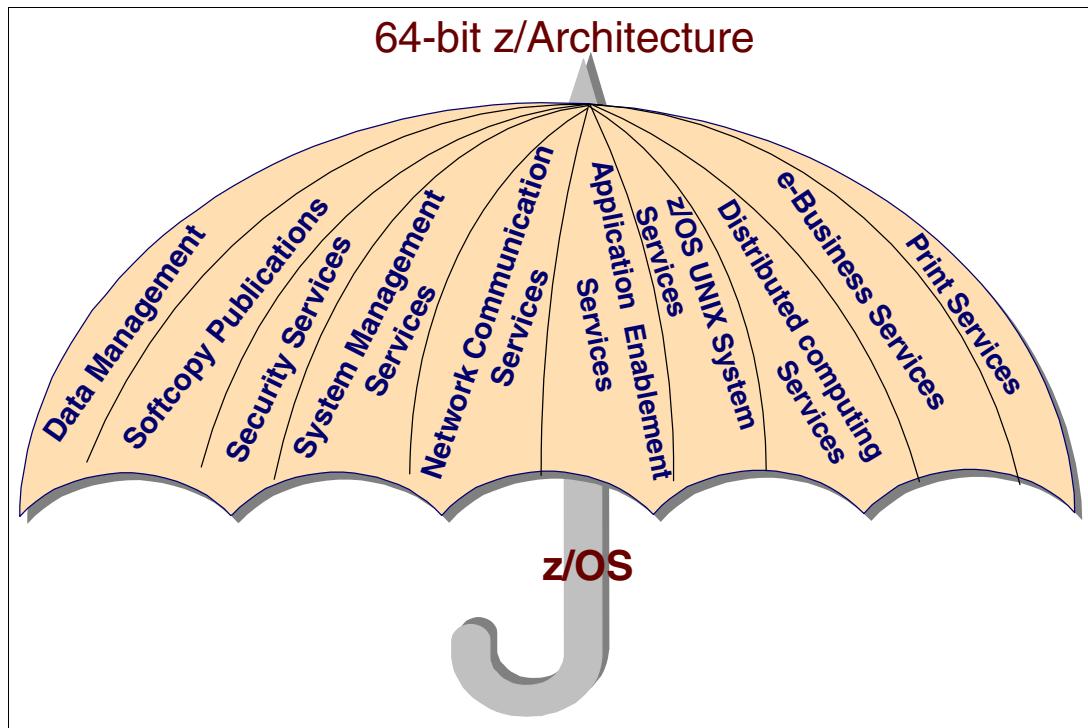


Figure 1-1 z/OS services

z/OS services

z/OS, as an operating system, provides program management services that let you create, load, modify, list, read, and copy executable programs.

The z/OS system provides solutions for the following major areas:

- ▶ **Data management:** z/OS provides a set of functions to support the following functions:
 - Manage storage resources on the system.
 - Support storage and retrieval of data on disk, optical, and tape devices.
 - Offer program management functions.
 - Supply device management functions to define and control the operation of input and output storage devices.Distributed FileManager (DFM) supports access to remote data and storage resources.
- ▶ **Softcopy publications services:** These services improve productivity in systems installation and management.
- ▶ **Security services:** Security and Cryptographic Services are a set of products and features used to control access to resources, and to audit and manage the accesses with appropriate centralized or decentralized control. These services form the basis for all security services for traditional applications, UNIX applications, and distributed systems.
- ▶ **System management services:** The functions and features provided with z/OS support robust control and automation of the basic processes of z/OS, therefore increasing availability, improving productivity of system programmers, and providing a consistent approach for configuring z/OS components of products.

- ▶ **Network communication services:** z/OS enables world-class Transmission Control Protocol/Internet Protocol (TCP/IP) and Systems Network Architecture (SNA) networking support, multivendor and multiplatform connectivity, connectivity to a broad set of users, and support for multiple protocols.
- ▶ **Applications enablement services:** These services provide a solid infrastructure in which you can build new applications, extend existing applications, and run online transaction processing (OLTP) and batch processes.
- ▶ **z/OS UNIX System Services:** z/OS contains the UNIX applications services (shell, utilities, and debugger) and the UNIX System Services (kernel and runtime environment). The shell and utilities provide the standard command interface familiar to interactive UNIX users. z/OS includes all the commands and utilities specified in the X/Open Portability Guide (XPG) 4.2.

With IBM Language Environment, z/OS supports industry standards for C programming, shell and utilities, client/server applications, and the majority of the standards for thread management, therefore enabling transparent data exchange and easy portability of applications in an open environment.

- ▶ **Distributed computing services:** These services are achieved by a set of features and functions. z/OS Network File System (NFS) acts as a file server to workstations, personal computers, or other authorized systems in an Internet Protocol network. Remote files are mounted from the mainframe (z/OS) to appear as local directories and files on the client system. DCE enables Data Encryption Standard (DES) algorithms and the commercial data masking facility (CDMF).

Distributed File Services (DFS) system-managed buffering (SMB) enables users to access data in a distributed environment across a wide range of IBM and non-IBM platforms. SMB can automatically handle the conversion between American Standard Code for Information Interchange (ASCII) and Extended Binary Coded Decimal Interchange Code (EBCDIC).

- ▶ **Online business services:** The IBM Hypertext Transfer Protocol (HTTP) Server provides for scalable, high-performance web serving for critical online business applications. It is exclusive to z/OS. This element was previously known as a base element of z/OS under the names IBM Lotus® Domino® Go, the Internet Connection Secure Server (ICSS), and the Internet Connection Server (ICS).
- ▶ **Print services:** Application output can be electronically distributed and printed, or presented over the web.

z/OS functional enhancements

z/OS Version 2 Release 1 was introduced as a replacement for the last release of z/OS Version 1. There were 13 releases of z/OS Version 1.

z/OS Version 1 Release 1 was introduced as a replacement for the last release of OS/390. There were ten releases of OS/390, with a new release issued every six months.

This book is based on z/OS Version 2 Release 1. This release was made generally available in September 2013.

z/OS Version 2 Release 1 (program number 5650-ZOS), is an OS designed to meet the on-demand challenges of the online business world. z/OS delivers the highest qualities of service for enterprise transactions and data, and extends these qualities to new applications using the latest software technologies.

1.2 IBM server and operating system evolution

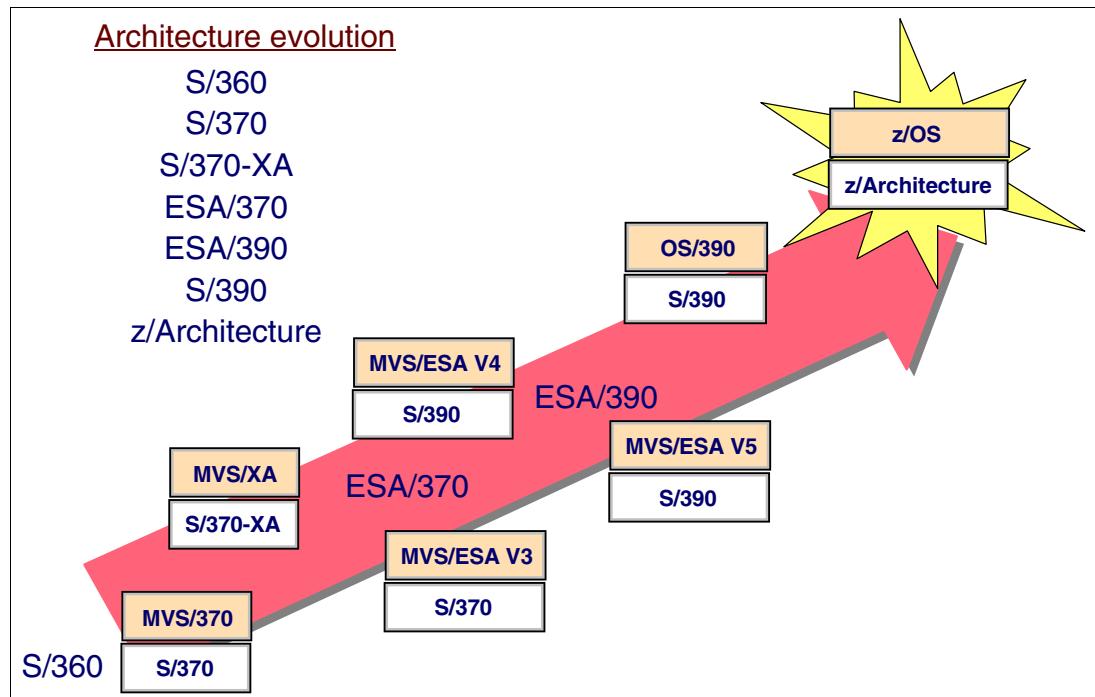


Figure 1-2 Architecture and operating system evolution

Software and hardware evolution

Enterprise system hardware was transformed by the introduction of the z/OS Parallel Servers. Based on the complementary metal oxide semiconductor (CMOS) technology, these parallel systems are smaller in size and larger in capacity than their predecessors, and deliver performance at a lower cost.

With the introduction of the IBM S/390 Parallel Servers in 1994, the mainframe was revived. The CMOS technology offers mainframe computing power at a lower cost. The zSeries CMOS systems can be connected with other zSeries or S/390 CMOS systems, or even with traditional ES/9000 systems, to form a Parallel Sysplex. The sysplex offers high availability (HA) and the option to add capacity in small increments.

The transformation of mainframe hardware, together with business requirements for information technology (IT), are the driving forces behind changes in the OS software. In 2001, the z/OS system was introduced as the zSeries server OS. z/OS extends the S/390 architecture to provide the enterprise-wide client/server infrastructure and tools that businesses need for fast, flexible deployment of new applications.

z/Architecture

IBM z/Architecture is the next step in the evolution from the IBM System/360 to the IBM System/370, IBM System/370 extended architecture (370-XA), IBM Enterprise Systems Architecture/370 (ESA/370), and IBM Enterprise Systems Architecture/390 (ESA/390).

z/Architecture includes all of the facilities of ESA/390 except for the asynchronous-pageout, asynchronous-data-mover, program-call-fast, and vector facilities.

z/Architecture also provides significant extensions, as follows:

- ▶ It provides 64-bit general registers and control registers.
- ▶ A 64-bit addressing mode, in addition to the 24-bit and 31-bit addressing modes of ESA/390, which are carried forward to z/Architecture.

Both operand addresses and instruction addresses can be 64-bit addresses. The program status word (PSW) is expanded to 16 bytes to contain the larger instruction address. The PSW also contains a newly assigned bit that specifies the 64-bit addressing mode.

- ▶ Up to three additional levels of dynamic address translation (DAT) tables, called *region tables*, for translating 64-bit virtual addresses.

A virtual address space can be specified either by a segment-table designation, as in ESA/390, or by a region-table designation. Either of these types of designation is called an *address-space-control element* (ASCE). An ASCE can alternatively be a real-space designation that causes virtual addresses to be treated as real addresses without the use of DAT tables.

- ▶ An 8 kilobyte (KB) prefix area for containing larger old and new PSWs and register save areas.
- ▶ Many new instructions, many of which operate on 64-bit binary integers.

Information: For a more detailed description of z/Architecture, see *ABCs of z/OS System Programming Volume 10*, SG24-6990.

1.3 z/OS operating systems and hardware

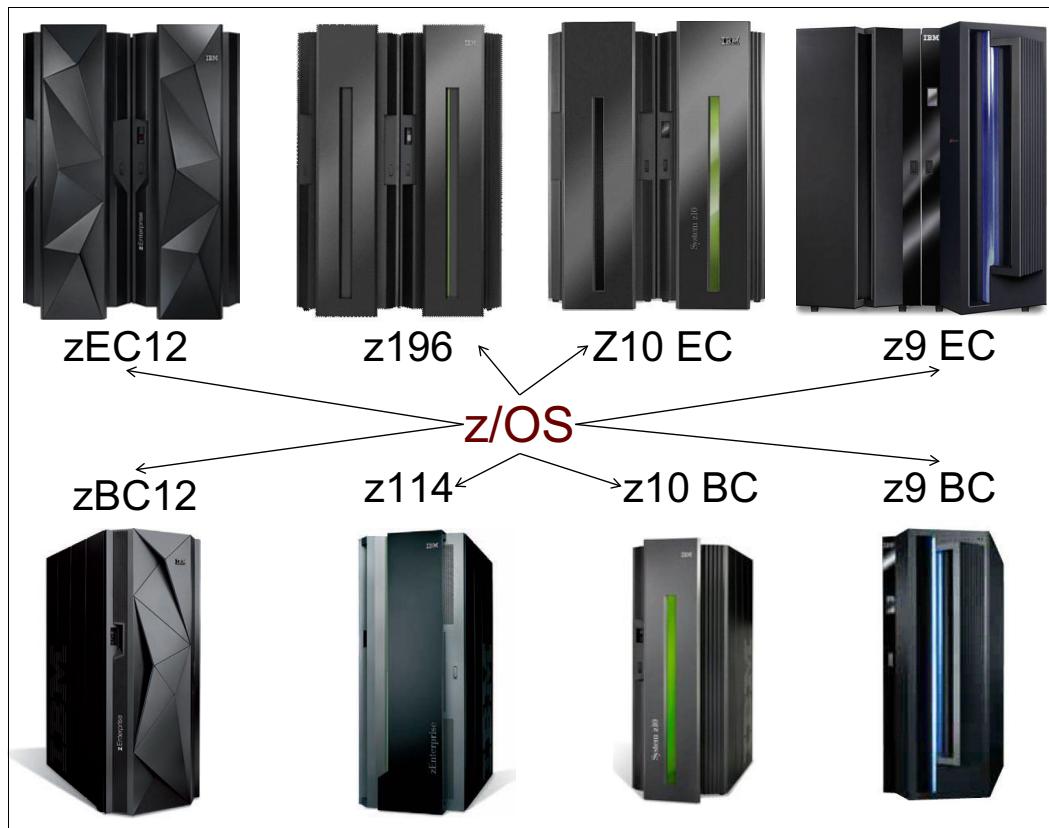


Figure 1-3 z/OS operating systems and hardware

z/OS operating systems and hardware

z/OS V2, the next generation of the z/OS operating system, enables you to manage the volatility of online business workloads. z/OS delivers the highest qualities of service for enterprise transactions and data, and extends these qualities to new applications using the latest software technologies.

Upon initial program load (IPL), custom parameters start an operating environment that is comparable to z/OS in all aspects of operation, service, management, reporting, and zSeries hardware functionality. z/OS V2R1 is supported on the IBM zEnterprise® EC12 (zEC12), zBC12, z196, z114, IBM z10™ Enterprise Class (EC), z10 Business Class (BC), z9 EC, and z9 BC IBM System z servers.

z/OS and workloads

The best platform for integrating web-based transaction processing, online business, and enterprise applications and database serving, is z/OS on zSeries, the premier platform for enterprise-scale workloads. Its qualities of service (QoS) in terms of reliability, scalability, security, availability are undisputed and unrivaled in the industry.

The value is two-fold, because z/OS is the most robust and reliable enterprise platform, and it has a history of managing a large percentage of the world's business data and transactions. The finance industry, the transportation industry, the health industry, and government all rely on z/OS to store, manage, and use their data efficiently and securely.

Restriction: Beginning with z/OS V1R6, z/OS can only be started in z/Architecture mode on z/Architecture servers.

1.4 Hardware requirements

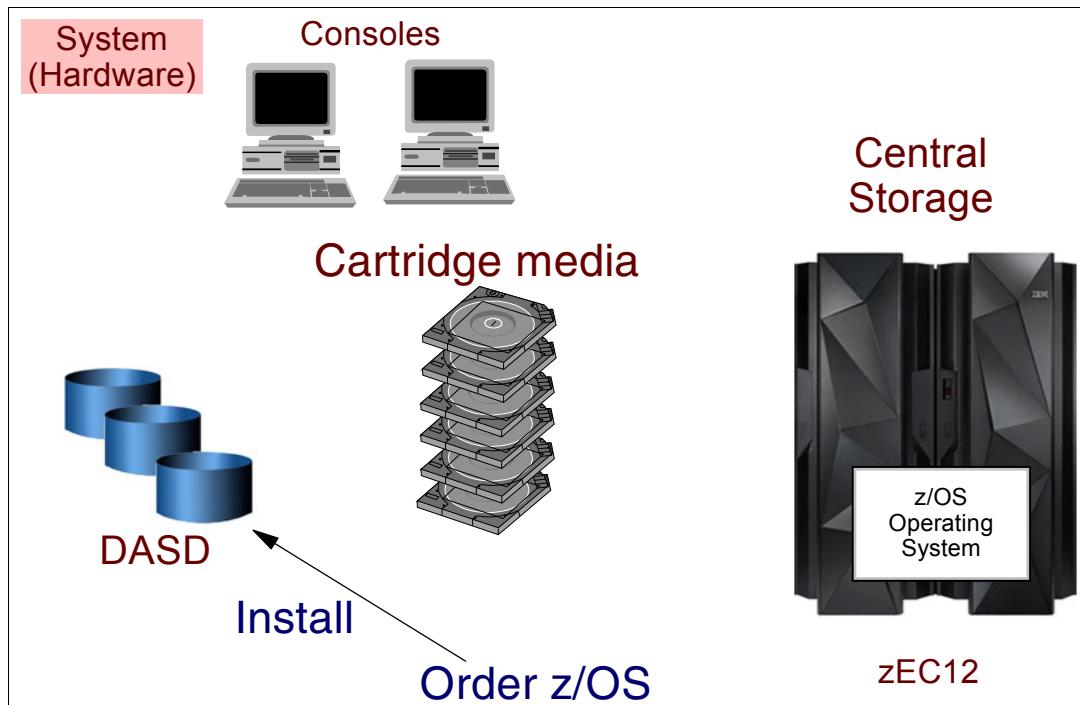


Figure 1-4 Hardware requirements

Hardware requirements

The base z/OS operating system runs in a processor and resides in the processor storage during execution. The z/OS operating system is commonly referred to as the system software. The hardware consists of the processors and other devices, such as a direct access storage device (DASD), tape, and consoles. Tape and DASD are used for system functions and by user programs that run in a z/OS environment.

When you order z/OS, you receive your order on tape cartridges. When you install the system from tape, the system code is then stored on DASD volumes. After the system is customized and ready for operation, system consoles are required to start and operate the z/OS system. Not shown in Figure 1-4 are the control units that connect the processor to the other tape, DASD, and console devices. The following list includes the main concepts shown here:

Software	The z/OS operating system consists of load modules, and is often called <i>executable code</i> . These load modules are placed onto DASD volumes in load libraries during a system install process.
Hardware	The system hardware consists of all of the devices, controllers, and processors that make up a z/OS complex.
Devices	Shown in Figure 1-4 are the tape, DASD, and console devices. There are many other types of devices that are discussed later in this document.
Storage	Central storage, often called <i>real storage</i> or <i>main storage</i> , is where the z/OS operating system runs. Also, all user programs share the storage of the processor with the OS.

1.5 IBM System z Server: Basic mode and logically partitioned mode

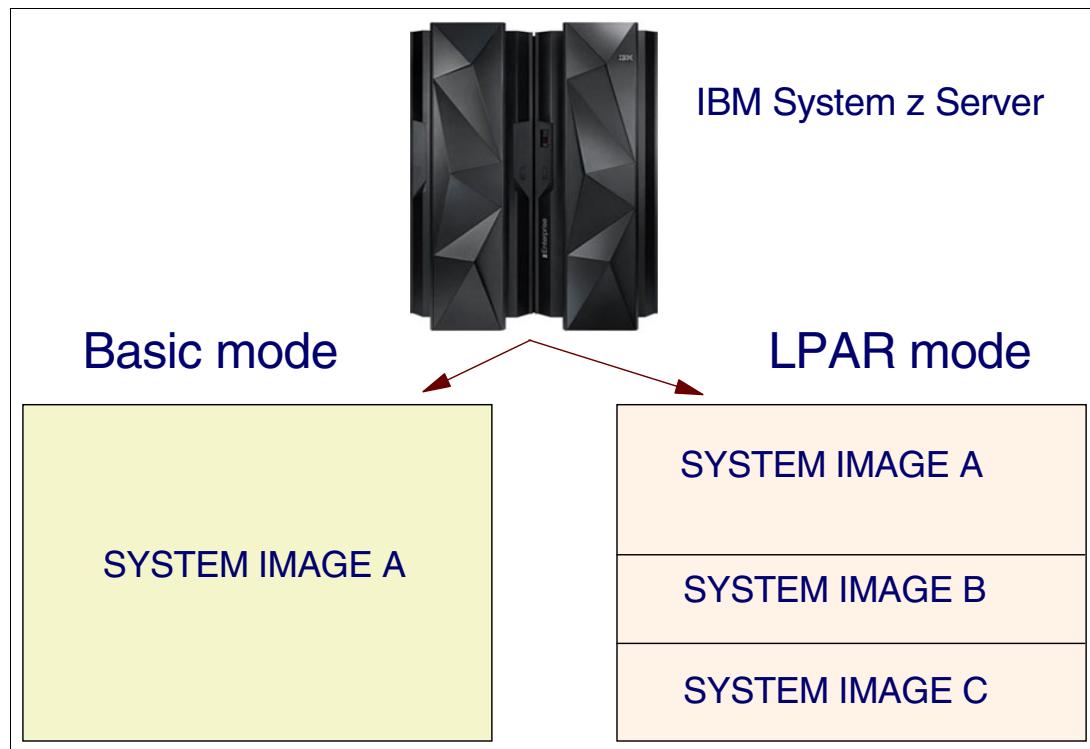


Figure 1-5 Basic mode and logically partitioned mode

Basic mode and logically partitioned mode

The z900 and z800 were the last System z servers that supported Basic mode. Current System z servers can only operate in logically partitioned mode (LPAR mode).

LPAR mode LPAR mode is a central processor (CP) Power-on Reset is a CP mode (established by power-on reset) that enables use of the IBM Processor Resource/System Manager (IBM PR/SM™) feature and enables an operator to allocate CP hardware resources (including CP, main storage, and channel paths) among multiple logical partitions (LPARs). z/OS as the OS runs in each LPAR in the machine with all server resources (CPs, storage, and channels).

Basic mode This is a CP mode that does not use logical partitioning, with the CP running one copy of the z/OS operating system. When a server is in Basic mode, all server resources (CPs, storage, and channels) are available to the one OS. All of the physical CPs are used in dedicated mode for the one OS. Any excess CP resource is wasted, because no other system has access to it.

1.6 z/OS V2R1 base elements

Alternate Library for REXX	ICKDSF
Base Control Program (BCP)	Integrated Security Services
Bulk Data Transfer base (BDT)	ISPF
BookManager Read	JES2
Common Information Model (CIM)	Language Environment
Communications Server	Library Server
Cryptographic Services	Metal C Runtime Library
DFSMSdfp	MICR/OCR
Distributed File Service	Network File System (NFS)
EREP	OSA/SF
ESCON Director Support	Run-Time Library Extensions
FFST	SMP/E
GDDM	TIOC
HCD	TSO/E
High Level Assembler (HLASM)	z/OS Font Collection
IBM HTTP Server	z/OS UNIX
IBM Tivoli Directory Server for z/OS	3270 PC File Transfer Program

Figure 1-6 z/OS base elements with z/OS V1R12

z/OS base elements

The z/OS system consists of base elements that deliver essential operating functions, in addition to the services provided by the Base Control Program (BCP) functions, such as communications support, online access, host graphics, and online viewing of publications. The base elements are listed in Figure 1-6.

Included with the z/OS system are the base OS, products, and features, for example, z/OS UNIX System Services and local area network (LAN) services. In addition to these features, the following list includes some of the products that provide essential OS functions and are included in the base:

- ▶ Time Sharing Option Extensions (TSO/E)
- ▶ Interactive System Productivity Facility (ISPF)
- ▶ Graphical Data Display Manager (GDDM)
- ▶ IBM BookManager® READ

Certain base elements can be dynamically enabled and disabled, because a client can choose to use a vendor product rather than IBM products.

The idea of the z/OS system is to have elements and features rather than licensed programs. This concept might be more easily explained by saying that z/OS consists of a collection of functions that are called *base elements* and *optional elements*. The optional elements (features) are either integrated or nonintegrated. It is important to note that these optional features, both integrated and nonintegrated, are also tested as part of the integration of the entire system.

Base elements

When you order z/OS you receive all of the base elements, as listed in Figure 1-6.

- ▶ **Alternate Library for REXX:** This is the Alternate Library portion of the priced product IBM Library for REXX on zSeries V1R4 (5695-014). The Alternate Library for REXX enables users to run complied Restructured Extended Executor (REXX) programs.

The Alternate Library consists of function modification identifiers (FMIDs) HWJ9143 (ENU) and JWJ9144 (JPN). These are the same FMIDs that are now in the z/OS base element. The no-fee web download Alternate Library for REXX on z/OS base element was new in z/OS V1R9. Including it as a base element in z/OS is intended to make it easier to use than the functionally equivalent programs previously listed.
- ▶ **BCP:** The exclusive base element BCP provides essential OS services. The BCP includes the following services:
 - The input/output (I/O) configuration program (IOCP)
 - The z/OS Workload Manager (WLM)
 - IBM System Management Facility (SMF)
 - The z/OS UNIX System Services (z/OS UNIX) kernel
 - The program management binder (FMID HPM7790)
 - IBM Health Checker for z/OS
 - Support for the Unicode Standard (FMID HUN7790)
 - z/OS Extensible Markup Language (XML) System Services (z/OS XML)
 - Capacity Provisioning (FMID HPV7790)
 - System REXX for z/OS Base
- ▶ **Bulk Data Transfer (BDT):** This is an exclusive base element that provides the base services that the optional BDT features (BDT File-to-File and BDT SNA network job entry, or BDT SNA NJE) need to transfer data from one computer system to another. You cannot activate any BDT functions until one or both of the optional BDT features is enabled.
- ▶ **BookManager READ:** This is a nonexclusive base element used to display, search, and manage online documents and bookshelves. A related optional feature is BookManager BUILD.
- ▶ **CIM:** The exclusive base element Common Information Model (CIM) is a standard data model for describing and accessing systems management data in heterogeneous environments. It enables system administrators to write applications that measure system resources in a network with other operating systems and hardware. To enable z/OS for cross-platform management, a subset of resources and metrics of a z/OS system are mapped into the CIM standard data model.

CIM was new in z/OS V1R7. In z/OS V1R9, a new component was added that contains a Java programming interface for CIM client applications. In z/OS V1R10, all CIM components were consolidated under a single FMID, HPG7750.

Transport Layer Security (TLS) encryption is performed for CIM by the base element Communications Server. CIM does not implement any of its own encryption algorithms.
- ▶ **Communications Server:** This exclusive base element can be dynamically enabled. Communications Server supports secure TCP/IP, SNA, and UNIX networking throughout an enterprise. It gives you the ability to connect subsystems and applications to each other, and to connect network devices (such as terminals and printers) to the system.

Communications Server consists of two components: IP Services and SNA Services. A related optional feature is Communications Server Security Level 3. For encryption, the IP Services component uses the Simple Network Management Protocol (SNMP) V3 DES 56-bit and IPSec DES 56-bit algorithms. IP Services also uses the System Secure Sockets Layer (SSL) component of Cryptographic Services for encryption services. The SNA Services component uses the limited DES algorithm for encryption.

- ▶ **Cryptography Services:** Cryptography is an exclusive base element. Cryptography is the transformation of data to conceal its meaning. In z/OS, the base element Cryptographic Services provides the following base cryptographic functions:

- Data secrecy
- Data integrity
- Personal identification
- Digital signatures
- The 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 optional feature z/OS Security Level 3. The Cryptographic Services element consists of the following components:

- Integrated Cryptographic Service Facility (ICSF)
- Open Cryptographic Services Facility (OCSF), last changed in z/OS V1R9
- The public key infrastructure (PKI) Services component, last changed in z/OS V2R1
- System SSL, last changed in z/OS V2R1

For encryption, the components use the following cryptography algorithms:

- OCSF uses the Rivest Cypher 2 (RC2)/RC4/RC5 40-56 bit and DES 56-bit algorithms.
- The PKI Services component uses the IBM Resource Access Control Facility (RACF) component of the optional Security Server feature, and the ICSF, OCSF, and System SSL components of base element Cryptographic Services. The PKI Services component uses the Rivest-Shamir-Adleman (RSA), digital signature algorithm (DSA), and DES 56-bit algorithms.
- System SSL uses the RC2/RC4, DES through 56-bit, Diffie-Hellman, RSA, and DSA algorithms.
- PKI Services and System SSL support keys generated with elliptic curve cryptography (ECC) algorithms.

- ▶ **DFSMSdfp:** The Data Facility Storage Management Subsystem (DFSMS) Data Facility Product (DFSMSdfp) component has the following features:

- DFSMS Removable Media Manager (DFSMSrmm)
- DFSMS Data Set Services (DFSMSdss)
- DFSMS Hierarchical Storage Manager (DFSMSshsm)
- DFSMS Transactional Virtual Storage Access Method, or VSAM (DFSMStvs)

The exclusive DFSMSdfp base element provides storage, data, program, and device management functions. Related optional features are DFSMSrmm, DFSMSdss, DFSMSshsm, and DFSMStvs. In z/OS V1R9, the FMID for the English panels was eliminated and the panels were merged into the base FMID, which is HDZ1190.

- ▶ **Distributed File Service:** This exclusive base element provides the following services:

- The IBM zSeries File System (zFS), which is a UNIX file system that can be used in addition to the hierarchical file system (HFS). zFS file systems contain files and directories that can be accessed with the z/OS HFS file application programming interfaces (APIs). zFS file systems can be mounted into the z/OS UNIX hierarchy along with other local (or remote) file system types, such as HFS, temporary file system (TFS), AUTOMNT, and NFS.

The zFS does not replace the HFS; it is complementary to the HFS. As of z/OS V1R7, you can use any combination of HFS and zFS file systems. zFS can be used for the root file system. Because zFS has higher performance characteristics than HFS and is the strategic file system, HFS might no longer be supported in future releases, in which case you will have to port the remaining HFS file systems to zFS.

The zFS provides significant performance gains in most environments that require files 8 KB in size or greater, and that are frequently accessed and updated. The access performance of smaller files is equivalent to that of the HFS. For all files, the zFS provides a reduced exposure to loss of updates.

The zFS is a logging file system with a write pattern to disk that reduces the points of failure after a system outage. For additional information about the zFS, including how to migrate data from the HFS to the zFS, see *z/OS Distributed File Service zSeries File System Administration*, SC24-5989.

As of z/OS V1R11, zFS has the capability of running sysplex-aware for read/write mounted file systems.

- Server message block (SMB) file/print serving support, which is based on the X/Open PC Interworking: SMB, Version 2. Included in the support is access to HFS, sequential, PDS, PDSE, and VSAM data sets from Microsoft Windows XP Professional, Windows Terminal Server on Windows 2003, SUSE Linux Enterprise Server (SLES) with Samba, and Red Hat Enterprise Linux (RHEL) with Samba. Windows workstation users can also use z/OS printer capabilities using the SMB file/print server interface to the z/OS Infoprint Server feature.
- ▶ **EREP:** As a nonexclusive base element, the Environmental Record Editing and Printing (EREP) program edits and prints reports for the records placed in the error-recording data set (ERDS), therefore helping IBM service representatives fix problems.
- ▶ **ESCON Director Support:** This exclusive base element Enterprise Systems Connection (ESCON) Director Support enables the reporting of ESCON director device errors to z/OS.
- ▶ **IBM FFST™:** IBM First Failure Support Technology™ (FFST) is an exclusive base element that provides immediate notification and first failure data capture for software events.
- ▶ **GDDM:** This nonexclusive base element provides presentation services and device-driving capability. It includes peripheral clock (PCLK) and REXX code. Related optional features are GDDM-Presentation Graphics Feature and GDDM-REXX.

Other GDDM-associated products are not included with z/OS, but are separately orderable with z/OS:

IVU	ImmerVision Extensible Markup Language (XML) User Interface
GKS	Graphical Kernel System
IMD	Interactive Map Definition

- ▶ **HCD:** Hardware Configuration Definition (HCD) is an exclusive base element that defines both the OS configuration and the processor hardware configuration for a system. A related optional feature is HCM (Hardware Configuration Manager).
- ▶ **HLASM:** High Level Assembler (HLASM) is a nonexclusive base element that integrates almost all functions of past assemblers, and provides extensions and improvements. A related optional feature is the HLASM Toolkit.
- ▶ **IBM HTTP Server:** An exclusive base element, IBM HTTP Server is the web server for z/OS. It provides scalable, high-performance web serving for critical online business applications. It supports SSL secure connections, dynamic caching using the Fast Response Cache Accelerator (FRCA), multiple IP addresses, proxy authentication, and double-byte character set characters.

IBM HTTP Server NA (North America) Secure is now a component of IBM HTTP Server. Before V1R6, it was an optional feature of z/OS. This packaging change was the only change to IBM HTTP Server in V1R6; there was no functional change.

- ▶ **IBM Tivoli® Directory Server for z/OS:** IBM Tivoli Directory Server for z/OS, as an exclusive base element, provides client access to an= Lightweight Directory Access Protocol (LDAP) server. It consists of an LDAP server, an LDAP client, and LDAP client utilities.
 For encryption, Tivoli Directory Server uses the DES (56-bit) and Message-Digest 5 (MD5) encryption algorithms.
- ▶ **ICKDSF:** Device Support Facility (ICKDSF) is a nonexclusive base element that enables you to perform functions needed for the installation and use of IBM DASD. ICKDSF was last changed in the z/OS V1R4 z990 Compatibility Support feature. This level was carried forward to the z/OS V1R4 z990 Exploitation Support feature and then to z/OS V1R5 and later, and is functionally equivalent to the R17 level of the ICKDSF product.
- ▶ **Integrated Security Services:** As an exclusive base element, Integrated Security Services provides base security functions for z/OS. It includes several components:
 - Enterprise Identity Mapping (EIM), which was last changed in z/OS V1R10, enables you to map a user's identity on one system to the user's identity on another system.
 - Network Authentication Service, which was last changed in z/OS V2R1, uses the DES algorithm for encryption.
 - Open Cryptographic Enhanced Plug-ins (OCEP), which was last changed in z/OS V1R9, enables applications to retrieve and use digital certificates and private keys that are stored in the RACF database.
 As of z/OS V1R11, LDAP Server is no longer a component of Integrated Security Services. IBM Tivoli Directory Server, a base element introduced in z/OS V1R8, contains a rewritten LDAP Server.
- ▶ As of z/OS V1R13, Distributed Computing Environment (DCE) and Distributed Computing Environment Security Server (DCE Security Server) are removed from the system. IBM suggests the IBM WebSphere® Application Server, the IBM Network Authentication Service, or the Tivoli Directory Server as replacement strategies for each of the DCE technologies. For more information, see the IBM Redbooks publication *DCE Replacement Strategies*, SG24-6935.
- ▶ **ISPF:** As an exclusive base element, Interactive System Productivity Facility (ISPF) has four major components:
 - Dialog Manager (DM), which provides services to dialogs and users. These services include display, variable services, input and output, user and application profiles, table management, system interface services, dialog testing aids, and debugging aids.
 - Program Development Facility (PDF) provides services to assist dialog or application developers. These include edit and browse functions, a wide range of foreground and batch compilers, data set and catalog utilities, TSO command interfaces, and data set search and compare functions.
 - Software Configuration and Library Manager (SCLM) is a tool that controls, maintains, and tracks all of the software components of the application throughout the development cycle.
 - The Client/Server component provides users who have a workstation running Windows or UNIX with a graphical user interface (GUI) to ISPF application panels.
- ▶ **JES2:** As an exclusive base element, IBM Job Entry Subsystem 2 (JES2) accepts the submission of work for the BCP. JES2 exercises independent control over its job processing functions, but JES3 exercises centralized control.
 z/OS V1R13 is the last release to support a staged migration for JES2 and JES3. Starting in z/OS V2R1, customers need to migrate to all elements of z/OS at the same time, including JES2, JES3, or both.

- ▶ **Language Environment:** An exclusive base element that provides the runtime environment for programs generated with C, C++, Common Business Oriented Language (COBOL), Fortran, and PL/I. Inclusion of Language Environment as a base element in z/OS does not replace the need for separate compilers. Language Environment uses the limited DES algorithm for encryption.
- ▶ **Library Server:** An exclusive base element that converts BookManager and information center documents to Hypertext Markup Language (HTML) for display through a web browser, and provides unified organization and searching of BookManager, PDF, and information center documents.
- ▶ **Metal C Runtime Library:** This exclusive base element is a set of link pack area (LPA)-resident C functions that can be called from a C program created using the z/OS XL C compiler METAL option. This base element was new in z/OS V1R9.
- ▶ **MICR/OCR:** This exclusive magnetic ink character recognition/optical character recognition (MICR/OCR) base element provides the device support code for various magnetic and optical devices.
- ▶ **NFS:** The exclusive base element acts as a file server to workstations, personal computers, or other authorized systems in an Internet Protocol network. It consists of a client and a server. It supports Berkeley sockets but not TCP/IP sockets. NFS uses the Network Authentication Service component of Integrated Security Services for encryption. NFS is always enabled, even when the alternate base configuration is ordered. In z/OS V1R8, the NFS server and client FMIDs were merged into one FMID.
- ▶ **OSA/SF:** The exclusive base element Open Systems Adapter/Support Facility (OSA/SF) provides a straightforward interface for monitoring and controlling the zSeries Open Systems Adapter feature, which provides network connectivity directly to LANs and wide area networks (WANs) that support IP and SNA protocols.

OSA/SF supports Gigabit (Gb), Token Ring, Fast Ethernet, 1000Base-T Ethernet, 10 Gigabit Ethernet (GbE), and asynchronous transfer mode (ATM) features, depending on the server on which z/OS runs. For details, see IBM zEnterprise 196, IBM System z10®, System z9 and IBM eServer zSeries *OSA-Express Customer's Guide and Reference*, SA22-7935.

OSA/SF was last changed in the z/OS V1R4 z990 Compatibility Support feature. This level was carried forward to the z/OS V1R4 z990 Exploitation Support feature, and then to z/OS V1R5 and later. OSA/SF became exclusive with the introduction of the z/OS V1R4 z990 Compatibility Support feature.

- ▶ **Run-Time Library Extensions:** This exclusive base element extends the runtime support provided by the Language Environment base element. It consists of:
 - Common Debug Architecture (CDA) libraries
 - Utilities
 - UNIX System Laboratories (USL) I/O Stream Library and USL Complex Mathematics Library
- ▶ **SMP/E:** This nonexclusive System Modification Program/Extended (SMP/E) base element is a tool for installing and maintaining software, and for managing the inventory of software that has been installed. The SMP/E product enables clients who are currently licensed for an earlier level of z/OS to order and install the current level of SMP/E without having to upgrade their entire operating system.

This enables products that run on z/OS to use the packaging and installation enhancements of SMP/E without requiring a later level of the OS. This also enables clients to use new electronic delivery and installation technologies in SMP/E sooner. The SMP/E product is available at no additional charge to clients.

The Planning and Migration Assistant (PMA), a component of SMP/E, can help you maintain, plan for, and order new releases of z/OS and other products. It provides reports that use IBM-supplied data, your SMP/E consolidated software inventory (CSI) data set, and a CustomPac inventory file. For more information, see the PMA website:

<http://www.ibm.com/systems/z/os/zos/smpe/pma/>

- ▶ **TIOC:** This exclusive terminal input/output controller (TIOC) base element enables console services and TSO/E to communicate with the terminal hardware.
- ▶ **TSO/E:** The exclusive base element provides an interactive terminal interface. As in prior releases of TSO/E, this element includes command lists (CLISTS) and REXX, but does not include a REXX compiler.
- ▶ **z/OS Font Collection:** The exclusive z/OS Font Collection base element, introduced with z/OS V2R1, consists of character sets, coded fonts, and Advanced Function Presentation (AFP) code pages for printing documents.

z/OS Font Collection (FMID HFNT110 and HFNT11J) provides selected fonts that are currently marketed and serviced for the z/OS environment into one package. Double-byte fonts for Chinese, Japanese, and Korean (CJK) are included. Fonts that are included in z/OS Font Collection are not available for ordering with z/OS V2R1, because they are provided with z/OS V2R1. The fonts in the z/OS Font Collection are:

- AFP Font Collection for IBM S/390 (5648-B33), includes Japanese, Korean, Traditional Chinese, and Simplified Chinese.
- IBM Infoprint Fonts for z/OS (5648-E76), includes Japanese, Korean, Traditional Chinese, and Simplified Chinese.
- IBM Print Services Facility™ (PSF) Compatibility Font feature (5655-M32), includes just the PSF feature for the compatibility fonts, not the executable code or the entire product.
- World type fonts that were not previously available in the z/OS environment. These fonts were part of the InfoPrint Font Collection V3.1 available on other platforms.
- Selected object fonts (not source), Pi and Special (5771-ABC), Math and Science (5771-ADT), Data1 Fonts (5771-ADA), A programming language (APL) R1.2 Bounded Box (5771-ADB), Son Serif Headline (5771-ADW), Senoran Serif (5771-ABA), Son Sans Serif (5771-ABB), Son Sans Serif Headline (5771-ADX), Son Sans Serif Cond (5771-AFL), Son Serif Expanded R1 (5771-AFN).

From a z/OS standpoint, this new element is assuming ownership of the parts using SMP/E SUP/DELETE logic, and therefore you no longer need to order the previous stand-alone products as of z/OS V2R1.

- ▶ **z/OS UNIX:** The exclusive base element z/OS UNIX System Services (z/OS UNIX) provides the standard command interface familiar to interactive UNIX users. This base element is made up of the following components:
 - Application Services, which was last changed in z/OS V2R1. This component includes the Shell and Utilities, Debugger, and Parallel Environment.
 - Integrated Call Level Interface (ICLI), which was last changed in z/OS V1R6, was removed from the z/OS V2R1 system.
 - Starting with z/OS V1R13, Connection Manager and Process Manager, which both last changed in OS/390 V2R7, are removed from the system.
- ▶ **3270 PC File Transfer Program:** This nonexclusive base element transfers files from the host to the workstation for offline data manipulation, updating, or correction, or for the transfer and storage of local data on the host system.

1.7 z/OS optional features

BDT File-to-File	HCM
BDT SNA NJE	HLASM Toolkit
BookManager BUILD	Infoprint Server
Communications Server Security Level 3	JES3
DFSMSdss	RMF
DFSMShsm	SDSF
DFSMSrmm	Security Server
DFSMStvs	XL C/C++
DFSORT	zEDC
GDDM-PGF	z/OS Security Level 3
GDDM-REXX	

Figure 1-7 z/OS optional features

z/OS optional features

In addition to the base elements, z/OS has optional features that are closely related to the base features. The optional features are orderable with z/OS, and provide additional OS functions. The optional features are listed in Figure 1-7.

Optional features are unpriced or priced:

- ▶ *Unpriced* features are issued to you only if you order them.
- ▶ *Priced* features are always included. These features are ready to use after you install z/OS (and customize them as needed). IBM enables the priced features you ordered and disables the priced features you did not order. Later on, if you decide to use them, you can notify IBM, and then you enable them dynamically (which is known as *dynamic enablement*). Dynamic enablement is done by updating the IFAPRDxx SYS1.PARMLIB member. Notify IBM by contacting your IBM representative.

Various *optional* features that support dynamic enablement are always shipped, including JES3, DFSMSdss, and DFMSHsm. If these features are ordered as part of the z/OS system order, they are included as enabled in the system. If they are not ordered, they are included but disabled. Later on you can enable them through a SYS1.PARMLIB member.

The other types of features are the optional features equivalent to optional licensed programs. Examples are RACF from the Security Server set of programs, the Resource Management Facility (RMF), the C/C++ compiler, and so on.

z/OS optional features

The z/OS optional features list follows:

- ▶ **BDT File-to-File:** This exclusive priced optional feature can be dynamically enabled. BDT File-to-File enables users at one z/OS system in an SNA network to copy data sets to or from another z/OS system in the network. This feature is related to the element BDT.
- ▶ **BDT SNA NJE:** This exclusive priced optional feature can be dynamically enabled. BDT SNA NJE enables JES3 users to transmit jobs, output, commands, and messages from one computer system to another in an SNA network. This feature is related to the element BDT and the feature JES3.

It enables JES3 users to transmit jobs, output, commands, and messages from one computer system to another in an SNA network. This feature is related to the JES3 feature.

- ▶ **BookManager BUILD:** This exclusive priced optional feature can be dynamically enabled. BookManager BUILD creates softcopy documents that can be used by any of the BookManager products, such as BookManager READ or Library Server.
- ▶ **Communications Server Security Level 3:** This exclusive priced optional feature can be dynamically enabled. It works in conjunction with the Communications Server base element to provide stronger encryption (greater than 64 bits) than that available without this feature. This feature uses the Triple Data Encryption Standard (TDES) and Advanced Encryption Standard (AES) algorithms for encryption.

The actual level of encryption that takes place with this feature installed can be configured to be less than the maximum level enabled by this feature. This feature is worldwide exportable, subject to US export regulations.

- ▶ **DFSMSdss:** This exclusive priced optional feature can be dynamically enabled. DFSMSdss copies and moves data for backup and recovery, and to reduce free-space fragmentation.
- ▶ **DFSMShsm:** This exclusive priced optional feature can be dynamically enabled. DFSMShsm provides automated DASD storage management, including space management for low and inactive data, and availability management for accidental data loss caused by local and site disasters. DFSMShsm also enables you to make effective use of tape media.

DFSMShsm requires DFSMSdss. For this reason, DFSMShsm is not available by itself. If you want to use DFSMShsm, you must order the DFSMShsm/DFSMSdss combination. (DFSMSdss is also available by itself for those who do not want DFSMShsm.)

- ▶ **DFSMSrmm:** This exclusive priced optional feature can be dynamically enabled. DFSMSrmm helps you manage your removable media as one enterprise-wide library across systems that can share DASD.
- ▶ **DFSMStvs:** This exclusive priced optional feature can be dynamically enabled. DFSMStvs enables batch jobs and IBM Customer Information Control System (IBM CICS®) online transactions to update shared VSAM data sets concurrently.
- ▶ **DFSORT:** This exclusive priced optional feature can be dynamically enabled. Data Facility Sort (DFSORT) provides fast and easy sorting, merging, copying, reporting, and analysis of your business information, and versatile data handling at the record, field, and bit level. DFSORT also includes the high-performance ICEGENER facility, the versatile ICETOOL utility, Symbols, and multiple output capability with the powerful OUTFIL feature.
- ▶ **GDDM-PGF:** This exclusive priced optional feature can be dynamically enabled. GDDM-Presentation Graphics Feature (PGF) is a set of programs for creating presentation material in a variety of styles. This feature is related to the base element GDDM.
- ▶ **GDDM-REXX:** This exclusive priced optional feature can be dynamically enabled. GDDM-REXX is a productivity tool that enables programmers to prototype GDDM applications, and to create small routines and utility programs, quickly and easily. This feature is related to the base element GDDM.
- ▶ **HCM:** The exclusive HCM priced optional feature, which can be dynamically enabled, is a workstation-based GUI to the base element HCD.
- ▶ **HLASM Toolkit:** The exclusive HLASM Toolkit priced optional feature, which can be dynamically enabled, provides tools to improve application development, debugging, and recovery. It is related to the base element HLASM.

- ▶ **Infoprint Server:** The exclusive Infoprint Server priced optional feature, which can be dynamically enabled, supports the ability to print files on z/OS printers from any workstation that has TCP/IP access. This feature consists of the following components:
 - IBM IP PrintWay™
 - NetSpool
 - Print Interface
 - Printer Inventory Manager
 - Transform Interface
 - z/OS Infoprint Central
- Use the Infoprint Server feature's IP PrintWay component rather than the Communications Server base element's Network Print Facility (NPF) to reroute print data to an IP network. IP PrintWay provides improved function, capacity, performance, and usability over NPF.
- ▶ **JES3:** The exclusive priced optional feature JES3, which can be dynamically enabled, accepts the submission of work for the BCP. JES3 exercises centralized control over its job processing functions, but JES2 exercises independent control. z/OS V1R13 is the last release to support a staged migration for JES2 and JES3. Starting in z/OS V2R1, customers need to migrate to all elements of z/OS at the same time, including JES2, JES3, or both.
- ▶ **RMF:** The exclusive RMF priced optional feature, which can be dynamically enabled, gathers data about z/OS resource usage and performance, and provides reports on any system in a sysplex.
- ▶ **SDSF:** The exclusive priced optional feature System Display and Search Facility (SDSF), which can be dynamically enabled, provides you with information to monitor, manage, and control your z/OS system. If you use a prior release of JES2 or JES3 with z/OS V1R12, you must use the same prior release of SDSF with z/OS V1R12.
- ▶ **Security Server:** The exclusive priced optional feature Security Server, which can be dynamically enabled, enables you to control access to protected resources. Security Server contains one component, RACF. Security Server uses the limited DES, Commercial Data Masking Facility (CDMF), RC 40-bit, RSA, and DSA algorithms for encryption.
- ▶ **XL C/C++:** This exclusive priced optional feature can be dynamically enabled. XL C/C++ consists of the XL C/C++ compiler and XL C/C++ application development utilities.
- ▶ **zEDC:** The IBM zEnterprise Data Compression (zEDC) for z/OS exclusive priced optional feature provides use of the IBM zEnterprise EC12 (zEC12) and IBM zEnterprise BC12 (zBC12) zEDC express adapter for hardware data compression.
- ▶ **z/OS Security Level 3:** This exclusive unpriced optional feature cannot be dynamically enabled. z/OS Security Server provides strong encryption for z/OS.

z/OS Security Level 3 includes IBM Tivoli Directory Server for z/OS Security Level 3 (new in z/OS V1R8; replaces LDAP Security Level 3), Network Authentication Service Security Level 3, OCSF Security Level 3, and System SSL Security Level 3.

1.8 z/OS Base Control Program

- ❑ Essential operating system services
 - Base control program and job entry subsystem (JES)
- ❑ BCP requires the following:
 - A security product (RACF is the IBM offering)
 - DFSMSdfp
 - Communications Server
 - SMP/E
 - TSO/E
 - z/OS UNIX System Services (z/OS UNIX) kernel
- ❑ Important BCP components
 - System Management Facilities (SMF)
 - Resource Management Facility (RMF)
 - Workload Manager (WLM)
- ❑ Interesting optional features
 - Infoprint Server

Figure 1-8 z/OS BCP functions

Base Control Program functions

The backbone of the z/OS system is the MVS BCP with either JES2 or JES3 as a primary job-entry subsystem. These provide the essential services that make z/OS the system of choice when you need to process your workloads reliably, securely, with complete data integrity, and without interruption.

The BCP includes the IOCP, the WLM, SMF, the z/OS UNIX System Services (z/OS UNIX) kernel, the program management binder, IBM Health Checker for z/OS, support for the Unicode Standard, z/OS XML System Services (z/OS XML), Capacity Provisioning, and System REXX for z/OS Base.

Tip: z/OS consists of the base elements shown in Figure 1-6 on page 10. Additional optional features can be added to the z/OS system, as listed in Figure 1-7 on page 17.

The components in Figure 1-8 are described in the following pages.

1.9 z/OS Security Server: RACF component

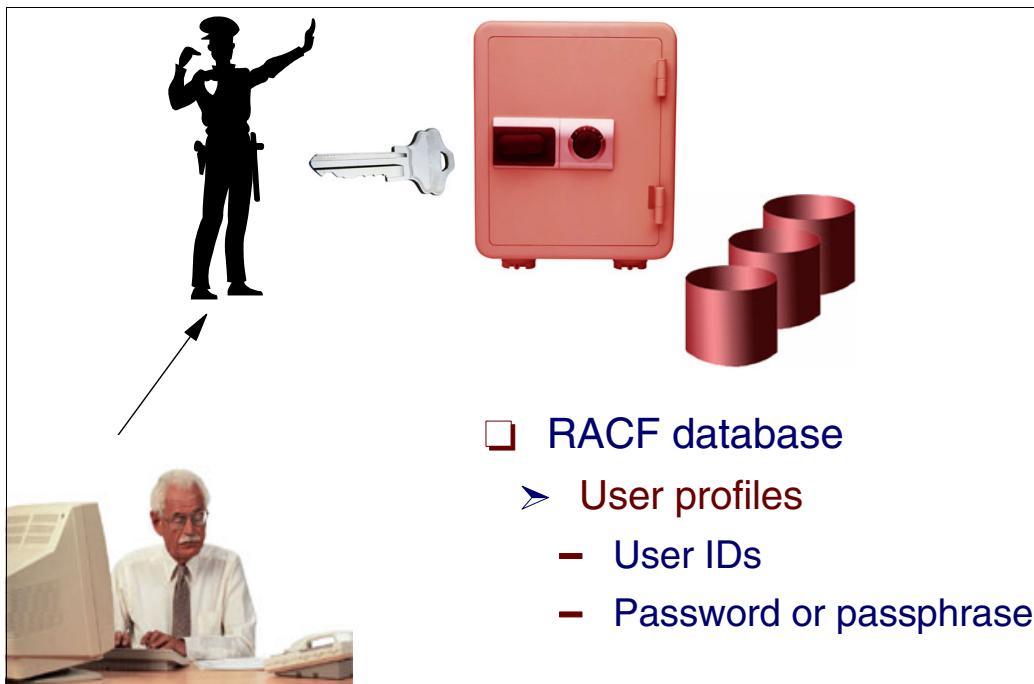


Figure 1-9 z/OS Security Server (RACF component)

Computer security

As general computer literacy and the number of people using computers has increased, the need for data security has taken on a new level of importance. No longer can the installation depend on keeping data secure simply because no one knows how to access the it. Further, making data secure does not mean simply making confidential information inaccessible to those without permission to see it; it also means preventing the inadvertent destruction of files by people who might not even know that they are improperly manipulating data.

RACF component

The z/OS Security Server is the IBM security product. The RACF product is a component of the z/OS Security Server, and it works together with the existing system features of z/OS to provide improved data security for an installation. If this product is to be installed in your environment, then RACF customization must be done.

RACF helps meet the need for security by providing the following functionality:

- ▶ Flexible control of access to protected resources
- ▶ Protection of installation-defined resources
- ▶ Ability to store information for other products
- ▶ Choice of centralized or decentralized control of profiles
- ▶ An ISPF panel interface
- ▶ Transparency to users
- ▶ Exits for installation-written routines

RACF security protection

RACF controls access to and protects resources. For a software access control mechanism to work effectively, it must first identify the person who is trying to gain access to the system, and then verify that the user is really that person.

RACF uses a user ID in a user profile in the RACF database, and a system-encrypted password or pass phrase (new with z/OS V1R8), to perform its user identification and verification. When you define a user to RACF, you assign a user ID and temporary password. The user ID identifies the person to the system as a RACF user. The password or pass phrase verifies the user's identity. The temporary password permits initial entry to the system, at which time the person is required to choose a new password.

RACF authorization checks

Having identified a valid user, the software access control mechanism must next control interaction between the user and the system resources. It must authorize what resources that user can access, but also in what way the user can access them, such as for reading only, or for updating and reading. This controlled interaction, or authorization checking, is done by RACF. Before this activity can take place, however, someone with the appropriate authority at the installation must establish the constraints that govern those interactions.

With RACF, you are responsible for protecting the system resources (data sets, tape and DASD volumes, IBM IMS™ and CICS transactions, TSO logon information, and terminals) and for issuing the authorities by which those resources are made available to users. RACF records your assignments in profiles stored in the RACF database. RACF then refers to the information in the profiles to decide if a user is to be permitted to access a system resource.

1.10 Data Facility Storage Management Subsystem

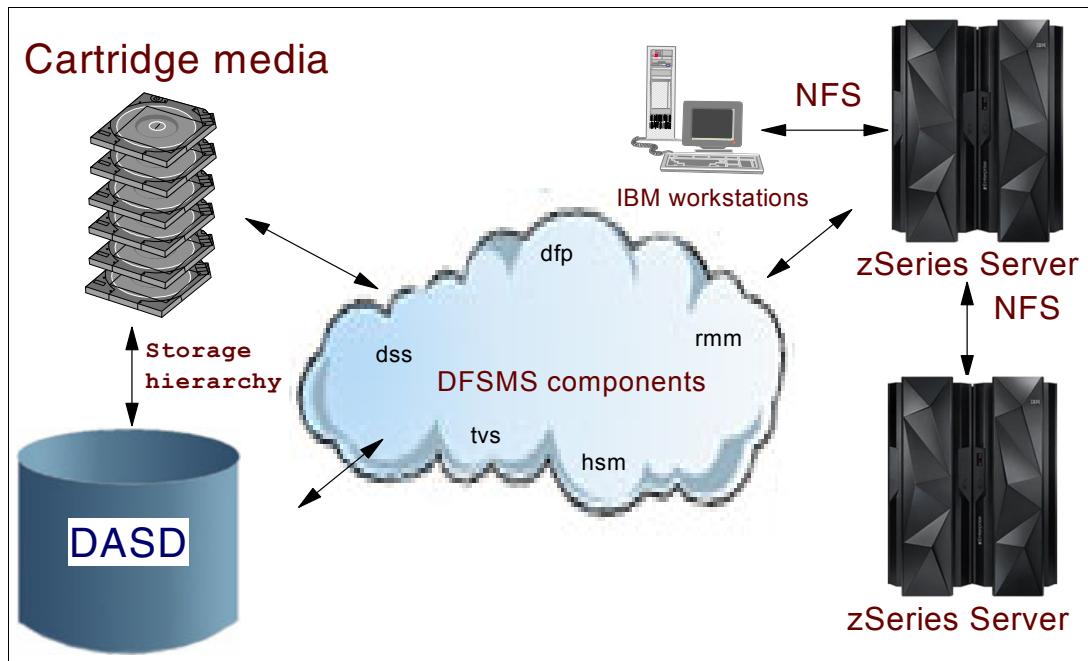


Figure 1-10 Data Facility Storage Management Subsystem (DFSMS)

DFSMS component

DFSMS consists of a suite of related data and storage management products for the z/OS system. DFSMS is an operating environment that helps automate and centralize the management of storage based on the policies that your installation defines for availability, performance, space, and security.

The heart of DFSMS is the storage management subsystem (SMS). Using SMS, the storage administrator defines policies that automate the management of storage and hardware devices. These policies describe data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements for the system.

DFSMS provides a range of automated data and space management functions:

- ▶ Eliminate, simplify, and automate tasks normally done by users or a storage administrator
- ▶ Improve storage space use
- ▶ Control external storage centrally
- ▶ Enable the storage administrator to manage storage growth

DFSMS makes it easier to convert to new device types, and takes advantage of what available hardware can do. DFSMS is a family of products, each one having specific function:

- ▶ **DFSMSdfp** is part of the z/OS base elements. Together with BCP, it forms the foundation of the z/OS operating system, performing the essential data, storage, and device management functions of the system:
 - Space allocation
 - Access methods
 - Support for the storage hardware to balance performance throughout the system
 - Program management tools
 - Applying the storage management policy throughout the SMS constructs

- ▶ **DFSMSdss** Data Set Services (dss) is a high-speed and high-capacity utility used to move data from disk to disk (copy) or disk to tape (dump) quickly and efficiently. It performs both logical dumps (to copy data) and physical dumps (to copy track images). DFSMSdss is an external interface to Concurrent Copy (CC) and SnapShot functions. DFSMSdss is also used by other DFSMS components to perform their functions.
- ▶ **DFSMShsm** Hierarchical Storage Manager (hsm) provides the following functions:
 - Full-volume dump and restore
 - Policy-based space management
 - Automatic and periodic data backup data set and volume levels
 - Data set backup (full and incremental)
 - Data set recovery
 - Aggregate backup and recovery support (ABARS)
 - Automatic or user-initiated migration and recall data sets from HSM database (disk or tape)

ABARS enables installations to define an aggregation of data to be backed up and restored as a logical entity. ABARS maintains the point in time relationship of the data in a specific aggregate group. In case of a disaster, when the application is brought online at the recovery location, the data is synchronized and the application can be started. The scope of the aggregation of data that is defined to ABARS is typically that of a critical application or applications.

ABARS finds where the data exists in the DFSMS storage device hierarchy (including user disk, user tape, DFSMShsm migration disk or tape volumes, and so on), and packages that data into 1 - 4 output files that can be physically or electronically sent to an offsite location.

ABARS also collects the metadata associated with the backed-up data, such as catalog information, generation data group (GDG) base information, or DFSMShsm control data set records for migrated data sets, and restores them along with the data.

- ▶ **DFSMSrmm** Removable Media Manager (rmm) is a full-function, construct-driven tape management feature that provides the following functions:
 - Manages tape volumes and data sets in systems (both automated and manual)
 - Keeps track of the locations where tapes are kept
 - Policy-driven Library Management

DFSMSrmm can manage the following components:

- A removable media library, which incorporates all other libraries:
 - System-managed tape libraries (for example, the automated IBM SystemStorage TS3500 Tape Library and the IBM SystemStorage Virtualization Engine TS700, formerly known as Virtual Tape Server)
 - Non-system-managed tape libraries or traditional tape libraries
- Storage locations that are onsite and offsite
- Storage locations defined as home locations

- ▶ **DFSMStvs** The Transactional VSAM Services (tvs) function enables batch jobs and CICS online transactions to update shared VSAM data sets concurrently.

Except for DFSMSdfp, which is a base element, all others features are exclusive and optional.

Network File System

Network File System (NFS) is a distributed file system that enables users to access UNIX files and directories that are located on remote computers as though they were local. NFS is independent of machine types, operating systems, and network architectures.

1.11 DFSMS Advanced Copy Services

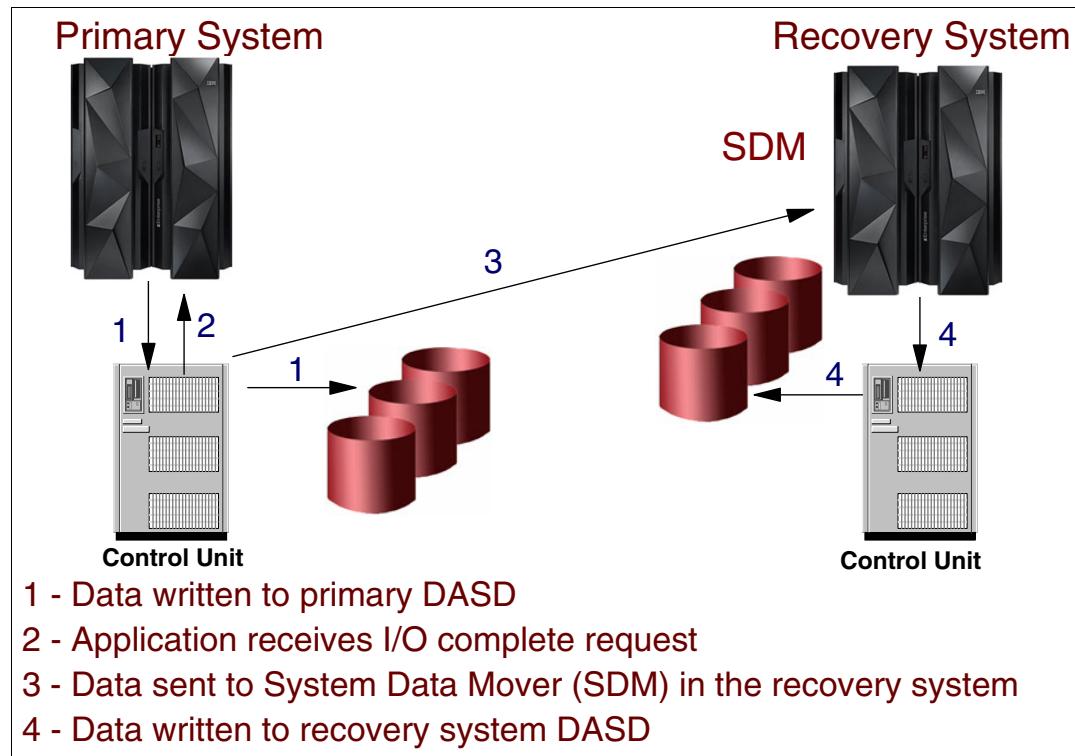


Figure 1-11 DFSMS Extended Remote Copy

DFSMS and Advanced Copy Services

DFSMS provides functions that aid installations in making copies of production data for the purpose of business continuance. DFSMS contains a software function called System Data Mover (SDM) that, when combined with the appropriate microcode on a disk storage subsystem (IBM or not), provides an extended distance remote copy capability called *extended remote copy* (XRC).

Mirror functions provide a consistent point-in-time copy of the data at the remote site, although the data at the remote site for Copy functions is not necessarily consistent.

The Advanced Copy Services functions shown in Table 1-1 fall into one of two groups. *Dynamic copy functions* constantly update the secondary copy as applications make changes to the primary data source. *Point-in-time copy functions* provide an instantaneous copy, or view, of what the original data looked like at a specific point in time.

Table 1-1 Advanced Copy Services functions

Type of Copy	Copy function name
Dynamic copy of data	XRC, coupled XRC (CXRC), Peer-to-Peer Remote Copy (PPRC), Global Mirror
Point-in-time copy of data	IBM FlashCopy®, SnapShot, and CC

Extended Remote Copy

XRC offers the highest levels of continuous data availability in a disaster recovery and workload movement environment. XRC provides asynchronous remote copy (mirror) of critical data over long distances to a second remote location with minimal effect.

XRC combines hardware and software to provide continuous data availability in a disaster recovery or workload movement environment. XRC provides an asynchronous remote copy solution for both system-managed and non-system-managed data to a second, remote location. XRC relies on the IBM TotalStorage Enterprise Storage Server®, IBM 3990, and DFSMSdfp.

XRC relies on the SDM, which is part of DFSMSdfp. SDM is a high-speed data movement program that efficiently and reliably moves large amounts of data between storage devices. XRC is a continuous copy operation, and it is capable of operating over long distances (with channel extenders). It runs unattended, without involvement from the application users. If an unrecoverable error occurs at your primary site, the only data that is lost is data that is in transit between the time when the primary system fails and the recovery at the recovery site.

Peer-to-Peer Remote Copy

PPRC is a hardware solution for rapid and accurate disaster recovery, and a solution to workload and DASD migration. Updates made on the primary DASD volumes are synchronously shadowed to the secondary DASD volumes.

In addition to the remote copy services functions, DFSMS provides three disk copy services:

- ▶ CC is a storage subsystem extended function that can generate a copy or a dump of data while applications are updating that data. CC uses DFSMSdss and works with SDM to control the process. DFSMSHsm uses the CC feature when starting DFSMSdss during its backup or dump processing.
- ▶ FlashCopy is a point-in-time copy services function that can quickly copy data from a source location to a target location.
- ▶ SnapShot is a point-in-time copy services function that enables you to “snap” (quickly copy) data directly from the source location to a target location.

1.12 z/OS Communications Server: TCP/IP

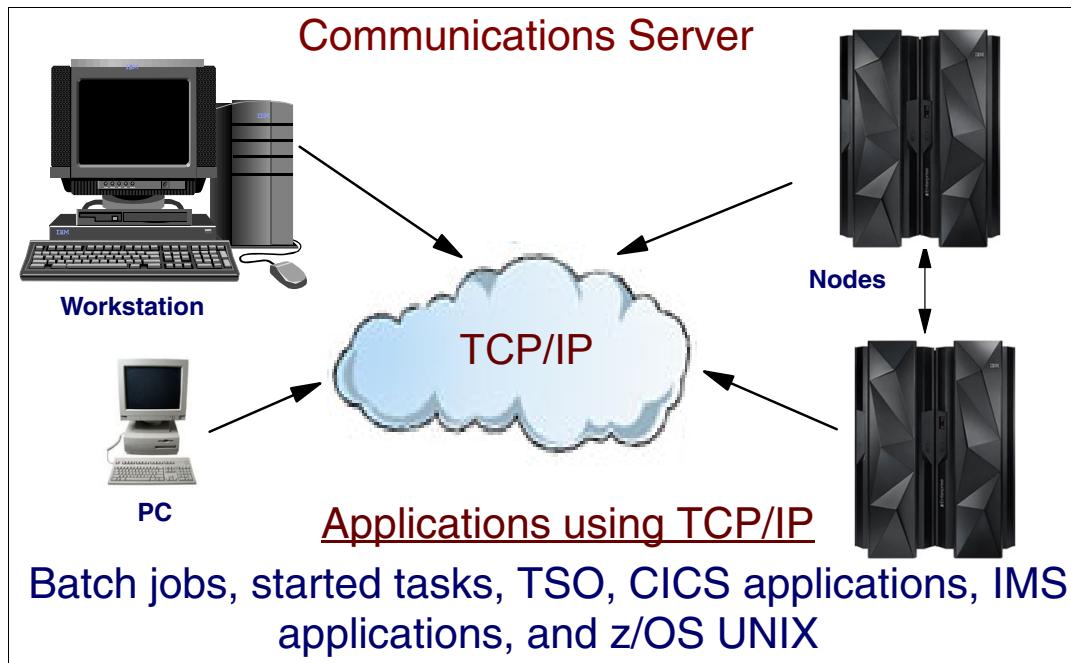


Figure 1-12 Transmission Control Protocol/Internet Protocol (TCP/IP)

Communications Server

z/OS Communications Server provides a set of communications protocols that support peer-to-peer connectivity functions for both LANs and WANs, including the most popular WAN, the Internet. z/OS Communications Server also provides performance enhancements that can benefit a variety of TCP/IP applications.

z/OS Communications Server provides both SNA and TCP/IP networking protocols for z/OS. The SNA protocols are provided by IBM VTAM, and include Subarea, Advanced Peer-to-Peer Networking, and High-Performance Routing (HPR) protocols.

The Communications Server protocol suite supports two TCP/IP environments:

- A native MVS environment where users can use the popular TCP/IP protocols in MVS application environments, such as batch jobs, started tasks, TSO, CICS applications, and Information Management System (IMS) applications.
- A z/OS UNIX environment that enables you to create and use applications that conform to the Portable Operating System Interface (POSIX) or XPG4 standard (a UNIX specification).

TCP/IP

TCP/IP is a set of protocols and applications that enable you to perform certain computer functions in a similar manner independent of the types of computers or networks being used. When you use TCP/IP, you are using a network of computers to communicate with other users, share data with each other, and share the processing resources of the computers connected to the Internet Protocol network.

Computer network

A computer network is a group of computer nodes electronically connected by a communication medium. Each node has the hardware and the programs necessary to communicate with other computer nodes across this communication medium. The node can be a personal computer (PC), workstation, departmental computer, or large computer system. The size of the computer is not important, but the ability to communicate with other nodes is.

Computer networks enable you to share the data and computing resources of many computers. Applications, such as departmental file servers, rely on networking as a way to share data and programs.

Many forms of communication media are available today. Each is designed to take advantage of the environment on which it operates. Communication media consist of a combination of the physical network used to connect to computer nodes and the language, or protocol, that they use to communicate with each other.

1.13 System Modification Program Extended

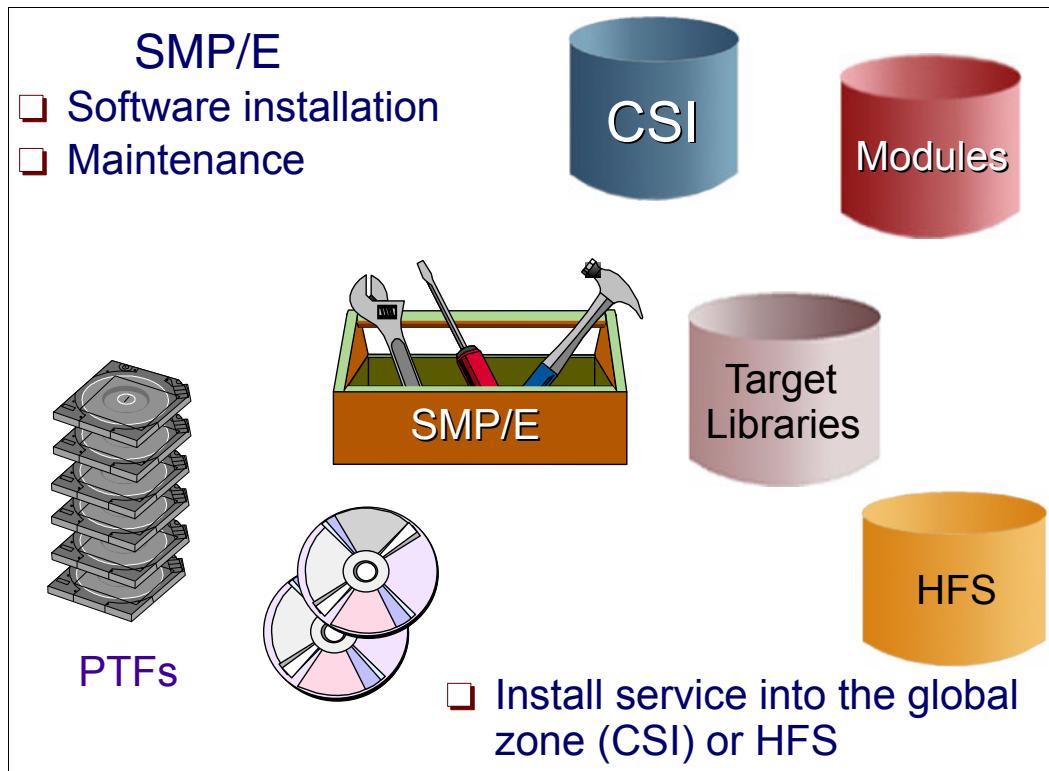


Figure 1-13 System Modification Program Extended

System Modification Program Extended

SMP/E is a software tool designed to manage the installation of software products on your z/OS system, and to track the modifications applied to those products. Usually, it is the system programmer's responsibility to ensure that all software products and modifications are correctly installed on the system, and that all products are installed at the correct level (for corrections and functions) so that all elements of the system can work together.

A z/OS system can appear to be one big block of code that drives the processor and the I/O. Actually, z/OS is a complex system comprising many separate smaller blocks of code. Each of those smaller blocks of code performs a specific function in the system. Each system function is composed of one or more load modules.

In a z/OS, a load module represents the basic unit of machine-readable executable code. Load modules are created by combining one or more object modules and processing them with a link-edit utility. Link editing of modules is a process that resolves external references and addresses. Functions on a system, therefore, are one or more object modules that have been combined and link edited.

Over time, you might need to change the elements of your system. These changes might be necessary to improve the usability or reliability of a product. You might need to add new functions to your system, upgrade the elements of your system, or modify elements for a variety of reasons. In all cases, you are making system modifications.

All executable software code is subject to errors. Any errors in the IBM software code are fixed by IBM and made available to installations in the form of either an authorized program analysis report (APAR) or program temporary fix (PTF). SMP/E ensures that all of the corrections are applied adequately, respecting co-requisites and precedence relations.

PTFs

When a problem with a software element is discovered, IBM supplies its clients with a tested fix for that problem. This fix comes in the form of a PTF. Although you might not have experienced the problem the PTF is intended to prevent, it is wise to install the PTF on your system. The PTF system modification (SYSMOD) is used to install the PTF, thereby preventing the occurrence of that problem on your system.

CSI data sets

The CSI data sets contain all of the information that SMP/E needs to track the distribution and target libraries. As an example, just as a card catalog contains a card for each book in a library, the CSI contains an entry for each element in its libraries. The CSI entries contain the element name, type, history, description of how the element was introduced into the system, and a pointer to the element in the distribution and target libraries. The CSI does not contain the element itself, but rather a *description* of the element that it represents.

In addition to the distribution and target zones, the SMP/E CSI also contains a global zone. The global zone contains the following elements:

- ▶ Entries needed to identify and describe each target and distribution zone to SMP/E
- ▶ Information about SMP/E processing options
- ▶ Status information for all SYSMODs that SMP/E has begun to process
- ▶ Exception data for SYSMODs requiring special handling, or that are in error

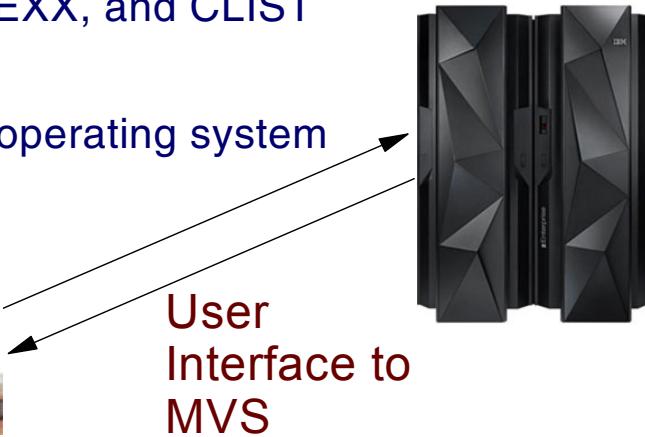
Installing service

Corrective service is installed into the target libraries. For z/OS UNIX, the PTFs are installed into the HFS. After you have installed the corrective service into the target libraries, you must decide whether you want to update the distribution libraries. Base this decision on the products involved, and on your processing requirements.

Information: For additional information about SMP/E, see the SMP/E chapter in *ABCs of z/OS System Programming Volume 2*, SG24-6982.

1.14 Time Sharing Option/Extended

- Communicate with other TSO/E users
- Create an office environment
- Develop and maintain programs in languages such as assembler, COBOL, FORTRAN, PASCAL, PL/I, REXX, and CLIST
- Process data
- Access the MVS operating system



User
Interface to
MVS

Figure 1-14 Time Sharing Option/Extended

Time Sharing Option/Extended

TSO/E is a base element of the z/OS operating system that enables users to interactively work with the system. It looks like the UNIX Shell. In general, TSO/E makes it easier for people with all levels of experience to interact with the z/OS system.

TSO/E provides benefits for a wide range of computer users. TSO/E users include system programmers, application programmers, information center administrators, information center users, TSO/E administrators, and others who access applications that run under TSO/E.

TSO/E environments

You can use TSO/E in any one of the following environments:

- ▶ Line Mode TSO/E. This involves using TSO/E commands typed on a terminal, one line at a time. This is a quick and direct way to use TSO/E, and it was how programmers originally used to communicate interactively with the z/OS operating system.
- ▶ ISPF/PDF. The ISPF/PDF works together with TSO/E to provide panels with which users can interact. ISPF provides a dialog management service that displays panels and enables a user to navigate through the panels. ISPF/PDF is a dialog of ISPF that helps maintain libraries (z/OS libraries included) of information in TSO/E and enables a user to manage the library through facilities, such as browse, edit, and utilities.
- ▶ Information Center facility. This is a set of panels that enables you to display services, such as mail and names directory, perform data analysis, and prepare documents, such as reports, graphs, and charts.

Command list language

The command list (CLIST) language is a high-level programming language that enables programmers to issue lists of TSO/E commands and job control language (JCL) statements in combination with logical, arithmetic, and string-handling functions provided by the language. The programs, called CLISTS, can simplify routine user tasks, start programs written in other languages, and perform complex programming functions.

REXX EXECs

REXX is a programming language that is extremely versatile. Aspects, such as common programming structure, readability, and free format, make it a useful language for beginners and general users. Furthermore, because the REXX language can be intermixed with commands to other host environments, it provides powerful functions and has extensive mathematical capabilities, so it is also suitable for more experienced computer professionals.

The TSO/E implementation of the REXX language enables REXX execs to run in any MVS address space. You can write a REXX exec that includes TSO/E services and run it in a TSO/E address space, or you can write an application in REXX to run outside of a TSO/E address space.

TSO/E commands and programs

Programs contain instructions that perform tasks. Common programming languages used under TSO/E include assembly language, COBOL, Fortran, Pascal, PL/I, REXX, and CLIST. REXX and CLIST are high-level interpretive programming languages that enable you to combine TSO/E commands with language statements.

The following commands can be used to compile or assemble source statements written in the various languages:

- ▶ **ASM** command
- ▶ **COBOL** command
- ▶ **FORT** command
- ▶ **PLI** command

1.15 z/OS UNIX System Services

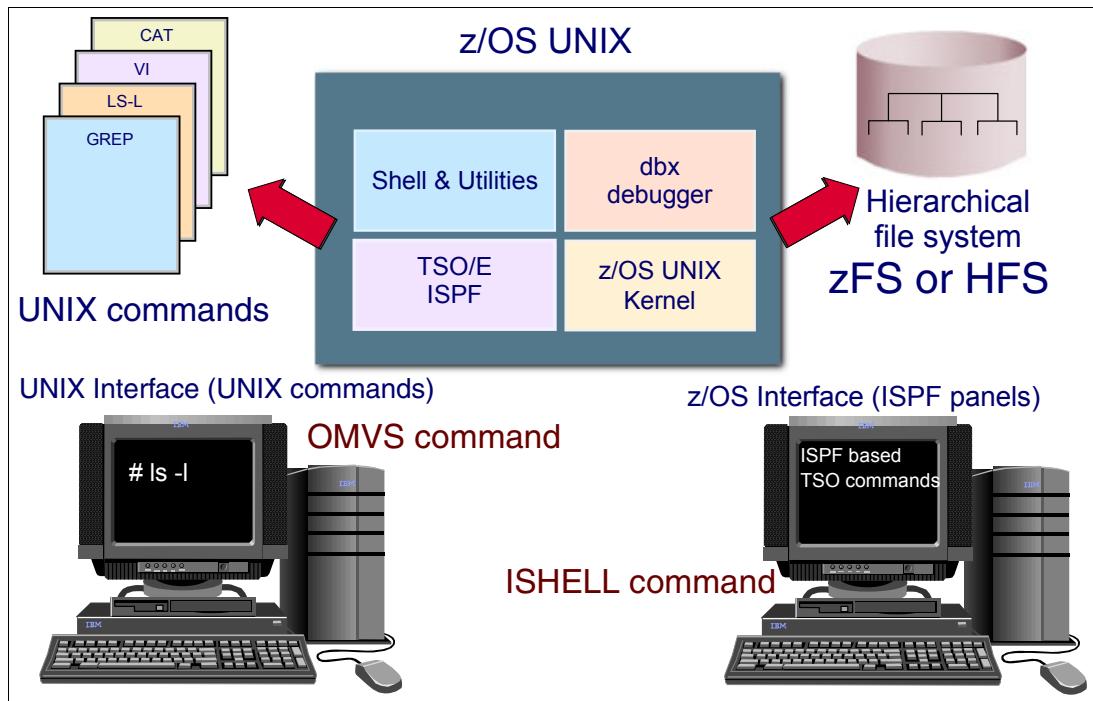


Figure 1-15 z/OS UNIX System Services

z/OS UNIX System Services

Beginning with OS/390 V2R4, z/OS UNIX has been merged with the BCP and is now part of the BCP FMID. The OpenEdition MVS (OMVS) address space (the kernel of z/OS UNIX) is started automatically at initial program load (IPL) by means of the OMVS statement in the IEASYSxx SYS1.PARMLIB member. The BPXOINIT address space is the started procedure that runs the initialization process.

ISPF shell

The ISPF shell is a panel interface that you can use rather than TSO/E commands or shell commands to perform certain tasks. For example, you can use the ISPF shell to display all mounted file systems or their attributes, such as total blocks.

HFS and zFS

When IBM created OpenEdition (now called z/OS UNIX) as part of MVS, the physical file system (PFS) that performed the I/O to the file system was called HFS. Beginning with z/OS V1R2, a second PFS called zFS was added. In current z/OS operating systems, both the HFS and zFS PFSs can be used to access file system data. See 1.16, “z/OS UNIX physical file systems” on page 35 for more information about these topics.

User-written programs use the POSIX API to issue file requests. These requests are routed by the logical file system (LFS) to the appropriate PFS through the PFS interface.

z/OS UNIX kernel

The kernel address space contains the MVS support for z/OS UNIX services. This address space can also be called the *kernel*, and is the part of z/OS UNIX that contains programs for such tasks as I/O, management, control of hardware, and the scheduling of user tasks.

ISHELL or OMVS shell

Interactive users of z/OS UNIX have a choice between using a UNIX-like interface (the shell), a TSO interface (TSO commands), and an ISPF interface (ISPF Common User Access (CUA) dialog). With these choices, users can choose the interface with which they are most familiar, and get a quicker start on z/OS UNIX.

The z/OS UNIX shell provides the environment that has the most functions and capabilities. Shell commands can easily be combined in pipes or shell scripts, and thereby become powerful new functions. A sequence of shell commands can be stored in a text file, which can be run. This is called a *shell script*. The shell supports many of the features of a regular programming language.

z/OS UNIX dbx debugger

The z/OS UNIX dbx debugger is an interactive tool for debugging C language programs that use z/OS UNIX. It is based upon the dbx debugger, which is regarded as an industry standard on UNIX systems. The dbx debugger provides the options to debug at source level or assembler level.

Source-level debugging enables you to debug C language programs. Assembler-level debugging enables you to debug executable programs at machine code level. The dbx debugger is a utility that is started from the z/OS UNIX shell. It cannot be started directly from TSO/E.

Shell and utilities

UNIX System Services Application Services provide the following functionality:

- ▶ A TSO/E command to enter the shell environment
- ▶ A shell environment for developing and running applications
- ▶ Utilities to administer and develop in a UNIX environment

z/OS UNIX interactions with z/OS

z/OS UNIX interacts with the following elements and features of z/OS:

- ▶ C/C++ Compiler, to compile programs
- ▶ Language Environment, to run the shell and utilities
- ▶ Data Facility Storage Management Subsystem, to access HFS/zFS files
- ▶ z/OS Security Server
- ▶ IBM Resource Measurement Facility™ (RMF)
- ▶ System Display and Search Facility (SDSF)
- ▶ Time Sharing Option Extensions (TSO/E)
- ▶ TCP/IP Services
- ▶ ISPF, to use the dialogs for OEDIT, or ISPF/PDF for the ISPF shell
- ▶ BookManager READ, to use the OHELP online help facility

1.16 z/OS UNIX physical file systems

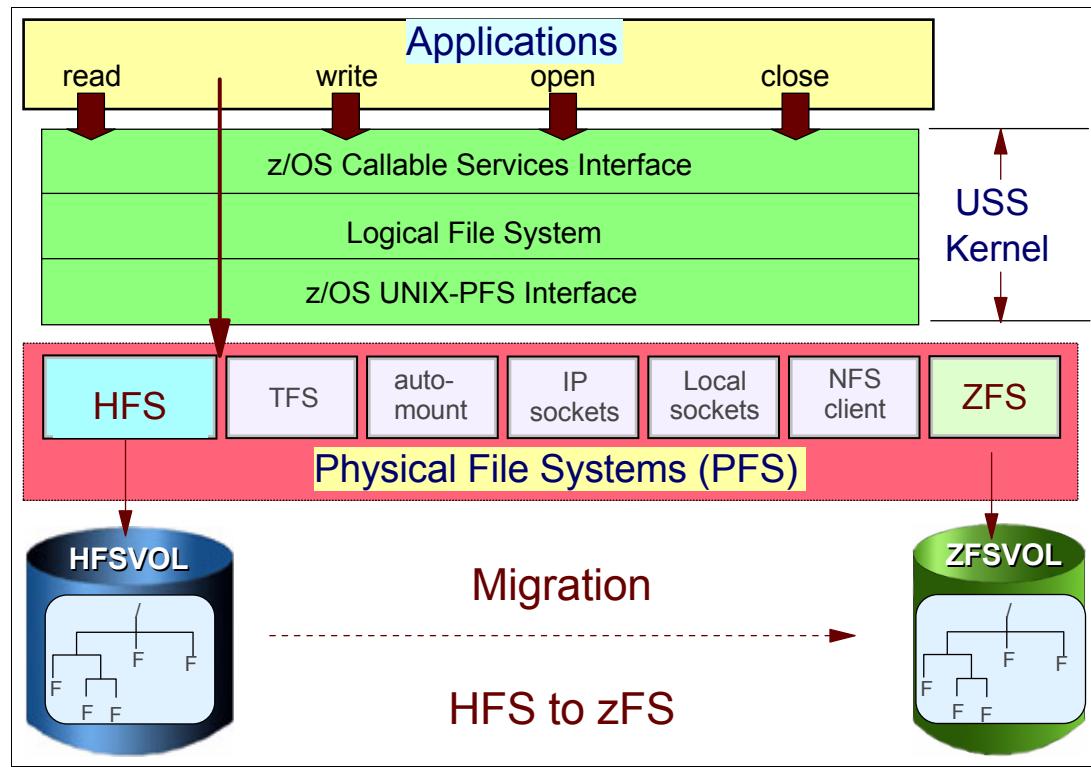


Figure 1-16 z/OS UNIX physical file systems

z/OS UNIX physical file systems

A PFS is packaged as one or more MVS load modules. These load modules must be installed in an authorized program facility (APF)-authorized MVS load library. The hierarchical file system is not available when a PFS is loaded, so it cannot be installed in the file system.

A PFS controls access to data. PFSs receive and act upon requests to read and write files that they control. The format of these requests is defined by the PFS interface. In a z/OS UNIX, the physical file systems are defined in the BPXPRMxx PARMLIB member. zFS, as a physical file system, is also defined in the PARMLIB member. Figure 1-16 shows all the physical file systems that can be defined in a z/OS UNIX System Services environment. The logical file system (LFS) is called by POSIX programs, non-POSIX z/OS UNIX programs, and VFS servers.

PFS interface

The PFS interface is a set of protocols and calling interfaces between the LFS and the PFSs that are installed on z/OS UNIX. PFSs mount and unmount file systems and perform other file operations.

There are two types of PFSs, those that manage files and those that manage sockets, as explained here:

- ▶ File management PFSs, such as HFS and zFS, deal with objects that have path names, and that generally follow the semantics of POSIX files.
- ▶ Socket PFSs deal with objects that are created by the `socket()` and `accept()` functions, and that follow socket semantics.

HFS-to-zFS migration

Because IBM has announced the stabilization of the HFS PFS, a migration of HFS file systems (both mounted and unmounted) to zFS file systems must occur over time.

Advantages of zFS

Like HFS, zFS is a UNIX file system. It contains files and directories that can be accessed with the APIs available for HFS. In general, the application view of zFS is the same as the application view of HFS. After a zFS file system is mounted, it is almost indistinguishable from any other mounted HFS. The benefits of using zFS are listed here:

- ▶ Improved performance

zFS provides significant performance gains in many client environments accessing files approaching 8K in size that are frequently accessed and updated. The access performance of smaller files is equivalent to HFS.

- ▶ Underlying architecture supports additional functions

Only zFS supports security labels. Therefore, in a multilevel-secure environment, you must use zFS file systems rather than HFS file systems.

As an optional function, zFS enables the administrator to make a read-only clone of a file system in the same data set. This clone file system can be made available to users to provide a read-only point-in-time copy of a file system.

zFS runs as a z/OS UNIX colony address space. Therefore, zFS can be stopped using the **p zfs** operator command. Before stopping zFS, unmount zFS file systems or move them to another sysplex member.

- ▶ Improved crash recovery

zFS provides a reduction in exposure to loss of updates. zFS writes data blocks asynchronously and does not wait for a sync interval. zFS is a logging file system. It logs metadata updates. If a system failure occurs, zFS replays the log when it comes back up to ensure that the file system is consistent.

1.17 System Management Facility



Figure 1-17 System Management Facility (SMF)

System Management Facility

SMF records in SYS1.MANxx data sets, system and job-related information that your installation can use in:

- ▶ Billing users
- ▶ Reporting reliability
- ▶ Analyzing the configuration
- ▶ Scheduling jobs
- ▶ Summarizing direct-access volume activity
- ▶ Evaluating data set activity
- ▶ Profiling system resource use
- ▶ Maintaining and auditing system security

SMF data collection

The data collection is run by several specific routines spread all over z/OS and other products. This collection is done into system-related or job-related records.

System-related SMF records include information about the configuration, paging activity, and workload. *Job-related* records include information about the processor time, SYSOUT activity, and data set activity of each job step, job, Advanced Program-to-Program Communication (APPC)/MVS transaction program, Started Task Control (STC) address space, and TSO/E session.

The volume and variety of information in the SMF records enables installations to produce many types of analysis reports and summary reports. For example, by keeping historical SMF data and studying its trends, an installation can evaluate changes in the configuration, workload, or job scheduling procedures. Similarly, an installation can use SMF data to determine system resources wasted because of problems, such as inefficient operational procedures or programming conventions.

Installation SMF routines

An installation can provide its own routines as part of SMF. These routines will receive control either at a particular point as a job moves through the system, or when a specific event occurs. For example, an installation-written routine can receive control when the processor time limit for a job expires, or when an initiator selects the job for processing. The routine can collect additional information, or enforce installation standards.

1.18 Resource Management Facility

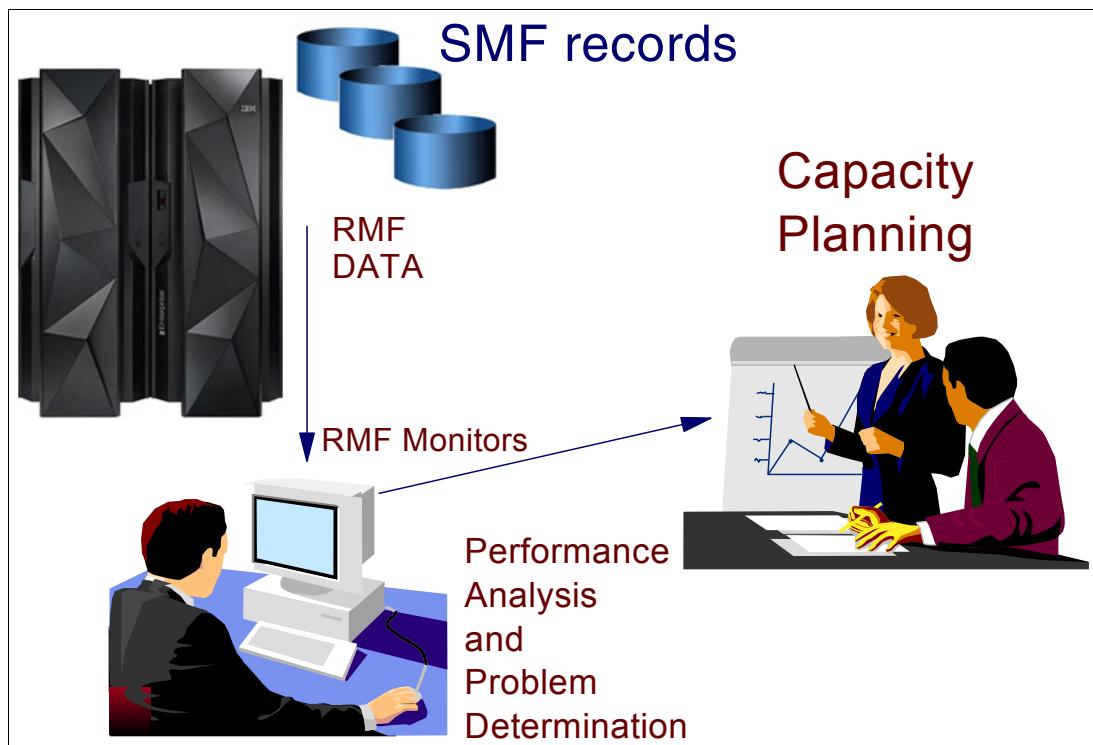


Figure 1-18 Resource Management Facility

Resource Management Facility

Many activities are required to keep your z/OS running smoothly, and to provide the best service on the basis of the available resources and workload requirements. The console operator, the service administrator, the system programmer, or the performance analyst does these tasks. z/OS Resource Management Facility (RMF) is the tool (online and in batch reports) that helps each of these people accomplish the job effectively, as far as the performance of the system is concerned. RMF gathers performance data using three monitors:

- ▶ Short-term data collection with Monitor III
- ▶ Snapshot monitoring with Monitor II
- ▶ Long-term data gathering with Monitor I

SMF records

Data is gathered for a specific cycle time, and consolidated data records are written (usually as SMF records) at a specific interval time. The default value for data gathering is one second and for data recording is 30 minutes. You can select these options according to your requirements and change them whenever the need arises.

Each SMF record contains information similar to the contents of the corresponding formatted report. For each system activity that you select, RMF collects data and formats an SMF record to hold the data it collects. Various totals, averages, and percentages are not explicitly contained in the SMF records, but are calculated from the SMF data.

Depending on the feedback options you select, RMF can write the SMF records to the SMF data set, use the data in the record to generate a printed report, or both. Regardless of the options you select, the format of the SMF record is the same.

RMF Monitor I, II, and III

RMF gathers data using three monitors:

- ▶ Short-term data collection with Monitor III
- ▶ Snapshot monitoring with Monitor II
- ▶ Long-term data gathering with Monitor I and Monitor III

Monitor I collects long-term data about system workload and resource usage, and covers all hardware and software components of your system:

- ▶ Processor
- ▶ I/O device and storage activities and usage
- ▶ Resource use
- ▶ Resource activity
- ▶ Performance of groups of address spaces

Monitor II simply takes snapshots of certain z/OS key fields. Monitor III uses the concept of Contention Analysis (address space using and delay states) for gathering data. An *address space* is a set of virtual addresses that a program can refer to. In z/OS, each set of logically related programs shares the same address space containing up to 2,000,000,000 addresses.

All three monitors can create reports.

1.19 Workload Manager

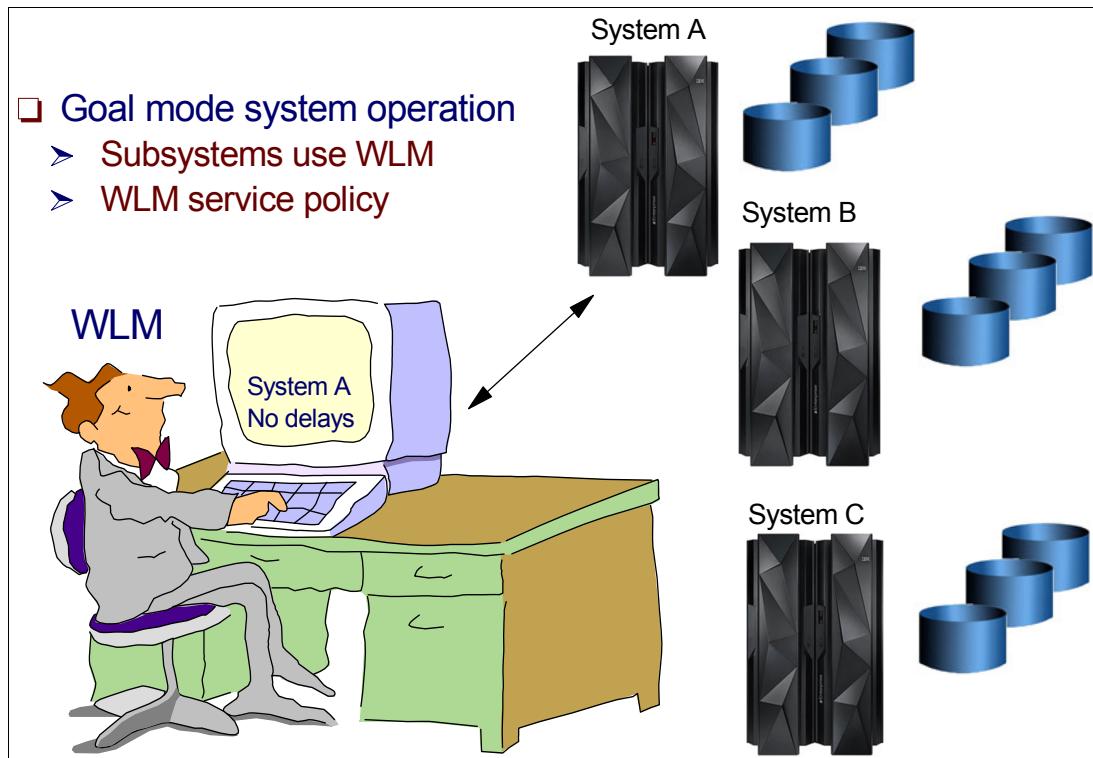


Figure 1-19 Workload Manager (WLM)

Workload management

Before the introduction of z/OS Workload Manager (WLM), the only way to inform the z/OS about your business goal (such as transaction response time) was to translate from high-level objectives about what transaction needs to be done into the extremely technical terms that the system can understand. This translation required highly skilled staff and could be protracted, error-prone, and eventually in conflict with the original business goals.

Additionally, it was often difficult to predict the effects of changing a parameter that might be required, for example, after a system capacity increase. This could result in unbalanced resource allocation. This method of operation, known as *compatibility mode*, was becoming unmanageable as new workloads were introduced, and as multiple systems were being managed together in Parallel Sysplex processing and data sharing environments.

WLM goal mode

When in *goal mode* system operation, WLM provides fewer, simpler, and more consistent system externals that reflect goals for transactions expressed in terms commonly used in business objectives, and WLM match resources (by setting priorities in resource queues) to meet those goals by constantly monitoring and adapting the system. WLM provides a solution for managing workload distribution, workload balancing, and distributing resources to competing workloads.

Subsystems using WLM

MVS workload management provides a solution for managing workload distribution, workload balancing, and distributing resources to competing workloads. MVS workload management is the combined cooperation of various subsystems (IBM CICS, IBM IMS/ESA®, JES, APPC, TSO/E, z/OS UNIX System Services, distributed data facility (DDF), IBM DB2®, System Object Model (SOM), LSFM, and Internet Connection Server) with the MVS WLM component.

WLM service policy

Workload management provides new MVS performance management externals in a service policy that reflects goals for work, expressed in terms commonly used in service level agreements (SLAs). Because the terms are similar to those commonly used in an SLA, you can communicate with users, with IBM Business Partners, and with MVS using the same terminology.

Using one common terminology, workload management provides feedback to support performance reporting, analysis, and modeling tools. The feedback describes performance achievements in the same terms as those used to express goals for work. Workload management eliminates the need to micro-manage each individual MVS image, thereby providing a way for you to increase the number of systems in your installation without greatly increasing the necessary skill level.

1.20 Infoprint Server

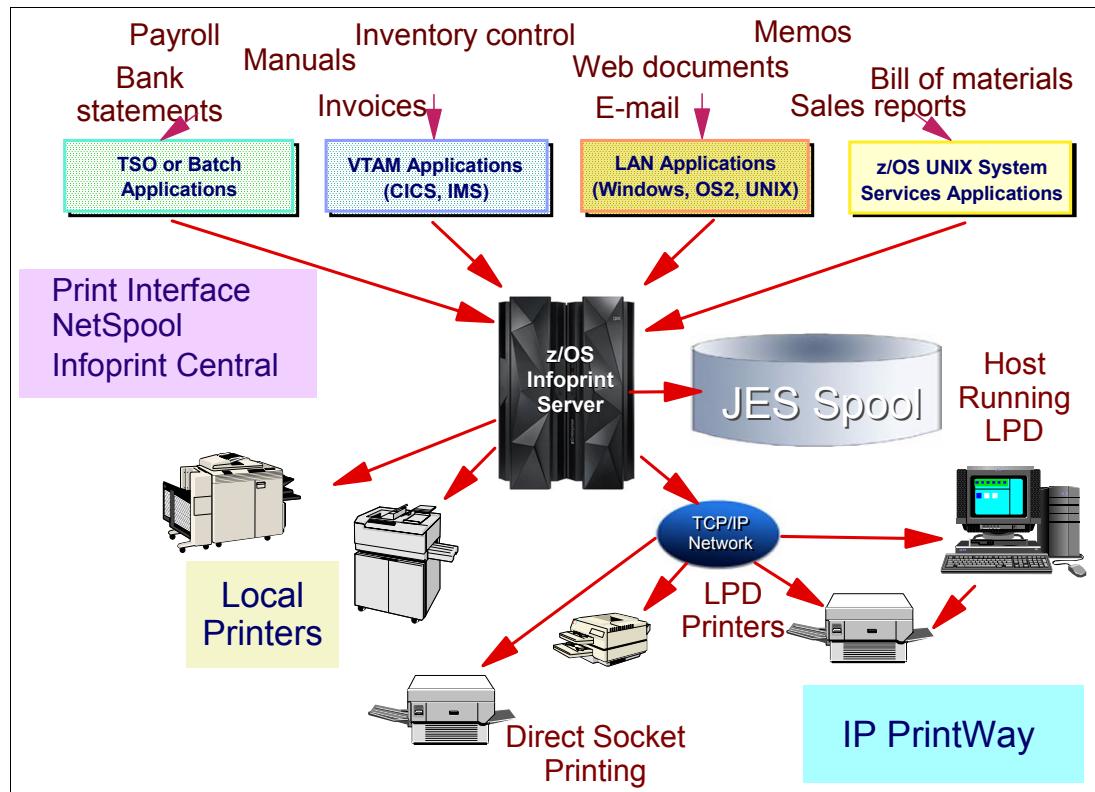


Figure 1-20 Infoprint functions and features

Infoprint Server overview

Infoprint Server is an optional feature of z/OS Version 1 Release 1 and higher. Infoprint Server is a UNIX application that uses z/OS UNIX System Services. This feature is the basis for a total print serving solution for the z/OS environment on an Internet Protocol network.

Infoprint Server enables users to submit print requests from remote workstations on an Internet Protocol network, from UNIX System Services applications, from batch applications, and from VTAM applications, such as CICS or IMS applications. It enables you to consolidate your print workload from the 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 and improve manageability, data retrievability, and usability.

Print Interface Print Interface is the component of Infoprint Server that processes print requests received from both remote clients and local users. When the Print Interface receives a print request, it allocates an output data set on the JES spool.

IP PrintWay IP PrintWay is the component of Infoprint Server that transmits output data sets from the JES2 or JES3 spool to network printers, or to other host systems in your Internet Protocol network. The remote printer or host system must support either the line printer requestor/line printer daemon (LPR/LPD) protocol, the Internet Printing Protocol (IPP), or direct socket printing.

IP PrintWay can give you fast access to TCP/IP-connected printers, and to VTAM-controlled printers.

NetSpool	NetSpool intercepts print data from VTAM applications, such as CICS and IMS; transforms the data streams to EBCDIC line data, printer control language (PCL), PDF, or other formats that the target printer accepts; and writes the output data set to the JES spool. JES or PSF can print the output data sets, or JES can transmit them to other locations for printing. IP PrintWay can transmit the output data sets to remote printers in your Internet Protocol network.
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: <ul style="list-style-type: none">▶ Work with print jobs▶ Work with printers▶ Work with NetSpool logical units (LUs)▶ Display printer definitions▶ Check system status

Information: For a detailed description of Infoprint Server, see *ABCs of z/OS System Programming Volume 7*, SG24-6987.

1.21 IBM z/OS Management Facility

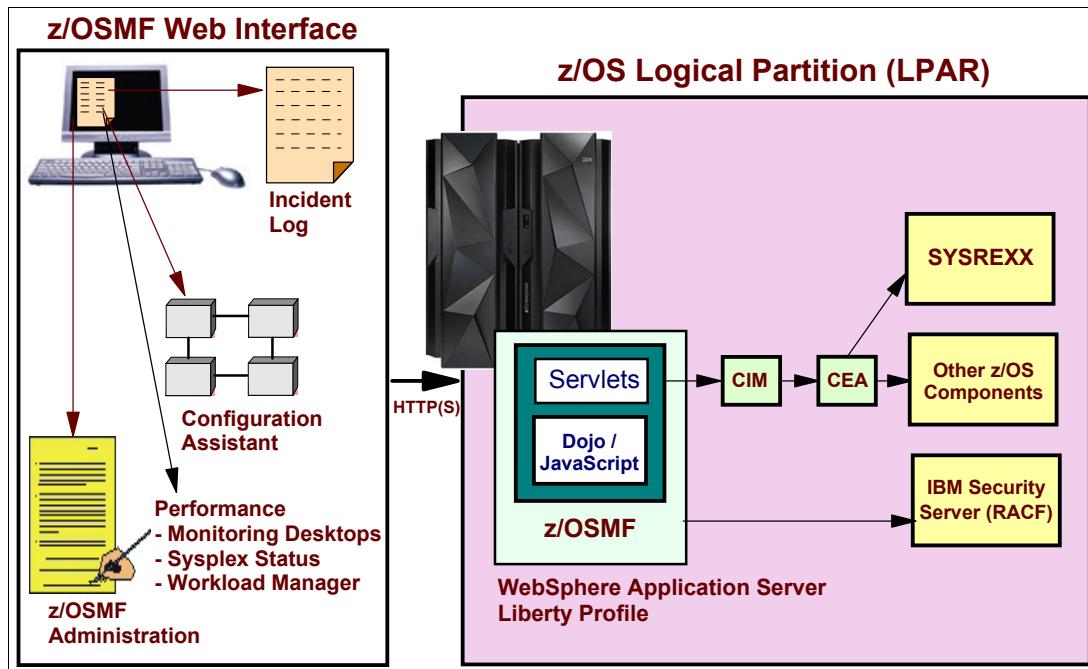


Figure 1-21 z/OSMF and related system components

z/OS Management Facility

The goal of the IBM z/OS Management Facility (z/OSMF) architecture is to provide simplified systems management functions through a common, easy-to-use GUI. Figure 1-21 shows a typical architecture and flow, starting with the user's browser session and continuing through z/OSMF and IBM WebSphere Application Server Liberty Profile, with information passed to various z/OS system components as needed.

z/OSMF is a central system management function for z/OS and is designed to provide better tools for managing systems and helping system programmers to be more productive. z/OSMF provides a framework for managing various aspects of z/OS systems through an intuitive Web 2.0 browser user interface and new enabling technologies on z/OS.

Related system components

Structurally, z/OSMF is a set of web applications hosted on your z/OS system. Depending on the system management task to be performed, z/OSMF interfaces with other z/OS components to offer a simplified interface for performing tasks. These components make up the environment necessary for using the z/OSMF functions. z/OSMF does not provide a separate client installation.

You will need to provide a compatible browser to access the z/OSMF web application. As an example, Figure 1-21 shows the architecture flow for a z/OSMF task, starting with z/OSMF and continuing through WebSphere Application Server Liberty Profile, with information passed to the CIM server, through CEA, and finally System REXX.

Economics

z/OSMF is a zero-priced product, so the decision to get started can be fast and easy. A large portion of the z/OSMF application is written in Java, and therefore, this portion is eligible for the IBM System z Application Assist Processor (zAAP).

In addition, portions of z/OSMF workload use the z/OS CIM server, and therefore, these portions are eligible for IBM System z Integrated Information Processor (zIIP).

Integrated Lightweight Application Server

As of z/OSMF V2R1, z/OSMF runs using the WebSphere Application Server Liberty profile, which is a dynamic profile of the WebSphere Application Server that enables the WebSphere Application Server to provision only the features required by z/OSMF. The WebSphere Application Server Liberty profile starts very quickly (a few seconds), has a much smaller footprint, and is included in the z/OSMF package.

Enhanced Security

As of z/OSMF V1R13, the security model was enhanced to provide better integration with Enterprise Security Management products. The z/OSMF System Authorization Facility (SAF) mode authorization support brings tighter integration with z/OS SAF-based authorization with the introduction of the resource class ZMFAPLA for z/OSMF task-based resources.

All z/OSMF tasks and links will be associated with resource names and resource class profiles under this resource class, and SAF groups are used to represent Roles. SAF mode further enables custom roles (by creating SAF groups at the customer's discretion).

Flexible Framework

z/OSMF provides a modular framework that z/OSMF administrators can use to bring together z/OS system management applications. z/OSMF supports the following types of integration.

Adding resources links

z/OSMF administrators can use the Links task to add links to non-z/OSMF resources, such as independent software vendor (ISV) products and commonly used installation websites, under any category in the z/OSMF navigation tree. The purpose here is to enable z/OSMF administrators to create a common list of frequently used information and disseminate it to the entire team so that operators and system programmers have the ability to access more information and tools quickly.

Creating application links

As of z/OSMF V1R13, z/OSMF administrators can create context-sensitive launch points between z/OSMF applications and other browser-based applications. Doing so can provide a more seamless experience for system programmers as they work with different tools and tasks on the z/OS system. For example, administrators can enable system programmers to go directly from the z/OSMF Incident Log task to ISPF browse to view the log snapshot for an incident.

Adding new functions

As of z/OSMF V2R1 with APAR PM98630, z/OSMF administrators can add installation-specific functions to z/OSMF in the form of plug-ins. The z/OSMF framework provides the infrastructure, security, and services needed to integrate a new web-based application into z/OSMF. Adding installation-specific functions to z/OSMF helps simplify the management of z/OS mainframe systems by reducing context shifts between disparate applications.

Improved Communication

As of z/OSMF V2R1, the Notifications task is provided, which notifies you when something occurs on the system that requires your awareness or response. Notifications might be informational in nature, or might be requests for action from other z/OSMF tasks. With the Notifications task, you can view and work with the notifications that are assigned to you.

RESTful Services

z/OSMF supports the use of Representational State Transfer (REST) APIs, which are public APIs that an HTTP client application can use to work with system resources and extract system data. The z/OSMF APIs support easy-to-use HTTP services that are language-independent and platform-independent, stateless, scalable, and easily parsed. z/OSMF provides the following APIs.

Application Linking Manager services

This API enables an HTTP client application to register and unregister event types and event handlers, to list all of the tasks that are eligible to be handlers, and to list the handlers that are registered for an event type.

Data persistence services

As of z/OSMF V2R1 with APAR PM98630, this API enables an HTTP client application to persist, retrieve, and delete user-specific and global application data.

TSO/E address space services

As of z/OSMF V2R1 with APAR PM98630, this API enables an HTTP client application to start, stop, ping, send messages to, and receive messages from TSO/E address spaces and to manage the applications running in a TSO/E address space.

z/OS data set and file REST interface

As of z/OSMF V2R1 with APAR PM98630, this API enables an HTTP client application to work with z/OS data sets and UNIX files on the z/OSMF host system. Specifically, HTTP client applications can list data set names and list z/OS UNIX directories or files.

z/OS jobs REST interface

This API enables an HTTP client application to perform operations with batch jobs on a z/OS system. Operations, such as submit a job to run on z/OS, obtain the status of a job, list the spool files for a job, retrieve the contents of a job spool file, cancel a job, or delete a job from the JES spool, can be performed.

Customization Options

To customize z/OSMF, use one or more of the following capabilities.

Systems and File Transfer Protocol Servers

As of z/OSMF V1R13 with APAR PM74502, you can use the Systems task to define the settings required for z/OSMF to access the systems in your installation and to specify the HTTP proxy settings for z/OSMF to use when establishing an HTTPS connection to another system. You can use the File Transfer Protocol (FTP) Servers task to provide the settings required for z/OSMF to access the FTP servers that are running on internal or external systems, and to specify the settings for z/OSMF to use when transferring files between systems.

Welcome properties file

z/OSMF administrators can use the sample Welcome properties file provided with z/OSMF to customize the z/OSMF Welcome panel. The purpose here is that z/OSMF administrators can provide users with information they should read before logging in to z/OSMF, such as installation-specific instructions. Administrators can even add a small image or graphic, such as your company logo and header and footer text suited for your installation.

1.22 z/OSMF Configuration Assistant task

Name	Description
ADNR	(VERIFY) IBM supplied: Automated Domain Name Registration traffic
Centralized_Policy_Server	(VERIFY) IBM supplied: Centralized Policy Server
CICS	(VERIFY) IBM supplied: CICS traffic
CSSMTP	(VERIFY) IBM supplied: CSSMTP traffic
FTP-Client	(VERIFY) IBM supplied: FTP Client traffic
FTP-Server	(VERIFY) IBM supplied: FTP Server traffic
LBA-Advisor	(VERIFY) IBM supplied: z/OS Load Balancing Advisor traffic
LBA-Agent	(VERIFY) IBM supplied: z/OS Load Balancing Advisor - Agent traffic
LDAP-Server	(VERIFY) IBM supplied: LDAP Server traffic
MSM	IBM supplied: MultiSystem Manager for IBM Tivoli Network Manager-IP traffic
NETCONV	IBM supplied: NETCONV traffic

Figure 1-22 z/OSMF Configuration Assistant task

z/OSMF Configuration Assistant task

The z/OSMF Configuration Assistant for z/OS Communications Server task provides assistance in configuring TCP/IP networking policies and can help dramatically reduce the amount of time required to create network configuration files. For example, it used to take hours to set up TCP/IP filters, but that can now potentially take as little as 30 minutes.

With the Configuration Assistant task, for any number of z/OS images and TCP/IP stacks, you can create configuration files for the following functions:

- ▶ Application transparent: Transport layer security
- ▶ IP security
- ▶ Intrusion detection services
- ▶ Network security services
- ▶ Quality of service (QoS)
- ▶ Policy-based routing
- ▶ Defense manager daemon

To simplify things even more, step-by-step tasks are provided, complete with setup for RACF security, started procedures, and Policy Agent configuration. Health checks are provided to ensure the consistency and accuracy of policy definitions.

1.23 z/OSMF Capacity Provisioning task

The screenshot shows the 'Capacity Provisioning' task within the IBM z/OS Management Facility. The top navigation bar includes 'IBM z/OS Management Facility', 'Welcome karan', 'Log out', and the 'IBM' logo. A search bar contains 'Capacity Pro...'. On the right, there are 'Help' and 'Capacity Provisioning' links. Below the search bar, a 'Capacity Provisioning' section is visible. Under 'Overview', it says: 'z/OS Capacity Provisioning helps you manage additional capacity of System z servers. The Provisioning Manager monitors the workload on a set of z/OS systems and organizes the allocation of On/Off Capacity on Demand and Defined Capacity. Define the systems to be observed in a domain configuration and the rules for additional capacity allocation in a policy.' It also states: 'Use this task to work with Capacity Provisioning. To get started, select one of the following actions.' A 'View Status and Define Connections' section lists 'Provisioning Manager' (link to status), 'Manage' (sections for 'Domain Configurations' and 'Policies'), and 'Settings' (link to time zone preferences). The 'Help' link is located at the top right of the main content area.

Figure 1-23 z/OSMF Capacity Provisioning task

z/OSMF Capacity Provisioning task

The Capacity Provisioning task, introduced with z/OSMF V1R13, provides a browser-based user interface for working with the Capacity Provisioning Manager on a z/OS system. With the Capacity Provisioning task, you can perform the following functions:

- ▶ Manage connections to a Provisioning Manager and use them to transfer provisioning policies and domain configurations to the Provisioning Manager, or to query various status reports.
- ▶ View the status of a Provisioning Manager by displaying reports about the domain status, the active configuration, and the active policy.
- ▶ Manage, create, modify, and delete provisioning policies and domain configurations from a central shared repository.
- ▶ Import and export domain configurations and policies.
- ▶ Install and activate policies and domain configurations for the domain that the Provisioning Manager controls.

1.24 z/OSMF System Status and Resource Monitoring task

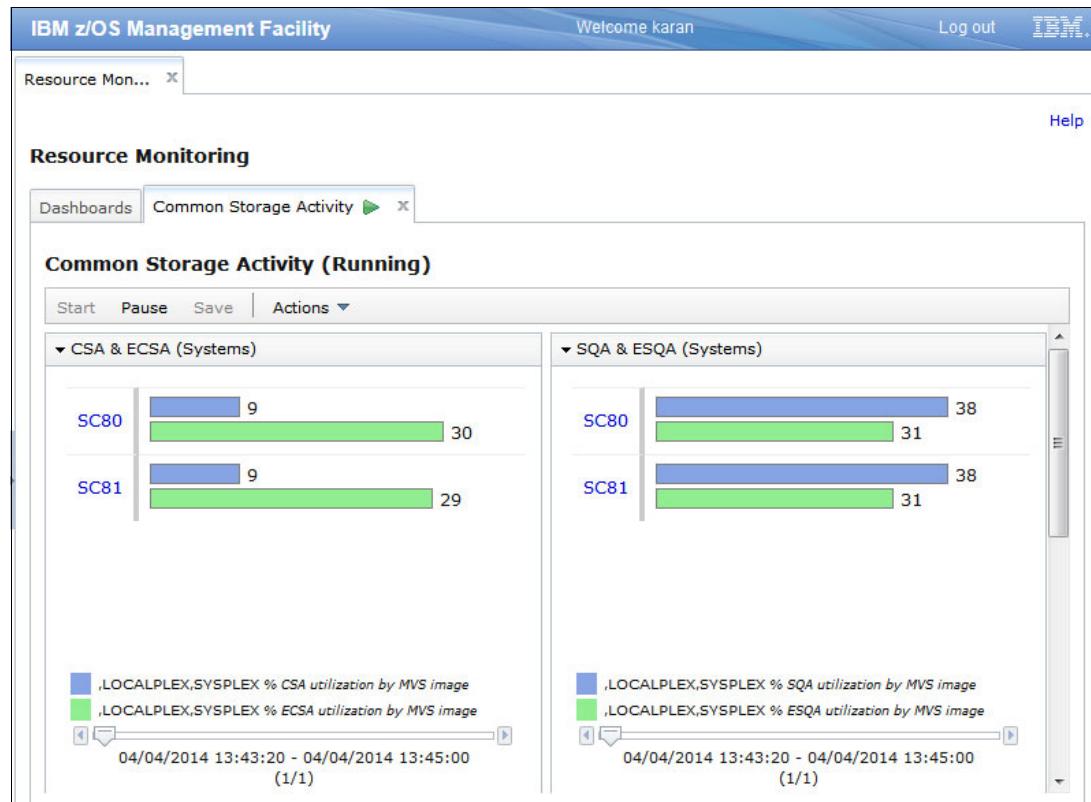


Figure 1-24 z/OSMF Resource Monitoring task

z/OSMF System Status and Resource Monitoring task

The System Status and Resource Monitoring tasks, introduced with z/OSMF V1R12, enable you to monitor the performance of your systems from the convenience of one location. What used to take many minutes and required significant skill, reading performance data for multiple systems, can be done in seconds with the System Status and Resource Monitoring tasks.

The System Status task, which requires the RMF feature of z/OS, is an RMF-based workload reporting application that displays information about the systems that you define. The System Status task enables the following functionality:

- ▶ Enables you to specify the z/OS sysplexes, IBM AIX® system complexes (IBM System p®), Windows system complexes (IBM System x®), and Linux system complexes (System z and System x) to be monitored by the Resource Monitoring task.
- ▶ Combines data from an entire sysplex into one performance indicator so that you can quickly assess the performance of the workloads running on the z/OS sysplexes in your environment.
- ▶ Enables you to link directly to the Resource Monitoring and Workload Management tasks so that you can view detailed performance data or view the active WLM service definition, the active WLM service policy, and the WLM status for the sysplex.

In the event of system performance issues, you can use the Resource Monitoring task to drill down to information about connectivity, performance index status, common storage, and more:

- ▶ Provides real-time performance status for the sysplexes and system complexes defined in the System Status task.
- ▶ Uses RMF Distributed Data Servers (DDSs) running on any supported z/OS release to gather data to display, including most of the data available in RMF Monitor III.
- ▶ Uses z/OS RMF Cross Platform (RMF XP) DDS, which enables you to monitor operating systems that can run on an IBM zEnterprise System, including the IBM zEnterprise BladeCenter® Extension (zBX).
- ▶ Enables you to combine performance metrics of various platforms in the same dashboard so that you can quickly view the overall status of the zEnterprise ensembles.
- ▶ Provides predefined monitoring dashboards that you can use to view different performance metrics, which can be grouped and arranged flexibly.
- ▶ Enables you to define and customize your own monitoring dashboards to create cross-references, correlations, and advanced filtering for focused monitoring.
- ▶ Enables you to link directly to the Workload Management task so that you can view WLM service definitions. This provides easier navigation, and enables a more productive experience for systems programmers, operators, and others using z/OSMF.
- ▶ Enables you to retrieve and view performance data that the RMF DDS has collected in the past for the metric groups contained in a monitoring dashboard. This capability is provided as of z/OSMF V2R1 with APAR PI08825.
- ▶ Enables you to export the data collected in monitoring dashboards into comma-separated values (CSV) format files on your local workstation, enabling further data evaluation using a spreadsheet application. This capability is provided as of z/OSMF V2R1 with APAR PI08825.

Remember: As of z/OSMF V1R13, the name of the Sysplex Status task was changed to *System Status*, and the name of the Monitoring Desktops task was changed to *Resource Monitoring*.

As of z/OSMF V2R1, the data gatherers for Windows system complexes (System x) are integrated into z/OSMF along with the data gatherers for AIX (System p) and Linux (System z and System x) system complexes, which were integrated in z/OSMF V1R13.

In-context application linking between the Resource Monitoring task and the Workload Management task, and between the System Status task and the Workload Management task, is provided as of z/OSMF V1R13 with APAR PM74508 and APAR PM74517.

1.25 z/OSMF Workload Management task

Name Filter	Description Filter	Activity Filter	Sysplex Filter	Messages Filter	Last Modified (GMT) Filter	Modified By Filter
<input checked="" type="checkbox"/> BVTTest	Modify Service Definition	⚠ Warning			May 16, 2011 4:48:35 PM	emurphy
<input type="checkbox"/> SYIMPORT	View Service Definition	⚠ Warning			May 16, 2011 4:48:35 PM	emurphy
<input type="checkbox"/> SYSPAUL	View Messages	⚠ Warning			Sep 15, 2012 2:59:49 AM	zmfusr1
<input type="checkbox"/> SYSPAUL	View History	⚠ Warning			Feb 9, 2012 11:03:18 AM	zmfusr3
<input type="checkbox"/> SYSTCP1	Print Preview	⚠ Warning			Jun 20, 2011 1:42:05 PM	zosmfad
<input type="checkbox"/> SYTEST	Install and Activate...	⚠ Warning			Feb 15, 2012 2:27:48 PM	jrl
<input type="checkbox"/> SYTEST	Copy...	⚠ Warning			May 16, 2011 4:48:35 PM	emurphy
<input type="checkbox"/> SYTEST	Delete...	⚠ Warning			May 16, 2011 4:48:35 PM	emurphy
<input type="checkbox"/> SYTEST	Export	⚠ Warning			May 16, 2011 4:48:35 PM	emurphy
<input type="checkbox"/> SYTEST	View WLM Status	⚠ Warning			May 16, 2011 4:48:35 PM	emurphy
<input type="checkbox"/> SYTEST	Cpy of ServDef - BVT 01/25/2012	UTCPLXCB	⚠ Warning		Sep 17, 2012 6:36:23 AM	zmfusr1
<input type="checkbox"/> SYTEST (installed & active)	Cpy of ServDef - BVT 04/25/2012	UTCPLXCB	⚠ Warning		Sep 17, 2012 6:36:23 AM	zmfusr1
<input type="checkbox"/> SYTESTC	Cpy of ServDef for use by COMBAT	UTCPLXCB	⚠ Warning		Sep 27, 2011 10:10:01 AM	zmfusr3

Figure 1-25 z/OSMF Workload Management task

z/OSMF Workload Management task

The Workload Management task, introduced with z/OSMF V1R12, can help simplify the creation, modification, and review of z/OS WLM service definitions.

For example, if you want to optimize a service definition based on best practices, it could take hours to read through the z/OS WLM-related manuals. With z/OSMF, this task can be potentially reduced to minutes because the preferred practices are built into the Workload Management task.

The Workload Management task offers the following functionality:

- ▶ Is an intuitive editor that enables you to select, edit, copy, and paste service definitions, manage and activate service policies, and monitor the WLM status in the sysplex.
- ▶ Has direct access to the WLM Couple Data Set to install and extract service definitions.
- ▶ Offers different views (table-based, HTML-based) that you can use to quickly analyze, review, or edit service definitions.
- ▶ Provides sorting and filtering capabilities that you can use to customize a view and display the optimal data for a particular edit or analysis task.
- ▶ Enables you to directly navigate between the different parts of the service definition using one click.
- ▶ Supports fine-grained authorization to better control which users can view, install, or modify service definitions. This capability is provided as of z/OSMF V1R13.

- ▶ Persists user settings and preferences between z/OSMF sessions, obviating the need to re-establish settings manually. This capability is provided as of z/OSMF V1R13.
- ▶ Provides in-context application linking between the Resource Monitoring task and the Workload Management task, and also between the System Status task and the Workload Management task. This provides easier navigation and enables a more productive experience for systems programmers, operators, and others using z/OSMF. This capability is provided as of z/OSMF V1R13 with APAR PM74508 and APAR PM74517.

1.26 z/OSMF Incident Log task

A screenshot of the IBM z/OS Management Facility Incident Log task interface. The window title is "IBM z/OS Management Facility" and the sub-title is "Welcome zmfusr3". The main area is titled "Incident Log". At the top left are "Welcome" and "Incident Log" buttons. Below is a table with columns: Incident Type Filter, Description Filter, Date and Time (GMT) Dates from Oct 19, 2012 12:00:00 AM, Sysplex Filter, System Filter, and Problem Number Filter. The table lists several incidents, including "User Initiated" entries for BUFFERUSAGE and "ABEND" entries for S00F4, S0EC6, S00C4, etc. An incident for ABEND S00F4 is selected, and a context menu is open over it, listing options: Set Tracking ID..., Set Problem Number..., Add Notes... (which is highlighted with a cursor), Delete Incident..., Send Diagnostic Data..., View Diagnostic Details, FTP Job Status, and Allow Next Dump...

Incident Type Filter	Description Filter	Date and Time (GMT) Dates from Oct 19, 2012 12:00:00 AM	Sysplex Filter	System Filter	Problem Number Filter
<input type="checkbox"/> User Initiated	BUFFERUSAGE AFTER DELETION	Oct 22, 2012 5:04:52 PM	UTCPLXCB	CB86	
<input type="checkbox"/> User Initiated	BUFFERUSAGE AFTER DELETION	Oct 22, 2012 4:56:34 PM	UTCPLXCB	CB86	
<input type="checkbox"/> User Initiated	BUFFERUSAGE BEFORE DELETION	Oct 22, 2012 3:52:12 PM	UTCPLXCB	CB86	
<input type="checkbox"/> User Initiated	BUFFERUSAGE TEST	Oct 22, 2012 2:52:00 PM	UTCPLXCB	CB86	
<input checked="" type="checkbox"/> ABEND S00F4	COMPID=DF115,CSECT=IGWDARD1+0D26,DATE=09/18/12,MAINTID=NONE ,ABND=0F4,RC=00000024,RSN=010E5ABB	Oct 22, 2012 5:05:46 AM	UTCPLXCB	CB88	12345,999,001
<input type="checkbox"/> ABEND S0EC6	COMPON=BPX,COMPID=SCPX1,ISSUER=BPXMIPCE,+157A,ABEND=S0EC6,REASON=0B010407	8:24 AM	UTCPLXCB	CB86	
<input type="checkbox"/> User Initiated	ALEX	8:47 PM	UTCPLXCB	CB86	
<input type="checkbox"/> ABEND S00C4	COMPON=HZR,COMPID=SCRTD,ISSUER=HZRMIREC,	8:58 AM	UTCPLXCB	CB86	
<input type="checkbox"/> User Initiated	ABEND=40D,RC=10,COMPON=RTM2,COMPID=SCRT - UNRECOVERABLE ABEND FAILURE	9:51 AM	UTCPLXCB	CB88	
<input type="checkbox"/> ABEND	COMPON=CTT TC=WLIJFSV29 ISSUER=CTTDIE - TE - CAT ADDRESS=22652AD0, JOB#=04000077	9:00 AM	UTCPLXCB	CB88	
<input type="checkbox"/> ABEND S01FB	COMPON=JSS-REC,COMPID=SC1B8,ISSUER=IEESB670,JOB SCHEDULING SUBROUTINE RECOVERY EXIT ROUTINE	Oct 19, 2012 8:06:54 PM	UTCPLXCB	CB89	

Figure 1-26 z/OSMF Incident Log task

z/OSMF Incident Log task

When problems occur, the task of gathering diagnostic data to be sent to IBM or another vendor's support team can be very time-consuming. You need to reduce risk to the business and get to the root of the problem quickly.

The z/OSMF Incident Log task is intended to facilitate problem data management tasks for new or less-skilled system programmers and system administrators. It can also provide procedural advantages for the experienced system programmer by providing an event log summary and detailed view of z/OS incidents.

With the Incident Log task, you can perform the following tasks:

- ▶ Manage the abnormal end of task (abend) and user-initiated SAN Volume Controller dumps, referred to as incidents, that occurred on a system or in a sysplex.
- ▶ Send diagnostic data to IBM or another vendor for further diagnostics.
- ▶ Break dumps into multiple data sets that can be sent across FTP in parallel to reduce transmission time. This capability is provided as of z/OSMF V1R12.
- ▶ Send additional data as attachments with an incident. This capability is provided as of z/OSMF V1R12.
- ▶ Add free-form notes to an incident. This capability is provided as of z/OSMF V1R12.
- ▶ Use the provided APAR search string to search the IBM RETAIN® database and find z/OS APARs related to an incident. This capability is provided as of z/OSMF V1R13.

- ▶ Browse the logs and FTP jobs associated with an incident. This capability is provided as of z/OSMF V1R13.
- ▶ Use the Problem Documentation Upload Utility that is provided with z/OS to send diagnostic data to IBM FTP sites. This capability is provided as of z/OSMF V1R13.
- ▶ Delete incidents that satisfy user-specified criteria using the ceatool command-line interface (CLI).

As of z/OS V1R13 with APAR OA38812, the ceatool CLI is provided with the z/OS common event adapter (CEA) component. With the ceatool, you can use a z/OS UNIX System Services shell, a JCL job, or a cron job to delete incidents and the associated diagnostic data.

- ▶ Use a four- to eight-character high-level qualifier for naming log snapshot data sets.

When you configure the Incident Log plug-in for z/OSMF, you specify a high-level qualifier to use for naming log snapshot data sets. By default, this value is CEA. z/OS V2R1 increases the allowable length of this high-level qualifier from four to eight characters through the HLQLONG statement in member CEAPRMxx.

1.27 z/OSMF ISPF task



Figure 1-27 z/OSMF ISPF task

z/OSMF ISPF task

The ISPF task, introduced with z/OSMF V1R13, enables you to open and start classic z/OS ISPF sessions directly from z/OSMF. You can open multiple sessions in multiple panes and go to all line mode ISPF applications from z/OSMF.

Additionally, this interface makes the ISPF applications Uniform Resource Locator (URL)-addressable, and therefore supports the ability to directly go to ISPF-based functions from z/OSMF tasks or from external applications. For example, you can start from the Incident Log task directly to browse a log snapshot data set in ISPF.

You can view up to four ISPF screen sessions at the same time. The panels appear similar to classic 3270 layout, and can be navigated using both mouse and keyboard. Additional tabs enable you to go to other ISPF panels, and TSO/E messages are displayed in a separate window.

1.28 z/OSMF Workflows task

The screenshot shows the 'z/OS customization for each z/OSMF plug-in' workflow in the z/OSMF Workflows task. The workflow consists of 43 steps, with 6 completed (12% complete) and 37 remaining. The steps are categorized by plug-in: Configuration Assistant, ISPF, Common Information Model (CIM), and Workload Management.

State	No.	Title	Owner	Skill Category	Assignees
Complete	1	Configuration Assistant plug-in	zmfusr3		
Complete	1.1	(Optional) Transfer existing backing store files into z/OSMF	zmfusr3	Network Administrator	zmfusr3
In Progress	2	ISPF plug-in			
In Progress	2.1	Ensure that users can access the ISPF task			
Ready	2.2	Assigning the TRUSTED attribute to CEA	zmfusr3	Security Administrator	zmfusr3
Complete	3	Common Information Model (CIM) server	zmfusr3		
Complete	3.1	CIM customization through ServerPac or the CIM sample jobs	zmfusr3	Security Administrator	zmfusr3
Complete	3.2	CIM security	zmfusr3	Security Administrator	zmfusr3
Complete	3.3	CIM file systems and directories	zmfusr3	System Programmer	zmfusr3
Complete	3.4	CIM server start-up	zmfusr3	System Programmer	zmfusr3
Complete	3.5	CIM server verification	zmfusr3	System Programmer	zmfusr3
In Progress	4	Workload Management plug-in			
Ready	4.1	Configure the Common Information Model (CIM) server	zmfusr3	System Programmer	zmfusr3

Figure 1-28 z/OSMF Workflows task

z/OSMF Workflows task

Configuring components and products for your z/OS system can be time-consuming and error-prone. A large number of interrelated steps might be involved, requiring work to be carried out by different members of your organization. Such projects require careful planning and coordination with your team to ensure the best results. As of z/OSMF V2R1, z/OSMF offers the Workflows task to assist you with these projects.

The Workflows task can help you by accomplishing the following preferred practices:

- ▶ Simplifying your software configuration efforts through guided, step-based workflows for common activities.
- ▶ Ensuring that consistent methods and tools are used for software configuration on z/OS, including the instructions and interfaces that you use to perform this work.
- ▶ Providing administrative functions for assigning workflow responsibilities and tracking progress.

More than merely a checklist of activities, the Workflows task provides a structured process for accomplishing work on z/OS:

- ▶ Guide the activities of system programmers, security administrators, and others at your installation who are responsible for managing the configuration of the z/OS system.
- ▶ Assign individual work items in the workflow (the steps) to performers and track their progress.
- ▶ Use wizards to assist your team with creating system objects (UNIX files and z/OS data set members) and submitting work to run on z/OS, such as batch jobs, REXX scripts, and UNIX shell scripts.

1.29 IBM Health Checker for z/OS

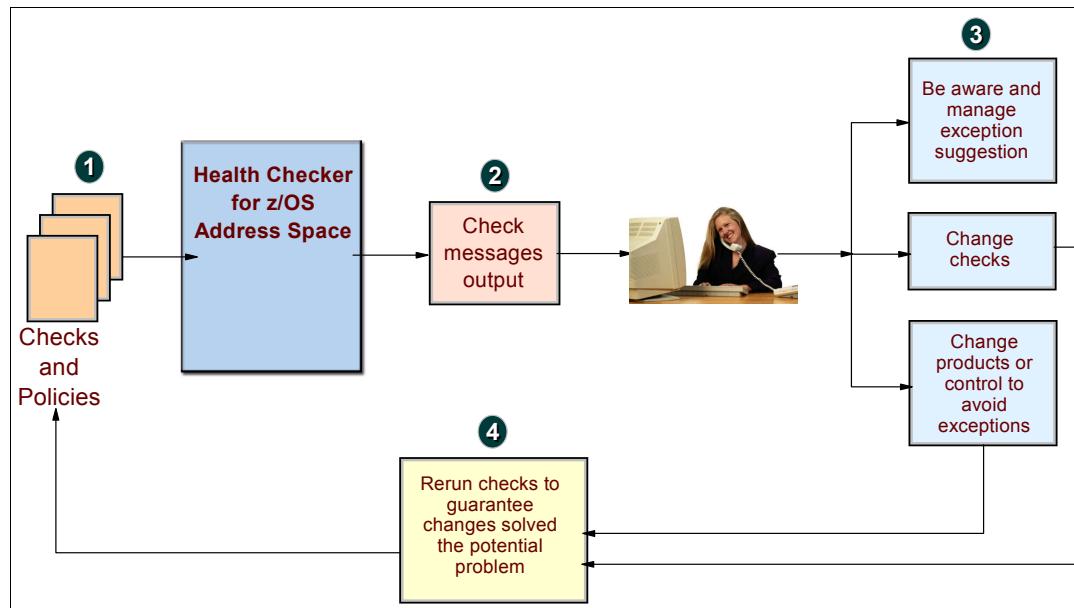


Figure 1-29 IBM Health Checker for z/OS

IBM Health Checker for z/OS

The objective of IBM Health Checker for z/OS is to identify potential problems before they affect your availability or, in worst cases, cause outages. It checks the current active z/OS and sysplex settings and definitions for a system and compares the values to those suggested by IBM or defined by you. It is not meant to be a diagnostic or monitoring tool, but rather a continuously running preventive tool that finds potential problems.

IBM Health Checker for z/OS produces output in the form of detailed messages to notify you of both potential problems and suggested actions to take. These messages do not mean that IBM Health Checker for z/OS has found problems that you need to report to IBM. Instead, IBM Health Checker for z/OS output messages inform you of potential problems so that you can take action on your installation.

Health Checker for z/OS processing

As illustrated in Figure 1-29, Health Checker for z/OS functions in the following way:

1. Check values provided by components.

Each check includes a set of predefined values:

- Interval, or how often the check will run
- Severity of the check, which influences how check output is issued
- Routing and descriptor codes for the check

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

2. Check output.

A check issues its output as write to operators (WTOs) and other messages, which you can view using SDSF, the HZSPRINT utility, or a log stream that collects a history of check output. If a check finds a deviation from leading practices or a potential problem, it issues a WTO message known as an *exception*, as previously mentioned.

Check exception messages include a description of the potential problem found, including the severity, but also information about what to do to fix the potential problem.

3. Resolve check exceptions.

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

4. If you solve an exception by changing a product setting or system control, it is a good policy to rerun the checks related to this action, to an Internet Protocol network, to ensure that the identified problems resolved.

1.30 z/OS release cycle

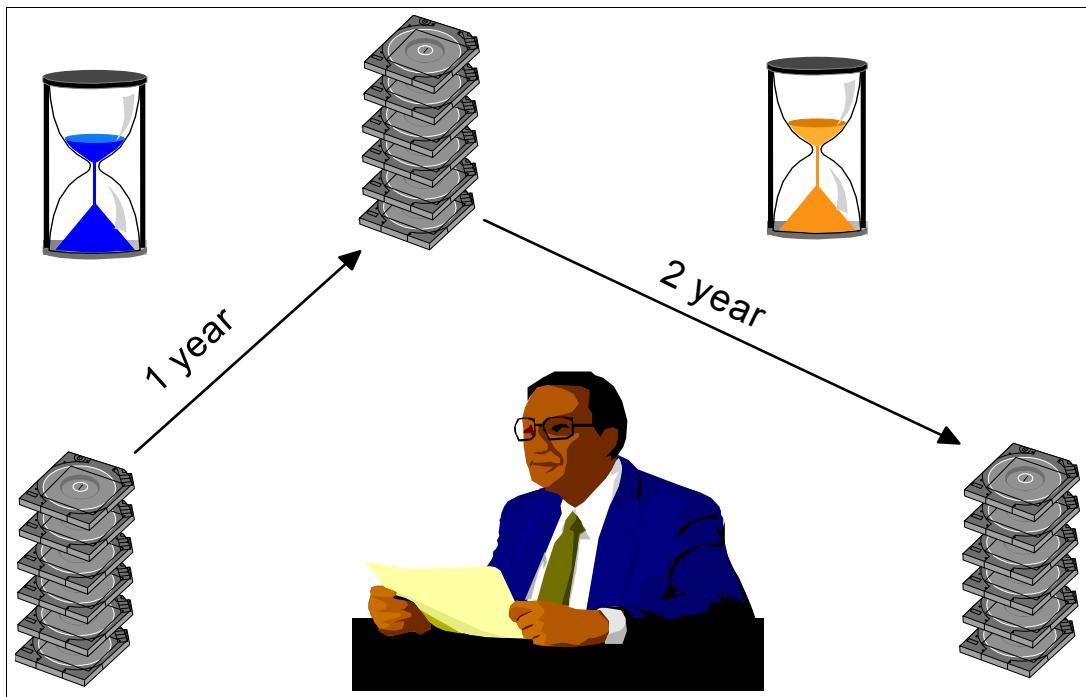


Figure 1-30 z/OS release cycle

z/OS release cycle

z/OS V1R13 was the last z/OS release that IBM delivered on an annual basis. The annual release cycle started in September 2004 with the general availability of z/OS V1R6. Before V1R6, a new release was delivered every six months. Starting with z/OS V2R1, IBM now delivers a z/OS release every two years.

New z/OS functions continue to be delivered between releases through the normal maintenance stream, or as web deliverables. In addition, significant new functions might be delivered between releases as features of the product.

This predictable release cycle can reduce your planning time. Because each new level is comprehensively tested, the quality of the operating system is improved. After your initial migration is complete, you can expect simplified ordering, planning, and installing of the next z/OS release.



TSO/E, ISPF, JCL, and SDSF

This chapter describes how to use the basic products that a system programmer needs to install and customize an IBM z/OS operating system. The following topics are covered:

- ▶ Time Sharing Option/Extensions (TSO/E)

As part of the install of z/OS process, a TSO/E session on the system that has undergone initial program load (IPL) must be established using a locally attached or network-attached terminal.

- ▶ Interactive System Productivity Facility and Program Development Facility (ISPF/PDF)

ISPF is a full-panel application navigated by a keyboard. ISPF includes a text editor and browser, and functions for locating files, listing files, and performing other utility functions when installing z/OS.

- ▶ Job control language (JCL)

Using batch JCL is the most common way to install a full volume dump order when ordering z/OS. When you have a z/OS driving system, you can install your order through a batch job that initializes and restores the direct access storage device (DASD) volumes.

- ▶ Spool Display and Search Facility (SDSF)

SDSF is useful during the install process of z/OS to be able to view the output of the many install JCL jobs that are used during this process.

2.1 z/OS facilities for system programmers

- Time Sharing Option/Extended - (TSO/E)
- Interactive System Productivity Facility - (ISPF)
- Job Control Language - (JCL)
- System Display and Search Facility - (SDSF)

Figure 2-1 z/OS facilities for system programmers

z/OS facilities for system programmers

As a system programmer, it is helpful to be familiar with the basic tools used in your daily job. These tools and their uses are listed here:

- ▶ TSO/E and ISPF are used to perform the following tasks:
 - Install and customize z/OS and other products
 - Communicate interactively with the operating system
 - Define and maintain user definitions
 - Create data sets and JCL, and submit jobs
 - Communicate with other TSO/E users
 - Develop and maintain programs in languages, such as assembly language, Common Business Oriented Language (COBOL), Fortran, Pascal, C, C++, Java, PL/I, Restructured Extended Executor (REXX), command list (CLIST), and so on
 - Manipulate data
- ▶ JCL enables you to submit jobs and allocate resources.
- ▶ SDSF is used to perform the following tasks:
 - Monitor
 - Jobs waiting for execution
 - Output waiting to be printed
 - System resources used by jobs
 - System resources available for use
 - Control
 - System resources, such as printers
 - Job priority and class
 - Job output priority and class

2.2 TSO/E

Installing z/OS

- Install and customize z/OS and other products
- Communicate interactively with the operating system
- Define and maintain user definitions
- Create data sets and JCL, and submit jobs

System Programmers

The diagram illustrates the interaction between system programmers and the z/OS system. At the top, a yellow box contains the text "System Programmers". Below it, two photographs show a man and a woman sitting at desks, each with a computer monitor. Arrows from both photographs point towards a large, dark server rack on the right side of the frame. The server rack has the word "IBM" printed on its front panel.

Figure 2-2 Using TSO/E during the install process

TSO/E

TSO/E is a base element of z/OS. TSO/E enables users to interactively share computer time and resources. In general, TSO/E makes it easier for people with all levels of experience to interact with the z/OS system.

TSO/E has undergone continuous enhancements during its lifecycle, and it has become the primary user interface to the z/OS operating system.

TSO/E provides programming services that you can use in system or application programs. These services consist of programs, macros, and CLISTS. TSO/E services support a wide range of functions that are useful in writing both system programs and application programs that use the full-screen capabilities of TSO/E.

CLISTS, REXX execs, servers, and command processors are specific types of programs that you can write to run in the TSO/E environment.

TSO/E users

TSO/E offers advantages to a wide range of computer users, including system programmers, application programmers, information center administrators, information center users, TSO/E administrators, and others who access applications that run under TSO/E. This book describes the major tasks that each of these users can perform using TSO/E.

2.3 TSO/E highlights

- Session Manager
- Commands
- Online help
- Console
- Support for z/OS UNIX
- CLIST
- REXX
- Data and notice handling
- Security

Figure 2-3 Key features of z/OS TSO/E

TSO/E highlights

Highlights of TSO/E are described in this section.

Session Manager

The TSO/E Session Manager is an interface to line mode TSO/E. It saves the commands that you enter and the responses that you receive and enables you to re-display or print them. You can correct or change a command that is displayed on the window without having to retype the entire command. By enabling you to re-display, change, and reuse your input, the Session Manager makes TSO/E easier to use.

Commands

TSO/E provides numerous commands for both users and programmers that enable them to interact with TSO/E and the IBM Multiple Virtual Storage (MVS) system. The **ALLOCATE**, **FREE**, and **EDIT** commands are examples of commands that enable users to manage their data sets.

The **TEST** and **TESTAUTH** commands enable programmers to test assembly language programs, including command processors, Advanced Program-to-Program Communication (APPC)/MVS transaction programs, and other programs written in assembly language. The **CONSOLE** command enables users with **CONSOLE** command authority to perform MVS operator activities from a TSO/E session.

Online help

Terminal users can obtain online help for most TSO/E commands. Information Center Facility users can obtain help for each panel and message. TSO/E Enhanced Connectivity Facility users can also obtain online help for terminal messages. Installations can also provide online help information to users in other languages.

Data and notice handling

TSO/E simplifies the way in which data and notices are sent and received. For example, the **TRANSMIT** and **RECEIVE** commands enable users to send data and messages to other users in a network. The broadcast data set or individual user logs contain messages that either the system or another user sends using the **SEND** command.

In addition, a recovery routine prevents broken mail chains that can occur when message handling is interrupted. Notices are also handled more efficiently during logon processing. TSO/E keeps a copy of notices in storage, thereby reducing the I/O operations needed to inform users of waiting messages when they log on.

Logon processing

TSO/E provides a full-screen logon panel that makes the logon process easier:

- ▶ Saving user attributes from one session to the next
- ▶ Enabling program function keys to be used during logon
- ▶ Enabling users to enter commands during logon
- ▶ Explaining the error when incorrect information is specified

The **LOGON** and **ACCOUNT** command processors enable users to request private areas of up to two gigabytes for each terminal session. Your installation can also customize the logon panel and the logon help panel, and customize logon processing using other exits.

Language enablement

TSO/E enable installations to provide TSO/E messages and the TRANSMIT full-screen panel to users in different languages. The TSO/E **CONSOLE** command also supports the display of translated system messages issued during a console session.

The logon authorized pre-prompt exit and the **PROFILE** command support the specification of languages to be used in displaying translated information. An installation can specify help data sets for different languages in the IKJTSOxx member of SYS1.PARMLIB. Support for logon panels and their help text in different languages is also available.

The TSO/E REXX external function, **SYSVAR**, provides support for new arguments that REXX execs can use to obtain language information. Execs can use this information together with the new **SETLANG** function to set the language in which REXX messages are displayed.

Security

TSO/E provides several enhancements to support the use of security labels. Installations can also control communication between users to protect the security classification of information. For example, installations can control and audit the use of the **SEND** command. **LISTBC** command processing enhancements make it possible for installations to restrict users from viewing messages for which they do not have the appropriate security.

CLIST language

The CLIST language is a high-level programming language that enables programmers to issue lists of TSO/E commands and JCL statements in combination with logical, arithmetic, and string-handling functions provided by the language. The programs, called CLISTS, can simplify routine user tasks, start programs written in other languages, and perform complex programming functions.

Restructured Extended Executor language support

REXX is a high-level procedures language that enables both inexperienced users and experienced programmers to write structured programs called *REXX execs*. REXX execs can be run in any MVS address space (both TSO/E and non-TSO/E). TSO/E also enables users to write APPC/MVS transaction programs in the REXX language.

Installations can acquire IBM Compiler and Library for REXX/370 or a functionally equivalent compiler. A compiled REXX exec runs more efficiently, because it does not need to be interpreted at run time.

TSO/E service facility

The TSO/E service facility enables TSO/E users to run authorized or unauthorized programs, TSO/E commands, or CLISTS from an unauthorized environment, while maintaining system integrity.

TSO Command Package

The TSO Command Package provides functions that help to improve productivity:

- ▶ Support for running terminal sessions as batch jobs
- ▶ Automatic saving of data
- ▶ Accounting facility
- ▶ Defaults for the user-attribute data set

Information Center Facility

The Information Center Facility eases users into the data processing environment by providing a series of conversational panels. These panels eliminate numerous command-driven interactions between the user and the system. Information Center Facility provides panels that enable an administrator to maintain the facility, enroll users, and add, modify, and delete products.

Enhanced Connectivity Facility

The Enhanced Connectivity Facility (ECF) enables you to customize the way in which host server programs and personal computer (PC) requester programs communicate. IBM products or customer-written programs can supply the services.

The user can access MVS host services from a PC using IBM System/370-to-IBM Personal Computer ECF. This enables a DOS/PC user to interact with MVS or IBM Virtual Machine/System Product (VM/SP) systems using PC commands.

Support for z/OS UNIX

Installations can use the functions provided by the TSO/E **ALLOCATE** and **FREE** commands to manipulate z/OS UNIX files.

2.4 TSO/E customization

- Define TSO/E to VTAM or TCAM
- Define the users allowed to log on to TSO/E
- Create TSO/E logon procedure for users

Figure 2-4 TSO/E required customization

TSO/E customization

TSO/E enables users to interactively work with the z/OS system. After the required customization, users are able to log on and issue commands from TSO/E.

Each user is defined to TSO/E by storing its user ID, logon procedure name, and the TSO/E resources which it has authority to use. This can be done in either of two ways:

- ▶ User Attribute data set (UADS), using the **ACCOUNT** command
- ▶ IBM Resource Access Control Facility (RACF) database

When RACF is installed, it can be used to control access to the system and store information about each TSO/E user. The RACF database contains profiles for every entity (user, data set, or group) defined to RACF. For more information about RACF, see *z/OS Security Server RACF System Programmer's Guide*, SA23-2287.

Customization of the TSO/E environment generally refers to making a TSO/E facility available or changing default values that affect TSO/E. You can customize the following areas:

- ▶ IBM Virtual Telecommunications Access Method (VTAM). You can change VTAM session protocols, provide substitute characters for unavailable keyboard characters, and override the default values used to start VTAM.
- ▶ Logon limits. You can limit and manage the maximum number of concurrent logons, the user's region size, and user access to applications.
- ▶ The logon/logoff process. You can change how often the system displays the logon proceeding message, limit the number of attempts a user can make at entering information in response to logon prompts, tailor the reconnect option, and suppress messages that are generated during the execution of the logon job.

You can also review factors that affect logon performance, such as using STEPLIBs in logon JCL, and you can write exits to further customize the logon/logoff process. Your installation can use security labels (SECLABEL) if the correct products are installed.

- ▶ ISPF/PDF and others products. You can make ISPF/PDF and others products that run in the TSO environment available to TSO/E users.
- ▶ Authorized commands and programs. You can select which authorized commands and programs users can use.

- ▶ Command/program invocation platform support. You can start TSO/E commands and programs on the command/program invocation platform. Both authorized and unauthorized commands and programs are supported.
- ▶ Command availability in the background. You can make specific commands unavailable for use in the background.
- ▶ **TRANSMIT** and **RECEIVE** availability. You can make the **TRANSMIT** and **RECEIVE** commands available.
- ▶ HELP data set usage. You can customize the use of HELP data set members.
- ▶ Host services availability. You can make host services available to PC users.
- ▶ Language support. You can provide information to users in their national language.

For more details about customization, see *z/OS TSO/E Customization*, SA32-0976.

2.5 TSO/E: Terminal control address space start procedure

```
//TSO      PROC MBR=TSOKEY00
//** LIB: CPAC.PROCLIB(TSO)
//** DOC: THIS IS THE CATALOGED PROCEDURE USED
//**        FOR STARTING TSO/VTAM TIME SHARING.  THE
//**        TERMINAL CONTROL ADDRESS SPACE (TCAS) MUST
//**        BE ACTIVE WHEN RUNNING TSO/VTAM TIME SHARING.
//**
//** NOTE: THE PARMLIB DD STATEMENT IDENTIFIES THE PARMLIB
//** DATA SET AND MEMBER THAT CONTAINS TSO/VTAM TIME
//** SHARING PARAMETERS.  THE MEMBER MAY EITHER BE
//** SPECIFIED ON THE START COMMAND OR DEFAULTED TO
//** TSOKEY00.
//*
//** THE PRINTOUT DD STATEMENT IDENTIFIES WHERE THE
//** TIME SHARING PARAMETERS THAT ARE USED SHOULD BE
//** LISTED.
//*
//** FREE=CLOSE IS CODED SO THAT THE DATA SETS ASSOCIATED
//** WITH BOTH THE PARMLIB AND PRINTOUT DD STATEMENTS
//** WILL BE DEALLOCATED WHEN THE DATA SETS ARE CLOSED.
//*
//STEP1   EXEC PGM=IKTCAS00,TIME=1440
//PARMLIB  DD  DSN=CPAC.PARMLIB(&MBR),DISP=SHR,FREE=CLOSE
//PRINTOUT DD  SYSOUT=*,FREE=CLOSE
//*
```

Figure 2-5 TSO/E: Terminal control address space start procedure

TSO/E: Terminal control address space start procedure

Before a user can log on to TSO/E, both VTAM and the terminal control address space (TCAS) must be active in the system. The system operator enters the **START** command to start VTAM.

After VTAM has been started, the system operator enters the **START** command to start TSO/E and activate TCAS. TCAS accepts logons from TSO/VTAM users and creates an address space for each user.

The TCAS (TSO) start procedure is usually stored in SYS1.PROCLIB. In the start procedure you specify the TSOKEYxx SYS1.PARMLIB member that contains the parameters to be used by TCAS to control the time-sharing buffers, maximum number of users, and other operational variables, as pointed by the parmlib DD card of the start procedure. If the parmlib DD card is not coded in the procedure, SYS1.PARMLIB is used.

When a user logs on, the VTAM terminal input/output (I/O) coordinator (VTIOC) is initialized. VTIOC controls the movement of data between TSO/E and VTAM. The parmlib member TSOKEY00 or an installation-defined alternative member contains parameters that are used during VTIOC initialization.

If a member other than TSOKEY00 is used, the operator must include the member name either on the **START** command or in the procedure that the **START** command starts. For a description of TSOKEY00, see *z/OS MVS Initialization and Tuning Reference*, SA23-1380.

2.6 TSO/E logon procedure

```
/*-----  
** SERVERPAC LOGON PROCEDURE  
**  
** THIS PROCEDURE ENABLES USERS TO LOG ON TO TSO/E.  
** THE CLIST ISPPDF, WHICH RESIDES IN CPAC.CMDPROC,  
** IS EXECUTED AT FIRST TO INVOKE THE ISPF.  
**-----  
//IKJACCNT PROC  
//IKJACCNT EXEC PGM=IKJEFT01,DYNAMNBR=500,PARM=ISPPDF  
//SYSPROC DD DISP=SHR,DSN=CPAC.CMDPROC  
//SYSHELP DD DISP=SHR,DSN=SYS1.HELP  
//          DD DISP=SHR,DSN=SYS1.SASFHELP  
//          DD DISP=SHR,DSN=ISF.SISFHELP  
//          DD DISP=SHR,DSN=REXX.V1R3M0.SEAGHENU  
//          DD DISP=SHR,DSN=REXX.V1R3M0.SFANHENU  
//          DD DISP=SHR,DSN=SYS1.SBDTHELP  
//          DD DISP=SHR,DSN=SYS1.HELPENP  
//          DD DISP=SHR,DSN=ISP.SISPHELP  
//SYSLBC DD DISP=SHR,DSN=SYS1.BRODCAST  
//SYSPRINT DD TERM=TS,SYSOUT=*  
//SYSTERM DD TERM=TS,SYSOUT=*  
//SYSIN   DD TERM=TS  
/*
```

Figure 2-6 TSO/E logon procedure

TSO/E logon procedure

A TSO/E logon procedure contains JCL statements that run the required program and allocate the required data sets to enable a user to acquire the resources needed to use TSO/E. To log on to TSO/E, a user must have access to at least one logon procedure.

The logon procedure is usually in data set SYS1.PROCLIB or another library identified in the PROCxx concatenation in the IBM Job Entry Subsystem 2 (JES2) startup procedure, or in the IATPLBxx DD statement in the JES3 startup procedure. TSO/E provides a logon procedure in SYS1.PROCLIB called IKJACCNT for system programmers to access the system, for example, during the initial installation or if there are problems with the RACF database. Figure 2-6 shows a sample logon procedure. The statements specify the following values:

PGM=IKJEFT01 This identifies the program to be run. IKJEFT01 is the TSO/E-supplied Terminal Monitor Program (TMP) that provides an interface between the user command processors and the TSO/E control program. It obtains commands, gives control to command processors, and monitors their execution. This program can also be run in the background by submitting JCL. Rather than the IKJEFT01 program, an installation can use the Session Manager program (ADFMDF03) or its own TMP.

PARM You can pass a command, CLIST, REXX, or program to IKJEFT01 to be interpreted as the first line of input from the terminal after the user has logged on. In the example, it runs a CLIST or a REXX named BRDCST.

DYNAMNBR	This defines the number of data sets that can be dynamically allocated at the same time. A constant of 2 is always added to the DYNAMNBR value that you specify. It enables data sets to be more quickly reallocated, because control blocks for data sets remain in storage, even after the data sets have been de-allocated. Choose the value for DYNAMNBR carefully. Make the value large enough that it is not readily exceeded by the number of dynamic allocation requests made during the user's session. However, the larger the value you specify for DYNAMNBR, the more virtual storage is used. The actual amount of virtual storage depends on the number of data sets that the user allocates and de-allocates in a session.
SYSIN	This specifies that SYSIN is the user's terminal.
SYSPRINT	This specifies that SYSPRINT is to be directed to the user's terminal.
Additional data sets can be allocated dynamically during the user's session or can be defined in the logon procedure. The following DD statements have special meaning, and can be included in the logon procedure:	
SYSPROC	This defines the current REXX exec or CLIST library to be searched when the user uses the implicit form of the EXEC command. Figure 2-6 on page 72 shows the explicit form of the EXEC command and the library name is specified in the command. The implicit form is BRDCST.
SYSEXEC	This defines the current REXX exec library concatenation to the EXEC command when users use the implicit form of the command. By default, the system searches SYSEXEC first, followed by SYSPROC.

The data sets described in SYSPROC and SYSEXEC DD statements must be partitioned, and have a record format of V, VB, F, or FB. You can allocate them dynamically using the **ALLOCATE** command and activate them with the **ALTLIB** command.

2.7 TSO/E logon process in a VTAM environment

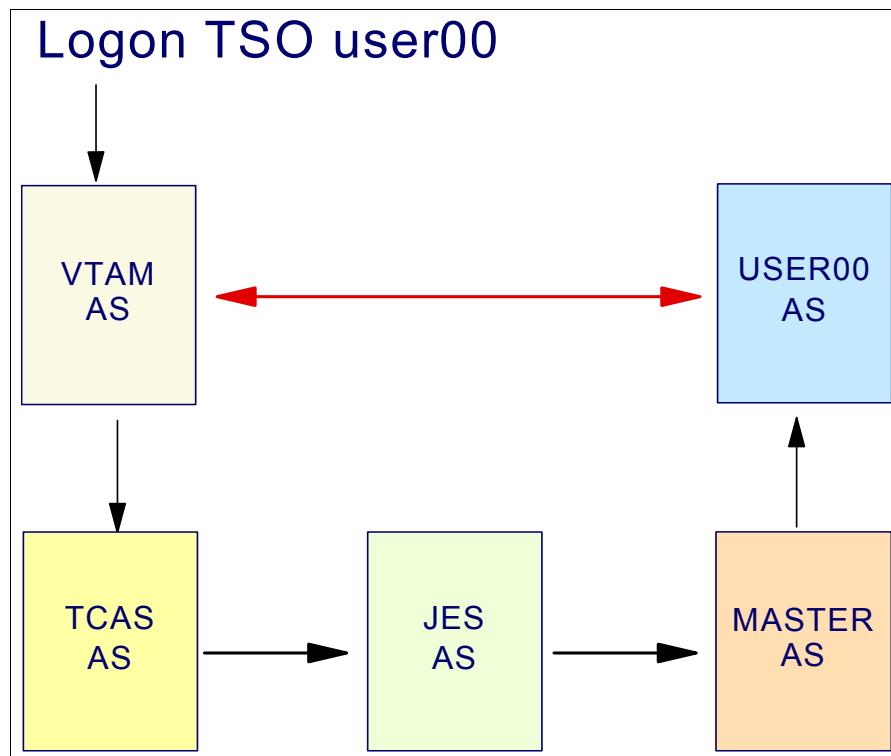


Figure 2-7 TSO/E logon process in a VTAM environment

TSO/E logon process in a VTAM environment

In a VTAM environment, when a user enters a **LOGON** command to the TSO applid the following actions occur:

1. VTAM receives the command and passes it to the TCAS address space.
2. If the maximum number of users logged on in the system is reached, the logon is rejected; if not, and the user ID was not specified, TCAS prompts for the user ID.
3. After the user ID is specified, TCAS verifies that the user has authority to use TSO/E. Depending on the installation customization, a full-screen logon panel is shown to the user. Figure 2-8 on page 75 shows the panel displayed when the user is RACF defined. The values shown in the fields PROCEDURE, ACCT NMBR, SIZE, and COMMAND are the same the user entered for the previous TSO/E session. If this is the first session, they are the default values. The command entered in the COMMAND field is run after any command entered in the PARM field on the EXEC statement of the logon procedure.
4. After the Enter key is pressed, TSO/E verifies the values entered, then the user ID and the logon procedure name is passed to JES. The JCL is interpreted and converted. The MASTER creates the user address space and the resources specified in the JCL are allocated.
5. The user receives a window with the READY prompt at the left top corner of the window. This is called *line-mode TSO/E*. Now TSO/E is ready to accept commands, and user interfaces, such as ISPF or SDSF can be called.

2.8 TSO/E full-screen logon panel

TSO/E LOGON			
Enter LOGON parameters below:			
Userid	====> ROGERS		
Password	====>		
Procedure	====> IKJACCNT		
Acct Nmbr	====> ACCNT#		
Size	====> 6072		
Perform	====>		
Command	====> ISPPDF		
Enter an 'S' before each option desired below:			
-Nomail	-Nonotice	S -Reconnect	-OIDcard
PF1/PF13 ==> Help PF3/PF15 ==> Logoff PA1 ==> Attention PA2 ==> Reshow You may request specific help information by entering a '?' in any entry field			

Figure 2-8 TSO/E full-screen logon panel

TSO/E full-screen logon panel

To log on to TSO/E, type LOGON <yourid> and press the Enter key. After you log on, a window similar to the one shown in Figure 2-8, which is known as a *panel*, is displayed. A panel is a predefined display image that fills your window. Notice that your user ID appears in uppercase letters to the right of the user ID arrow, and that other information required by your installation appears in uppercase letters to the right of other arrows.

The computer generally re-displays information in uppercase regardless of how you typed it. The area to the right of an arrow is called an *input field*. You can type information only in input fields. If you type anywhere else on the window, the keyboard locks. To unlock the keyboard, press the Reset key.

Important: The values that are required to be entered on the logon panel depend on your installation, and are determined by the system programming staff.

Enter a user ID

The user ID identifies the user that the system is to notify. The user ID is 1 through 7 alphanumeric or national (dollar sign (\$), number sign (#), or at sign (@)) characters. The first character must be an alphabetic or national (\$, #, @) character.

Entering a password

In the Password input field, you must enter the password that your administrator gave you. The characters do not appear on the window when you enter your password. Suppressing the password in this way prevents others from seeing it as you type it.

Change password

Optionally, you can also supply the following items:

- ▶ A new password (if you want to change it)
- ▶ A logon procedure, if you do not want to use the installation default
- ▶ An account number, if required by the installation
- ▶ An RACF group
- ▶ An identification (that you are connected to)
- ▶ The region size, if you need more than the installation default
- ▶ The first command to be run after your user ID is logged on

Command line

Figure 2-8 on page 75 shows that a command **ISPPDF** was specified. In this example, ISPPDF is an installation CLIST that allocates the required data sets and calls ISPF. In such cases, rather than entering TSO/E in line-mode, the user receives the ISPF Primary Menu panel and is in full-screen mode.

2.9 TSO/E line-mode

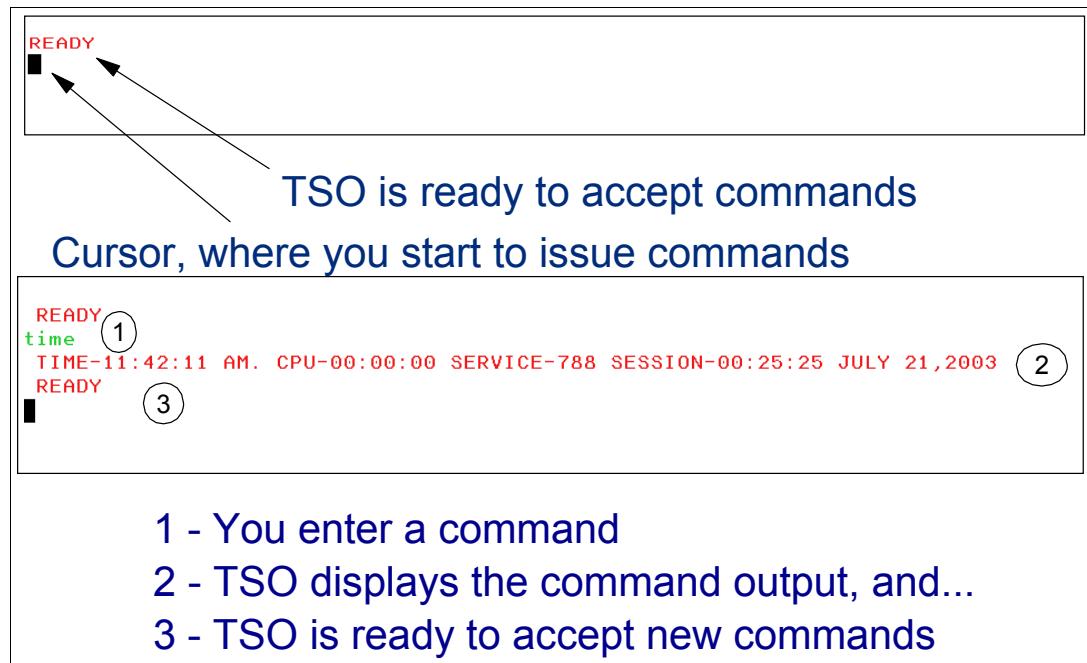


Figure 2-9 TSO/E line-mode

TSO/E line-mode

When you do not enter a command name in the panel shown in Figure 2-8 on page 75, you enter TSO/E in line-mode. When you log on, you see the panel shown in Figure 2-9. The word READY in the corner indicates that TSO is ready to accept your commands.

In TSO/E line-mode you type TSO/E commands one line at a time. It is a quick and direct way to use TSO/E, and was the way programmers originally used to communicate interactively with the z/OS operating system.

You probably will not use TSO/E in line-mode. The user interface provided by ISPF is a more friendly way to work with TSO/E. The following sections contain hints to help you when you are using TSO/E and ISPF.

For more information, see *z/OS TSO/E Primer*, SA32-0984, and *z/OS TSO/E User's Guide*, SA32-0971.

Interrupting a TSO/E function

The Attention Interrupt key enables you to interrupt or end a process that is taking place. If you are in a process and you want to stop or see a message requesting information you do not have, you can press the Attention Interrupt key to end the process.

The Attention Interrupt key often is labeled PA1. Sometimes it is called an *escape* key and is labeled Esc.

2.10 Using TSO/E as batch job

```
//MIRIAM2 JOB 19,MIRIAM,NOTIFY=&SYSUID,MSGCLASS=T,  
// MSGLEVEL=(1,1),CLASS=A  
//*-  
//** THIS IS A SAMPLE TSO BATCH JCL *  
//**-*  
//STEP1 EXEC PGM=IKJEFT01  
//SYSPROC DD DSN=SYS1.OS390.CLIST,DISP=SHR  
//SYSTSIN DD *  
    DELETE 'MY.DATASET'  
    PROFILE  
/*  
//SYSTSPRT DD SYSOUT=*  
  
DD name for command output  
  
DD name where you put the TSO commands
```

Figure 2-10 TSO/E batch

Using TSO/E as batch job

Rather than waiting at a terminal for your job to run, you can use the terminal to prepare a job containing the commands and data you would have entered at the terminal, then use the **SUBMIT** command to run the job. In this case, you are using the facilities of TSO/E exactly as though you submitted the commands individually at the terminal. You need the following JCL statements to submit your job:

- ▶ A JOB statement to identify your job
- ▶ An EXEC statement with the name of the TSO/E terminal monitor program (IKJEFT01, IKJEFT1A, or IKJEFT1B)
- ▶ At a minimum, the following ddnames:
 - SYSTSPRT, which is used to control the output for your job. You can specify this DD as SYSOUT.
 - SYSTSIN, which is used as input for your TSO/E commands. It can be in stream (use a slash asterisk (*) to indicate the end of the stream).
- ▶ The DDNAMEs required by the application you intend to run

2.11 TSO/E Profile command

```
READY
1 profile
CHAR(0) LINE(0)      PROMPT   INTERCOM   NOPAUSE NOMSGID MODE      WTPMSG    NORECO
VER PREFIX(MIRIAM) PLANGUAGE(ENU) SLANGUAGE(ENU)
DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL
READY
2 delete old.dataset.sample1
ENTRY (A) MIRIAM.OLD.DATASET.SAMPLE1 DELETED
READY
3 profile noprefix
READY
4 profile
CHAR(0) LINE(0)      PROMPT   INTERCOM   NOPAUSE NOMSGID MODE      WTPMSG    NORECO
VER NOPREFIX          PLANGUAGE(ENU) SLANGUAGE(ENU)
DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL
READY
5 delete old.dataset.sample2
ENTRY (A) OLD.DATASET.SAMPLE2 DELETED
READY
6 prof msgid list
IKJ56688I CHAR(0) LINE(0)      PROMPT   INTERCOM   NOPAUSE MSGID    MODE      WTPMS
G    NORECOVER NOPREFIX          PLANGUAGE(ENU) SLANGUAGE(ENU)
IKJ56689I DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL
READY
7 profile prefix(newpref) list
IKJ56688I CHAR(0) LINE(0)      PROMPT   INTERCOM   NOPAUSE MSGID    MODE      WTPMS
G    NORECOVER PREFIX(NEWPREF) PLANGUAGE(ENU) SLANGUAGE(ENU)
IKJ56689I DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL
READY
```

Figure 2-11 TSO/E PROFILE command

TSO/E Profile command

Under TSO/E you always have a TSO profile, which is kept from one session to another. The way you issue the command to see your TSO profile depends on which environment you are running in:

- ▶ In TSO/E line-mode environment, type **PROFILE** and press the Enter key.
- ▶ In ISPF/PDF environment, type **TSO PROFILE** in the command line.

Initially, a user profile is prepared for you when arrangements are made for you to use the system. The authorized system programmer creates your user ID and your user profile. The system programmer is restricted to defining the same user profile for every user ID that the programmer creates.

This typical user profile is defined when a user profile table (UPT) is initialized to hexadecimal zeroes for any new user ID. Therefore, your initial user profile is made up of the default values of the operands discussed under this command.

Use the **PROFILE** command or the **PROFILE** subcommand of **EDIT** to establish, change, or list your user profile. The information in your profile tells the system how you want to use your terminal.

Changing a TSO profile

To change the TSO profile, type the **PROFILE** command followed by its operands and values. **PROFILE** can be shortened to **PROF**.

Figure 2-11 on page 79 shows the command in line-mode:

1. The command output is shown at 1.
2. When you use PREFIX in your TSO profile, all data sets you refer to, when *not* embedded in single quotation marks, are prefixed with your prefix, as shown at 2.
3. To deactivate the prefix, enter the command **PROFILE NOPREFIX**, as shown at 3.
4. Now, you do not see the prefix name displayed in the output of the **PROFILE** command.
5. Now, you do not have to use quotation marks, but you have to inform the complete data set name, as shown at 5. Otherwise, you will receive an error message or delete a wrong data set.
6. You can issue **PROFILE MSGID** to have all diagnostic messages you receive identified by their IDs, as shown at 6. You can add the **LIST** operand in your **PROFILE** command to list the new profile after the change, as shown at 6.
7. Use to **PROFILE** command to define a new PREFIX.

For more information about the **PROFILE** command and its operands, see *z/OS TSO/E Command Reference*, SA32-0975.

2.12 TSO/E languages

□ REXX	□ CLIST
<pre>/*REXX */ parse upper arg ax Address 'ISPEXEC' "LIBDEF ISPPLIB DATASET ID('VAINI.U.PANELS')" if rc = 0 then do Address "TSO" aplst = "VAINI.U.CLIST" if aplst ~= "" then , "ALTLIB ACT APPL(CLIST) DS("aplst")" Address "ISPEXEC" "SELECT CMD(DR" ax ") NEWAPPL(DREQ) PASSLIB " Address "TSO" if aplst ~= "" then , "ALTLIB DEACT APPL(CLIST)" Address 'ISPEXEC' "LIBDEF ISPPLIB " end else say '*** ERROR*** DR must be run under ISPF' Exit 0</pre>	<pre>PROC 0 DB IF .&DB = .DB THEN + CONTROL LIST SYMLIST CONLIST MSG PROMPT CONCATD F(SYSPROC) DA('DB2V710S.NEW.SDSNCLST') BEFORE CONCATD F(ISPPLIB) DA('DSN710.QPP.RUNLIB.LOAD') BEFORE ISPEXEC SELECT CMD(DSNECPRI SSID(DB2S)) NEWAPPL(DSNE) DECON F(ISPPLIB) DA('DSN710.QPP.RUNLIB.LOAD') DECON F(SYSPROC) DA('DB2V710S.NEW.SDSNCLST')</pre>

Figure 2-12 TSO/E languages

TSO/E languages

There are two languages available in the TSO/E environment: REXX and CLIST. REXX and CLIST can be used to customize and tailor your TSO/E environment specifically for the applications you want to use. Figure 2-12 shows a REXX exec and a sample CLIST procedure.

REXX

REXX is a programming language that is extremely versatile. With aspects, such as common programming structure, readability, and free format, it is a useful language for beginners and general users. Yet because the REXX language can be intermixed with commands to various host environments, provides powerful functions, and has extensive mathematical capabilities, it is also suitable for more experienced computer professionals.

The TSO/E implementation of the REXX language enables REXX execs to run in any MVS address space. You can write a REXX exec that includes TSO/E services and run it in a TSO/E address space, or you can write an application in REXX to run outside of a TSO/E address space.

Note: There is also a set of z/OS UNIX extensions to the TSO/E REXX language which enable REXX programs to access z/OS UNIX callable services. The z/OS UNIX extensions, called *syscall commands*, have names that correspond to the names of the callable services that they start, for example, access, chmod, and chown. For more information about the z/OS UNIX extensions, see *z/OS Using REXX and z/OS UNIX System Services*, SA23-2283.

CLIST

A CLIST is an interpretative language that helps you to work more efficiently with TSO/E. It is a command list language because the most basic CLISTS are lists of TSO/E commands.

When started, it issues the TSO/E commands in sequence. The CLIST language includes the programming tools that you need to write extensive, structured applications. CLISTS can perform several complex tasks, from displaying a series of full-screen panels to managing programs written in other languages.

You can include TSO/E commands and subcommands (and user-written commands and subcommands) in a CLIST at any point where the specific functions (for example, allocate, free, and so on) are required. For certain applications, a CLIST might consist entirely of commands and subcommands.

2.13 Interactive System Productivity Facility

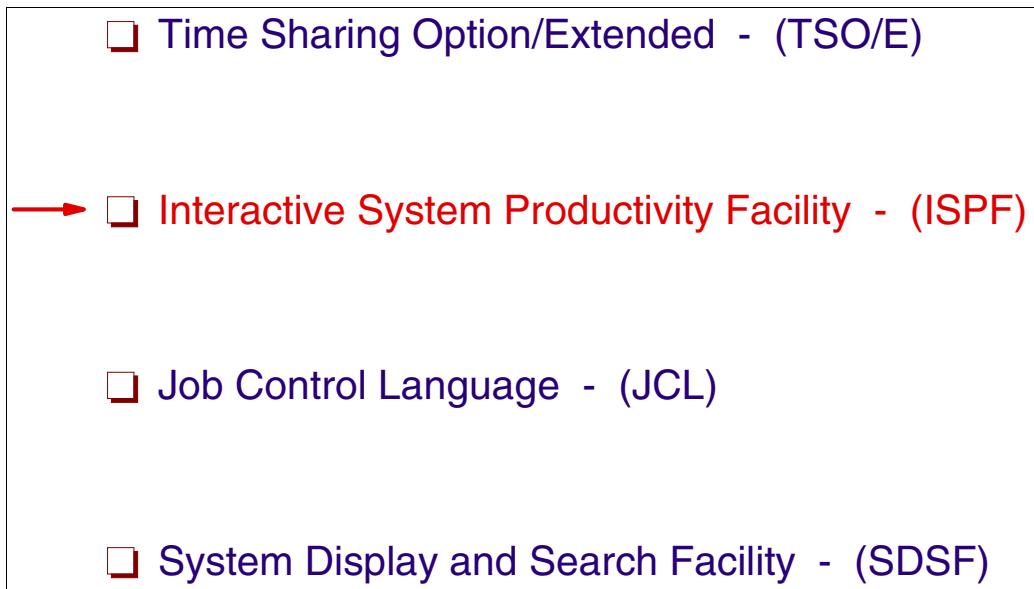


Figure 2-13 Interactive System Productivity Facility

Interactive System Productivity Facility

The ISPF/PDF is a set of panels that help you manage libraries of information about the z/OS system. The libraries are made up of units called *data sets* that can be stored and retrieved. You can have various kinds of information in data sets. Several examples are shown here:

- ▶ Source code
- ▶ Data, such as inventory records, personnel files, or a series of numbers to be processed
- ▶ Load modules

Using ISPF

ISPF is a multifaceted development tool set for the z/OS operating system. Since 1975, MVS programmers have used ISPF for host-based application development productivity. ISPF forms the basis of many TSO and CMS applications and provides extensive programmer-oriented facilities as well.

ISPF can be used in many ways, as these examples illustrate:

- ▶ Users can edit, browse and print data.
- ▶ Data processing administrators and system programmers can use ISPF to:
 - Monitor and control program libraries.
 - Communicate with MVS through TSO commands, CLISTS, or REXX execs.
- ▶ Programmers can use ISPF to develop a batch, interactive, or any other type of program and its documentation.
- ▶ Terminal users can start a wide range of utilities, such as search, compare, compilers, and so forth.

ISPF components

ISPF helps programmers develop interactive applications called *dialogs*. Dialogs are interactive because ISPF uses them to communicate with terminal users through a series of panels while the users perform application development tasks.

ISPF panels provide the following functions:

- ▶ Provide access to ISPF functions through menus.
- ▶ Request information from users through data entry panels.
- ▶ Provide information from users through scrollable data displays.

ISPF components

These are the main components of ISPF:

- ▶ Dialog Manager (DM)

DM provides services to dialogs and users. These include display, variable services, input and output, user and application profiles, table management, system interface services, dialog testing and debugging aids, and other services.

- ▶ Program Development Facility (PDF)

PDF provides services to assist the dialog or application developer. These include the edit and browse functions, data set and catalog utilities, TSO or CMS command interfaces, and data set search and compare functions.

- ▶ Software Configuration and Library Manager (SCLM)

The SCLM facility provides library management capabilities, such as versioning, auditing, and promotion. It also provides configuration management capabilities to track how all of the pieces of an application fit together, including source code, objects, load modules, test cases, documentation, and other items. The Build function tracks and starts the necessary compilers, assemblers and linkage editors.

- ▶ Client/Server component

The Client/Server component enables the users of ISPF applications to use a workstation running Windows or UNIX to display the panels of an ISPF application. It does this using the graphical user interface (GUI) of the workstation. Because of this, there is no requirement to change your existing ISPF applications to run in a GUI environment.

2.14 ISPF: Data set types supported

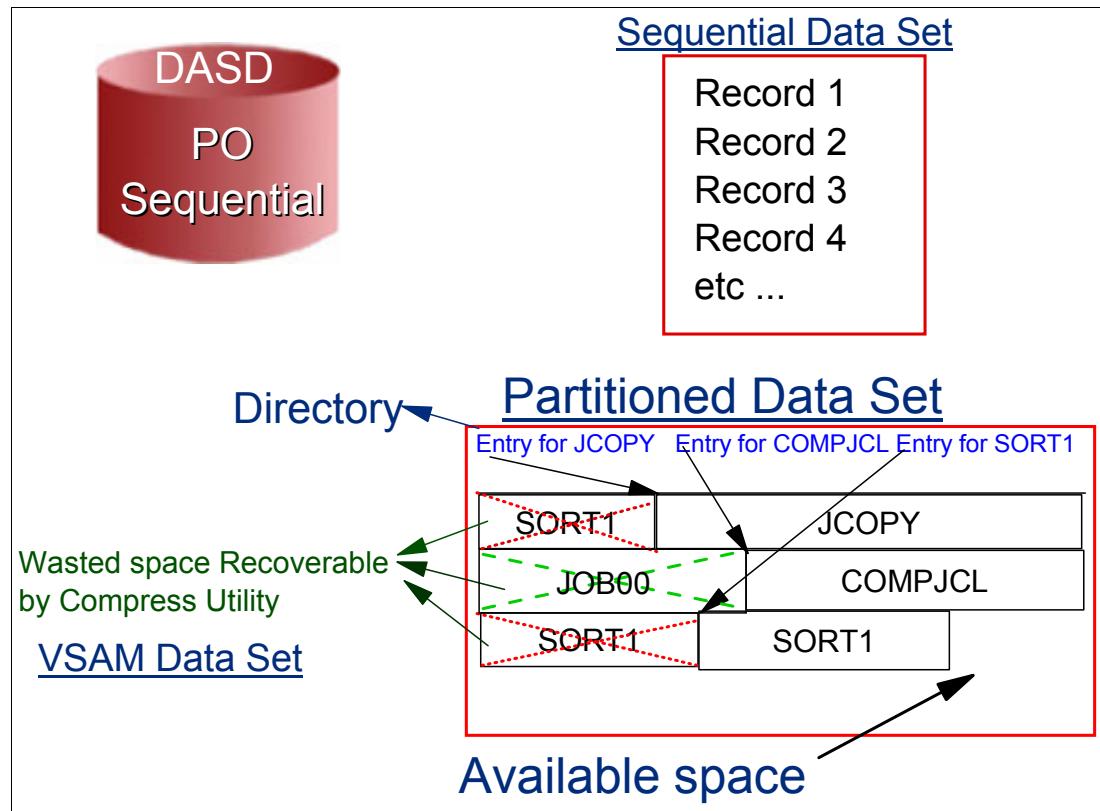


Figure 2-14 ISPF and ISPF/PDF data set types supported

ISPF: Supported data set types

A *data set* is a collection of logically related data; it can be a source program, a library of macros, or a file of data records used by a processing program. *Data records* are the basic unit of information used by a processing program. ISPF supports the following data set types for any ISPF options, such as Edit, Browse, and Delete:

- ▶ Sequential data set

A sequential data set is a data set whose records are organized on the basis of their successive physical positions. In a sequential data set, records are stored and retrieved in a sequential order.
- ▶ Partitioned data set and partitioned data set extended

A partitioned data set (PDS) is a data set on direct access storage that is divided into partitions called *members*. Each member can contain a program, part of a program, or data.

A *partitioned data set* is like a collection of sequential data sets, where the individual members each have a unique name. A directory index is used to locate members in the partitioned data set. The directory consists of 256-byte records, each one containing directory entries. There is one directory entry for each member.

For Virtual Storage Access Method (VSAM) data sets, ISPF supports creation, obtaining data set information, and deletion. Browse and Update are supported by DITTO, an IBM licensed program installed under ISPF.

Tip: You can create and delete VSAM data sets and obtain VSAM data set information. VSAM data sets are supported for Edit, Browse, and View if the ISPF Configuration table has been customized to enable such support.

ISPF does *not* support some of the following data set types:

- ▶ Record format variable block spanned (VBS) data sets
- ▶ Direct access (DA) data sets
- ▶ Tape data sets
- ▶ Generation data group (GDG) base data sets

2.15 ISPF Option 2

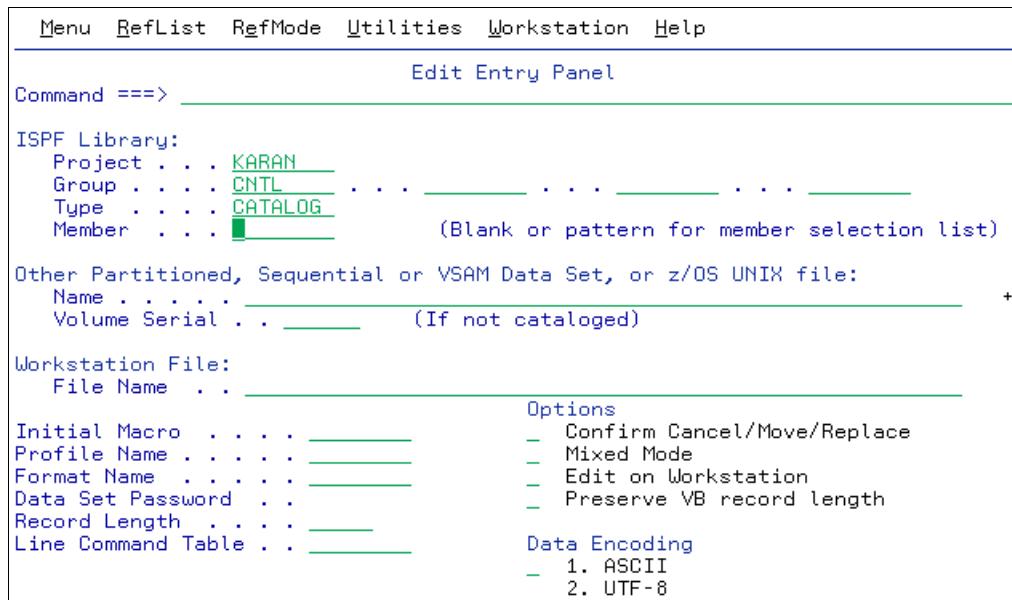


Figure 2-15 ISPF Edit Option 2

ISPF primary menu panel

The primary option menu panel, shown in Figure 2-16 on page 88, is the first panel that displays when you start ISPF.

Attention: The panelid shown is ISR@PRIM. To display the panelids on all ISPF panels, when you Enter ISPF, type panelid on the command line.

ISPF edit mode (Option 2)

You can use Edit to create or change source data, such as program code and documentation, using the ISPF full-screen editor. You can scroll the data up, down, left, or right. You can change the data by using Edit line commands, which are entered directly on a line number, and primary commands, which are entered on the command line.

Member name

On the panel shown in Figure 2-15, you can enter a Member name. Otherwise, you can press the Enter key from the panel to display all the member names, as shown in Figure 2-17 on page 88.

Members created through ISPF must follow the following naming conventions:

- ▶ The first character must be alphabetic or special (@ # \$).
- ▶ Characters 2-8 must be alphabetic, numeric, or special (@ # \$).

The primary option menu panel is shown in Figure 2-16.

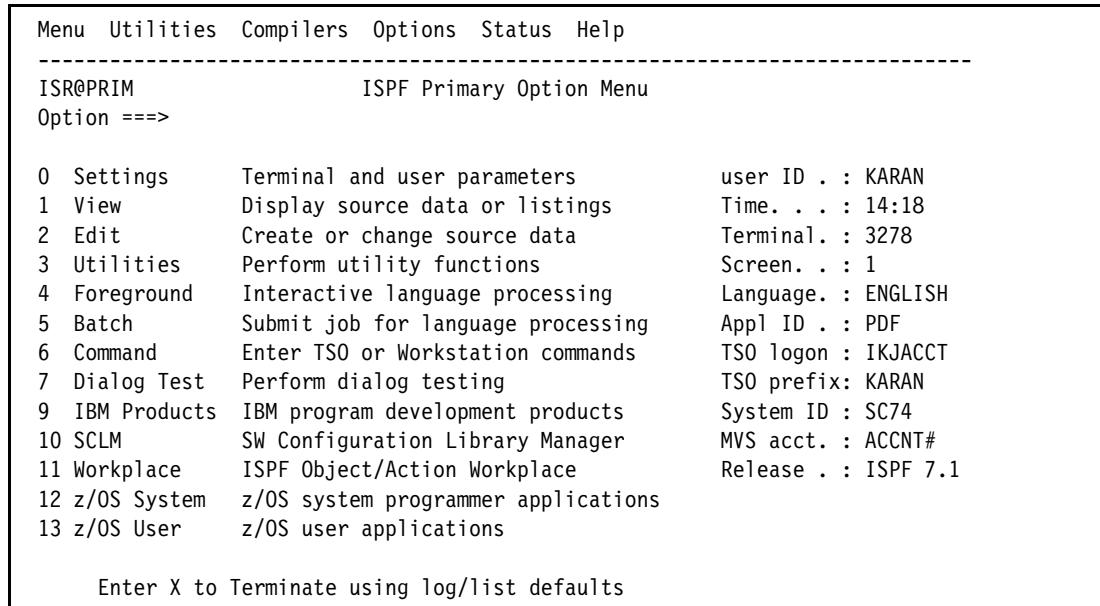


Figure 2-16 ISPF primary option menu

ISPF Option 2

When you select Option 2 from the primary option menu shown in Figure 2-16, the edit entry panel is displayed. You use this panel to enter the name of an ISPF library, an “other” data set or z/OS UNIX file, or a workstation file. If more than one data set or file name is specified, ISPF accepts the workstation file name first, then the “other” data set name, and then the ISPF library name.

In Figure 2-15 on page 87, the data set name (PDS member) displayed is KARAN.CNTL.CATALOG). If you press the Enter key, the members in that partitioned data set are displayed; see Figure 2-17.

Menu	functions	Utilities	Help
EDIT	KARAN.CNTL.CATALOG		Row 0000001 of 0000041
Command	====>		Scroll ==> CSR
Name	Prompt	Size	Created
.	\$CINDEX	36	2014/02/26 2014/03/04 16:36:34
.	#CCAS	60	2014/02/28 2014/02/28 15:24:17
.	#CDELRCV	19	2014/02/28 2014/02/28 13:19:05
.	#CMRGCAT	21	2014/02/28 2014/02/28 12:18:27
.	#CMVECAT	36	2014/02/28 2014/02/28 13:00:53
.	#CRACF	46	2014/02/28 2014/02/28 14:25:58

Figure 2-17 Part of the members displayed from PDS ROGERS.JCL.VERS5

Edit Option 2

The Edit Option 2 enables you to create, display, and change data stored in ISPF libraries, other partitioned or single-volume or multivolume sequential data sets. You can also set the data encoding options to Unicode Transformation Format, 8-bit (UTF-8) or American Standard Code for Information Interchange (ASCII) to view or edit a data set containing data encoded as Unicode or ASCII.

2.16 ISPF: Other Sequential data set

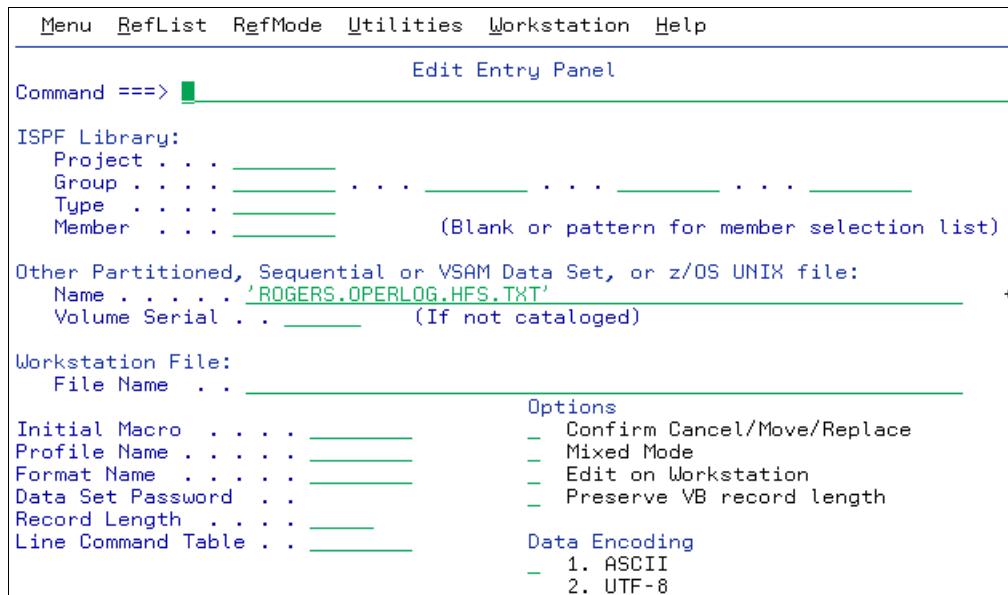


Figure 2-18 Specifying an Other Sequential data set

Other Sequential data set specification

ISPF enables you to create data sets and member names that follow the ISPF naming convention shown in Figure 2-18, ('ROGERS.OPERLOG.HFS.TXT').

All data sets and member names created in ISPF are converted to uppercase. If you create members outside of ISPF that do not meet these conventions, they are displayed in ISPF member lists and can be selected from those lists. These member names can also be specified for the Browse service, with the exception of member names containing lowercase alphabetic characters. (ISPF converts the member name to uppercase before searching for the member and therefore cannot process a lowercase member.)

Member names not meeting the ISPF naming convention are not supported for the other ISPF services. The same applies to data sets.

ISPF has useful online help and tutorial facilities (available when using ISPF). For example, if you need help completing the data requested by an ISPF utility, you can use the tutorial to help you understand the data entry requirements for that utility. For information about using the tutorial, see *z/OS ISPF Dialog Developer's Guide*, SC19-3619.

Data set name convention

The data set name must be a unique name with a maximum of 44 characters and with a maximum of 22 name segments or level qualifiers. The first name in the left is called the high-level qualifier (HLQ). The last name in the right is the low-level qualifier (LLQ).

Level qualifiers are separated by a dot (.), and each level qualifier can be from 1 up to 8 characters. The first character of a qualifier must be alphabetical (A-Z) or a national character (@ # \$). The 7 remaining characters can be alphabetical, national, numeric (0-9), or a hyphen (-); for example:

MYID.JCL.FILE2 HLQ: MYID 3 qualifiers

Member name of a partitioned data set

The member name can be 8 bytes, where the first byte must be alphabetical (A-Z) or a national character (@ # \$). The 7 remaining characters can be alphabetical, national, or numeric (0-9).

Important: All member names created in ISPF are converted to uppercase. If you create members outside of ISPF that do not meet these conventions, they are displayed in ISPF member lists and can be selected from those lists.

These member names can also be specified for the Browse service with the exception of member names containing lowercase alphabetic letters. (ISPF converts the member name to uppercase before searching for the member and therefore cannot process a lowercase member.) Member names not meeting the ISPF naming convention are not supported for the other ISPF services.

2.17 ISPF components



Figure 2-19 ISPF components

ISPF components

ISPF consists of four major components, DM, PDF, SCLM, and Client/Server, as shown in Figure 2-19. These components are considered one element in all releases of z/OS.

ISPF Dialog Manager

DM provides services to dialogs and users. A *dialog* is the interaction between a person and a computer. It helps a person who is using an interactive display terminal to exchange information with a computer. The user starts an interactive application through an interface that the system provides. The dialog with the user begins with the computer displaying a panel and asking for user interaction. It ends when the task for which the interactions were initiated is completed. Dialog Manager is composed of six elements:

- ▶ Functions. A function is a command procedure or a program that performs processing requested by the user. It can start ISPF dialog services to display panels and messages, build and maintain tables, generate output data sets, and control operational modes. They can be written as:
 - REXX or CLIST command procedures
 - Programs
- ▶ Panel definitions. A panel definition is a programmed description of the panel. It defines both the content and format of a panel. Most panels prompt the user for input. The user's response can identify which path is to be taken through the dialog, as on a selection panel. The response can be interpreted as data, as on a data-entry panel.

- ▶ Message definitions. Message definitions specify the format and text of messages to users. A message can confirm that a user-requested action is in progress or completed, or it can report an error in the user's input.
- ▶ Table. Tables are two-dimensional arrays that contain data and are created by dialog processing. They can be created as a temporary data repository, or they can be retained across sessions. A retained table can also be shared among several applications. The type and amount of data stored in a table depends on the nature of the application. Not all dialogs use tables.
- ▶ File tailoring skeletons. Skeletons work like a fill-in-the-blank exercise. They take dialog variables and put them into a data set containing statements that control the output format. The output data set can be used to drive other processes. File skeletons are frequently used to produce job data sets for batch execution. Various dialogs can use this kind of resource.
- ▶ Dialog variables. ISPF services use variables to communicate information among the various elements of a dialog application. ISPF provides a group of services for variable management. Variables can vary in length from zero to 32K bytes.

Program Development Facility

PDF provides View, Browse, Edit, and library access services that can be combined in a dialog with any of the ISPF services. The library access services run functions involving members of a programming library. These functions include adding, finding, and deleting members, and displaying member lists.

Software Configuration Library Manager

SCLM is a software tool that helps you develop complex software applications. Throughout the development cycle, SCLM automatically controls, maintains, and tracks all of the software components of the application. You can lock the version being edited in a private library and then promote it. Use SCLM to create, control, maintain, and track software components for a project. For more information about SCLM, see *z/OS ISPF SCLM Project Manager's and Developer's Guide*, SC19-3625.

ISPF Client/Server - the Workstation Agent component

The Workstation Agent (WSA) component enables you to run ISPF on a programmable workstation and display the panels using the display function of your workstation operating system. Manuals in the ISPF library refer to this as running in GUI mode. The ISPF WSA is supported on the following platforms:

- ▶ Microsoft Windows
- ▶ AIX
- ▶ HP-UX
- ▶ Solaris

Connecting to a workstation for data access has a direct effect on your installation's processor processing time. One reason for using the ISPF Client/Server function is to offload processor cycles from the host to a less expensive workstation. But even if that is not your goal, an added benefit is that your users can use the connection for distributed editing.

Therefore, they can use their favorite editor to work with your data, whether that means using a host editor on host and workstation files, or using a workstation editor on the same files. By making the connection to the workstation, a user can edit workstation files on ISPF, or host files on his workstation. The distributed edit function can be used in standard 3270 mode, or in ISPF GUI mode.

2.18 Sample CLIST to allocate ISPF and SDSF data sets

```
ALLOC FI(SYSEXEC) DS('ISP.SISPEXEC' 'ISF.SISFEXEC') REUS SHR
ALLOC FI(SYSPROC) DS('USER.CLISTS.DSN' 'ISP.SISPCLIB') REUS SHR
ALLOC FI(ISPPLIB) DS('ISP.SISPENU' 'ISF.SISFPLIB') SHR
ALLOC FI(ISPMLIB) DS('ISP.SISPMENU' 'ISF.SISFMLIB') SHR
ALLOC FI(ISPSLIB) DS('ISP.SISPENU' ) SHR REUS
ALLOC FI(ISPTLIB) DS('ISP.SISPTENU' 'ISF.SISFTLIB') SHR REUS
ALLOC FI(SYSHHELP) DS('ISP.SISPHHELP' 'ISF.SISFTLIB') SHR REUS
ALLOC FI(ISPTABL) DS('USERID.PROFILE.DSN') SHR
ALLOC FI(ISPPROF) DS('USERID.PROFILE.DSN') SHR
ALLOC FI(ISPLIST) SYSOUT(A) LRECL(121) RECFM(F B A)
ALLOC FI(ISPLOG) SYSOUT(A) LRECL(121) RECFM(V A)
ALLOC FI(ISPCTL0) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(80) +
BLKSIZE(800) RECFM(F B)
ALLOC FI(ISPCTL1) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(80) +
BLKSIZE(800) RECFM(F B)
ALLOC FI(ISPCTL2) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(80) +
BLKSIZE(800) RECFM(F B)
ALLOC FI(ISPWRK1) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(256) +
BLKSIZE(2560) RECFM(F B)
ALLOC FI(ISPWRK2) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(256) +
BLKSIZE(2560) RECFM(F B)
ALLOC FI(ISPLST1) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(121) +
BLKSIZE(1260) RECFM(F B A)
ALLOC FI(ISPWRK2) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(121) +
BLKSIZE(1260) RECFM(F B A)
ISPSTART PANEL(ISR@PRIM) NEWAPPL(ISR)
```

Figure 2-20 Sample CLIST to allocate ISPF and SDSF data sets

Sample CLIST to allocate ISPF and SDSF data sets

ISPF runs in a TSO/E environment. Before ISPF can be started, data sets must be available. This can be achieved in any of the following ways:

1. Allocate them in the logon procedure.
2. Use dynamic allocation through the TSO/E **ALLOC** command, as shown in Figure 2-20.
3. Save all commands shown in Figure 2-20 as a command and start during the logon process as the **ISPPDF** command shown in Figure 2-8 on page 75.

Required DDNAMEs

The required DDNAMEs are listed here:

- SYSEXEC for partitioned data sets containing REXX execs
- SYSPROC for partitioned data sets containing CLIST and REXX execs
- ISPPLIB for panel definitions libraries
- ISPTLIB for input table definitions libraries
- ISPMLIB for message libraries
- ISPPROF for the *user profile* data set under ISPF/PDF environment. ISPF/PDF use this data set for storing variables and settings to be used from one TSO/ISPF session to another.

Other important DDNAMEs

The DDNAMEs for ISPTABL (for output tables) and ISPSLIB (for skeletons) might be requested for specific applications.

For various data sets, if not preallocated, ISPF allocates them:

- ▶ ISPLOG DD name. The data set where ISPF logs commands issued by the user and some ISPF functions log errors detected.
- ▶ ISPLIST DD name. ISPF uses this data set when user requests printed output.

The DDNAMEs ISPCTLx (where x can be 1 - 9, A - W) are used by ISPF as a temporary data set. ISPF can use one for each logical window to generate JCL or utility control statements or to generate listings. ISPF can run up to 32 logical screens at one time. The default value is 8. The installation can change the default value by modifying the ISRCFG table. ISPCTL0 is used only by Edit for the **SUBMIT** command.

The DDNAMEs ISPWRKx are used by ISPF for file tailoring services with ISPFILE allocated to a PDS. The DD names ISPLSTx are used for generated listings. The same preallocation can be done by the **ALLOCATE** command in a CLIST or in a REXX exec to be run before the ISPF start.

2.19 ISPF primary option menu

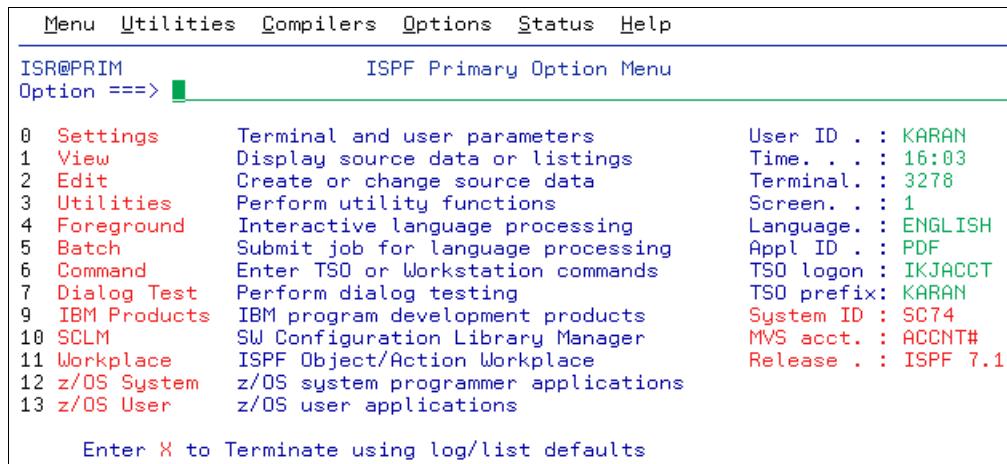


Figure 2-21 ISPF primary option menu

ISPF primary option menu

ISPF is started in a TSO/E environment through an ISPF, or PDF, or ISPSTART command. The ISPF Primary Option Menu contains the options that you can use to create your own applications online. If your installation has a customized ISPF Primary Option Menu, the menu might not contain all of options shown in Figure 2-21, or it might contain certain installation-specific options. Most ISPF panels have action bars at the top; many panels also have point-and-shoot text fields. The following list describes the panel options:

- 0 Settings displays and changes selected ISPF parameters, such as terminal characteristics and function keys.
- 1 View displays data using the View or Browse function. You can use View or Browse to look at (but not change) large data sets, such as compiler listings. You can scroll the data up, down, left, or right. If you are using Browse, a FIND command, entered on the command line, enables you to search the data for a character string. If you are using View, you can use all the commands and macros available to you in the Edit function.
- 2 Edit enables you to create or change source data, such as program code and documentation, using the ISPF full-screen editor. You can scroll the data up, down, left, or right. You can change the data by using Edit line commands, which are entered directly on a line number, and primary commands, which are entered on the command line.
- 3 Utilities perform library and data set maintenance tasks, such as moving or copying library or data set members, displaying or printing data set names and volume table of contents (VTOC) information, comparing data sets, and searching for strings of data.
- 4 Foreground calls IBM language processing programs in the foreground.
- 5 Batch calls IBM language processing programs as batch jobs. ISPF generates JCL based on information that you enter and submits the job for processing.
- 6 Command calls TSO commands, CLISTS, or REXX execs under ISPF.
- 7 Dialog Test tests individual ISPF dialog components, such as panels, messages, and dialog functions (programs, commands, menus).
- 9 IBM Products enable you to select other installed IBM program development products on your system.

The following products are supported:

- ▶ Tivoli Information Management
 - ▶ COBOL Structuring Facility foreground dialog (COBOL/SF)
 - ▶ Screen Definition Facility II (SDF II) licensed program
 - ▶ Screen Definition Facility II-P (SDF II-P) licensed program
- 10** SCLM controls, maintains, and tracks all of the software components of an application.
- 11** Workplace gives you access to the ISPF Workplace, which combines many of the ISPF functions onto one object-action panel.
- 12** z/OS System gives you access to the z/OS System Programmer Primary Option Menu. It contains options for z/OS elements that are used by system programmers and administrators:
 - ▶ Graphical Data Display Manager (GDDM) Print Queue Manager
 - ▶ Hardware Configuration Definition (HCD) I/O configuration
 - ▶ DCE configuration
 - ▶ APPC Administration
 - ▶ z/OS Workload Manager (WLM)
 - ▶ IBM First Failure Support Technology (FFST)
 - ▶ Infoprint Server
 - ▶ z/OS Resource Management Facility (RMF)
 - ▶ System Modification Program/Extended (SMP/E)
 - ▶ Transmission Control Protocol/Internet Protocol (TCP/IP) Network Print Facility (NPF)
- 13** z/OS User gives you access to the z/OS Applications panel. It contains options for z/OS elements that are used by most ISPF users:
 - ▶ BookManager Build
 - ▶ BookManager Read
 - ▶ BookManager Index Creation
 - ▶ Data Facility Storage Management Subsystem (DFSMS) Removable Media Manager (DFSMSSrmm)/Interactive Storage Management Facility (ISMF)
 - ▶ DFSMS Data Facility Product (DFSMSdfp)/ISMF
 - ▶ Data Facility Sort (DFSORT)
 - ▶ Bulk Data Transfer (BDT) File-to-File
 - ▶ Interactive Problem Control System (IPCS)
 - ▶ z/OS UNIX Browse
 - ▶ z/OS UNIX Edit
 - ▶ z/OS UNIX Shell
 - ▶ Security Server
 - ▶ TSO/E Information Center Facility
 - ▶ SDSF
- X** EXIT leaves ISPF using the log and list defaults. You can change these defaults from the Log>List pull-down on the ISPF Settings panel action bar.

2.20 ISPF panel areas

The figure shows the z/OS Primary Option Menu. Area 1 highlights the menu bar with options: Menu, Utilities, Compilers, Options, Status, Help. Area 2 highlights the 'Option' input field. Area 3 highlights the list of menu options and their descriptions. Area 4 highlights the function key definitions at the bottom.

Option	Description	Setting
0 Settings	Terminal and user parameters	User ID . : ROGERS
1 View	Display source data or listings	Time. . . : 13:22
2 Edit	Create or change source data	Terminal. : 3278
3 Utilities	Perform utility functions	Screen. . . : 2
4 Foreground	Interactive language processing	Language. : ENGLISH
5 Batch	Submit job for language processing	Appl ID . : PDF
6 Command	Enter TSO or Workstation commands	TSO logon : IKJACCT
7 Dialog Test	Perform dialog testing	TSO prefix: ROGERS
9 IBM Products	IBM program development products	System ID : SC75
10 SCLM	SW Configuration Library Manager	MVS acct. : ACCNT#
11 Workplace	ISPF Object/Action Workplace	Release . : ISPF 6.3
12 z/OS System	z/OS system programmer applications	
13 z/OS User	z/OS user applications	

Enter X to Terminate using log/list defaults

F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel

Figure 2-22 ISPF panel areas

ISPF panel areas

ISPF panels can contain the following areas:

1. Action bar. This is the area at the top of an ISPF panel that contains choices that give you access to actions available on that panel.
2. Option. The fields in this column are point-and-shoot text fields. For example, if you want to select the Edit Panel, to create or change source data, you type it in the option input field and press Enter. You can also type commands in the Option field.
3. Dynamic status area. You can specify what you want to be displayed in this area; in Figure 2-22 it displays descriptions of all options you can take from this panel.
4. Function Keys. ISPF uses Common Use Access (CUA)-compliant definitions for function keys F1-F12. For example, PF1 is used to start the ISPF Help function. You can eliminate this area by typing PFSHOW OFF in area 2, the Option field.

2.21 Action bars

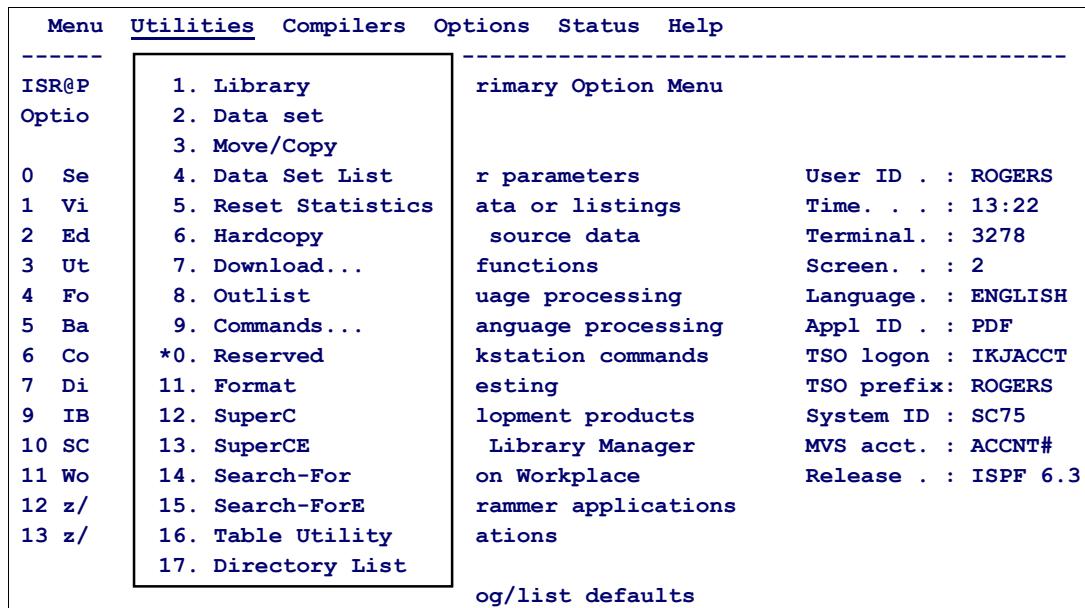


Figure 2-23 Action bars

Action bars

Action bars give you another way to move through ISPF. If the cursor is located somewhere on the panel, there are several ways to move it to the action bar:

- ▶ Use the cursor movement keys to manually place the cursor on an action bar choice.
- ▶ Type ACTIONS on the command line and press Enter to move the cursor to the first action bar choice.
- ▶ Press F10 (Actions) or the Home key to move the cursor to the first action bar choice.

Restriction: ISPF does not provide a mouse emulator program. You can use Select in conjunction with point-and-shoot text fields and action bar choices to simulate moving the cursor to a field and pressing Enter. Each ISPF panel has different action bars.

When you make an action bar selection, the selected item is highlighted and ISPF displays a pull-down menu, that is, a list of numbered choices extending from the selection you made on the action bar. Figure 2-23 shows the result when you select **Utilities** on the panel.

You can select an action either by typing in its number and pressing Enter, or by selecting the action with your cursor. ISPF displays the requested panel. If your choice contains an ellipsis (...), ISPF displays a window. When you exit this panel or window, ISPF closes the pull-down menu and returns you to the panel from which you made the initial action bar selection.

Utilities action bar

This choice is available from many panels in ISPF and displays the options listed on the Utility Selection panel. The following choices are available from the Utilities pull-down menu:

1. Library Displays the Library Utility panel
2. Data set Displays the data set Utility panel
3. Move/Copy Displays the Move/Copy Utility panel

- 4. Data set List** Displays the data set List Options panel
- 5. Reset Statistics** Displays the Reset ISPF Statistics panel
- 6. Hardcopy** Displays the Hardcopy Utility panel
- 7. Download** Displays the panel that enables you to download workstation clients and other files from the host
- 8. Outlist** Displays the Outlist Utility panel
- 9. Commands** Displays the Command Table Utility panel
- *0 Reserved** Reserved for future use by ISPF; an unavailable choice
- 11. Format** Displays the Format Specification panel
- 12. SuperC** Displays the SuperC Utility panel
- 13. SuperCE** Displays the SuperCE Utility panel
- 14. Search-for** Displays the Search-For Utility panel
- 15. Search-forE** Displays the Search-ForE Utility panel
- 16. Table Utility** Displays the ISPF Table Utility panel
- 17. Directory List** z/OS UNIX Directory List Utility

2.22 Customizing a TSO/ISPF/PDF session

- ❑ The Options action bar choice displays the settings available for customization
 - Select Option 1. General Settings
 - Pull-down provides access to many ISPF settings, including CUA attributes and colors, keylists, and point-and-shoot fields, and the Dialog Test Application ID pop-up

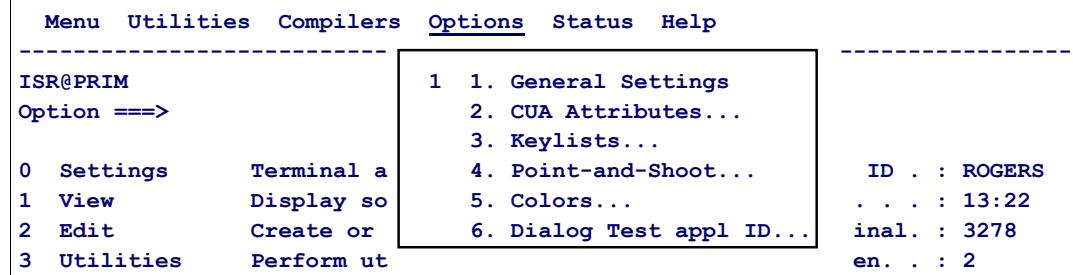


Figure 2-24 Customizing a TSO/ISPF/PDF session

Customizing a TSO/ISPF/PDF session

You can customize a TSO/ISPF/PDF session by selecting **Options** → **General Settings** as shown in Figure 2-24. The Options pull-down provides access to many ISPF settings, including CUA attributes and colors, keylists, and point-and-shoot fields, and the Dialog Test Application ID pop-up.

For more information, see the topic about General Settings (Option 1) in Figure 2-25 on page 101. The first line shows other action fields where you can choose to change default settings, such as colors, function keys, and so on.

ISPF Help function

ISPF has a powerful Help function that can teach you how to use all of the default ISPF options. Access it by pressing the Help program function key or typing `help` in the command line. You can see and change the program function key assignments by using the **KEYS** command.

You can learn how to use ISPF by just using the help function. At the Primary Option Menu press Help to learn about the options. In the Help panel, press Help again to learn how to move through help panels. Several Help options are cursor-sensitive, meaning you have to move the cursor to that option to get more information.

A useful source of information for beginners in the ISPF environment is *Interactive System Productivity Facility Getting Started*, SC34-4440.

2.23 Customizing an ISPF session

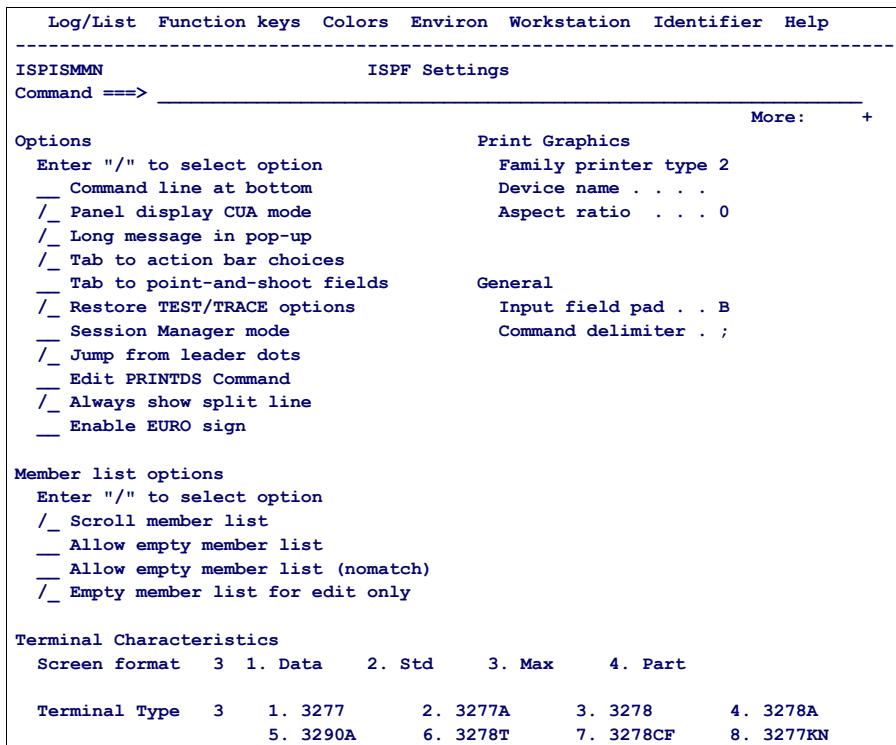


Figure 2-25 Customizing an ISPF session

Customizing an ISPF session

The General Settings option, specified in Figure 2-24 on page 100, enables you to display and change a variety of ISPF parameters at any time during the ISPF session. Changes remain in effect until you change the parameter again, and ISPF saves them from session to session. This topic explains how to use the fields on the ISPF Settings panel and the action bar choices.

Tip: If you select Option 0 on the ISPF Primary Option Menu, you will also see panelid ISPISMNN.

For a complete description of all options you can use to customize a session, see *z/OS Interactive System Productivity Facility (ISPF) User's Guide Volume II*, SC19-3628.

Important: The panel shown in Figure 2-25 has been customized and does not show the initial defaults.

2.24 Allocating data sets: Utility option

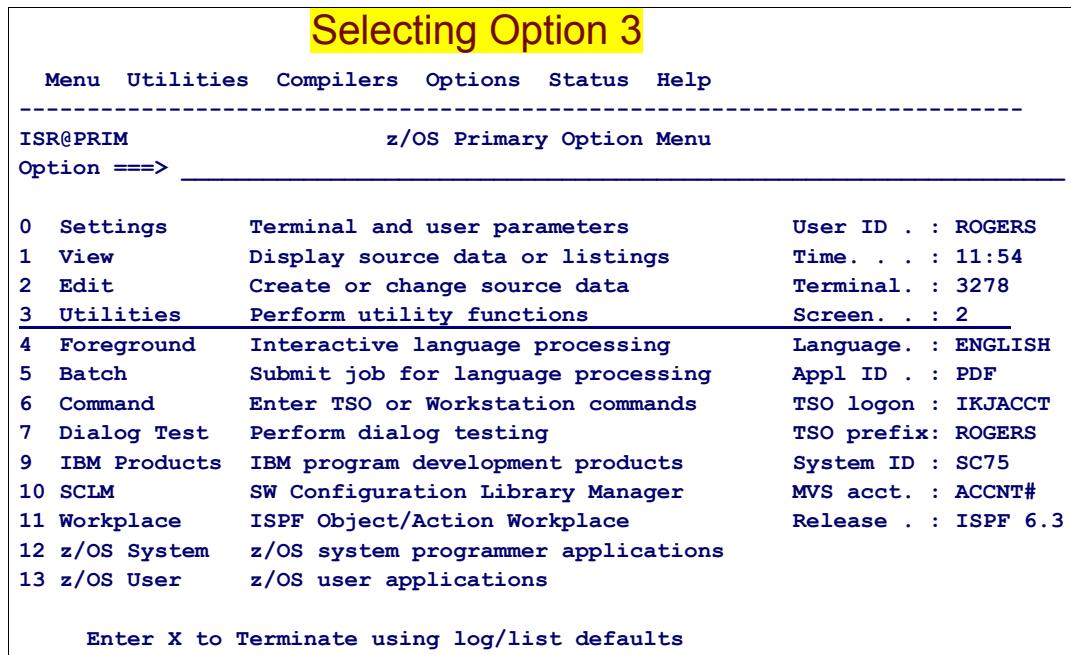


Figure 2-26 Allocating a data set: Utility option

Allocating data sets: Utility option

A data set, or file, is an area that is reserved on either disk or tape to enable you to write programs and store data. Before you can edit or store data, you must instruct the system to allocate space on disk, often referred to as *direct access storage devices* (DASD), and provide information to identify the format of this data set.

There are two ways to allocate a data set:

- ▶ By using ISPF dialogs
- ▶ By using JCL

Utilities option 3

In the ISPF Primary Option Menu, type 3 in the command line and press the Enter key. Option 3 displays the utilities panel, as shown in Figure 2-27 on page 103. The Utilities option (3) provides a variety of functions for library, data set, and catalog maintenance.

2.25 Utility Selection Panel

```
Menu Help
-----
ISRUTIL          Utility Selection Panel
Option ==> 2

1 Library      Compress or print data set. Print index listing. Print,
                rename, delete, browse, edit or view members
2 Data Set     Allocate, rename, delete, catalog, uncatalog, or display
                information of an entire data set
3 Move/Copy    Move, or copy members or data sets
4 Dslist       Print or display (to process) list of data set names.
                Print or display VTOC information
5 Reset        Reset statistics for members of ISPF library
6 Hardcopy     Initiate hardcopy output
7 Transfer     Download ISPF Client/Server or Transfer data set
8 Outlist      Display, delete, or print held job output
9 Commands     Create/change an application command table
11 Format      Format definition for formatted data Edit/Browse
12 SuperC      Compare data sets                               (Standard Dialog)
13 SuperCE     Compare data sets Extended                   (Extended Dialog)
14 Search-For   Search data sets for strings of data      (Standard Dialog)
15 Search-ForE  Search data sets for strings of data Extended (Extended Dialog)
16 Tables      ISPF Table Utility
17 Uplist      Print or display (to process) z/OS UNIX directory list
```

Figure 2-27 Utility Selection Panel option 2

Utility Selection Panel

The Utility Selection Panel gives you the ability to select utilities to perform a variety of functions:

- ▶ Compress or print a data set
- ▶ Create, rename, or delete a data set
- ▶ Move or copy data sets or members
- ▶ Search for strings in a data set
- ▶ Compare data sets

ISPF option 3.2 enables you to reserve space on a storage device and identify this space with a data set name, often referred to as a DSN or DSNAME.

In this example you will allocate a data set. To accomplish this, type 2 in the Option field and press Enter.

Fast path: From the ISPF primary panel, from the command line, you can go directly to the panel shown in Figure 2-28 on page 104 by specifying (=3.2) on the command line, which is essentially specifying option 3, then option 2.

2.26 Data set Utility: Allocating a data set

```
Menu RefList Utilities Help
-----
Option ==> a

A Allocate new data set          C Catalog data set
R Rename entire data set        U Uncatalog data set
D Delete entire data set        S Short data set information
blank Data set information     V VSAM Utilities

ISPF Library:
Project . . . ROGERS           Enter "/" to select option
Group . . . PRIVATE             / Confirm Data Set Delete
Type . . . JCL

Other Partitioned, Sequential or VSAM Data Set:
Name . . . . .                 (If not catalogued, required for option "C")
Volume Serial . . . . .         (If password protected)

Data Set Password . . . . .     (If password protected)
```

Figure 2-28 Allocating a data set - data set Utility panel

Data set Utility panel: Allocating a data set

The data set Utility panel presents a variety of actions you can perform. You can allocate, delete, rename, catalog, uncatalog, or obtain information about a specific data set. The **V** option enables you to create and obtain information about VSAM data sets easily.

From the ISPF Primary Option Menu you can go directly to this panel by typing =3.2 in the Option field, as discussed previously. To allocate a new data set, type **A** on the Option field, as shown in Figure 2-28. Next, specify the name of data set you want to create. In Figure 2-28, the following information was entered:

Project . . . ROGERS
Group . . . PRIVATE
Type JCL

A data set with a three-level name can be entered in the Project, Group, and Type fields, as shown in area 1 in Figure 2-28. Type one level of the name in each field. ISPF does not add your prefix as HLQ. The prefix or HLQ in this example is ROGERS.

Data set name and volume serial

You can specify the data set name and optionally a volume, as shown in area 2 of the panel:

Other Partitioned, Sequential or VSAM data set:
Data set Name. . . 'HLQ.XXX.XXXX.XXX'
Volume Serial . . .

Important: Embed the data set name in single quotation marks unless you want the HLQ to equal your prefix.

If both a library (1) and a data set name (2) are specified on the same panel, the data set name takes priority. Therefore, to specify a library, leave the data set Name field blank. We did not enter the data set name option field. Therefore, the values specified in the library name are used. See Figure 2-18 on page 89 for data set naming conventions.

Data set name standards are site-dependent and can be protected by RACF or another security product. If you do not have the authority to allocate a DSN with a specific name, your request fails and you receive an error message.

The following list shows examples of data set names:

- ▶ SYS1.PARMLIB
- ▶ SYS1.PARMLIB.BACKUP.D99156
- ▶ ROGERS.PRIVATE.JCL
- ▶ GEORGE.TEST.NEW.SYSTEM.JCLLIB

High-level qualifier

The HLQ is the first part of the DSN. In the examples, the HLQs are SYS1 (usually reserved for MVS system DSNs), and MIRIAM, which can be your user ID. Several system data sets must be named as specified, but for personal or in-house data sets, use names that are meaningful and easy to associate with a user or application, and to enable efficient security and DASD maintenance strategies to be maintained.

2.27 Allocate New Data Set panel

```
Menu RefList Utilities Help
-----
ISRUUAASE          Allocate New Data Set      Invalid combination
Command ==>                                More: +
Data Set Name . . . : ROGERS.PRIVATE.JCL

Management class . . . . . (Blank for default management class)
Storage class . . . . . (Blank for default storage class)
Volume serial . . . . . (Blank for system default volume) **
Device type . . . . . (Generic unit or device address) **
Data class . . . . . (Blank for default data class)
Space units . . . . . CYLS (BLKS, TRKS, CYLS, KB, MB, BYTES
                           or RECORDS)
Average record unit (M, K, or U)
Primary quantity . . . 5 (In above units)
Secondary quantity 1 (In above units)
Directory blocks . . . 10 (Zero for sequential data set) *
Record format . . . . FB
Record length . . . . 80
Block size . . . . . 3120
Data set name type PDS (LIBRARY, HFS, PDS, LARGE, BASIC, *
                        EXTREQ, EXTPREF or blank)
Extended Attributes (NO, OPT or blank)
Expiration date . . . (YY/MM/DD, YYYY/MM/DD
Enter "/" to select option YY.DDD, YYYY.DDD in Julian form
Allocate Multiple Volumes DDDD for retention period in days
or blank)
```

Figure 2-29 Allocate New data set panel

Allocate New Data Set panel

After pressing the Enter key, the Allocate New Data Set panel is displayed. This panel shows the information you have to provide to allocate the new data set.

In this example you are allocating a PDS. This type of data set supports individual members in the data set. Most environments now use DFSMS to control data set allocation; therefore, it is not necessary to specify management class, storage class, or data class information. In most cases the defaults are satisfactory. You must, however, specify the following information:

Space units These are the allocation units for your data set. The allocation units can be in blocks (BLKS), tracks (TRKS), cylinders (CYLS), kilobytes (KB), megabytes (MB), bytes, or records.

Primary quantity This is the number of primary space units that you want to allocate.

Secondary quantity This is the space that can be allocated for secondary extents, if the primary quantity fills up. Non-extended data sets can have a maximum of 16 extents, which include up to five multiple extents that might have been used to satisfy the primary extents.

Important: The exception to the 16-extent limitation is PDSE data sets, which can have up to 123 extents.

Directory blocks	<p>These must be specified for PDS. A <i>directory</i> is an index used to locate members in the partitioned data set. Each directory record consists of 256 bytes containing directory entries. There is one directory entry for each member.</p> <p>The directory is written at the beginning of the primary space. It must fit in the first extent of the data set.</p> <p>For PDS, be sure to request enough directory space to support growth of the data set. You cannot lengthen the directory after the data set is created. If the directory runs out of space, you must re-create the data set.</p>
	<p>Remember: The number of member entries that fit in a directory block is listed here:</p> <ul style="list-style-type: none"> ▶ For a data set with ISPF statistics: six entries per block ▶ For a data set without ISPF statistics: 21 entries per block ▶ For a load module data set: 4 to 7 entries, depending upon attributes
Record format	<p>The format can be any valid combination of the following options:</p> <p>F Fixed-length records V Variable length records U Undefined format records B Blocked records A ASA printer control characters M Machine code printer control characters S Standard (for F) or spanned (for V): sequential data sets only T Track-overflow feature</p> <p>The option we used for a partitioned data set was FB.</p>
Record length	<p>This is the logical record length, in bytes, of the records to be stored in the data set. In the case of a JCL or program library, this value is 80 bytes.</p>
Block size	<p>This is the block size (physical record length), in bytes, of the blocks to be stored in the data set. If records are specified, the block size specifies the average record length.</p>
Data set version	<p>Some types of data sets, such as PDSE, support different versions. z/OS V2R1 introduces PDSE V2. This field can be left blank to use the installation default or specify a 1 for a PDSE V1 or a 2 for a PDSE V2 data set.</p>
Num of generations	<p>A PDSE V2 data set supports member generations. If enabled you can specify a nonzero number to indicate the number of generations to keep for each member. A 0 indicates no generations are kept.</p>
Extended Attributes	<p>Many data sets can be on extended address volumes (EAVs). Specify No to indicate that the allocation is not to be on the extended address space (EAS) on an EAV. Specify OPT to have the system determine whether the data set is to be allocated in EAS space. Leave blank to take the default based on data type.</p>

Data set allocated

After pressing the PF3 (Exit) key, the successful response to the allocation is indicated on the data set Utility panel that reappears. You are returned to this after processing the Allocate New Data Set panel. The upper right of the panel indicates data set allocated.

You have now created a partitioned data set with the name, ROGERS.PRIVATE.JCL This is an empty data set, so the first thing you do is add a member to this data set.

2.28 Edit profile for a user

Profile Name

File Edit Edit_Settings Menu Utilities Compilers Test Help

```
ISREDDE2 ROGERS.PRIVATE JCL(GENER) - 01.01 Columns 00001 00072
Command ==> Scroll ==> PAGE
***** ***** Top of Data *****
=PROF> ....JCL (FIXED - 80)....RECOVERY ON....NUMBER OFF.....
=PROF> ....CAPS ON....HEX OFF....NULLS OFF....TABS OFF.....
=PROF> ....AUTOSAVE OFF PROMPT....AUTONUM OFF....AUTOLIST OFF....STATS ON.....
=PROF> ....PROFILE UNLOCK....IMACRO NONE....PACK OFF....NOTE ON.....
=PROF> ....HILITE OFF CURSOR FIND.....
000001 //ROGERSQ JOB (POK,999),MSGCLASS=T,NOTIFY=ROGERS,CLASS=A
000002 //MAIN SYSTEM=SC65
000003 //PRINTA OUTPUT DEFAULT=YES,INTRAY=2,CLASS=I
000004 //PRINT EXEC PGM=IEBGENER
000005 //SYSPRINT DD SYSOUT=I
000006 //SYSUT2 DD SYSOUT=I
000007 //SYSUT1 DD DISP=SHR,DSN=ROGERS.JCL.VERS5(GENER)
000008 //*APE DD DSN=(ROGERS.A),DISP=(NEW,KEEP),UNIT=3490
000009 //SYSIN DD DUMMY
000010 /*
000011 */
***** ***** Bottom of Data *****
```

Figure 2-30 Edit profile for a user

User edit profile

When you are in edit mode for a data set, the **PROFILE** command displays the information shown in Figure 2-30.

An edit profile for a user controls that user's edit session through modes and temporary lines. For example, these modes and lines convert data to uppercase (caps mode), automatically renumber lines of data (autonum mode), or specify the left and right boundaries used by other commands (=BNDS> line).

Using edit profile types

Various kinds of data can have various edit profiles. For example, you can set up one edit profile for COBOL programs, another edit profile for memos, and a third edit profile for test data. The installation system programmer determines how many separate edit profiles are available to a user. Typically, 25 edit profiles are available.

Edit modes

The edit modes control how your edit session operates. To set these modes, use the associated primary commands. For example, if you are editing a COBOL program that is in uppercase and you want all your input to be converted to uppercase, set caps mode on by entering CAPS ON.

Primary commands

The primary commands listed here summarize the primary commands you use to display and change your edit profile. Notice that many of these command settings are displayed in the user profile shown in Figure 2-30 on page 109:

PROFILE	This displays the current setting of each mode in this list and controls whether changes to these settings are saved.
AUTOLIST	This controls whether a copy of the saved data is automatically stored in the ISPF list data set.
AUTONUM	This controls whether lines of data are automatically renumbered when the data is saved.
AUTOSAVE	This controls whether data is saved when you enter END.
CAPS	This controls whether alphabetic characters are stored in uppercase when the data is saved.
HEX	This controls whether data is displayed in hexadecimal format.
HILITE	This controls the use of enhanced edit color.
IMACRO	This names an edit macro used at the start of the edit session.
NOTES	This controls whether tutorial notes are included in an Edit model.
NULLS	This controls whether blank spaces at the end of a line are written to the panel as blanks or nulls. The difference is that nulls allow you to insert data; blanks do not.
NUMBER	This controls the generation of sequence numbers in a data set.
PACK	This controls whether ISPF packs (compresses) the data when it is saved.
RECOVERY	This controls the recovery of an edit session after a system failure.

Edit primary commands

Primary commands affect the entire data set being edited. Line commands usually affect only a single line or block of lines. To enter a primary command, perform one of the following actions:

- ▶ Type the command on the command line and press Enter.
- ▶ Press the function key to which the command is assigned.

Information: There are many other commands. For a complete description of the commands, see *z/OS Interactive System Productivity Facility (ISPF) Edit and Edit Macros*, SC19-3621.

Modifying an edit profile

You modify an edit profile by entering commands to set various modes, options, and temporary lines. Whenever you change an edit profile value, ISPF saves the value (unless the edit profile is locked). The next time you edit data using the edit profile, the data is retrieved and the environment is set up again. This is easier than it sounds. First, there are defaults for all of the modes, and, in most cases, you do not need to change them. Second, if you decide that you want to change a mode, you simply enter the appropriate command. The edit profile is automatically changed and saved.

Attention: In Figure 2-30 on page 109, the AUTOSAVE primary command was changed from its default by issuing the following command:

```
autosave off prompt
```

2.29 Edit function: Option 2

```
Menu Utilities Compilers Options Status Help
-----
ISR@PRIM          z/OS Primary Option Menu
Option ==> 2

0 Settings      Terminal and user parameters      User ID . : ROGERS
1 View          Display source data or listings    Time. . . : 13:19
2 Edit           Create or change source data     Terminal. . : 3278
3 Utilities     Perform utility functions        Screen. . . : 2
4 Foreground    Interactive language processing   Language. . : ENGLISH
5 Batch          Submit job for language processing Appl ID . : PDF
6 Command        Enter TSO or Workstation commands  TSO logon : IKJACCT
7 Dialog Test   Perform dialog testing           TSO prefix: ROGERS
9 IBM Products  IBM program development products System ID : SCT75
10 SCLM         SW Configuration Library Manager MVS acct. : ACCNT#
11 Workplace    ISPF Object/Action Workplace
12 z/OS System  z/OS system programmer applications
13 z/OS User    z/OS user applications
                                         Release . : ISPF 6.3

Enter X to Terminate using log/list defaults
```

Figure 2-31 Edit: Option 2

Edit function: Option 2

When you select Option 2 from the Primary Option Menu, the edit entry panel is displayed, as shown in Figure 2-32 on page 112. You use that panel to enter the name of an ISPF library, an “other” data set, a z/OS UNIX file, or a workstation file. If more than one data set or file name is specified, ISPF accepts the workstation file name first, then the “other” data set name, and then the ISPF library name.

You can add a new member to a PDS, or modify an existing member, by using ISPF Option 2 (Edit) in the ISPF Primary Option Menu.

2.30 Edit Entry panel

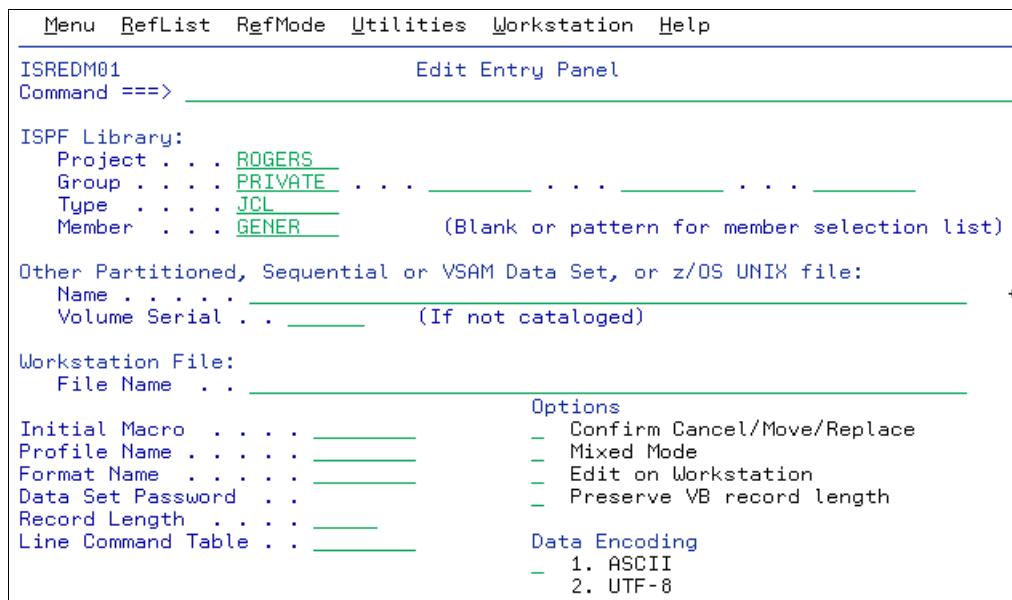


Figure 2-32 Edit Entry panel

Edit Entry panel

In the Edit Entry panel you have to supply the name of the PDS to which you want to add a new member. In this example, the PDS name is ROGERS.PRIVATE.JCL (the data set you allocated previously). If you are creating a new member or want to edit an existing member, you can enter the member name on the panel.

In Figure 2-32, a new member called GENER is entered. After entering the corresponding input fields in this panel, press Enter.

As discussed previously, if your data set has three qualifiers, you can use the Project/Group/Type fields to identify your data set. The advantage is that ISPF stores this information in your profile data set, and the next time that you enter this panel the fields are set by the saved information. If your data set does not have three qualifiers, you must use the Other Partitioned or Sequential data set field, embedding the data set name in single quotes, unless you want ISPF to add your prefix as the HLQ.

Remember: If you do not specify a member name and use the enter key, all members that exist in the Project/Group/Type fields that exist are displayed.

2.31 Editing a data set

The screenshot shows a window titled 'ISREDDE2 ROGERS.PRIVATE.JCL(GENER) - 01.00'. The menu bar includes 'File', 'Edit', 'Edit_Settings', 'Menu', 'Utilities', 'Compilers', 'Test', and 'Help'. Status bars at the top right show 'Columns 00001 00072' and 'Scroll ==> PAGE'. The main area displays JCL code. The first few lines are:

```
ISREDDE2 ROGERS.PRIVATE.JCL(GENER) - 01.00
Command ===>
***** ***** Top of Data *****
000001 //ROGERSQ JOB (POK,999),MSGCLASS=T,NOTIFY=ROGERS,CLASS=A
000002 //*MAIN SYSTEM=SC65
000003 //XXX OUTPUT DEFAULT=YES,INTRAY=2,CLASS=I
000004 //PRINT EXEC PGM=IEBGENER
000005 //SYSPRINT DD SYSOUT=I
000006 //SYSUT2 DD SYSOUT=I
000007 //SYSUT1 DD DISP=SHR,DSN=ROGERS.JCL.VERS5(GENER)
000008 //*APE DD DSN=(ROGERS.A),DISP=(NEW,KEEP),UNIT=3490
000009 //SYSIN DD DUMMY
000010 /*
000011 //
.....
.....
```

At the bottom, it says '***** Bottom of Data *****'

Figure 2-33 Editing a data set

Editing a data set

Because the member name GENER is a new member, the window displayed in Figure 2-33 will have no entries. On the first line, you might see the following first lines:

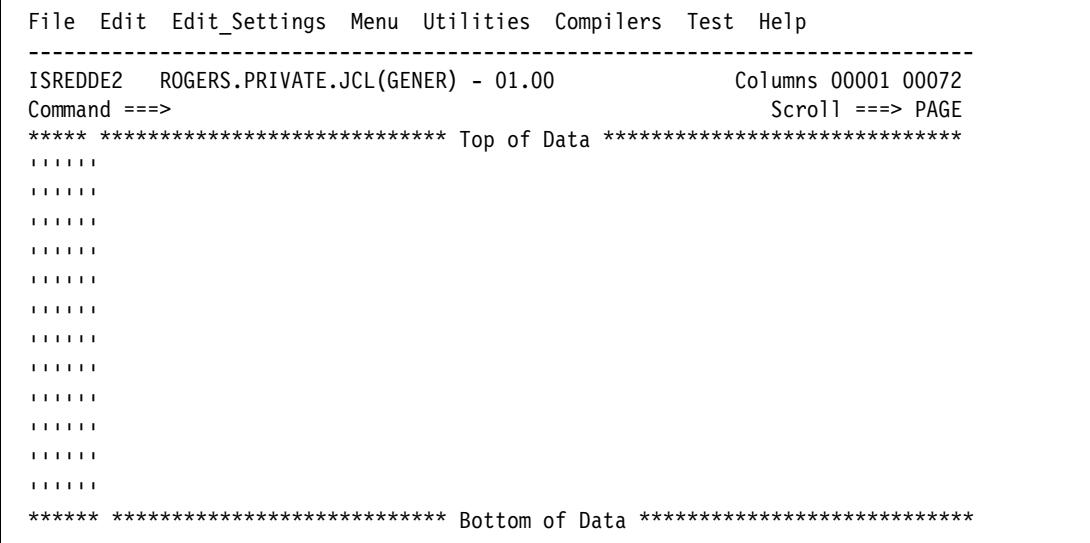
```
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
```

On the command line, enter the command **RESET** to remove those two lines. Now, you will have no lines to create your file. On the top line, to create lines, you can enter **i12**, which will insert 20 lines showing seven commas (,,,...,) in the non-text portion of the panel; see Figure 2-34 on page 114.

Important: The **RESET** primary command can restore line numbers in the line command field when those line numbers have been replaced by labels, pending line commands, error flags, and change flags. **RESET** can also delete special lines from the display, re-display excluded lines and excluded lines messages, and temporarily disable the highlighting of FIND strings.

Type the records that you want to create in the new member. In Figure 2-33, there are coded JCL statements filling the first 11 lines of the 12 lines that were inserted by the **i12** command and you can see the remaining initial lines that were created by the **i20** command.

Figure 2-34 shows inserted lines.



The screenshot shows the ISPF Edit panel interface. The menu bar includes File, Edit, Edit_Settings, Menu, Utilities, Compilers, Test, and Help. Below the menu is a dashed line separator. The title bar displays 'ISREDDE2 ROGERS.PRIVATE.JCL(GENER) - 01.00' and 'Columns 00001 00072'. The command line shows 'Command ==>' and 'Scroll ==> PAGE'. The main area contains 20 lines of data, each consisting of a six-column Line Command field followed by a 72-column data field. The data is represented by a series of vertical dots ('....') for the command field and a series of asterisks ('*****') for the data field. The top of the data is marked with '***** * Top of Data *****' and the bottom with '***** * Bottom of Data *****'.

Figure 2-34 Several of the 20 lines inserted

Using the Edit panel

You can use the Edit function to create, display, or change data stored in the partitioned data set member or sequential data sets. When the editor displays existing data, each line consists of a six-column Line Command field followed by a 72-column data field. To view data that is not displayed, use the scroll commands. The following list shows the PDF default values for function keys:

- ▶ F7/19 Scrolls up
- ▶ F8/20 Scrolls down
- ▶ F10/22 Scrolls left
- ▶ F11/23 Scrolls right

Using the PFK keys

To see the function key values, type **KEYS** in the Command line and press Enter. You can issue line commands and primary commands in edit mode.

Information: The ISPF line commands are described in Figure 2-35 on page 115.

The **i** command was used in the text described here.

2.32 ISPF edit: Various line commands

Command	Description
I	Insert lines
D	Delete lines
R	Repeat lines
C	Copy lines
M	Move lines
A	After line
B	Before line
(Shift right columns
<	Shift right data
)	Shift left columns
>	Shift left data
X	Exclude lines

Figure 2-35 ISPF edit: various line commands

ISPF edit line commands

Line commands affect only a single line or block of lines. You enter line commands by typing them in the line command field on one or more lines and pressing Enter. The line command field is usually represented by a column of 6-digit numbers on the far left side of your display. When you are editing an empty data set or member, however, the line command field contains quotation marks. This field can also be used to define labels and to display flags that indicate special lines, such as the =NOTE= flag, which indicates a note line.

With line commands, you can perform the following tasks:

- ▶ Insert or delete lines
- ▶ Repeat lines
- ▶ Rearrange lines or overlay portions of lines
- ▶ Simplify text entry and formatting
- ▶ Define an input mask
- ▶ Shift data
- ▶ Include or exclude lines from the display
- ▶ Control tabs and boundaries for editing
- ▶ Convert various types of special temporary lines to data lines

Figure 2-35 shows various line commands you can use. Press Help again and a long message appears.

Rules for entering line commands

Enter a line command by performing one of these actions:

- ▶ Typing the command in the line command field and pressing Enter.
- ▶ Placing the cursor in the data or line command field and pressing a function key to which the command is assigned.

These rules apply to all line commands:

- ▶ You can type several line commands and make multiple data changes before you press Enter.
- ▶ The editor displays an error message if the line command is ambiguous. Because the line commands are processed from top to bottom, it is possible to have one error message appear that masks a later error condition.
- ▶ Only the first error condition found is displayed. After you have corrected that error condition, processing can continue and the next error condition, if any, is displayed. If you type a line command incorrectly, you can replace it before you press Enter by retyping it, blanking it out, or entering RESET.

2.33 ISPF edit: Repeating and deleting lines

```
Command ==>                                         Scroll ==> PAGE
***** ***** Top of Data *****
000001 LINE 1
d00002 LINE 2
***** ***** Bottom of Data *****
Results in:

Command ==>                                         Scroll ==> PAGE
***** ***** Top of Data *****
000001 LINE 1
***** ***** Bottom of Data *****
Issuing a R3 command - Results in:

ISREDDE2    ROGERS.PRIVATE.JCL(NEW) - 01.00          Columns 00001 00072
Command ==>                                         Scroll ==> PAGE
***** ***** Top of Data *****
r30001 LINE 1
***** ***** Bottom of Data *****
Results in:

Command ==>                                         Scroll ==> PAGE
***** ***** Top of Data *****
000001 LINE 1
000002 LINE 1
000003 LINE 1
000004 LINE 1
***** ***** Bottom of Data *****
```

Figure 2-36 ISPF edit: Repeating and deleting lines

ISPF edit: repeating and deleting lines

You can delete lines by issuing the line command **D**, which will delete one line. To delete multiple lines, issue **Dxx**, where *xx* is the number of lines to delete. You can also delete a block of lines by using **DD** at the beginning of the block and at the end of the block. In Figure 2-36 we deleted one line by entering **D** in the corresponding command line.

Repeating lines can be done by issuing **R** (one line), **Rxx** for *xx* lines, or **RR** for a block. Figure 2-36 shows the result of the execution of an **R5** command: The line is repeated five times.

2.34 ISPF edit: Copying lines

The screenshot shows three examples of copying lines in ISPF edit mode:

- Example 1:** Shows the command `C` being used to copy lines 00001 and 00002. The copied lines are then pasted at the bottom of the data set.
- Example 2:** Shows the command `CC` being used to copy lines 00001 through 00003. The copied lines are then pasted at the bottom of the data set.
- Example 3:** Shows the command `CC` being used to copy lines 00001 through 00005. The copied lines are then pasted at the bottom of the data set.

Each example includes a "Results in:" section showing the final state of the data set after the copy operation.

Figure 2-37 Edit: Copying line

Edit: Copying lines

Copying lines can be done by issuing **C** to copy one line; **Cxx**, to copy xx lines; or **CC** to copy a block, with **CC** marking the first line and the last lines of the block to be copied. After indicating the lines, you must tell the Editor where to copy the lines to. This is done by typing either a **B** (for before) or an **A** (for after) on the line following or preceding where you want the data to be copied.

Copying one or more lines

To copy one or more lines in the same data set or member, perform these tasks:

1. Type **C** in the line command field of the line to be copied. If you also want to copy one or more lines that immediately follow this line, type a number greater than 1 after the **C** command.
2. Next, specify the destination of the line to be copied by using the **A** (after), the **B** (before), or the **O** (overlay) line command.
3. Press Enter, and the line (or lines) are copied to the new location.

Copying blocks of lines

To copy a block of lines in the same data set or member, perform these steps:

1. Type **CC** in the line command field of both the first and last lines to be copied. You can scroll (or use **FIND** or **LOCATE**) between typing the first **CC** and the second **CC**, if necessary.

2. Use the **A** (after), the **B** (before), or the **OO** (overlay) command to show where the copied lines are to be placed.

When you use the block form of the **C** command (**CC**) to copy and overlay lines, also use the block form of the **O** command (**OO**).

3. Press Enter, and the lines that contain the two **CC** commands and all of the lines between them are copied to the new location.

Copying lines to another data set or member

You can copy lines from another data set or member into the data set you are currently working with by performing the following tasks:

1. Type either CREATE or REPLACE on the command line.
2. Use one of the forms of the **C** command described previously.
3. Press Enter.
4. On the next panel that PDF displays, type the name of the data set or member that you want to create or replace.
5. Press Enter, and the lines are copied to the data set or member that you specified.

2.35 ISPF/PDF edit: Primary commands

```
ISR2M000 ----- Edit Primary Commands ----- Tutorial
Option ==>

The following topics are presented in sequence, or may be selected by number.
Individual commands may be selected by entering the command name.

0 - Primary commands (general information)
1 - Miscellaneous commands
2 - FIND/CHANGE/EXCLUDE commands
3 - Number control commands
4 - Display mode control commands
5 - Termination control commands
6 - External data commands
7 - Macro control commands
8 - Data editing commands
9 - Edit settings command

Note: Select >CANCEL to see information for the CANCEL command.
```

Figure 2-38 ISPF/PDF edit: Primary commands tutorial

ISPF/PDF edit: Primary commands

Primary commands affect the entire data set. You type them on the Command ==> field. You use a primary command to perform the following tasks:

- ▶ Control your editing environment
- ▶ Find a specific line
- ▶ Find and change a character string
- ▶ Combine several members into one
- ▶ Split a member into two or more members
- ▶ Submit data to the job stream
- ▶ Save the edited data or cancel without saving
- ▶ Sort data
- ▶ Delete lines
- ▶ Access dialog element models
- ▶ Run an edit macro

An easy way to access a tutorial about all Edit primary commands is:

1. Type P in the Command line and press Enter. You receive the message COMMAND P NOT FOUND. Press Enter; you receive a short message at the upper right of the panel.
2. Press the Help function key; now you receive a long message.
3. Press the Help function key again; now you are in the tutorial of all Edit primary commands showing how each command works, and you receive the panel displayed in Figure 2-38.

2.36 ISPF/PDF edit: Saving new or updated files

- Saving your file**
 - Enter **SAVE** on the command line
 - Use PF3 to save the file with profile option **AUTOSAVE ON**
- Canceling updates to a file**
 - Enter **CANCEL** on the command line
- While in edit mode, issue the PROFILE command:**
 - Check the **AUTOSAVE** option

```
=PROF> ....JCL (FIXED - 80)....RECOVERY ON....NUMBER OFF.....  
=PROF> ....CAPS ON....HEX OFF....NULLS OFF....TABS OFF.....  
=PROF> ....AUTOSAVE OFF PROMPT....AUTONUM OFF....AUTOLIST OFF....STATS ON.....  
=PROF> ....PROFILE UNLOCK....IMACRO NONE....PACK OFF....NOTE ON.....  
=PROF> ....HILITE OFF CURSOR FIND.....
```

Figure 2-39 ISPF/PDF: Save command

ISPF/PDF edit: Saving new or updated files

While you are editing a file, saves of the file can be performed without closing the file. The **SAVE** command saves edited data without ending your edit session. Generally, you do not need to use **SAVE** if recovery mode is on.

A **SAVE** command writes the data to the same data set from which it was retrieved unless you specified a concatenated sequence of partitioned data sets on the Edit Entry panel. In that case, the data is saved in the first library in the concatenation sequence, regardless of which library it came from. For a sequential data set, the complete data set is rewritten. For a partitioned data set, the member is rewritten with the same member name. If stats mode is on, the library statistics for the member are automatically updated.

Updating a file

To save your data, issue the **SAVE** primary command. With the profile option **AUTOSAVE ON**, pressing the F3/F15 END key ends the edition and saves the data.

Information: See 2.28, “Edit profile for a user” on page 109 to determine the settings for the **AUTOSAVE** command.

You can cancel your updates by issuing the **CANCEL** primary command. This action removes all changes made since the last **SAVE** was performed, or since the data set was edited.

2.37 ISPF Data Set List Utility option

```
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:
Ddbname Level . . . ROGERS
Volume serial . .

Data set list options
Initial View           Enter "/" to select option
2 1. Volume           / Confirm Data Set Delete
2. Space              / Confirm Member Delete
3. Attrib             / Include Additional Qualifiers
4. Total              / Display Catalog Name
                      Display Total Tracks
                      Prefix Ddbname Level

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 2-40 ISPF option 3.4 panel

ISPF Data Set List Utility option 3.4

When you select the ISPF option 3.4 from the primary panel, the Data Set List Utility panel (Figure 2-40) is displayed. You can either display or print lists of ISPF libraries, data sets, or volume table of contents (VTOC) information.

The Data Set List Utility is a useful option in your daily job. From the Menu, to use this list utility to manage all data sets that you have access to, as shown in Figure 2-41. Move the cursor to the Ddbname Level field and enter the high-level qualifier of the data sets that you want to work with. First, take a look in the options available:

- ▶ You can obtain a list of all data sets in a DASD volume. To do this, use option V and enter a volume name.
- ▶ You can obtain a data set list.
- ▶ You can customize what kind of data set information you want.

```
Menu Options View Utilities Compilers Help
-----
ISRUDSLO data sets Matching ROGERS          Row 18 of 42
Command ==>                                Scroll ==> PAGE

Command - Enter "/" to select action
-----                                     Tracks %Used XT
                                         75   1   1
-----                                     ROGERS.PRIVATE.JCL
```

Figure 2-41 Option 3.4 list of data sets for HLQ ROGERS

Dsname Level

This field is used to specify the level (or levels) of any data set that you want ISPF to list or print for you. An optional installation exit, called the data set list (DSLIST) exit, can control whether a data set name appears in the list. See *z/OS ISPF Planning and Customizing*, GC19-3623, for more information about this exit.

When you specify the Dsname Level, you are defining the level qualifiers for the data set names to be included in the list.

An ISPF library typically has a three-level name:

project, group, and type

The Dsname Level field supports the inclusion of system symbols. ISPF retains the information you put in this field and displays it the next time you use this panel.

Using a data set name list

For a data set name list, the data set name qualifiers can be partially specified using asterisks (*) as global file-name characters and percent signs (%) as placeholders:

- * A single asterisk indicates that at least one qualifier is needed to occupy that position. A single asterisk in a qualifier indicates that zero or more characters can occupy that position.
- ** A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk in a qualifier is invalid.
- % A single percent sign indicates that any one character can occupy that position. One to eight percent signs can be specified in each qualifier.

For example, entering ABC.DDD%*.E%%%F.** lists all data sets with ABC as HLQ, second qualifier starts with DDD, the third starts with E and ends with F and has at least three characters between E and F. After that it can have any number of qualifiers. In Figure 2-40 on page 122 we used the ROGERS HLQ.

2.38 Working with a data set list

Menu Options View Utilities Compilers Help					
ISRUDSL0 Data Sets Matching ROGERS				Row 18 of 42	
Command ==>				Scroll ==> PAGE	
Command - Enter "/" to select action				Tracks	%Used
E	ROGERS.PRIVATE.JCL			75	1
	ROGERS.RMFZOS18.ADMGDF			5	20
	ROGERS.SC75.ISPF42.ISPPROF			6	100
	ROGERS.SC75.SPFLIST.LIST			9	0
	ROGERS.SPFLIST.LIST			9	11
	ROGERS.SRCHFOR.LIST			10	10
	ROGERS.STDPARM			1	100
	ROGERS.WLM.SAVE01			5	40
	ROGERS.ZFS				
	ROGERS.ZFS.DATA			75	?
	ROGERS.ZFS.INSTALL.TXT			1	100
	ROGERS.ZFS.JOB.CNTL			6	66
	ROGERS.ZFS.JOB.CNTL.UNLOAD.BIN			3	100
	ROGERS.ZFS.MIGRTOOL			13	84
	ROGERS.ZFS.MIGRTOOL.LOADLIB			16	100

Figure 2-42 Data set list for ROGERS

Working with a data set list

Working with a data set list shows all cataloged data sets having the high-level qualifier ROGERS. You can select for any of the data sets for processing by entering by placing a forward slashmark (/) at the left of the data set name shown in Figure 2-42. This displays a menu on the window, as shown in Figure 2-44 on page 125. As you can see, this is a comprehensive list of action characters to enable you to manage your data sets.

If you place a slash (/) next to data set ROGERS.PRIVATE.JCL in Figure 2-42, the list of action characters shown in Figure 2-44 on page 125 enables you to access that data set by selecting one of the action codes by its number.

If you select the number 1, Figure 2-43 is displayed showing the members of the data set, which is the equivalent of an E, as shown in Figure 2-42. Using the slash (/) enables you to see all of the possible actions.

Menu functions Confirm Utilities Help					
ISRUDSM EDIT ROGERS.PRIVATE.JCL				Row 00001 of 00002	
Command ==>				Scroll ==> PAGE	
Name	Prompt	Size	Created	Changed	ID
GENER		11	2011/03/23	2011/03/24 09:29:21	ROGERS
NEW		5	2011/03/24	2011/03/24 09:07:13	ROGERS
End					

Figure 2-43 Members in data set ROGERS.PRIVATE.JCL

As mentioned, the menu shown in Figure 2-44 provides a list of action characters that you can use to manage your data sets.

```
ISRUDABC          data set List Actions
More:      +
Data set: ROGERS.PRIVATE.JCL

DSLISIT Action
1. Edit           15. Reset
2. View           16. Move
3. Browse          17. Copy
4. Member List     18. Refadd
5. Delete          19. Exclude
6. Rename          20. Unexclude 'NX'
7. Info            21. Unexclude first 'NXF'
8. Short Info      22. Unexclude last 'NXL'
9. Print            23. SuperC 'SC'
10. Catalog         24. SuperCE 'SCE'
11. Uncatalog        25. Search-For 'SF'
12. Compress         26. Search-ForE 'SFE'
13. Free             27. Allocate
14. Print Index
```

Figure 2-44 Data set List Actions

Typing E next to data set ROGERS.PRIVATE.JCL displays a list of the members contained in that data set. After the member list is displayed, you can select a member by typing S in front of the member name. Also, you can perform many actions against the members, such as copy, rename, delete, and so forth. To see all actions available in the member list, follow these steps:

1. Type a question mark (?) at the left of any member name. You receive an error message.
 2. Press the Help function key; you receive a long error message.
 3. Press the Help function key again; the tutorial for the commands available is displayed.
- Note that in the upper right of the panel the following information appears:

More: +

It indicates a scrollable panel; you can go forward and backward throughout this panel using PF7 and PF8 (default keys).

2.39 Job control language

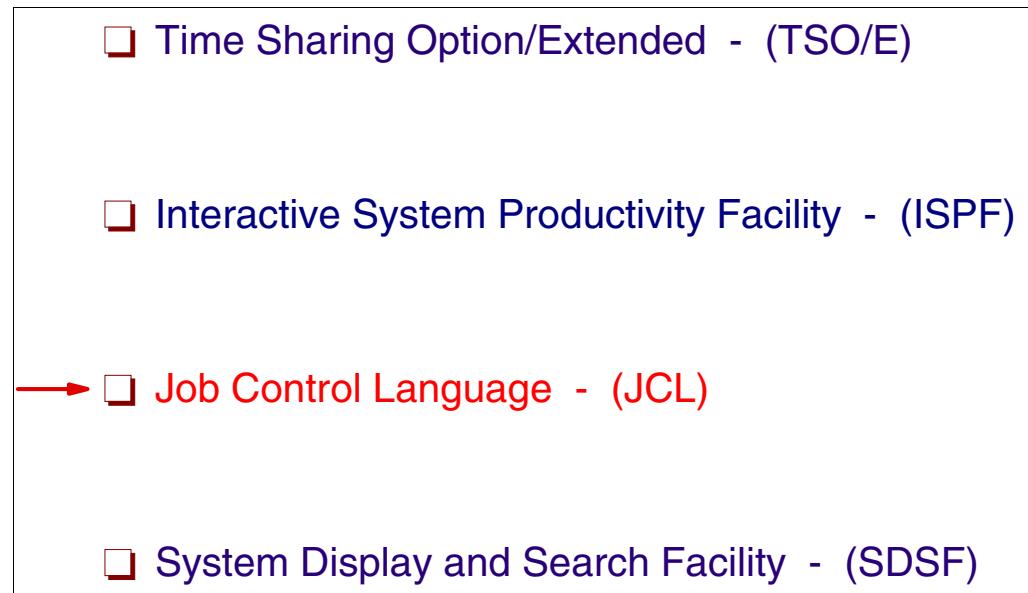


Figure 2-45 Job control language

Job control language

For your program to run on the computer and perform the work you designed it to do, your program must be processed by your operating system.

Your operating system consists of a z/OS Base Control Program (BCP) with a job entry subsystem (JES2 or JES3) and DFSMSdfp installed with it.

For the operating system to process a program, programmers must perform certain job control tasks. These tasks are performed through the job control statements:

- ▶ JCL statements
- ▶ JES2 control statements
- ▶ JES3 control statements

The JES2 and JES3 statements are called Job Entry Control Language (JECL) statements.

2.40 JCL introduction

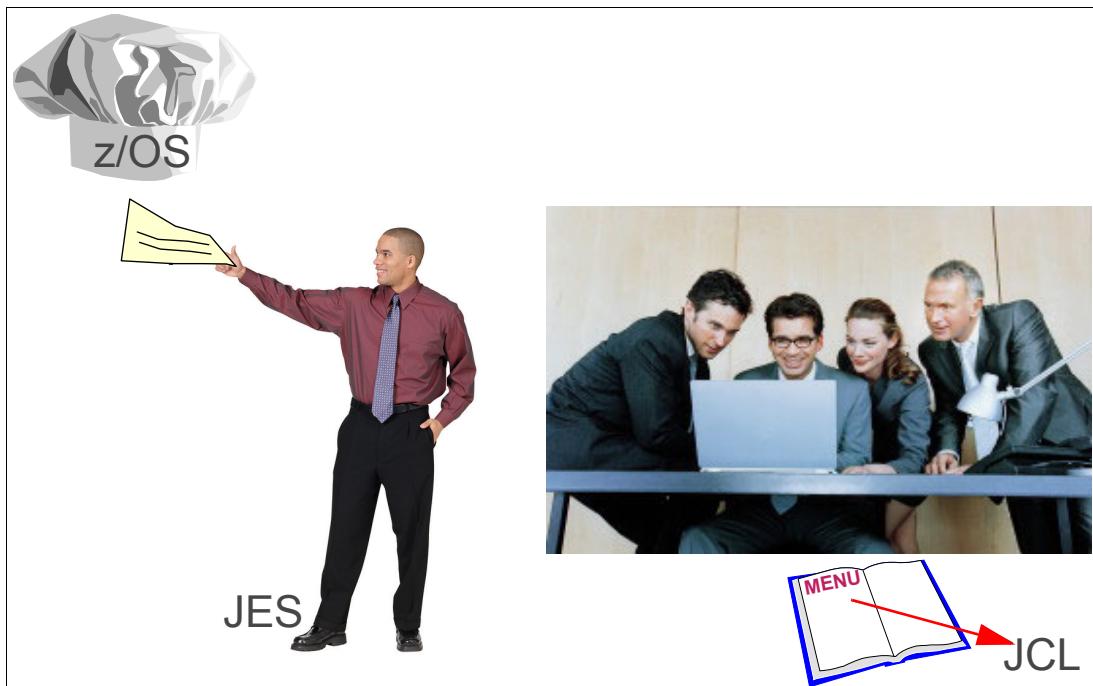


Figure 2-46 Job Control Language

JCL introduction

To get your MVS system to accomplish work for you, you must describe to the system the work you want done and the resources your work needs. You use JCL to provide this information to MVS.

One way of thinking about JCL is to compare it to a menu in a restaurant. If you are a customer at a restaurant, you and the other customers do not simply walk into the kitchen and start cooking your own dinners; that defeats the purpose of going to a restaurant. Instead, from a menu describing all that the restaurant has to offer, you select items to make up an order, specifying which entrées you want, which salad dressing you prefer, and any other special requests you have. You then ask the waiter to take your order to the kitchen.

In the kitchen, a team of chefs divides up the work and the appropriate ingredients to prepare each dish as quickly and efficiently as possible. While the meals are being prepared, you and your friends can ignore what is going on in the kitchen, engaging instead in dinner conversation and catching up on the latest news. When the waiter brings out your meal, you concentrate on your enjoyment of the meal.

Now imagine yourself back at the office using your MVS system, and think of JCL as the menu. In the same way that you and the other diners select items from the menu and place orders for the waiter to take to the team of chefs, you and other MVS users use JCL to define work requests (called jobs), and use a JES to submit those jobs to MVS.

Using the information that you and the other users provide with JCL statements, MVS allocates the resources needed to complete all of your jobs just as the kitchen chefs divided up the work to prepare the orders of all the customers.

And just as the chefs worked in the kitchen while you and the other diners devoted your attention to what was going on at your tables, MVS completes the submitted jobs in the background of the system, enabling you and the other users to continue working on other activities in the foreground. Also, just as the waiter conveys the results of the chefs' work to you, JES presents the output of the jobs to you.

For a complete description of the process, see *z/OS MVS JCL User's Guide*, SA23-1386.

2.41 JCL-related actions

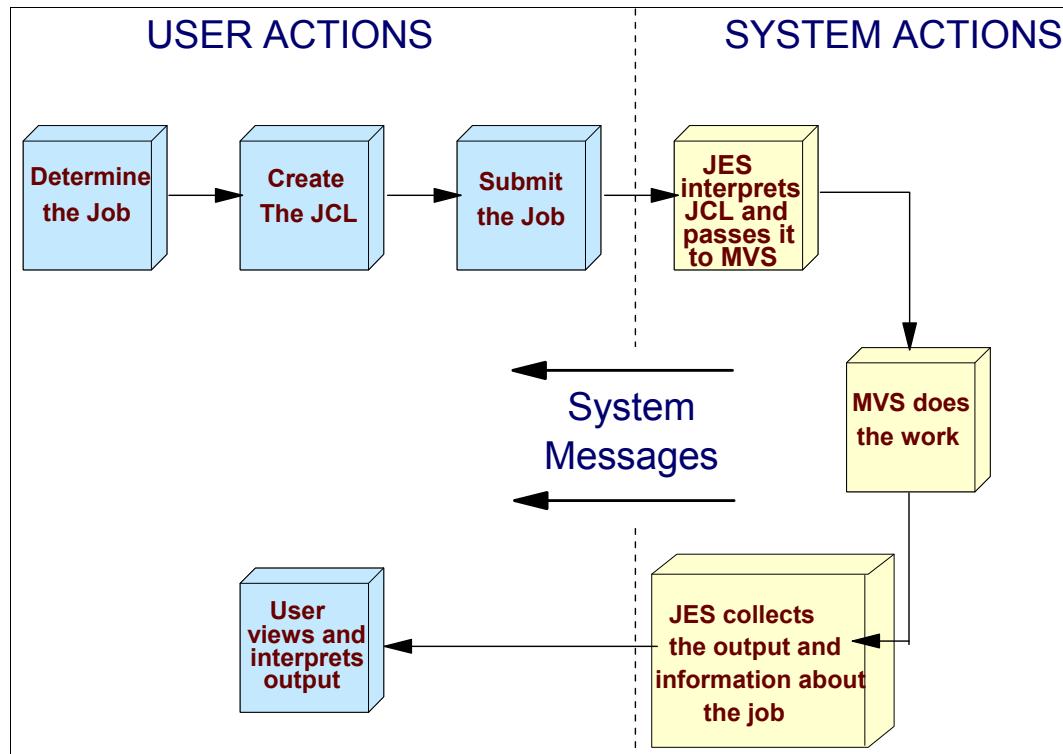


Figure 2-47 JCL-related actions

JCL-related actions

Figure 2-47 shows an overview of the job submission process. The user performs the activities on the left side of the figure, and the system performs those on the right. In this example, MVS and JES make up *the system*.

For every job that you submit, you need to tell MVS where to find the appropriate input, how to process that input (that is, what program or programs to run), and what to do with the resulting output.

JCL statements

You use JCL to convey this information to MVS through a set of statements known as *job control statements*. JCL's set of job control statements is quite large, enabling you to provide a great deal of information to MVS.

Most jobs, however, can be run using a small subset of these control statements. After you become familiar with the characteristics of the jobs you typically run, you might find that you need to know the details of only some of the control statements.

In each job, the control statements are grouped into job steps. A job step consists of all the control statements needed to run one process, for example a sort, a copy, or an application program. If a job needs to run more than one process, the job will contain another job step for each of those programs. A job can have from one up to 255 steps.

2.42 Required control statements

□ Control statements

- Job (JOB)
- Execute (EXEC)
- Data Definition (DD)

```
//ROGERSQ JOB    (POK,999),MSGCLASS=T,NOTIFY=ROGERS,CLASS=A  
//PRINT      EXEC PGM=IEBGENER  
//SYSPRINT   DD SYSOUT=I  
//SYSUT2     DD SYSOUT=I  
//SYSUT1     DD DISP=SHR,DSN=ROGERS.JCL.VERS5(GENER)  
//TAPE        DD DSN=(ROGERS.A),DISP=(NEW,KEEP),UNIT=3490  
//SYSIN      DD DUMMY  
/*
```

Figure 2-48 Required control statements

Required control statements

Every job must contain, at minimum, the following two types of control statements:

- ▶ A JOB statement

This statement marks the beginning of a job and assigns a name to the job. The JOB statement is also used to provide certain administrative information, including security, accounting, and identification information. Every job has one and only one JOB statement.

- ▶ An EXEC (run) statement

This statement marks the beginning of a job step, to assign a name to the step and identify the program or procedure to be run in the step. You can add various parameters to the EXEC statement to customize the way the program runs. Every job has at least one EXEC statement.

In addition to the JOB and EXEC statements, most jobs also contain:

- ▶ One or more data definition (DD) statements

These statements identify and describe the input and output data to be used in the step. The DD statement can be used to request a previously created data set, to define a new data set, to define a temporary data set, or to define and specify the characteristics of the output.

2.43 JCL streams and jobs

- ❑ Programmers create jobs
 - Using JCL statements
- ❑ Jobs are submitted to and processed by:
 - JES2 or JES3
 - z/OS operating system
- ❑ Jobs have one or more job steps

Figure 2-49 JCL streams and jobs

JCL streams and jobs

For the operating system to process a program, system programmers or application programmers must perform certain job control tasks. These tasks are performed through the job control statements:

- ▶ JCL statements, as mentioned before
- ▶ JES2 control statements *or* JES3 control statements

Job Entry Subsystems

MVS uses a JES to receive jobs into the operating system, schedule them for processing by MVS, and control their output processing. JES2 is descended from Houston Automatic Spooling Program (HASP). HASP is defined as a computer program that provides supplementary job management, data management, and task management functions, such as scheduling, control of job flow, and spooling. HASP remains in JES2 as the prefix of most module names and the prefix of all messages sent by JES2 to the operator.

Job Entry Subsystem 2

JES2 is a functional extension of the HASP II program that receives jobs into the system and processes all output data produced by the job. So, JES2 is that component of MVS that provides the necessary functions to get jobs into, and output out of, the MVS system. It is designed to provide efficient spooling, scheduling, and management facilities for the MVS operating system.

However, none of this explains why MVS needs a JES. Basically, by separating job processing into several tasks, MVS operates more efficiently. At any point in time, the computer system resources are busy processing the tasks for individual jobs, while other tasks are waiting for those resources to become available.

In its most simple view, MVS divides the management of jobs and resources between the JES and the base control program of MVS. In this manner, JES2 manages jobs before and after running the program; the BCP manages them during processing.

JES2 compared to JES3

IBM provides two JESs from which to choose: JES2 and JES3. In an installation that has only one processor (computer), JES2 and JES3 perform similar functions. That is, they read jobs into the system, convert them to internal machine-readable form, select them for processing, process their output, and purge them from the system. However, for an installation that has more than one processor in a configuration, there are noticeable differences in how JES2 exercises independent control over its job processing functions.

That is, in the configuration, each JES2 processor controls its own job input, job scheduling, and job output processing. In a sysplex environment, it is possible to configure JES2 to share spool and checkpoint data sets with other JES2 systems in the same sysplex. This configuration is called Multi-Access Spool (MAS).

Job Entry Subsystem 3

In contrast, JES3 exercises centralized control over its processing functions through a single global JES3 processor. This global processor provides all job selection, scheduling, and device allocation functions for all the other JES3 systems. The centralized control that JES3 exercises provides increased job scheduling control, deadline scheduling capabilities, and increased control by providing its own device allocation.

2.44 JES control statements in JCL

- ❑ Controls the input and output processing of jobs
 - JES2
 - /*JOBPARM
 - /*MESSAGE
 - /*ROUTE
 - /*XEQ
 - etc...
 - JES3
 - ///*OPERATOR
 - ///*ROUTE XEQ
 - ///*PROCESS
 - ///*MAIN
 - etc ...

Figure 2-50 Coding JES2 and JES3 control statements in a JCL

JES control statements in JCL

Code JES control statements after JOB card JCL statement. The exception is for ///*PAUSE, JES3 appears before the JOB statement or between its continuation. JES2 and JES3 ignore control statements in a procedure.

The rules for coding JES2 control statements are the same as the rules for JCL statements, with the following additions:

- ▶ Columns 1 and 2 always contain the characters slash asterisk (/*).
- ▶ Where comments are needed, code a JCL comment statement, which consists of two slashes and an asterisk (//*).
- ▶ If you code the same parameter on the same statement more than one time, JES2 uses the value in the last parameter.

The rules for coding JES3 control statements are the same as the rules for JCL statements, with the following additions:

- ▶ Generally, the characters two slashes and an asterisk (//*) are in columns 1 to 3. Various JES3 control statements *must* contain only a single slash and asterisk (/*) in columns 1 and 2.
- ▶ Columns 3 and 4 must not be blank.
- ▶ To code a comment on a JES3 control statement, code a blank after the control statement, and end the comment before column 72.

2.45 Introduction to JCL: Creating a data set

Coding rules

Creating a data set using JCL

- Data sets are defined as new in statements MYDS and MDS1

```
//ROGERS1 JOB (MVS,POK),CLASS=A,MSGCLASS=H,  
//                  MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID  
//ALLOC EXEC PGM=IEFBR14  
//MYDS   DD  DSNAME=ROGERS.A.B,DISP=(NEW,KEEP,DELETE),  
//          UNIT=3390,VOLUME=SER=123456,SPACE=(CYL,(10,5))  
//MYDS1  DD  DSNAME=ROGERS.B.C,DISP=(NEW,CATLG,DELETE),  
//          UNIT=SYSDA,VOLUME=SER=654321,SPACE=(TRK,(20,5))  
//
```

Figure 2-51 Creating jobs: An introduction to JCL

Introduction to JCL: Creating a data set

First, you need to know the basic JCL coding rules. The JCL statements follow these conventions:

- ▶ Start in column 1, and are identified by two slashes (//) at the beginning of the line.
- ▶ The commentary lines are identified by two slashes and an asterisk (/*) at the start of the line.
- ▶ The statements are coded from column 1 to column 71.
- ▶ A comma (,) indicates that the statement has continuation.
- ▶ A continuation of a statement *must* start between columns 4 and 16.
- ▶ Two slashes (//) and the rest of a statement with blanks indicates the end of the job.
- ▶ Key words are in uppercase.

Comments must be separated from the operators parameter by at least one blank.

Creating a data set using JCL

In 2.24, “Allocating data sets: Utility option” on page 102, you learned how to allocate a data set using ISPF dialogs. Next, you see how you can create a data set using JCL control statements. In Figure 2-51, the new data set being created is coded as follows:

```
//ROGERS1 JOB (MVS,POK),CLASS=A,MSGCLASS=H,  
//                  MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID  
//ALLOC EXEC PGM=IEFBR14  
//MYDS   DD  DSNAME=ROGERS.A.B,DISP=(NEW,KEEP,DELETE),  
//          UNIT=3390,VOLUME=SER=123456,SPACE=(CYL,(10,5))  
//MYDS1  DD  DSNAME=ROGERS.B.C,DISP=(NEW,CATLG,DELETE),  
//          UNIT=SYSDA,VOLUME=SER=654321,SPACE=(TRK,(20,5))
```

/* Required job statements

Every job must contain a minimum of the following two types of control statements:

JOB statement

This statement marks the beginning of a job and assigns a name to the job. The JOB statement is also used to provide certain administrative information, including security, accounting, and identification information. Every job has one and only one JOB statement.

EXEC (run) statement

This statement marks the beginning of a job step, to assign a name to the step, and to identify the program or procedure to be run in the step. You can add various parameters to the EXEC statement to customize the way the program runs. Every job has at least one EXEC statement.

In addition to the JOB and EXEC statements, most jobs usually also contain the following statements:

DD statements

A job can contain one or more DD (data definition) statements, to identify and describe the input and output data to be used in the step. The DD statement can be used to request a previously-created data set, to define a new data set, to define a temporary data set, or to define and specify the characteristics of the output.

Job JCL statements

In addition to submitting jobs to a system, you can use JCL statements to create, delete, catalog, or uncatalog data sets. Figure 2-51 on page 134 shows the JCL that you use to create a data set and to review various fundamental JCL statements:

//ROGERS1 JOB The job statement marks the beginning of a job and assigns a name to the job (ROGERS1).

//ALLOC EXEC The EXEC statement marks the beginning of a job step; assigns a name to the step; identifies the program or the catalogued or in-stream procedure to be run in this step.

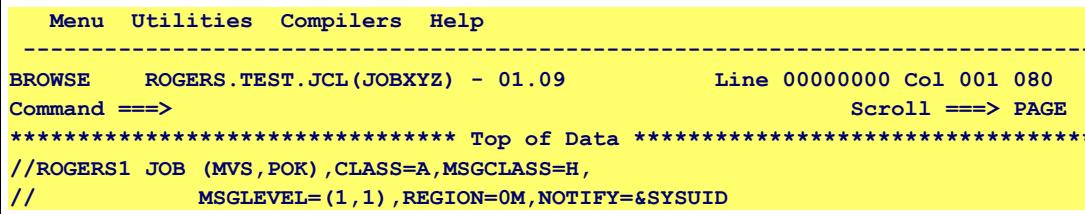
//MYDS DD The DD statement identifies and describes a data set.

//MYDS1 DD The DD statement identifies and describes a data set.

For more details, see *z/OS MVS JCL Reference*, SA23-1385.

2.46 JCL: JOB statement

- Create a member using ISPF edit
 - The member name is ROGERS1
- Create JCL statements
 - JOB card statement
 - EXEC statement
 - DD statement



```
Menu Utilities Compilers Help
-----
BROWSE    ROGERS.TEST.JCL (JOBXYZ) - 01.09      Line 00000000 Col 001 080
Command ===>                               Scroll ===> PAGE
***** Top of Data *****
//ROGERS1 JOB (MVS,POK),CLASS=A,MSGCLASS=H,
//                         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
```

Figure 2-52 The JOB card

JCL: JOB statement

Create a member to insert your JCL control statements. You already know how to create a member in a partitioned data set (see 2.30, “Edit Entry panel” on page 112 to review). The first statement you code is the JOB statement, also known as the JOB card. The JOB statement marks the beginning of a job and tells the system how to process the job.

In this exercise of creating a job using JCL, we used the JCL shown in the highlighted box in Figure 2-52. The JOB card and its parameters are highlighted by the inner box (lines 000001 and 000002).

Remember: A JOB statement has two kinds of parameters: positional and keyword. Parameters are optional; however, your installation might require the accounting information parameter and the programmer's name parameter.

The first line of the JOBXYZ member is the job statement. It specifies parameters to be used by the job entry subsystem to schedule this job for processing. The format of the job card, and the importance of the data specified in the job card vary from installation to installation. The following fields are important:

Jobname	The first field is the jobname, in this case ROGERS1. It can have up to eight characters. Various sites perform security checking against the job name to ensure standards, usually the ID of the user who submitted the job. Let us suppose the standard is: User ID suffixed with at least one alphanumeric character. If ROGERS is the user ID, ROGERS1 matches the standards. Other sites might not have any job name restrictions.
JOB	This field identifies the job to the system, when submitted. It must be present, must follow the job name, and there must be at least one space between them.

ACCOUNT	Various sites use this field for accounting and job processing information. In the example, the value is 19.
Programmer name	The installation specifies if a programmer's-name parameter is required on the JOB statement. The installation decides what the parameter must contain.
NOTIFY	This field tells the system where to send "job complete" information. &SYSUID tells the system to automatically insert your user ID here, so the information will be sent to you.
MSGCLASS	MSGCLASS= assigns the job log to an output class. The output class and its characteristics are identified in a parameter file used at JES initialization.
MSGLEVEL	This field tells the system to reproduce this JCL code in the output, and to include allocation messages.
CLASS	The CLASS= field identifies the JES job class that this job will run under. In the example, CLASS=A is the JES job class. Many sites do not use this option, and the JES class is set according to your user ID. Job classes are set up at JES initialization.

2.47 JCL: EXEC statement

- EXEC statement
 - Region size
 - REGION=4096K
 - Process conditions from previous steps
 - COND=(0,NE)
 - COND=(0,NE,STEP1)

```
//ALLOC EXEC PGM=IEFBR14
//MYDS   DD DSNAME=ROGERS.A.B,DISP=(NEW,KEEP,DELETE),
//          UNIT=3390,VOLUME=SER=123456,SPACE=(CYL,(10,5))
//MYDS1  DD DSNAME=ROGERS.B.C,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,VOLUME=SER=654321,SPACE=(TRK,(20,5))
//
```

Figure 2-53 EXEC statement parameters

JCL: EXEC statement

The EXEC Statement identifies the step and the program to be run. This job only has one step, arbitrarily identified by the name STEP1, and it starts the IEFBR14 program. This program is provided in all installations and its only task is to receive control and give it back to the system; for this reason it is useful for several JCL capabilities.

In the EXEC you specify:

- Step name** In the example, STEP1 gives a name to the step. You might have called the step name IEFBR14 or any name that will help you identify the step. In a large job with many steps, unique step names can assist you when diagnosing problems. The choice is up to you.
- EXEC** This identifies a step job. It must be present.
- PGM=** This specifies the name of the program to be run. In this case the program name is IEFBR14.

EXEC statement parameters

Two other parameters that you might use on the EXEC are:

- REGION** The REGION parameter specifies the quantity of virtual storage (or central storage when ADDRSPC=REAL is coded) that a step requires. If no REGION parameter is specified, the system uses an installation default specified at JES initialization. Various programs can need more storage than is allowed by default. To enable the program to get more storage, you can code the REGION parameter as follows:

```
//STEP1 EXEC PGM=progrname,REGION=4096K
```

This enables the programs to get up to 4 megabytes (MB) of storage below 16 MB and up to the installation default storage above 16 MB (IBM default is 32 MB).

COND The COND parameter is used to inform the system to test return codes from previous job steps and determine whether to bypass this job step. You can specify one or more tests on the COND parameter, and you can test return codes from particular job steps or from every job step that has completed processing.

If the test conditions are satisfied, the system evaluates the COND parameter as true and *bypasses* the job step. If the test conditions are not satisfied, the system evaluates the COND parameter as *false* and *runs* the job step. For example, consider the following statement:

```
//STEP2 EXEC PGM=IEFBR14,COND=(0,NE)
```

With this statement, the system checks the return code from all previous steps. If they were *not* equal to zero, then the system does not run this step. You can also check for return codes that are *EQ* (equal to), *GT* (greater than), *LT* (less than), *GE* (greater than or equal to), and *LE* (less than or equal to). You can check the return code in a specific step by coding the following statement:

```
//STEP2 EXEC PGM=IEFBR14,COND=(0,NE,STEP1)
```

With this statement, if STEP1 ends with return code zero, then STEP2 is run.

2.48 JCL: DD statement

- DD statement** 
 - ddname as referred by the program
- DSN=**
 - The data set name
- DISP=**
 - Disposition of the data set
- DSNTYPE=**
- SPACE=**

```
//MYDS    DD  DSNAME=ROGERS.A.B,DISP=(NEW,KEEP,DELETE) ,
//          UNIT=3390,VOLUME=SER=123456,SPACE=(CYL,(10,5))
//MYDS1   DD  DSNAME=ROGERS.B.C,DISP=(NEW,CATLG,DELETE) ,
//          UNIT=SYSDA,VOLUME=SER=654321,SPACE=(TRK,(20,5))
```

Figure 2-54 JCL: DD statement parameters

JCL: DD statement

To reiterate, the program IEFBR14 actually does nothing. However, it enables you to submit a valid job and have the system process the DD statements identified in the JCL. This enables you to allocate, delete, catalog, and uncatalog data sets by using a batch process. At job initialization, after the program IEFBR14 has been located, the system allocates the data sets specified in the DD statements, then the control is passed to program IEFBR14, which does nothing, and you get notified that your job is done.

The DD (data definition) statement describes a data set and specifies the input and output resources needed for the data set. The DD statement is highlighted in Figure 2-54 and identifies the data set that you want to create.

This statement shows several of the parameters that can be coded in a DD statement:

- ddname** This parameter is used to give a name to the DD statement. NEWDD is the name assigned in the example. The programs refer to DD names rather than dsnames. So, unless a program allocates dynamically, all ddnames referred by a program must be coded. For example, if a program in a step needs a file identified as OUTFILE, you must code a DD named OUTFILE, identifying the relevant data set. In the example, IEFBR14 does not use data sets, so you can choose any ddname you want.
- DSN** This parameter is used to identify the data set. In the example MIRIAM.IEFBR14.TEST.NEWDD is the data set.

DD statement: DISP parameter

The DISP parameter describes the status of a data set to the system and tells what to do with the data set after termination of the step or job. You specify this value for both normal and abnormal termination.

The first field identifies the status of the data set and how to control access to it:

- NEW** This field indicates that the data set will be created and the job will have exclusive control of the data set. No other job can access this data set until the last step in this job that refers to this data set ends. NEW is the default.
- OLD** This field indicates the data set exists and the job requires exclusive access to it.
- MOD** This field indicates that if the data set exists, data will be appended to the end of the data set; otherwise, a new data set will be created. The job requires exclusive access to the data set.
- SHR** This field indicates that the data set can be shared by other users.

The second field in the DISP parameter indicates to the system what to do with the data set when the step finishes NORMAL. It can be:

- CATLG** Catalog the data set
- UNCATLG** Uncatalog the data set
- DELETE** Delete the data set
- PASS** Pass the data set to the subsequent steps
- KEEP** Keep the data set intact

The third field in the DISP parameter indicates the ABNORMAL completion action. It can be: DELETE, CATLG, UNCATLG, or KEEP.

In the example, the status field specifies to create the data set; in the normal termination of the step, to catalog; and in abnormal termination to delete the data set. To delete a data set that exists you code its DSN and DISP=(OLD,DELETE,DELETE).

Fast Path: A *catalog* is a data set that contains information about other data sets. It provides users with the ability to locate a data set by name, without knowing where the data set is. By cataloging data sets, your users will need to know less about your storage setup. Therefore, data can be moved from one device to another, without requiring a change in JCL DD statements that refer to an existing data set.

Cataloging data sets also simplifies backup and recovery procedures. Catalogs are the central information point for data sets; all data sets must be cataloged. In addition, all SMS-managed data sets must be cataloged.

DD statement: UNIT parameter

This parameter identifies the device or type of device on which the data set will be allocated. You use this parameter to specify the number of devices to be used. If the data set exists, you only need to specify the device type if the data is not cataloged. Most installations now administer disk storage with the Storage Management System (DFSMS). With SMS, you do not need to use the UNIT parameter to specify a device for SMS-controlled data sets. Several common examples are shown here:

- UNIT=SYSDA** Allocates the data set on a DASD
- UNIT=3390** Allocates the data set on a 3390 type disk
- UNIT=SYSALLDA** Allocates the data set on a DASD
- UNIT=TAPE** Allocates the file on a TAPE device

DD statement: SPACE parameter

The *SPACE* DD parameter is required for allocating data sets on DASD. It identifies the space allocation required for your data set. Before a data set can be created on disk, the system must know how much space the data set will require and how the space is to be measured. You can code it as shown in the following example:

```
SPACE=(type,(primary-qty,second-qty,directory))
```

In this statement, *type* can be any of the following values:

TRK	Requests that space be allocated in tracks.
CYL	Requests that space be allocated in cylinders.
block length	Specify here the block length to request an allocation in number of blocks. Indicates that the values specified for primary and secondary allocations are block quantities, and directs the system to compute the number of tracks to allocate using a block length. For example, <i>SPACE</i> =(3150,(5,1)) means that the system will allocate five blocks, each block having 3150 bytes, as primary space and one block as secondary space.
record length	Specifies that the average record length in bytes will be used to allocate space. This is only applicable if SMS is active and the AVGREC parameter is coded.

The system allocates DASD space in whole tracks. The number of tracks required depends on how the records are blocked:

primary-qty	Specifies the initial allocation amount.
secondary-qty	Specifies an additional allocation amount. The system does not allocate additional space until it is needed.
directory	You must code for a partitioned data set, to indicate the number of blocks the system must reserve for the directory. Partitioned data sets Extended (PDSE) grow dynamically; if you specify directory size, SMS uses the size you specify only if you later convert the PDSE to a PDS. Omit this parameter for sequential data sets.

Other DD statement SPACE parameter options

The following list includes several examples showing how you can code:

SPACE=(TRK,4)	To allocate space to a sequential data set, requesting four tracks, only primary space.
SPACE=(CYL,(5,2))	To allocate space to a sequential data set, five cylinders as primary allocation space and two cylinders as secondary.
SPACE=(CYL,(10,,100))	To allocate space to a partitioned data set, only primary allocation of 100 cylinders and 100 directory blocks.

LRECL and BLKSIZE parameters

Programs access data sets through ddnames. The ddnames are defined in the programs. Like them, data set characteristics, such as data set organization, logical record size, and record size, are also defined for the programs.

During the allocation process you have to specify several of these characteristics. You can use the following DD statement parameters to specify the data set characteristics:

LRECL Identifies the data set logical record size.

BLKSIZE Specifies the maximum length, in bytes, of a block.

RECFM	Identifies the record format.
DSORG	Identifies the data set organization. If you do not specify DSORG, the system uses the information in SPACE to determine if the data set is to be sequential or partitioned

These parameters are part of data control block (DCB) information and can be coded in the JCL as shown in the following statement:

```
// DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=P0)
```

They can also be coded as individual parameters, without the need to specify the DCB=(....) parameter:

```
//      LRECL=80,
//      BLKSIZE=3120,
//      RECFM=FB,
//      DSORG=P0
```

If BLKSIZE is not specified, the system will determine what it considers to be an optimum block size for the device type on which the data set will be allocated. For a data set with fixed record format (all records with same size), the BLKSIZE must be a multiple of LRECL. For variable record size, the BLKSIZE must be a multiple of the greatest record plus four.

2.49 Submitting a job

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
EDIT      ROGERS.JCL.VERS5(IEHLIST) - 01.06          Columns 00001 00072
Command ==> sub                                Scroll ==> PAGE
***** ***** Top of Data *****
000010 //ROGERSW JOB    (POK,999),MSGCLASS=T,NOTIFY=ROGERS
000200 //STEP1    EXEC PGM=IEHLIST
000650 //SYSPRINT DD SYSOUT=*
000660 //TOTDCI   DD DISP=SHR,VOL=SER=SBOXB4,UNIT=3390
000670 //SYSIN     DD *
000680           LISTVTOC VOL=3390=SBOXB4,FORMAT
000690 /*
000691 //
***** ***** Bottom of Data *****

JOB ROGERSW(JOB21266) SUBMITTED
***
```

Figure 2-55 Submitting a job

Submitting a job

At this point, a job has been created in the data set ROGERS.JCL.VERS5(IEHLIST), as shown in the JCL member IEHLIST. Now this JCL is ready to be submitted for execution by using the **SUBMIT** command.

To submit the JCL stream, enter **submit(sub)** on the command line. The TSO/E command processor sends the JCL statements to the JES subsystem for execution.

After entering the command, you receive the following message indicating that your job was submitted successfully, as shown in Figure 2-55:

JOB jobname (jobnumber) SUBMITTED

2.50 System Display and Search Facility

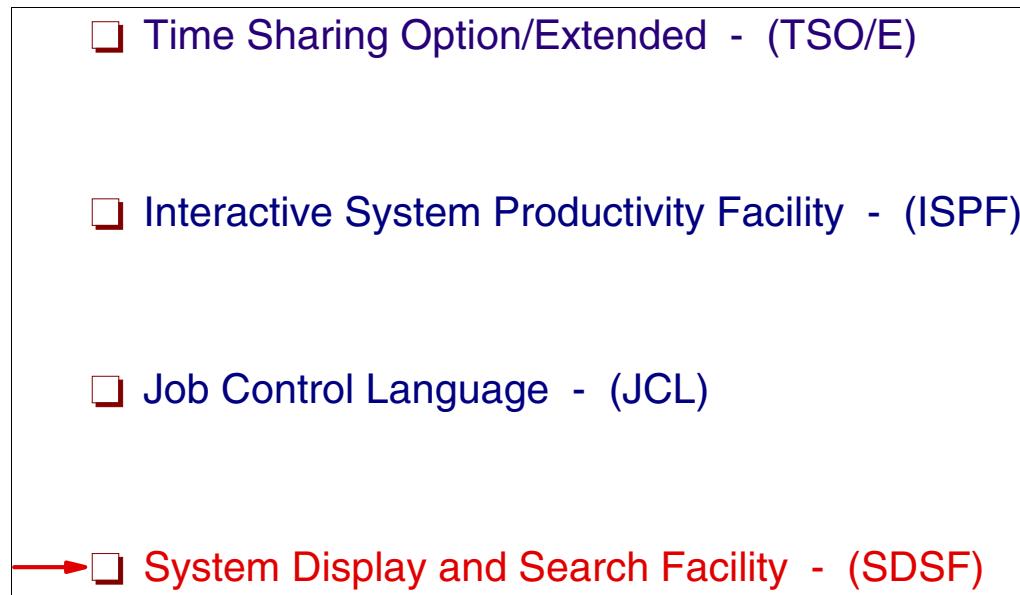


Figure 2-56 System Display and Search Facility (SDSF)

System Display and Search Facility

SDSF is a licensed program that helps authorized users efficiently monitor and control the operation of a JES2 or JES3 subsystem.

With SDSF you can authorize your JES users to perform the following tasks:

- ▶ Control job processing (hold, release, cancel, and purge jobs)
- ▶ Monitor jobs while they are being processed
- ▶ Display job output before deciding to print it
- ▶ Manage the system's workflow
- ▶ Control the order in which jobs are processed
- ▶ Determine the number of output jobs and the total number of records to be printed
- ▶ Control the order in which output is printed
- ▶ Control printers and initiators
- ▶ View the system log online and use commands to search for specific information in the log
- ▶ Dynamically change job data set output descriptors
- ▶ Issue JES and MVS commands that affect their jobs
- ▶ Print selected lines of the JES output data set
- ▶ Edit JCL direct from spool

2.51 SDSF: Panels hierarchy

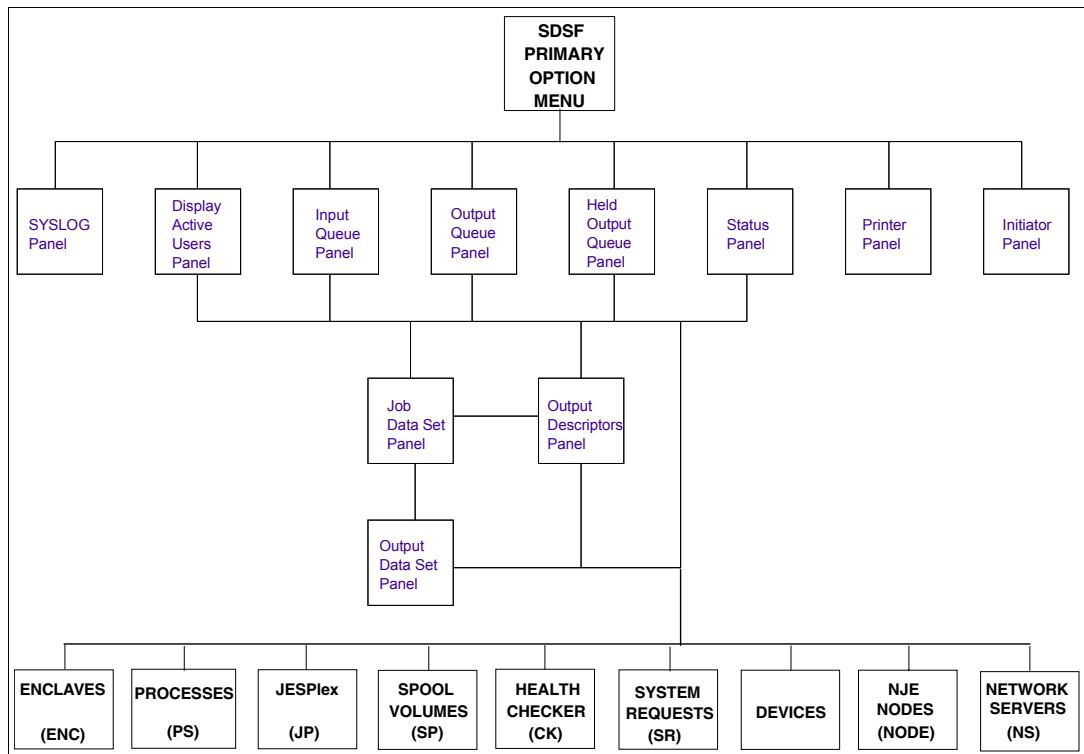


Figure 2-57 SDSF: JES2 Panel hierarchy

SDSF: JES2 panel hierarchy

SDSF consists of panels that provide immediate information about jobs, printers, queues, and resources in a JES2 system. The SDSF panel hierarchy is illustrated in Figure 2-57. From these panels, authorized users can enter SDSF commands to control the processing of jobs and the operation of system resources. Authorized users also can issue MVS and JES2 system commands from the SDSF panels.

SDSF provides an easy way to manage JES2 jobs, which can help you work more efficiently. It gives immediate, current, sysplex-wide information about jobs waiting to be processed or in execution, such as the following items:

- ▶ The status, class, priority, date, and time of a specific job
- ▶ All jobs on a specific queue, such as the input or held output queue
- ▶ Detail for a job no matter where it is in the sysplex
- ▶ Reasons why a job might be delayed
- ▶ Output from a job as it is created

Using the SDSF panels, SDSF commands and action characters, and by typing over panel fields, you can hold or release jobs, cancel jobs, filter the jobs displayed to show just the jobs that interest you, or change a job's priority, class, or destination.

2.52 JES2 SDSF Primary Option Menu

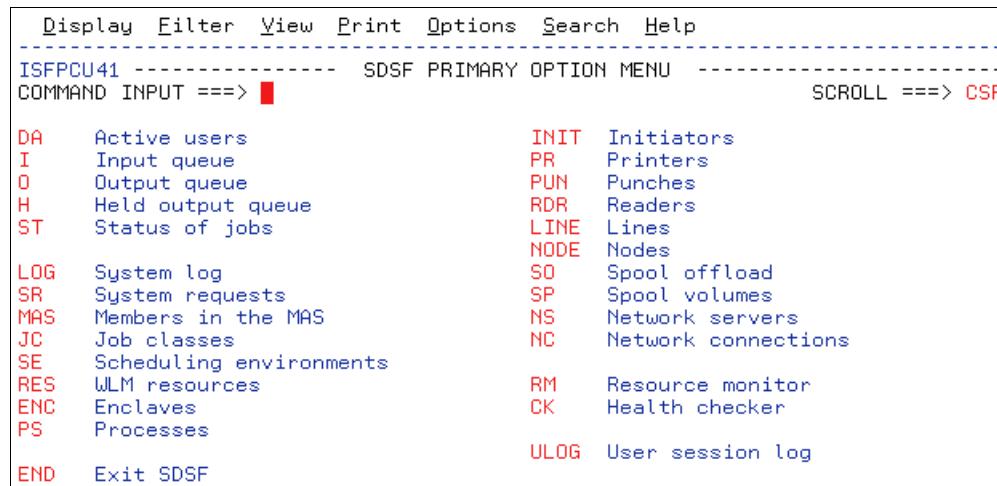


Figure 2-58 JES2 SDSF Primary Option Menu

JES2 SDSF Primary Option Menu

This section provides information about the SDSF facility and how you can use its functions to monitor and manage your workloads. You see how SDSF can be used to display job input and output data, and purge (delete) jobs that are on the input, output, or held queues. You review the monitoring functions of SDSF that enable you to evaluate the current workload, and enable you to cancel, hold, or reschedule work.

SDSF can be started from ISPF menus, but the setting of the options is often customized by each site differently. You will have to review your site's ISPF menus to find the SDSF option. Alternatively, issuing the **TSO SDSF** command from the Command ==> line starts SDSF. After choosing this option, the panel you receive will be similar to the one in Figure 2-58.

However, it might not have all the same options shown in the figure; the options can vary according to the security level of the user. The authority to perform functions in these options also varies according to the security level of the user. It is possible to control most system functions by using the SDSF facility. The scope of the functions includes reviewing job output, controlling the processing of jobs (both their input and output), printer control, operator functions, and system administration.

2.53 SDSF: JES3 panel hierarchy

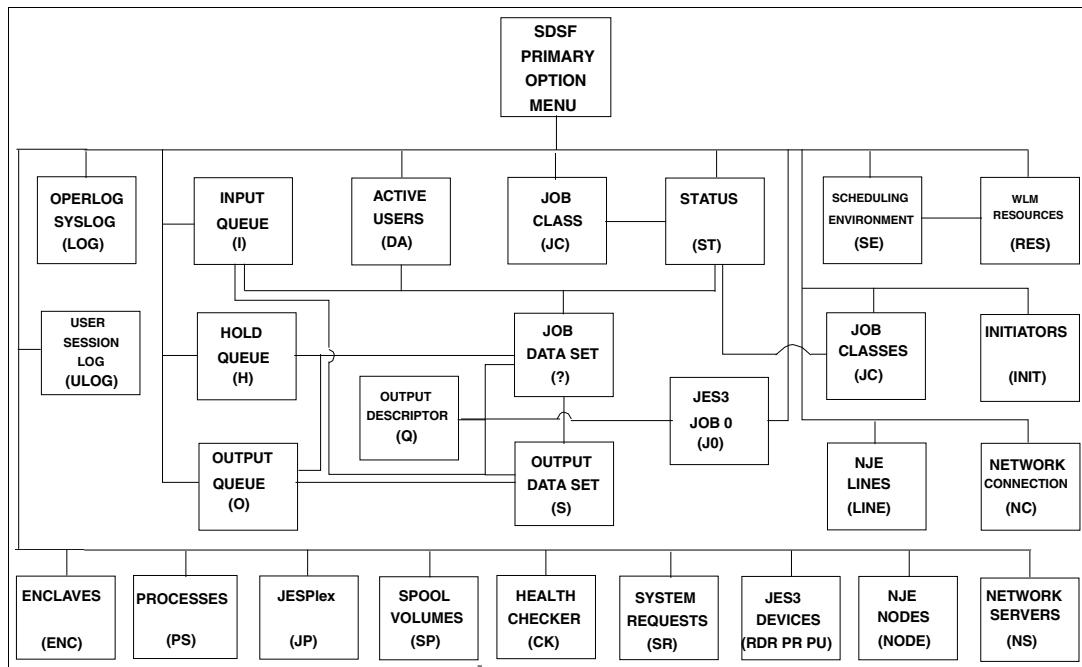


Figure 2-59 SDSF: JES3 panel hierarchy

SDSF: JES3 panel hierarchy

SDSF consists of panels that provide immediate information about jobs, printers, queues, and resources in a JES3 system. The SDSF panel hierarchy is illustrated in Figure 2-59. From these panels, authorized users can enter SDSF commands to control the processing of jobs and the operation of system resources. Authorized users also can issue MVS and JES3 system commands from the SDSF panels.

SDSF provides an easy way to manage JES3 jobs, which can help you work more efficiently. It gives immediate, current, sysplex-wide information about jobs waiting to be processed or in execution, such as the following items:

- ▶ The status, class, priority, date and time of a specific job
- ▶ All jobs on a specific queue, such as the input or held output queue
- ▶ Detail for a job no matter where it is in the sysplex
- ▶ Reasons a job might be delayed
- ▶ Output from a job as it is created

Using the SDSF panels, SDSF commands and action characters, and by typing over panel fields, you can hold or release jobs, cancel jobs, filter the jobs displayed to show just the jobs that interest you, or change a job's priority, class, or destination.

2.54 JES3 SDSF Primary Option Menu

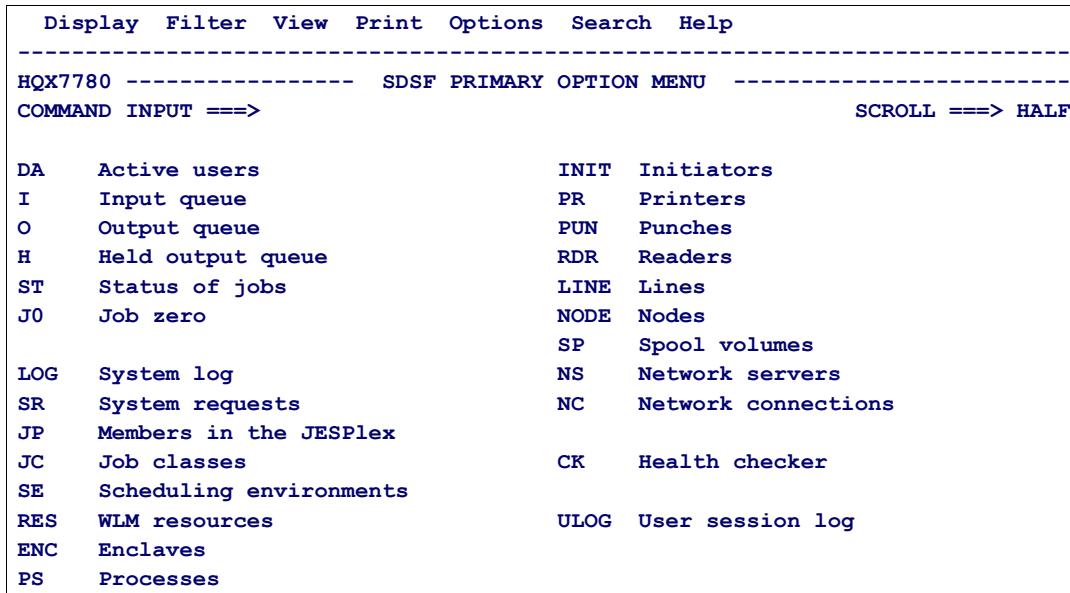


Figure 2-60 JES3 SDSF Primary Option Menu

JES3 SDSF Primary Option Menu

This section provides information about the SDSF facility and how you can use its functions to monitor and manage your workloads. You see how SDSF can be used to display job input and output data, and purge (delete) jobs that are on the input, output, or held queues. You review the monitoring functions of SDSF that enable you to evaluate the current workload, and enable you to cancel, hold, or reschedule work.

SDSF can be started from ISPF menus, but the setting of the options is often customized by each site differently. You will have to review your site's ISPF menus to find the SDSF option. Alternatively, issuing the **TSO SDSF** command from the Command ==> line starts SDSF. After choosing this option, the panel you receive will be similar to the one in Figure 2-60.

However, it might not have all of the same options shown in the figure; the options vary according to the security level of the user. The authority to perform functions in these options also varies according to the security level of the user. It is possible to control most system functions by using the SDSF facility. The scope of the functions includes reviewing job output, controlling the processing of jobs (both their input and output), printer control, operator functions, and system administration.

2.55 SDSF: Options menu

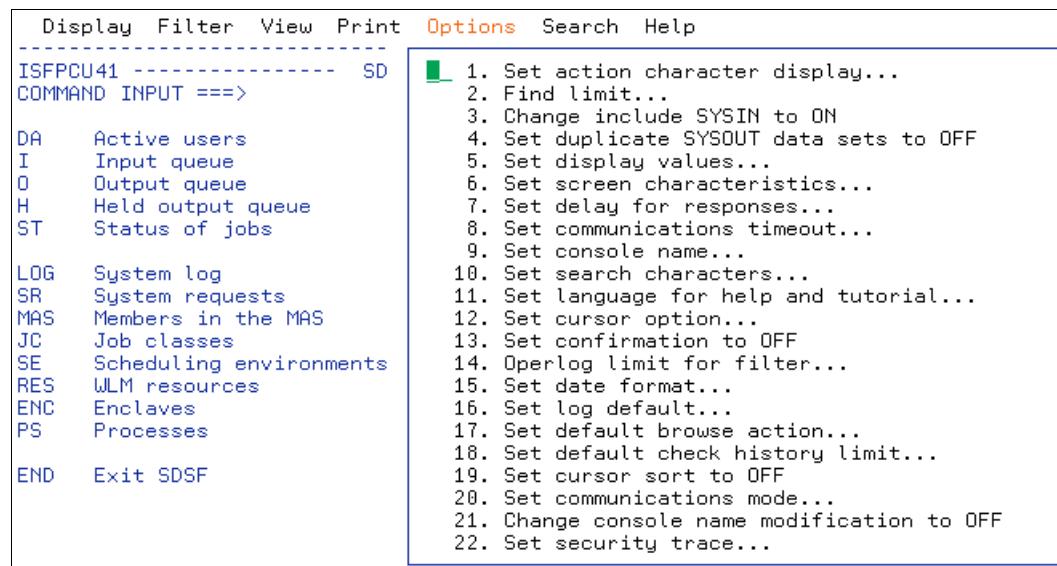


Figure 2-61 SDSF: Options menu pull-down

SDSF: Options menu pull-down

Before choosing any option, place the cursor in the menu action bar, select **Options**, and press Enter. Option 5 shows Set display values to OFF. If you then choose this option, the display options value will be set to OFF. You get the same result by issuing the SDSF command **SET DISPLAY OFF** in the COMMAND INPUT line.

You can customize your SDSF panels by choosing the **View** option in the action bar and then the **Arrange** option. You do that for each panel where the **Arrange** option is available, and choose the fields in the order you want them. SDSF stores the information in your ISPF/PDF profile data set, and uses them the next time you enter any SDSF options.

2.56 SDSF: viewing the output files

```
Display Filter View Print Options Search Help
-----
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 6,678      LINE 1-1 (1)
COMMAND INPUT ===>                               SCROLL ===> HALF
PREFIX=ROGERS* DEST=(ALL) OWNER=* SYSNAME=*
ACTION//=-Block,--Repeat,+-Extend,-?JDS,Q-OutDesc,S-Browse,SB-ISPFBrowse,
ACTION=SE-ISPFEdit,SJ-JCLEdit,X-Print,XC-PrintClose,XD-PrintDS,
ACTION=XDC-PrintDSClose,XF-PrintFile,XFC-PrintFileClose,XS-PrintSysout,
ACTION=XSC-PrintSysoutClose
NP   JOBNAME JobID    Owner     Prty C Dest          Tot-Rec Tot-Page F
?    ROGERSW JOB21266 ROGERS     1 T ANYLOCAL        6,678           S

-----
Display Filter View Print Options Search Help
-----
SDSF JOB DATA SET DISPLAY - JOB ROGERSW (JOB21266)      LINE 1-4 (4)
COMMAND INPUT ===>                               SCROLL ===> HALF
PREFIX=ROGERS* DEST=(ALL) OWNER=* SYSNAME=*
NP   DDNAME StepName ProcStep DSID Owner     C Dest          Rec-Cnt Page
JESMSGLG                      2 ROGERS   T ANYLOCAL       9
JESJCL                         3 ROGERS   T ANYLOCAL      11
JESYSMSG                       4 ROGERS   T ANYLOCAL      17
SYSPRINT STEP1                 10 ROGERS  T ANYLOCAL    6,641
```

Figure 2-62 SDSF: Viewing the JES2 output files

SDSF: viewing the output files

To be able to see the JES output data sets created during the execution of the job, the output is saved on the JES spool as data sets. The JES data sets can be displayed by using the following SDSF JES queues: input (**i** SDSF option), execution queue (**DA** option), output queue (**O** option) or held queue (**H** option).

For output and held queues, you cannot see those JES data sets you requested to be automatically purged by setting a MSGCLASS out sysout CLASS that has been defined to not save output. Also, depending on the MSGCLASS you chose in the JOB card, the sysouts can be in the Output queue or in the Held queue.

The first window shown in Figure 2-62 displays the job that was submitted in Figure 2-55 on page 144 and whose output was directed to the HOLD (Class T) queue, as identified in the MSGCLASS=T parameter in the job card. Because only one job was submitted and run, then only one job is displayed on the hold queue.

Using the question mark (?) action character

Issuing a question mark (?) command in the NP column displays the output files generated by job 21266. Figure 2-62 displays ddnames:

- ▶ JES message log file
- ▶ JES JCL file
- ▶ JES system messages file
- ▶ The SYSPRINT file

This option is useful when you are seeing jobs with many files directed to SYSOUT and you want to display one associated with a specific step. You issue an **S** in the NP column to select a file you want to view.

Viewing the output files for a job

To see all files, then rather than a question mark (?), type **S** in the NP column. The result is shown in Figure 2-63 on page 153.

Figure 2-63 on page 153 shows part of the output for the job. The following list includes some of the most important items to note:

- ▶ IEF142I ROGERSW STEP1 - STEP WAS run - COND CODE 0000.
- ▶ The RC or Return Code value is *00*, which indicates a successful completion of the step.
- ▶ The COND CODE or Condition Code is *0000*, which indicates a successful completion of the job.
- ▶ The following messages show that the data set ADH.SC63.INTERVAL.DATA was allocated on disk volume SBOXB4, as shown in Figure 2-63 on page 153:

ADH.SC63.INTERVAL.DATA	SBOXB4	1	2009.034	1999.365
------------------------	--------	---	----------	----------

```

15:22:19 ---- IAT6853 THE CURRENT DATE IS WEDNESDAY, 20 APR 2011 ----
IRR010I user ID ROGERS IS ASSIGNED TO THIS JOB.
15:22:19 IAT2000 JOB ROGERSW (JOB21266) SELECTED SC75 GRP=A
15:22:19 ICH70001I ROGERS LAST ACCESS AT 15:21:55 ON WEDNESDAY, APRIL 20, 20
15:22:20 - -----TIMINGS (MINS.)---
15:22:20 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB VECT VAFF
15:22:20 -ROGERSW STEP1 00 794 .00 .00 .00 .00
15:22:20 -ROGERSW ENDED. NAME- TOTAL CPU TIME=.00 TO
//ROGERSW JOB (POK,999),MSGCLASS=T,NOTIFY=ROGERS 00001001
//STEP1 EXEC PGM=IEHLIST 00020000
//SYSPRINT DD SYSOUT=* 00065005
//TOTDCI DD DISP=SHR,VOL=SER=SBOXB4,UNIT=3390 00066006
//SYSIN DD *
/*
1 //ROGERSW JOB (POK,999),MSGCLASS=T,NOTIFY=ROGERS
2 //STEP1 EXEC PGM=IEHLIST
3 //SYSPRINT DD SYSOUT=*
4 //TOTDCI DD DISP=SHR,VOL=SER=SBOXB4,UNIT=3390
5 //SYSIN DD *

ICH70001I ROGERS LAST ACCESS AT 15:21:55 ON WEDNESDAY, APRIL 20, 2011
IEF236I ALLOC. FOR ROGERSW STEP1
IEF237I JES3 ALLOCATED TO SYSPRINT
IEF237I 8333 ALLOCATED TO TOTDCI
IEF237I JES3 ALLOCATED TO SYSIN
IEF142I ROGERSW STEP1 - STEP WAS run - COND CODE 0000
IEF285I ROGERS.ROGERSW.JOB21266.D000000A.? SYSOUT
IEF285I SYS1110.T152219.RA000.ROGERSW.R0200199 KEPT
IEF285I VOL SER NOS= SBOXB4.
IEF285I ROGERS.ROGERSW.JOB21266.D0000009.? SYSIN
IEF373I STEP/STEP1 /START 201110.1522
IEF032I STEP/STEP1 /STOP 201110.1522
CPU: 0 HR 00 MIN 00.09 SEC SRB: 0 HR 00 MIN 00.00 SEC
VIRT: 84K SYS: 212K EXT: 8K SYS: 10728K
IEF375I JOB/ROGERSW /START 201110.1522
IEF033I JOB/ROGERSW /STOP 201110.1522
CPU: 0 HR 00 MIN 00.09 SEC SRB: 0 HR 00 MIN 00.00 SEC
SYSTEMS SUPPORT UTILITIES---IEHLIST
DATE: 2011.110 TIME: 15.22.19
CONTENTS OF VTOC ON VOL SBOXB4 <THIS VOLUME IS NOT SMS MANAGED>
THERE IS A 2 LEVEL VTOC INDEX
Data sets ARE LISTED IN ALPHANUMERIC ORDER
FORMAT 4 DSCB NO AVAIL/MAX DSCB /MAX DIRECT NO AVAIL NEXT ALT FORMAT 6 LAS
VI DSCBS PER TRK BLK PER TRK ALT TRK TRK(C-H) (C-H-R) DSC
81 4497 50 45 0 0 0 8
-----data set NAME----- SER NO SEQNO DATE.CRE DATE.EXP
ADH.SC63.INTERVAL.DATA SBOXB4 1 2009.034 1999.365
SMS.IND LRECL KEYLEN INITIAL ALLOC 2ND ALLOC EXTEND LAST BLK(T-R-L
0 CYLS 1
EATTR
NS
EXTENTS NO LOW(C-H) HIGH(C-H)
0 347 0 347 14
---UNABLE TO CALCULATE EMPTY SPACE.
----- the rest of the output is not displayed here-----

```

Figure 2-63 SDSF displaying output for the job

2.57 SDSF filters for panel displays

- ❑ You can limit your SDSF panels to display only what you want to see by using these SDSF commands that filter the information to be displayed
- ❑ SDSF commands for filtering of data:
 - DEST
 - OWNER
 - PREFIX
 - SYSNAME

Figure 2-64 SDSF filter commands

SDSF filter commands

Use a **FILTER** command to limit rows on the tabular panels to only those that you want to display. The filters you set are displayed on any panel you are currently using:

```
Display Filter View Print Options Search Help
-----
SDSF OUTPUT ALL CLASSES ALL FORMS    LINES 1,289,060   LINE 1-20 (724)
COMMAND INPUT ===>                      SCROLL ===> HALF
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=*
```

DEST Use the **DEST** command to limit jobs on the SDSF panels to those with particular destinations. Type **DEST** on the command line or select it from the Filter pull-down menu. It affects only the H, I, O, PR, PUN, and ST panels.

- ▶ With no parameters, **DEST** displays jobs for all authorized destinations, if no **IDEST** list is specified in ISFPARMS. If both **IDEST** and **DEST** lists are specified in ISFPARMS, the jobs for authorized destination names in the **IDEST** list are displayed.

destination name This name can be from one to four destination names that replace the current destination list, in any format acceptable to JES.

- +** Using a plus (+) sign followed by a destination adds the destination name that follows to the current destination list.
- Using a minus (-) sign followed by a destination deletes the destination names that follow from the current destination list.
- ?** The question mark (?) displays the current setting for DEST.

Note: COMMAND INPUT ===> **dest chicago omaha**

- ▶ This command displays all jobs with destinations of CHICAGO and OMAHA. This list replaces the current destination list.

COMMAND INPUT ===> **dest + laramie**

- ▶ This command adds all jobs with the destination of LARAMIE.

OWNER Use the **OWNER** command to limit jobs displayed on the SDSF panels to the owning user IDs for those jobs. Jobs with writer IDs that match the owning user IDs are also displayed. Type **OWNER** on the command line or select it from the Filter pull-down menu. It only affects jobs on the DA, H, I, O, and ST panels.

- ▶ With no parameters, **OWNER** displays all jobs for all owner IDs.

owner-pattern This is the owning user ID for the job, or the netmail ID, which is in either the WTR field or is part of the DEST field in the O panel. When specified, SDSF displays only those jobs or output with owning user IDs or netmail IDs that match the specified owner-pattern.

The owner-pattern can be up to eight characters and might include the special pattern matching characters.

? This character displays the current setting for **OWNER**.

Example: COMMAND INPUT ===> **owner ***

- ▶ With no other filtering in effect, this command displays all jobs for all owner IDs.

COMMAND INPUT ===> **owner rogers**

- ▶ With no other filtering in effect, this command displays only jobs for that owner.

PREFIX Use the **PREFIX** command to limit your panels to jobs whose names match a specific character string. Type **PREFIX** on the command line or select it from the Filter pull-down menu. It only affects the DA, I, O, H, and ST panels.

- ▶ With no parameters - **PREFIX** displays all jobs, except on the H panel, where it displays all jobs with names that begin with your user ID.
- ▶ To display all jobs on the H panel, enter the **H ALL** command, and then enter the **PREFIX * command**. If you do not enter the **PREFIX** command, the **PREFIX** setting defaults to the **PREFIX** setting in ISFPARMS.

string-pattern This is a character string of up to eight characters that can limit which jobs are displayed on the DA, I, ST, O, and H panels. Only those jobs whose names match string-pattern are displayed, and on the O panel, jobs whose netmail ID match string-pattern are also displayed.

? The ? displays the current **PREFIX** setting of string-pattern. You can retain that setting by pressing Enter or change it by entering a new parameter.

Example: COMMAND INPUT ===> **prefix ieb***

- ▶ When this command is in effect, the DA, I, ST, O, and H panels display only jobs whose names begin with the character string IEB.

SYSNAME Use the **SYSNAME** command to select the systems in the sysplex displayed on the DA panel. Type **SYSNAME** on the command line or select it from the Filter pull-down menu. It only affects the DA panel.

- ▶ With no parameters, **SYSNAME** selects only address spaces running on the system you are logged on to.

system-name The system-name is a character string of up to eight characters to limit the systems displayed on the DA panel. Only those address spaces running on systems whose names match system-name are displayed.

? The question mark (?) displays the current setting for **SYSNAME**.

Examples: Be aware of the following points:

COMMAND INPUT ===> **sysname system10**

- ▶ This command selects only SYSTEM10 for display on the DA panel.

COMMAND INPUT ===> **sysname**

- ▶ This command displays all address spaces running on the system the user is logged on to.

COMMAND INPUT ===> **sysname ***

- ▶ This command displays all address spaces running on all systems.

2.58 SDSF: Display Active Users (DA command)

Display Filter View Print Options Search Help														

SDSF DA SC75 (ALL) PAG 0 CPU/L/Z 4/ 1/ 0 LINE 1-25 (155)			SCROLL ==> HALF											
COMMAND INPUT ==>														
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=*														
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP Real Paging						
					STC03929	+MASTER+	NS FF	6880 0.00						
	MASTER						NS FF	192 0.00						
	PCAUTH	PCAUTH					NS FF	529 0.00						
	RASP	RASP					NS FF	627 0.00						
	TRACE	TRACE					NS FF	909 0.00						
	DUMPSRV	DUMPSRV	DUMPSRV				NS FF	10T 0.00						
	XCFAS	XCFAS	IEFPROC				NS FF	21T 0.00						
	GRS	GRS					NS FF	1844 0.00						
	SMSPDSE	SMSPDSE					NS FF	2026 0.00						
	SMSPDSE1	SMSPDSE1					NS FF	3151 0.00						
	SMSVSAM	SMSVSAM	IEFPROC				NS FF	7045 0.00						
	CONSOLE	CONSOLE					NS FF	8849 0.00						
	WLM	WLM	IEFPROC				NS FE	1447 0.00						
	ANTMAIN	ANTMAIN	IEFPROC				NS FE	1327 0.00						
	ANTAS000	ANTAS000	IEFPROC				NS FF	511 0.00						
	DEVMAN	DEVMAN	IEFPROC				NS FF	27T 0.00						
	OMVS	OMVS	OMVS				NS FE	709 0.00						
	RESOLVER	RESOLVER	IEFPROC				NS FF	89 0.00						
	IEFSCHAS	IEFSCHAS					NS FF	648 0.00						
	JESXCF	JESXCF	IEFPROC				NS FF	3367 0.00						
	ALLOCAS	ALLOCAS					NS FF	649 0.00						
	JES2MON	JES2MON	IEFPROC											

Figure 2-65 SDSF - Display Active Users (DA command)

SDSF: Display Active Users (DA command)

SDSF provides the ability to monitor the current system workload. The **DA** command displays the active tasks and provides information about each task. This information includes processor usage for each task, the amount of processor time that a task has used, and the I/O-related Execute Channel Program (EXCP) statistics. Figure 2-65 displays part of the data that this facility captures. Press PF11 to move to the right and see all the available fields.

Attention: Notice that in the upper right corner of the figure, there are 155 active address spaces in the sysplex that has two logical partitions (LPARS), one JES2 and the other JES3. Also there are many columns for this display that are not shown.

Use the **DA** command to view jobs running on any system in a sysplex. You can list the active users and display information about MVS address spaces for jobs, started tasks, initiators, or TSO users on the DA panel. When RMF is installed, SDSF uses it as the source of data for the panel.

The **DA** command accepts parameters to limit the address spaces that are displayed according to type and position (location). The parameters control the type (jobs, started tasks, TSO users, or initiators) and position (swapped in, swapped out, in transition, or ready) of address spaces that are displayed.

Other DA command options

When you enter the **DA OTSU** command, SDSF displays only TSO users. The **DA OJOB** command shows only the JES jobs running, and the **DA OSTC** command shows only the active started tasks.

Using action characters

SDSF provides action commands you can use in the NP column. For example, from the Active Users panel, a user can enter S in the NP column next to a job to look at the output data set for that job. SDSF displays the output data set on the Output data set panel. You might not have the authority to issue various of the available commands. In this case, SDSF issues a message in the upper right of the panel. When you choose an action command, SDSF issues the system command that corresponds to the action you chose.

To display which action commands can be used in an SDSF panel, issue the **HELP** command in each option panel. Then choose option **3 - Action characters**. A panel displays, listing all of the action commands that you can use in that option. Figure 2-66 displays the action characters for JES2 for the DA panel.

```
ACTION=//-Block,=-Repeat,+Extend,?-JDS,A-Release,C-Cancel,CA-CancelARM,  
ACTION=CD-CancelDump,CDA-CancelARDump,D-Display,DL-DisplayLong,E-Restart,  
ACTION=EC-RestartCancel,ES-RestartStep,ESH-RestartStepHold,H-Hold,K-SysCancel,  
ACTION=KD-SysCancelDump,L-List,LL-ListLong,P-Purge,PP-PurgeProtected,Q-OutDesc,  
ACTION=R-Reset,RQ-ResetQuiesce,S-Browse,SB-ISPFBrowse,SE-ISPFEdit,SJ-JCLEdit,  
ACTION=W-Spin,X-Print,XC-PrintClose,XD-PrintDS,XDC-PrintDSClose,XF-PrintFile,  
ACTION=XFC-PrintFileClose,XS-PrintSysout,XSC-PrintSysoutClose,Y-SysStop,  
ACTION=Z-SysForce
```

Figure 2-66 Action characters for JES2 for the DA panel

Figure 2-67 displays the action characters for JES3 for the DA panel.

```
ACTION=//-Block,=-Repeat,+Extend,?-JDS,A-Release,C-Cancel,CA-CancelARM,  
ACTION=CD-CancelDump,CDA-CancelARDump,CP-CancelPrint,D-Display,  
ACTION=DE-DisplayEstimates,DL-DisplayLong,DSD-DisplayDDDnames,  
ACTION=DSH-DisplaySpoolHold,DSP-DisplaySpoolPartition,DX-DisplayExtended,  
ACTION=E-Restart,H-Hold,K-SysCancel,KD-SysCancelDump,L-List,LB-ListBDT,  
ACTION=LH-ListHold,LT-ListTCP,P-Purge,Q-OutDesc,R-Reset,RQ-ResetQuiesce,  
ACTION=S-Browse,SB-ISPFBrowse,SE-ISPFEdit,SJ-JCLEdit,W-Spin,X-Print,  
ACTION=XC-PrintClose,XD-PrintDS,XDC-PrintDSClose,XF-PrintFile,  
ACTION=XFC-PrintFileClose,XS-PrintSysout,XSC-PrintSysoutClose,Y-SysStop,  
ACTION=Z-SysForce
```

Figure 2-67 Action characters for JES3 for the DA panel

2.59 Issuing MVS and JES commands

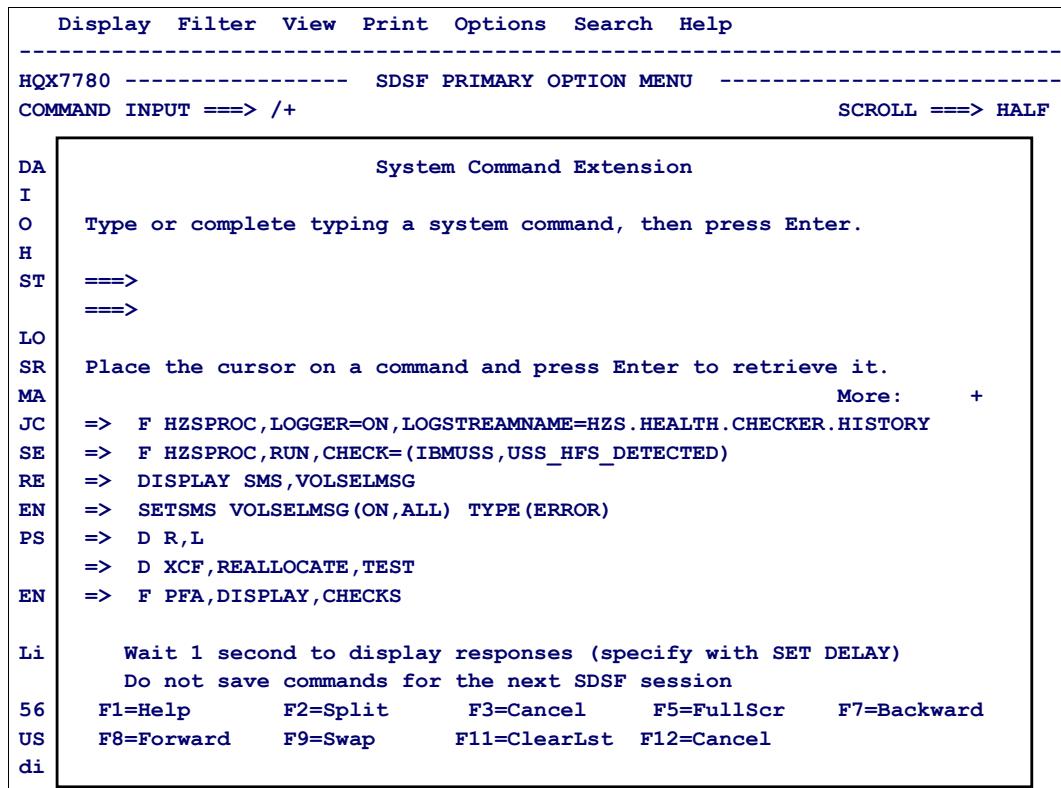


Figure 2-68 Using SDSF to issued MVS and JES Commands

Issuing MVS and JES commands

If you are authorized, then on the COMMAND INPUT ==> line of any SDSF panel you can issue any MVS, JES2, or JES3 command following a forward slash (/ followed by the command):

- ▶ /\$da a JES2 command
- ▶ /*i j=21266 a JES3 command
- ▶ /d a,1 a MVS command

Long commands

If the command is too long for the command line, type /+ on the command line and two more lines will appear to enable you to type the rest of the command, as shown in Figure 2-68. Using /* on the command line displays the system command extension pop-up menu, shown in Figure 2-68, which enables you to enter longer system commands. If you have already entered command text on the command line, it appears on the pop-up menu and you can continue typing the command.

User log (ULOG) panel

The ULOG panel enables you to display all MVS, JES3, and JES2 commands and responses (including commands generated by SDSF) you issued during your session. The log is deleted when you end the SDSF session.

If you have authority, you can use the **ULOG** command option to see only your commands and their response.

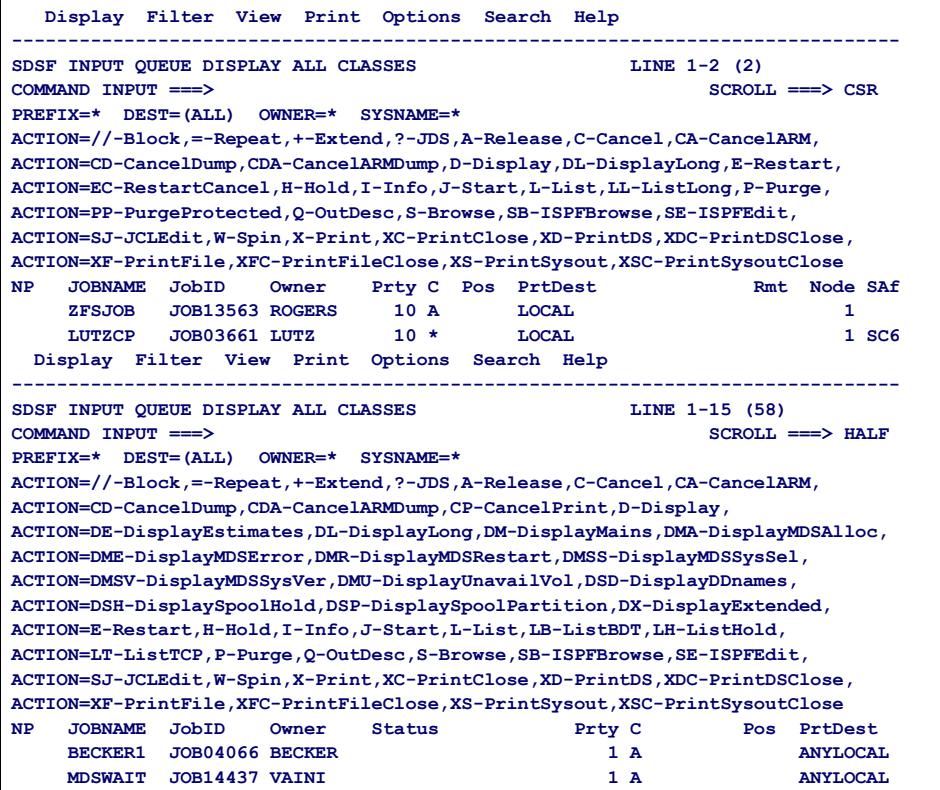
Viewing the ULOG

Use the **ULOG** command to browse the user log to see all of the system commands and responses issued during your current SDSF session in chronological order. A ULOG is displayed in Figure 2-69.

```
ISF031I CONSOLE ROGERS ACTIVATED
-$DA
$HASP612 NO ACTIVE JOBS
-SC75*I J=21266
IAT8674 JOB ROGERSW (JOB21266) P=01 CL=A      OUTSERV(PENDING WTR)
IAT8699 INQUIRY ON JOB STATUS COMPLETE,        1 JOB DISPLAYED
-D A,L
IEE114I 11.07.42 2011.111 ACTIVITY 629
JOBS   M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM    OAS
00006  00031  00012     00037    00019   00012/00030    00026
LLA    LLA    LLA     NSW S NET      NET      NET    NSW S
VLF    VLF    VLF     NSW S RRS      RRS      RRS    NSW S
OPTSO  OPTSO  OPTSO   OWT S APPC    APPC    APPC   NSW S
ASCH   ASCH   ASCH   NSW S RMF      RMF      IEFPROC NSW S
SDSF   SDSF   SDSF   NSW S HZSPROC HZSPROC HZSTEP NSW SO
RACF   RACF   RACF   NSW S JES2      JES2      IEFPROC NSW S
TSO    TSO    STEP1   OWT S TCPIP    TCPIP    TCPIP  NSW SO
RMFGAT RMFGAT IEFPROC NSW SO CBDQDISP CBDQDISP TDIS    OWT SO
WEBPW  WEBPW  WEBSRV  IN  SO INETD1  STEP1    OMVSKERN OWT AO
IOASRV IOASRV SERVER  OWT SO PORTMAP  PORTMAP  PORTMAP OWT SO
TN3270 TN3270 TN3270  NSW SO FTPOE1  STEP1    TCPIP    OWT AO
FTPMVS1 STEP1   TCPIP   OWT AO PFA      PFA      PFA    OWT SO
```

Figure 2-69 Viewing the ULOG

2.60 SDSF: Input queue panel



The figure shows two SDSF Input Queue panels side-by-side. The left panel is for JES2 and the right panel is for JES3. Both panels have a header with menu options: Display, Filter, View, Print, Options, Search, Help. Below the header is a command input line. The JES2 panel has a scroll field labeled 'SCROLL ===> CSR'. The JES3 panel has a scroll field labeled 'SCROLL ===> HALF'. The main display area shows job information. In the JES2 panel, the jobs listed are ZFSJOB (JobID: JOB13563, Owner: ROGERS) and LUTZCP (JobID: JOB03661, Owner: LUTZ). In the JES3 panel, the jobs listed are BECKER1 (JobID: JOB04066, Owner: BECKER) and MDSWAIT (JobID: JOB14437, Owner: VAINI). The bottom of each panel has a footer with the same menu options as the top.

NP	JOBNAME	JobID	Owner	Prtv	C	Pos	PrtDest	Rmt	Node	Saf
ZFSJOB	ZFSJOB	JOB13563	ROGERS	10	A	LOCAL			1	
LUTZCP	LUTZCP	JOB03661	LUTZ	10	*	LOCAL			1	SC6

NP	JOBNAME	JobID	Owner	Status	Prtv	C	Pos	PrtDest	
BECKER1	BECKER1	JOB04066	BECKER		1	A		ANYLOCAL	
MDSWAIT	MDSWAIT	JOB14437	VAINI		1	A		ANYLOCAL	

Figure 2-70 SDSF: Input queue panel

SDSF: Input queue panel

The Input queue panel provides information about jobs, TSO users, and started tasks that are on the JES input queue or are being processed by the system. Users display this information by entering I on the command input line of any SDSF panel. An example of the panels for JES2 and JES3 is shown in Figure 2-70.

Using the Input queue command

When you specify I with no parameters, the command displays the input queue for all jobs in the job classes, and in the converter queue. You must use special characters to display TSO users and started tasks.

You can display the following information using these parameters:

input-class The input class can be a list of up to seven input job classes that displays only jobs with those classes. There is no blank between I and an input class or between input classes.

H This parameter displays only jobs that are held.

NH This parameter displays only jobs that are not held.

Remember: Figure 2-70 shows only a subset of the fields of information that are available on the Input Queue panel. You can scroll right, left, up, and down throughout this panel. Also you can control the scroll using the SCROLL ==> field. You can use, for example, C (CSR), to better control the scroll.

Specifying multiple input classes

Enter the list of up to seven input job classes without a space after the **I** command. Only jobs in these input classes are displayed.

To display tasks and TSO users, you must use special characters. They are as follows:

Asterisk (*)	Converter queue
Dollar sign (\$)	TSO users
Number sign (#)	Started tasks
At sign (@)	Jobs waiting to be transmitted to another node for execution
Exclamation mark (!)	Hardcopy queue

Tip: The hardcopy queue contains all jobs that have any type of output in the system. Accessing the hardcopy queue by using the **I** command enables you to find output for a job, whether it is on a held or non-held JES2 output queue. You can also use the hardcopy queue to display output that has been printed but that remains in the JES spool.

Input command examples

The following list includes examples of the **I** command:

- ▶ COMMAND INPUT ===> **i**
This command displays the input queue for all jobs in the all classes and the converter queue.
- ▶ COMMAND INPUT ===> **iak nh**
This command displays the input queue for jobs in classes A and K that are not held.
- ▶ COMMAND INPUT ===> **i\$**
This command displays the input queue for all TSO users.
- ▶ COMMAND INPUT ===> **iabc**
This command displays the input queue for jobs in classes A, B, and C that are held and not held.
- ▶ COMMAND INPUT ===> **ia\$#**
This command displays jobs in class A, TSO users, and started tasks.

2.61 SDSF: Output queue panel

Display Filter View Print Options Search Help						
SDSF OUTPUT ALL CLASSES ALL FORMS					LINE 1-281,964	LINE 1-19 (704)
COMMAND INPUT ==>					SCROLL ==> HALF	
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=SC74 ACTION=// -Block,--Repeat,+-Extend,-?JDS,A-Release,C-Cancel,H-Hold,L-List, ACTION=LL-ListLong,P-Purge,Q-OutDesc,S-Browse,SB-ISPFBrowse,SE-ISPFEdit, ACTION=SJ-JCLEDit,X-Print,XC-PrintClose,XD-PrintDS,XDC-PrintDSClose, ACTION=XF-PrintFile,XFC-PrintFileClose,XS-PrintSysout,XSC-PrintSysoutClose						
NP	JOBNAME	JobID	Owner	Prty C Forms	Dest	Tot-Rec
	DAVISRD	TSU00425	DAVISRD	144 A STD	LOCAL	64
	DAVISRD1	JOB00887	DAVISRD	144 A STD	LOCAL	458
	DAVISRD1	JOB00888	DAVISRD	144 A STD	LOCAL	430

Display Filter View Print Options Search Help						
SDSF OUTPUT ALL CLASSES ALL FORMS					Lines 6,891,756	Cursor not on choice
COMMAND INPUT ==>					SCROLL ==> HALF	
PREFIX=* DEST=(ALL) OWNER=* SORT=JobID/A SYSNAME=*						
ACTION=// -Block,--Repeat,+-Extend,-?JDS,Q-OutDesc,S-Browse,SB-ISPFBrowse, ACTION=SE-ISPFEdit,SJ-JCLEDit,X-Print,XC-PrintClose,XD-PrintDS, ACTION=XDC-PrintDSClose,XF-PrintFile,XFC-PrintFileClose,XS-PrintSysout, ACTION=XSC-PrintSysoutClose						
NP	JOBNAME	JobID	Owner	Prty C Forms	Dest	Tot-Rec T
	PELEGDSF	JOB17189	PELEG	1 A STD	ANYLOCAL	52
	PELEGDSF	JOB17190	PELEG	1 A STD	ANYLOCAL	55
	PELEGSP3	JOB17191	PELEG	1 A STD	ANYLOCAL	32
	PELEGSP3	JOB17192	PELEG	1 A STD	ANYLOCAL	32

Figure 2-71 SDSF: Output queue panels for JES2 and JES3

SDSF: Output queue panel

The SDSF **0** command displays the *non-held* output queue. The Output Queue panel provides information about output data sets for jobs, TSO users, and started tasks that are on a JES2 and JES3 output queue. Users display this information by entering **0** on the command input line of any SDSF panel. Examples of these panels are shown in Figure 2-71.

When you specify the **0** command with no parameters, it displays information about all jobs, started tasks, and TSO users on any non-held output queue.

output-class The output-class can be a list of up to seven output classes. Specify the output class without a blank immediately after the **0** command. Only data sets in those non-held output classes are displayed. For example, if you want to display sysout class e, specify **o(e)**.

Selecting output for display

You might want to use the **PREFIX** command to limit the entries in the display, as shown in the following list:

- ▶ **PREFIX**, with no parameter, displays all jobs for which you are authorized.
- ▶ **PREFIX MQ*** displays all jobs that start with the name MQ.
- ▶ **PREFIX M%%R*** displays all jobs that begin with M and have R in the fourth position.

Using the prefix command

When you first enter SDSF, the default is your logon ID prefix. If SDSF is not displaying what you expect, then issue the **SET DISPLAY ON** command. This controls the display of values you have set for PREFIX, DEST, OWNER, SORT, and FILTER. It displays as shown in Figure 2-71 on page 163 and as specified in the following command:

```
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=SC74
```

SDSF displays using a prefix command

The output display shown in Figure 2-72 used the **prefix ROG*** command.

Display Filter View Print Options Search Help							
<hr/>							
SDSF OUTPUT ALL CLASSES ALL FORMS							
COMMAND INPUT ==>							
PREFIX=ROG*	DEST=(ALL)	OWNER=*	SORT=JobID/A	SYSNAME=*			SCROLL ==> HALF
ACTION//=Block,=-Repeat,+=Extend,?-JDS,Q-OutDesc,S-Browse,SB-ISPFBrowse,							
ACTION=SE-ISPFEdit,SJ-JCLEdit,X-Print,JCLEdit,XC-PrintClose,XD-PrintDS,							
ACTION=XDC-PrintDSClose,XF-PrintFile,XFC-PrintFileClose,XS-PrintSysout,							
ACTION=XSC-PrintSysoutClose							
NP	JOBNAME	JobID	Owner	Prty C Forms	Dest		Tot-Rec
ROGERS	JOB20171	ROGERS		15 E STD	ANYLOCAL		2,432
ROGERS	JOB20174	ROGERS		15 E STD	ANYLOCAL		2,859
ROGERS	JOB20380	ROGERS		15 E STD	ANYLOCAL		413
ROGERS	JOB20402	ROGERS		15 A STD	ANYLOCAL		128
ROGERS	JOB20402	ROGERS		15 E STD	ANYLOCAL		7,823
ROGERS	JOB20792	ROGERS		15 E STD	ANYLOCAL		2,925
ROGERS	JOB20873	ROGERS		15 E STD	ANYLOCAL		442

Figure 2-72 Using a prefix of ROG*

Using a prefix of DB2* displays as shown here:

NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP	Real	Paging	SI0
DB2GDBM1	DB2GDBM1	IEFPROC	STC04424	STC		NS	FE	9826	0.00	0.00	
DB2GDIST	DB2GDIST	IEFPROC	STC04425	STC		NS	FE	2313	0.00	0.00	
DB2GIRLM	DB2GIRLM		STC04423	STC		NS	FE	284	0.00	0.00	
DB2GMSTR	DB2GMSTR	IEFPROC	STC04422	STC		NS	FE	1887	0.00	0.00	
DB2GSPAS	DB2GSPAS	IEFPROC	STC04426	STC		NS	FB	982	0.00	0.00	

2.62 SDSF: Held Output queue panel

Display Filter View Print Options Search Help						
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 27					LINE 1-1 (1)	
COMMAND INPUT ==>					SCROLL ==> CSR	
PREFIX=COMPRESS DEST=(ALL) OWNER=* SYSNAME=* ACTION//=-Block,--Repeat,+-Extend,?-JDS,A-Release,C-Cancel,H-Hold,L-List, ACTION=LL-ListLong,O-Release,OK-ReleaseKeep,P-Purge,Q-OutDesc,S-Browse, ACTION=SB-ISPFBrowse,SE-ISPFEdit,SJ-JCLEdit,X-Print,XC-PrintClose,XD-PrintDS, ACTION=XDC-PrintDSClose,XF-PrintFile,XFC-PrintFileClose,XS-PrintSysout, ACTION=XSC-PrintSysoutClose						
NP	JOBNAME	JobID	Owner	Prty C ODisp Dest	Tot-Rec	Tot-
				144 T HOLD LOCAL	27	
<hr/>						
Display Filter View Print Options Search Help						
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 6,678					Cursor not on choice	
COMMAND INPUT ==>					SCROLL ==> HALF	
PREFIX= DEST=(ALL) OWNER=* SYSNAME=* ACTION//=-Block,--Repeat,+-Extend,?-JDS,Q-OutDesc,S-Browse,SB-ISPFBrowse, ACTION=SE-ISPFEdit,SJ-JCLEdit,X-Print,XC-PrintClose,XD-PrintDS, ACTION=XDC-PrintDSClose,XF-PrintFile,XFC-PrintFileClose,XS-PrintSysout, ACTION=XSC-PrintSysoutClose						
NP	JOBNAME	JobID	Owner	Prty C Dest	Tot-Rec	Tot-Page F
ROGERSW	JOB21266	ROGERS		1 T ANYLOCAL	6,678	S

Figure 2-73 SDSF: Held Output queue panel

SDSF: Held Output queue panel

The Held Output queue panel provides information about output that is being held on any JES2 output queue. This display shows you information about SYSOUT data sets for jobs, started tasks, and TSO users on any held JES2 output queue. Users display this information by entering **H** on the command input line of any SDSF panel. An example of this panel is shown in Figure 2-73. This figure shows only a subset of the information available on the Held Output queue panel. From this panel, a user can look at the output for a specific job and then decide to print or purge the output.

Selecting held output

When you specify **H** with no parameters, the command displays all jobs with names that are prefixed with your user ID. The held output class that you want to display can be specified using the **H** command as follows:

output-class You can request held output classes using the **H** command by specifying a list of up to seven output classes. **H** limits the panel to jobs with those classes. There is no blank between **H** and an output class or between output classes.

ALL Specifying the subparameter **ALL** displays all held output classes for all jobs. Issuing the command **H ALL** displays all held output classes.

string-pattern This subparameter is a character string that limits the panel to jobs with names that match that character string. The string-pattern might include the special pattern matching characters, where the string pattern is **HAI***. The command is **H HAI*** and is shown here:

NP	JOBNAME	JobID	Owner	Prty C ODisp Dest	Tot-Rec	Tot-
	HAIMO	TSU04529	HAIMO	144 S HOLD LOCAL	371	
	HAIMO	TSU04544	HAIMO	144 S HOLD LOCAL	299	

The following example shows the string-pattern is DSS* and the command is H DSS*.

NP	JOBNAME	JobID	Owner	Prty	C	ODisp	Dest	Tot-Rec	Tot-
	DSSMOVE	JOB03008	HAIMO	144	T	HOLD	LOCAL	98	
	DSSMOVE	JOB03009	HAIMO	144	T	HOLD	LOCAL	98	
	DSSMOVE	JOB03010	HAIMO	144	T	HOLD	LOCAL	97	
	DSSMOVE	JOB07986	HAIMO	144	T	HOLD	LOCAL	79	
	DSSCOPY1	JOB04310	HAIMO	144	T	HOLD	LOCAL	71	
	DSSREST1	JOB04312	HAIMO	144	T	HOLD	LOCAL	73	
	DSSCOPY1	JOB04317	HAIMO	144	T	HOLD	LOCAL	67	
	DSSREST1	JOB04319	HAIMO	144	T	HOLD	LOCAL	73	

2.63 SDSF: Status panel

JES2

NP	Jobname	JobID	Owner	Prty	Queue	C	Pos	SAff	ASys	Status
DYMOKELA	JOB05117	DYMOKEL		8	EXECUTION	A				HOLD
LAFITTE	TSU04622	LAFITTE		15	EXECUTION		SC74	SC74		
RODOLFI	TSU04628	RODOLFI		15	EXECUTION		SC74	SC74		
KOZAKOS	TSU04634	KOZAKOS		15	EXECUTION		SC74	SC74		
DIANAG	TSU04635	DIANAG		15	EXECUTION		SC74	SC74		
ROGERS	TSU04640	ROGERS		15	EXECUTION		SC74	SC74		
HWIBCPPII	STC04566	IBMUSER		15	EXECUTION		SC74	SC74		
SYSLOG	STC04567	+MASTER+		15	EXECUTION		SC74	SC74		
RMF	STC04570	RMF		15	EXECUTION		SC74	SC74		
SDSF	STC04574	IBMUSER		15	EXECUTION		SC74	SC74	ARMELEM	

JES3

NP	Jobname	JobID	Owner	Prty	Queue	C	Pos	SAff	ASys	S	SC75
NJECONS	JOB00497			15	EXECUTION					S	SC75
NJERDR	JOB00498			15	EXECUTION					S	SC75
SNARJP	JOB06905			15	EXECUTION					S	SC75
MONITOR	JOB13803	VJUHA		15	EXECUTION					S	SC75
INTRDR	JOB20589			15	EXECUTION					S	SC75
TCP	JOB20820	VAINI		15	EXECUTION					S	SC75
CR	JOB21023	VJUHA		15	EXECUTION					S	SC75
DC	JOB21193	VAINI		15	EXECUTION					S	SC75
ASCHINT	JOB21282	IBMUSER		15	EXECUTION	A				S	SC75

Figure 2-74 SDSF: Status panel

SDSF: Status panel

The Status panel provides information about jobs, started tasks, and TSO users that are on any of the JES2 queues. Users display this information by entering **st** on the command input line of any SDSF panel. An example of this panel is shown in Figure 2-74. This figure shows only a subset of the fields of information available on the Status panel.

Selecting the Status panel

When you specify **ST**, you can process a job from this panel even if it has been printed or processed (and not yet purged). Active jobs are highlighted on the panel.

The format of the command is shown in the following statement:

ST(classes) (string)

In this statement, define the variables as shown in the following list:

classes *Classes* displays information for a specific class.

- ▶ With JES2: Enter up to six classes with no blanks. For jobs in execution, use A-Z or 0-9.
- ▶ With JES3: Enter a single class, up to 6 characters. To filter the panel using more than one class, use the **FILTER** command.

Note that you can use these special characters for class (JES2 and JES3):

Asterisk (*)	Converter queue
Dollar sign (\$)	TSO users in execution
Number sign (#)	Started tasks in execution
Exclamation mark (!)	Hard-copy queue
Plus sign (+)	Output queue
Minus sign (-)	Input queue
Question mark (?)	Purge queue
Left parenthesis ()	Receiver queue
Equals sign (=)	Spin queue
Forward slash (/)	Setup queue
At sign (@)	Jobs waiting to be transmitted to another queue
string	<i>String</i> is a character string that limits the panel to jobs with names that match the character string. The string can be up to eight characters in length, including the following special characters:
Asterisk (*)	Represents any character or string of characters
Percentage sign (%)	Represents any single character

Important: ST with no parameters displays all jobs. The information displayed might be limited by your authorization and by settings for SDSF filters, such as FILTER and PREFIX.

JES2 and JES3 action characters for the ST panel

The following figures show the action characters for the JES panels. Figure 2-75 displays the JES2 action characters for the ST panel.

```
ACTION=// -Block,=-Repeat,+-Extend,?-JDS,A-Release,C-Cancel,CA-CancelARM,  
ACTION=CD-CancelDump,CDA-CancelARDump,D-Display,DL-DisplayLong,E-Restart,  
ACTION=EC-RestartCancel,ES-RestartStep,ESH-RestartStepHold,H-Hold,I-Info,  
ACTION=J-Start,L-List,LL-ListLong,O-Release,P-Purge,P0-PurgeOutput,  
ACTION=PP-PurgeProtected,Q-OutDesc,S-Browse,SB-ISPFBrowse,SE-ISPFEdit,  
ACTION=SJ-JCLEdit,W-Spin,X-Print,XC-PrintClose,XD-PrintDS,XDC-PrintDSClose,  
ACTION=XF-PrintFile,XFC-PrintFileClose,XS-PrintSysout,XSC-PrintSysoutClose
```

Figure 2-75 JES2 action characters for the ST panel

Figure 2-76 displays the JES3 action characters for the ST panel.

```
ACTION=// -Block,=-Repeat,+-Extend,?-JDS,A-Release,C-Cancel,CA-CancelARM,  
ACTION=CD-CancelDump,CDA-CancelARDump,CP-CancelPrint,D-Display,  
ACTION=DE-DisplayEstimates,DL-DisplayLong,DM-DisplayMains,DMA-DisplayMDSAalloc,  
ACTION=DME-DisplayMDSError,DMR-DisplayMDSRestart,DMSS-DisplayMDSSysSel,  
ACTION=DMSV-DisplayMDSSysVer,DMU-DisplayUnavailVol,DSD-DisplayDDnames,  
ACTION=DSH-DisplaySpoolHold,DSP-DisplaySpoolPartition,DX-DisplayExtended,  
ACTION=E-Restart,H-Hold,I-Info,J-Start,L-List,LB-ListBDT,LH-ListHold,  
ACTION=LT-ListTCP,P-Purge,Q-OutDesc,S-Browse,SB-ISPFBrowse,SE-ISPFEdit,  
ACTION=SJ-JCLEdit,W-Spin,X-Print,XC-PrintClose,XD-PrintDS,XDC-PrintDSClose,  
ACTION=XF-PrintFile,XFC-PrintFileClose,XS-PrintSysout,XSC-PrintSysoutClose
```

Figure 2-76 JES3 action characters for the ST panel

2.64 SDSF HELP panel

HELP: SDSF -- Table of Contents		
COMMAND INPUT ==>		
Select a topic by number, or press Enter to view topics in sequence.		
1 - What's new	13 - Action bar	
2 - Job and workload panels	14 - PF keys and keylists	
3 - Device and resource panels	15 - Action characters	
4 - Search and scroll commands	16 - Overtypable fields	
5 - Filter commands	17 - JES and MVS commands	
6 - View commands	18 - Online library	
7 - Print command and actions	19 - SDSF messages	
8 - Options commands	REXXHelp - SDSF REXX (ISPF only)	
9 - Other commands		
10 - Server commands		
11 - SDSF command		
12 - Help commands		
SEARCH - Search the help (ISPF only)		
F1 = Help	F10 = Previous	Enter = Forward
F3 = Exit	F7 = Up	TOC = Menu

Figure 2-77 SDSF HELP panel

SDSF HELP panel

You can use the SDSF HELP panel to become familiar with all of the SDSF functions. This panel is displayed when you type the command `help` on the command line. The tutorial shows you the basic functions of SDSF. These panels will help you learn how SDSF works.

The help command

The `help` command or the PF1 key provides information about how to use SDSF or how to use each panel. The following list describes reasons for the help screens and PF keys:

- ▶ Purpose. Displays online help for SDSF.
- ▶ Where used. Any SDSF panel, including help and tutorial panels.
- ▶ Format. HELP.
 - Press Enter to see the next help panel in a sequence.
- ▶ You can also use the following options:
 - SEARCH. Search the help (ISPF only). Enter outside of help.
 - I or F11. Display an index to help (ISPF only).
 - TOC. Display the table of contents for help.
 - TUTOR or F4. Display an interactive tutorial (ISPF only).
- ▶ Several PF keys have special uses in help:
 - F4 (Tutor). Display an interactive tutorial (ISPF only).
 - F5 (Extended help). Display the general help for the topic (ISPF only).
 - F7 (Up). Display the start of the current topic or, if you are already at the start, display the previous menu.
 - F10 (Previous). Back up one help panel.
 - F11 (Index). Display the help index (ISPF only).

Information: Under ISPF, you can obtain more information about a highlighted phrase by placing the cursor on the phrase and pressing Enter.



The z/OS system programmer

Because the base elements and optional features of IBM z/OS are integrated into a single package with compatible service levels, you must install, with few exceptions, the entire z/OS product. You can install z/OS using one of several IBM packages.

This chapter presents the available z/OS delivery options to install z/OS, as listed here:

- ▶ IBM z/OS ServerPac (ServerPac)
- ▶ Custom-Built Product Delivery Option (CBPDO)
- ▶ IBM SystemPac
- ▶ Entry Server Offering

In addition, this chapter briefly describes the download process and installation steps for using the ServerPac installation option.

Attention: When you order any of the installation packages, you receive a comprehensive installation guide that details the installation tasks step-by-step from the beginning of the installation until you start your system.

For example, if you choose the ServerPac installation package, you receive *ServerPac: Installing Your Order* documentation that is tailored to your order for installation. This document is unique to your environment and is based on what you have ordered.

In a mainframe IT installation, system programmers are responsible for installing, customizing, and maintaining the z/OS operating system (OS). Their responsibilities also include installing or upgrading all additional products that run on the system, including middleware products. *Middleware* is a software layer between the OS and the user, or user application. Middleware supplies major functions that are not provided by the operating system.

This chapter describes the tasks involved with the install process.

3.1 z/OS Operating System: The role of a system programmer

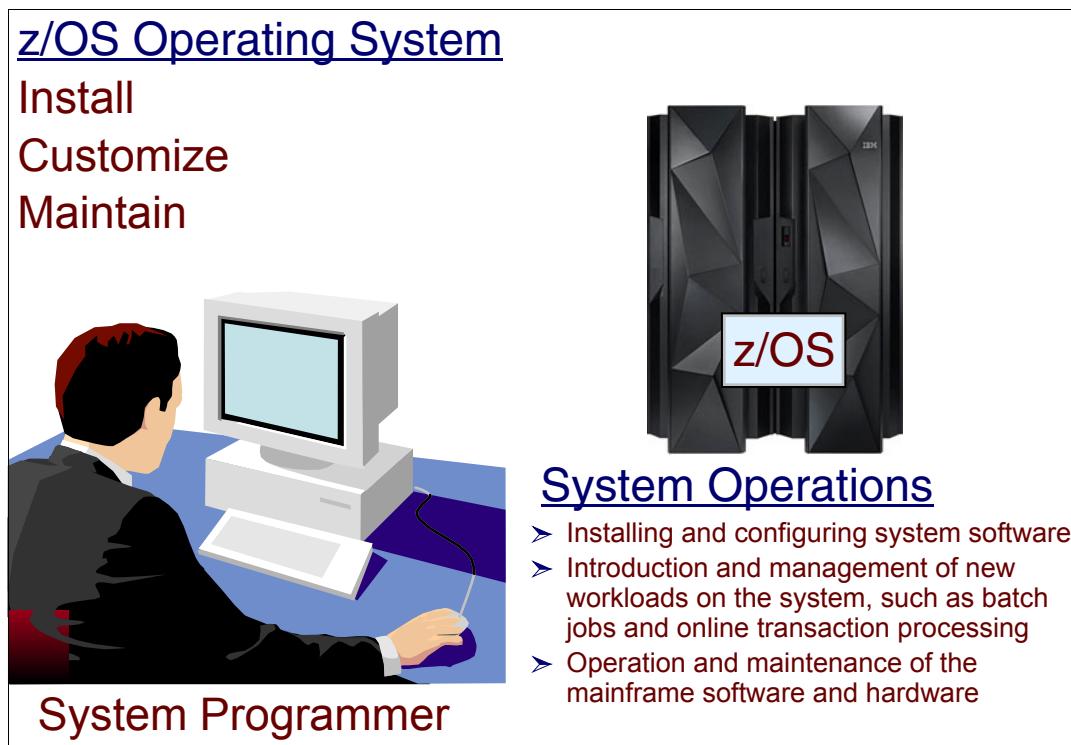


Figure 3-1 z/OS Operating System: The role of a system programmer

The role of a system programmer

As a system programmer, and to meet the specific requirements of your installation, you can customize z/OS functions and interfaces to take advantage of new functions after installation. The role of the system programmer is to install, customize, and maintain the operating system. z/OS runs on various hardware configurations. A system programmer must define the hardware input/output (I/O) configuration resources that are to be available to z/OS. The hardware used can be either IBM or other manufacturer machines. As a z/OS system programmer, you must be aware of the following items:

- ▶ Storage concepts
- ▶ Virtual storage and address spaces concepts
- ▶ Device I/O configurations
- ▶ Processor configurations
- ▶ Console definitions
- ▶ System libraries where the software is placed
- ▶ System data sets and their placement
- ▶ Customization parameters that are used to define a z/OS configuration
- ▶ Execution of the performance analysis task through the use of performance monitors, such as the z/OS Resource Management Facility (RMF)

System operations

Installing and configuring system software is a task that occurs frequently. For the various products installed, different system programmers might be assigned to different products and subsystems. Introduction and management of new workloads on the system, such as batch jobs and online transaction processing, might also involve individual system programmers being assigned to support the various types of workloads that run in the complex.

System programmers must be skilled at debugging problems with system software. These problems are often captured in a copy of the computer's memory contents called a *dump*, which the system produces in response to a failing software product, user job, or transaction.

Using the dump and specialized debugging tools, the system programmer can determine where the components have failed. When the error occurs in a software product, the system programmer works directly with the software vendor to report the problem and wait for a potential fix to install.

z/OS operating system

Several IBM packages are available for installing z/OS. Some are entitled with the product (as part of your z/OS license, at no additional charge). Others are available for an additional fee.

Important: Because the base elements and optional features are integrated into a single package with compatible service levels, you must install, with few exceptions, the entire z/OS product.

Installing z/OS

One of the following methods can be used to install z/OS:

- ▶ ServerPac (entitled with z/OS)

ServerPac is an entitled software delivery package consisting of products and service for which IBM has performed the System Modification Program/Extended (SMP/E) installation steps and some of the post-SMP/E installation steps. To install the package on your system and complete the installation of the software that it includes, you use the IBM CustomPac Installation dialog.

The CustomPac Installation dialog generates tailored installation jobs and saves detailed definitions of volume, catalog, and data set configurations, which can be tailored, saved, and merged to install subsequent ServerPacs. The CustomPac Installation dialog is the same dialog that is used for all of the CustomPac offerings, including SystemPac (dump-by-data-set format), IBM ProductPac®, and IBM RefreshPac.

- ▶ CBPDO (entitled with z/OS)

Custom-Built Product Delivery Option (CBPDO) is an entitled software delivery package consisting of uninstalled products and unintegrated service. There is no dialog program to help you install, as there is with ServerPac. You must use SMP/E to install the individual z/OS elements and features, and their service, before you can perform an initial program load (IPL). Installation instructions are in the publication z/OS Program Directory.

- ▶ SystemPac (additional charge with z/OS)

SystemPac is a software package, available for an additional fee and offered worldwide, that helps you install z/OS and subsystems (IBM DB2, IBM IMS, IBM Customer Information Control System (IBM CICS), Network Control Program (NCP), and IBM WebSphere Application Server). SystemPac is tailored to your specifications; it is manufactured according to parameters and I/O definition file (IODF) definitions that you supply during order entry.

The goal is to have the system tailored to your specifications, and to have products enabled according to your specified configuration. Parameters are collected by telephone. Using a printed questionnaire as a guide, you tell an IBM representative your responses. Upon completion, a printout showing all of the parameters and definitions that you specified is sent to you for reference.

3.2 z/OS system programmer management overview

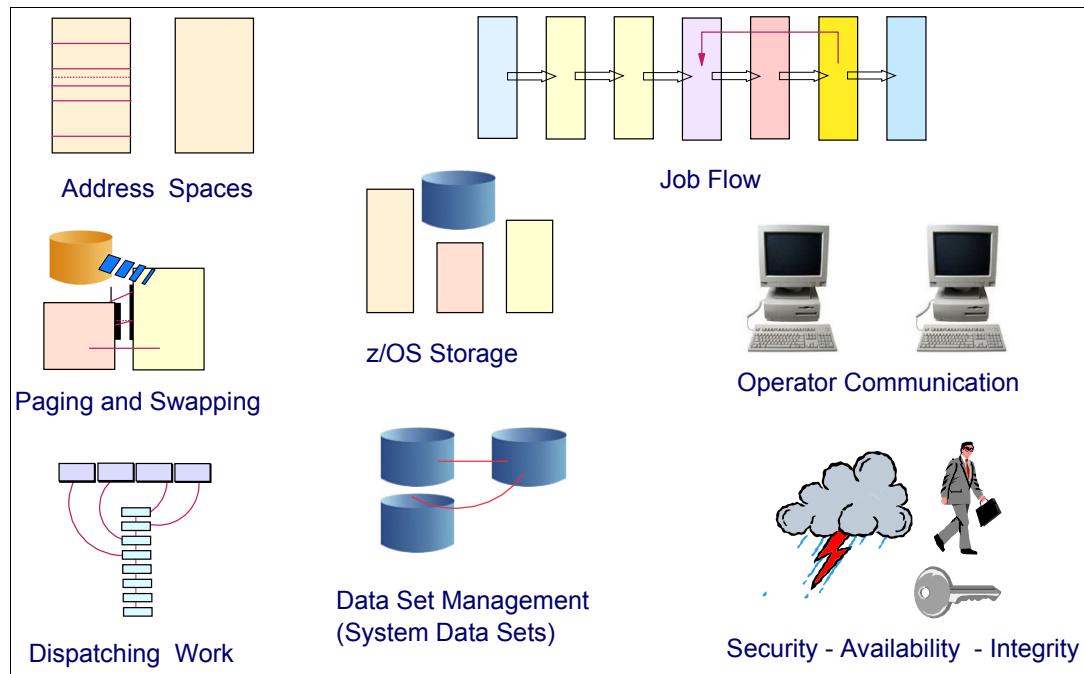


Figure 3-2 z/OS system programmer management overview

z/OS system programmer management overview

As a z/OS system programmer, you need to be involved in the customization of the items illustrated in Figure 3-2. These items are explained in the following list:

- ▶ **Address spaces.** When z/OS is started, z/OS establishes system component address spaces. During the IPL, the first address space started is the master scheduler address space (*MASTER*). There are other system address spaces for various subsystems and system components.
- ▶ **Paging.** Page data sets contain the paged-out portions of address spaces, the common service area (CSA), pageable link pack area (PLPA), and the data written to virtual I/O (VIO) data sets.
- ▶ **Dispatching work.** The scheduling of address spaces as dispatchable units to run on a central processor (CP) in the z/OS system is done by the z/OS dispatcher component. The z/OS dispatcher is responsible for finding and dispatching (run on a CP) the highest priority dispatchable unit in the system (service request blocks (SRBs) and tasks).
- ▶ **Job flow.** z/OS uses IBM Job Entry Subsystem (JES) to receive jobs (also called *batch*, which is a non-interactive type of transaction) into the OS, to schedule them for processing by z/OS, and to control their output processing. JES is the component of the OS that provides supplementary job management and data management.
JES also provides supplementary task management functions, such as scheduling, control of job flow, and spooling (storing output on direct access storage device (DASD) spool volumes rather than printing them at the moment they are produced).
- ▶ **z/OS storage.** The system programmer must be aware of all storage considerations when installing and customizing a z/OS environment. The initialization process begins when the system operator selects the LOAD (IPL) function at the system console. z/OS locates all of the usable main storage that is online and available to the system, and creates a virtual environment for the building of various system areas.

This initialization phase allocates the system's minimum virtual storage for the system queue area (SQA) and the extended SQA, allocates virtual storage for the extended local system queue area (extended LSQA) for the master scheduler address space, and allocates virtual storage for the CSA and the extended CSA. The amount of storage allocated depends on the values specified on the CSA system parameter read during the IPL.

- ▶ System data sets. Each installation must incorporate required system data sets into the system by allocating space for them on appropriate direct access devices (DASD) during system installation. The DEFINE function of Access Method Services is used to define both the space requirements and the volume for each system data set. Some data sets must be allocated on the system residence volume (the volume that has the kernel of z/OS code). Other data sets can be placed on other direct access volumes.
- ▶ Operator communication. The operation of a z/OS system involves the following elements:
 - Console operations, or how operators and system programmers interact instantaneously with z/OS to monitor or control the hardware and software.
 - Message (produced by z/OS) processing and command (produced by an operator) processing that forms the basis of operator interaction with z/OS, and the basis of z/OS automation.
 - Managing hardware, such as processors and peripheral devices (including the consoles where operators or system programmers do their work), and software, such as the z/OS operating control system, JES, subsystems (such as IBM NetView®) that can control automated operations, and all of the applications that run on z/OS.
- ▶ Security. Data security is the protection of data against unauthorized disclosure, transfer, modification, or destruction, whether accidental or intentional. A security system, such as IBM Resource Access Control Facility (RACF), must be installed in your OS by a system programmer to maintain the resources necessary to meet the security objectives. The system programmer has the overall responsibility, using the technology available, to transform the objectives of the security policy into a usable plan.
- ▶ Availability. The software products supporting system programmers and operators in managing their systems heavily influence the complexity of their job and their ability to keep system availability at a high level. Performance management is the system management discipline that most directly impacts all users of system resources in an enterprise and can be achieved, for example, by using RMF.
- ▶ Integrity. An OS is said to have system integrity when it is designed, implemented, and maintained to protect itself against unauthorized access, and does so to the extent that security controls specified for that system cannot be compromised. Specifically for z/OS, there must be no way for any unauthorized program, using any system interface, defined or undefined, to perform the following actions:
 - Bypass store or fetch protection.
 - Bypass password use, Virtual Storage Access Method (VSAM) password, or RACF security checking.
 - Obtain control in an authorized state.

3.3 The system programmer and z/OS operations

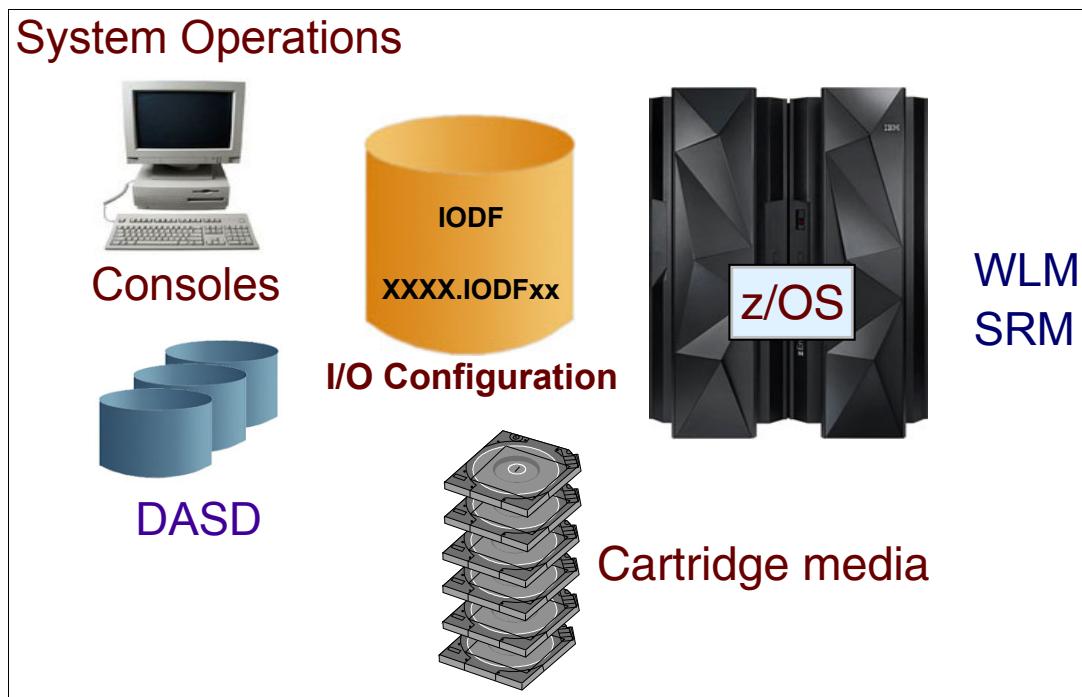


Figure 3-3 The system programmer and z/OS operations

The system programmer and z/OS operations

A system programmer has to plan the following operations areas:

- ▶ **z/OS Workload Manager**

z/OS Workload Manager (WLM) provides a solution for managing workload distribution, workload balancing, and distributing resources to competing workloads. Managing workloads is possible due to the combined cooperation of various subsystems (CICS, IBM IMS/ESA, JES, Advanced Program-to-Program Communication (APPC), Time Sharing Option Extensions (TSO/E), z/OS UNIX System Services, distributed data facility (DDF), DB2, System Object Model (SOM), LAN Server for MVS (LSFM), and Internet Connection Server) with the WLM component.

- ▶ **System performance**

The task of tuning a system is an iterative and continuous process. The controls offered by system resources manager (SRM) are only one aspect of this process. Initial tuning consists of selecting appropriate parameters for various system components and subsystems. After the system is operational and criteria have been established for the selection of jobs for execution by job classes and priorities, SRM will control the distribution of available resources according to the parameters specified by the installation.

WLM, however, can only deal with available resources. If these are inadequate to meet the needs of the installation, even optimal distribution might not be the answer. Other areas of the system should be examined to determine the possibility of increasing available resources.

When requirements for the system increase and it becomes necessary to shift priorities or acquire additional resources, such as a larger processor, more storage, or more terminals, the WLM goals might have to be adjusted to reflect changed conditions.

- ▶ I/O device configuration

As a system programmer, you must define an I/O configuration to the operating system (software) and the channel subsystem (hardware). The Hardware Configuration Definition (HCD) component of z/OS consolidates the hardware and software I/O configuration processes under a single interactive user interface. The validation checking that HCD does as you enter data helps to eliminate errors before you attempt to use the I/O configuration.

The output of HCD is an IODF, which contains the server, the logical partitions and the I/O configuration data. An IODF is used to define multiple hardware (servers) and software configurations to z/OS. When you activate an IODF, HCD defines the I/O configuration to the channel subsystem or the operating system.

With the HCD activate function or the z/OS **ACTIVATE** operator command, you can make changes to the current configuration without having to restart the software or perform a Power-on Reset (POR) of the hardware. Making changes while the system is running is known as *dynamic configuration* or *dynamic reconfiguration*.

- ▶ Console operations

The operation of a z/OS system involves the following elements:

- Console operations, or how operators interact with z/OS to monitor or control the hardware and software
- Message and command processing that forms the basis of operator interaction with z/OS and the basis of z/OS automation

Operating z/OS involves managing hardware, such as processors and peripheral devices (including the consoles where your operators do their work), and software, such as the z/OS operating control system, the job entry subsystem, subsystems (such as IBM Tivoli NetView for z/OS) that can control automated operations, and all of the applications that run on z/OS.

Planning z/OS operations for a system must take into account how operators use consoles to accomplish their work and how you want to manage messages and commands.

Because messages are also the basis of automated operations, understanding message processing in an z/OS system can help you plan z/OS automation.

System operations

Also involved are the business goals and policies established to enable the installation to grow and handle work efficiently. These needs, of course, vary from installation to installation, but they are important when you plan your z/OS operations.

Managing the complexity of z/OS requires you to think about the particular needs of your installation. However, installations can consider the following goals when planning z/OS operations.

- ▶ Increasing system availability

Many installations need to ensure that their system and its services are available and operating to meet service level agreements (SLAs). Installations with 24-hour, 7-day operations need to plan for minimal disruption of their operation activities. In terms of z/OS operations, how the installation establishes console recovery or whether an operator must restart a system to change processing options are important planning considerations.

- ▶ Controlling operating activities and functions

As more installations make use of multisystem environments (as in IBM Parallel Sysplex), the need to coordinate the operating activities of those systems becomes crucial. Even for single z/OS systems, you must consider controlling communication between functional areas (such as a tape-pool library and the master console area, for example).

In both single and multisystem environments, the commands that operators can issue from consoles can be a security concern that requires careful coordination. As a planner, ensure that the correct people are performing the correct tasks when they interact with z/OS. If your installation uses remote operations to control target systems, you also need to consider how to control those activities from the host system.

- ▶ Simplifying operator tasks

Because the complexity of operating z/OS has increased, the tasks and skills of operators also require careful consideration. How operators respond to messages at their consoles and how you can reduce or simplify their actions are important to operations planning. Further, planning z/OS operator tasks in relation to any automated operations that help simplify those tasks is also needed.

- ▶ Streamlining message flow and command processing

In thinking about operator tasks, consider how to manage messages and commands. Operators need to respond to messages. Routing messages to operator consoles, suppressing messages to help your operators manage increased message traffic, and selecting messages for automated operations can all help you manage system activity efficiently.

- ▶ Single system image

Single system image enables the operator, for certain tasks, to interact with several images of a product as though they were one image. For example, the operator can issue a single command to all z/OS systems in the sysplex rather than repeating the command for each system.

- ▶ Single point of control

Single point of control enables the operator to interact with a suite of products from a single workstation. An operator can accomplish a set of tasks from a single workstation, thereby reducing the number of consoles that the operator has to manage.

3.4 Requirements for z/OS installation

- TSO/E and ISPF/PDF
 - > Batch job JCL
- Storage concepts
- Device I/O configurations
- Processor configurations
- Console definitions
- System libraries management
- DASD space management
- Customization parameters - SYS1.PARMLIB
- Data set placement

Figure 3-4 Requirements for z/OS installation

Requirements for z/OS installation

To be able to install and customize a z/OS operating system, a system programmer has to possess certain basic skills and functions. Figure 3-4 lists some of the areas that a programmer needs to be familiar with, and these areas are explained here:

TSO/E	TSO/E is an option of the z/OS Operating System that enables users to interactively share computer time and resources. TSO/E is an integral part of z/OS, and serves as a platform for other elements, such as IBM BookManager READ, HCD, and Interactive System Productivity Facility (ISPF)/Program Development Facility (PDF). See “TSO/E” on page 65 for a complete description of TSO/E.
ISPF/PDF	ISPF/PDF work under TSO/E to provide panels with which users can interact. ISPF provides the underlying dialog management service that displays panels, and enables a user to navigate through the panels. PDF is a dialog of ISPF that helps maintain libraries of information in TSO/E and enables a user to manage the library through facilities, such as browse, edit, and utilities. See “Interactive System Productivity Facility” on page 83 for a complete description of ISPF.
JCL	During the install phase of z/OS, many batch jobs are required to be submitted. The job control language (JCL) for these jobs needs to be updated for your environment. Therefore, it is essential that a system programmer be very familiar with JCL, batch job submission from TSO/E, and using ISPF. See “Introduction to JCL: Creating a data set” on page 134.
Storage	Storage concepts must be understood by the system programmer in setting up a z/OS environment. See “z/OS storage concepts” on page 247.

Device I/O	An I/O configuration is the definition of hardware resources that are available to the operating system and the connections between these resources. The resources include: <ul style="list-style-type: none"> – Channels (IBM ESCON and IBM FICON®) – ESCON and FICON Directors (switches) – Control units – Devices <p>When you define a configuration, you need to provide both physical and logical information about these resources. For example, when defining a device you provide physical information, such as its type and model, in addition to logical information, such as the identifier you will assign in the configuration definition.</p> <p>You must define an I/O configuration to the OS (software) and the channel subsystem (hardware). The HCD component of z/OS consolidates the hardware and software I/O configuration processes under a single interactive user interface.</p>
Processors	When more than one processor exists in a complex, or more than one logical partition (LPAR) exists in a complex, z/OS is required to be defined in multisystem mode or a sysplex.
Consoles	A console configuration consists of the various consoles that operators use to communicate with z/OS. Your installation first defines the I/O devices it can use as consoles with the HCD. HCD manages the I/O configuration for the z/OS system. After you have defined the devices, indicate to z/OS which devices to use as consoles by specifying the appropriate device numbers in the CONSOLxx parmlib member.

System libraries management

When installing z/OS, the current z/OS system becomes the driving system that you use to install the target system. The target system is the system software libraries and other data sets that you are installing. You log on to the driving system and run jobs there to create or update the target system. After the target system is built, it can be restarted on the same hardware (same LPAR or same processor), or on different hardware than that used for the driving system.

DASD space management

The space required by system software data sets, except for partitioned data set extended (PDSE) data sets, is affected by the block sizes you choose for those data sets. Generally, data sets with larger block sizes use less space to store the same data than those with smaller block sizes. Data sets that store more data in less space usually offer better DASD performance than those that use more space to store the same data. The DASD space required to install z/OS includes:

- ▶ All elements
- ▶ All features that support dynamic enablement, regardless of your order
- ▶ All unpriced features that you ordered

SYS1.PARMLIB data set

SYS1.PARMLIB is a required partitioned data set (PDS) that contains IBM-supplied and installation-created members, which contain lists of system parameter values.

3.5 z/OS system with SMP/E

- ❑ These elements of the system fit together, and they can be installed, modified, and tracked using SMP/E

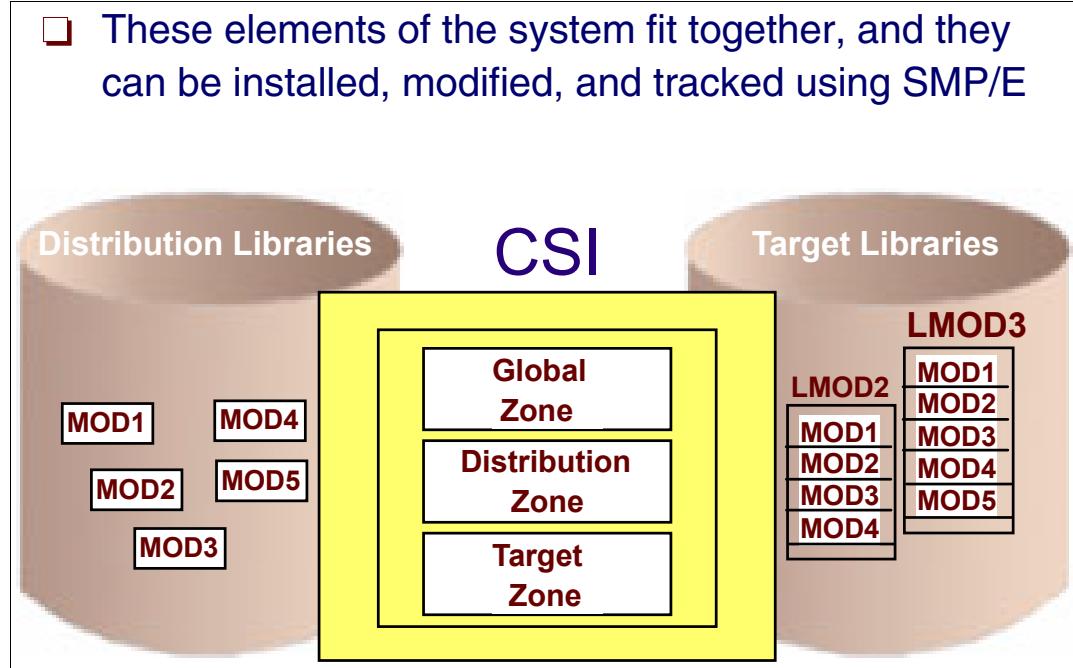


Figure 3-5 z/OS terminology for maintenance of the SMP/E libraries

z/OS terminology for the install

SMP/E is a tool designed to manage the installation of software products on your z/OS system, and to track the modifications that you make to those products. Usually, it is the system programmer's responsibility to ensure that all software products and their modifications are properly installed on the system.

The system programmer also has to ensure that all products are installed at the correct level, so that all elements of the system can work together. At first, that might not seem too difficult, but as the complexity of the software configuration increases, so does the task of monitoring all of the elements of the system.

Remember: All of the information found in the global zone, combined with the information found in the distribution and target zones, represents the data that SMP/E needs to install and track your system software.

The following SMP/E functions need to be understood during any install process:

LMOD

The load module (LMOD) entry contains all of the information needed to replace or update a given load module. This includes information, such as whether the load module is link-edited or copied during the system generation process, any link-edit statements required to relink the load module, the link-edit attributes of the load module, and the libraries in which it exists. An LMOD entry is generally created by one of the following methods:

- ▶ Installing a system modification (SYSMOD) that adds the load module
- ▶ Processing JCLIN commands

ACCEPT	<p>The ACCEPT function is an SMP/E control statement that controls the placement (installing) of SYSMODs into the distribution libraries. Processing is similar to that during APPLY except that the distribution zone is updated, not the target zone, and JCLIN data is not processed by ACCEPT.</p> <p>If the installation is successful, any entries in the SMPSCDS data set (SCDS) created by APPLY are deleted, as are temporary libraries created by RECEIVE. Therefore, after a SYSMOD is accepted, it can no longer be removed by SMP/E.</p>
APAR	<p>An authorized program analysis report (APAR) identifies IBM-supplied fixes of a temporary corrective nature to elements of IBM-supplied function SYSMODs. APAR fixes are intended to solve problems currently being experienced by an installation.</p> <p>The APAR fix is usually in the form of either a modification to a load module or an update to card-image data. It is intended as a temporary arrangement until a program temporary fix (PTF) is issued to fix the problem permanently. This PTF will supersede the APAR fix, and specifies this relationship on its ++VER statement.</p> <p>To get an APAR SYSMOD accepted into the distribution libraries, the APARS keyword must be specified in the ACCEPT control statement, which protects against inadvertent updating of distribution libraries that are to be kept free of temporary fixes.</p> <p>The ++VER statement in the APAR SYSMOD must specify the function modification identifier (FMID) of the function that “owns” the elements being updated.</p> <p>Note the following examples:</p> <pre>++APAR(AP12345) ++VER(C150) FMID(HCI6300)</pre> <p>It is better to avoid accepting APARs into the distribution library, however, because the relevant PTF will become available in due course as a more permanent form of service.</p>
APPLY	<p>The APPLY function is a function of SMP/E. It is an SMP/E control statement that applies SYSMODs to the CICS target libraries, where they can be tested. If the tests are not satisfactory, you can remove all or selected SYSMODs using the RESTORE function. If the test is successful, you can use the ACCEPT function to store the elements from the SYSMOD into the distribution libraries.</p> <p>During JCLIN processing, every affected entry in the target zone is saved in the SCDS, in case the target system libraries and the target zone must be restored to their original status.</p>
CSI	<p>Consolidated software inventory (CSI) is a keyed VSAM data set, logically divided by SMP/E into zones. For further information about the CSI and the logical structure of zones, see <i>System Modification Program Extended: User's Guide</i>, SA23-2277.</p>
Distribution zone	<p>The distribution zone describes the structure and contents of a set of distribution libraries.</p>
FMID	<p>FMID is a keyword of SYSMODs that is used in identifying the release and option to which a SYSMOD is applicable for a particular product.</p>

Global zone	The global zone is a logical division of the SMP/E CSI, containing data such as the following information: <ul style="list-style-type: none"> ▶ Definitions of all other related zones ▶ Descriptions of the SYSMODs present in the PTF temporary store (PTS) ▶ Descriptions of the system utilities to be started during SMP/E processing ▶ Data definition (DD) entries for use by dynamic allocation
Load module	In the context of SMP/E, a load module (LMOD) is <i>an executable load module in a target library</i> (such as hlq.SDFHLOAD).
PTF	A PTF contains IBM-supplied fixes to elements of IBM-supplied function SYSMODs. PTFs are intended for installation by all users to avoid possible problems. A PTF can contain fixes for several different problems, so several APAR fixes reported in IBM RETAIN might all be superseded by the more permanent PTF. Every PTF is introduced by a ++PTF header statement, and contains the FMID keyword on its ++VER modification control statement, identifying for example, CICS (HCI6300) as the owner of the modules being serviced.
	Note the following examples: ++PTF(UP12345) ++VER(C150) FMID(HCI6300)
PTS	PTS is an SMP/E primary data set used to store temporary SYSMODs that are in RECEIVE or APPLY status; that is, they have not been rejected or accepted.
RECEIVE	RECEIVE is a function of SMP/E. It is an SMP/E control statement that initiates processing of a SYSMOD. RECEIVE reads the SYSMODs from the SMPPTFIN data set. Each SYSMOD must have been received before any other function can be run. RECEIVE updates the SMPPTS data set and performs syntax checking on input. Before any SYSMOD for a product can be received, the global zone must have been initialized with a global zone entry. A service SYSMOD can be received into the PTS before the function to which it applies has been received. They can be maintained there until the function is received. This supports all service for a product, such as CICS, to be installed with the base product.
REJECT	REJECT is a function of SMP/E. It is an SMP/E control statement that removes SYSMODs from the PTS data set and deletes any temporary libraries that SMP/E might have allocated when the SYSMOD was received (RELfiles). If the SELECT or EXCLUDE option is not coded on the REJECT control statement, then all SYSMODs not applied or accepted will be removed from the PTS. This is called a <i>mass rejection</i> . All other SYSMOD processing functions (RECEIVE, APPLY, RESTORE, and ACCEPT) can have SELECT or EXCLUDE specified, or can default to mass-processing mode.

RESTORE	RESTORE is a function of SMP/E. It is an SMP/E control statement that removes SYSMODs from the target system libraries after they have been applied, and restores the target libraries to their status before application of the SYSMODs. If necessary, RESTORE reconstructs the target zone entries from the SCDS. If you select mass restore , all SYSMODs that have been applied but not accepted will be removed from the target libraries.
SYSMOD	SMP/E can install a large variety of system updates, provided they are packaged as a SYSMOD. A SYSMOD is the actual package of elements and control information that SMP/E needs to install and track system modifications.
Target zone	The target zone describes the structure and contents of a set of target system libraries.
UCLIN	UCLIN is a function of SMP/E. It is an SMP/E control statement that can be used to manipulate the various data sets that make up the SMP/E database. The most common use of this function is to initialize the SMP/E database before the first attempt to use it.
USERMOD	<p>USERMOD (user modification) is a user-supplied modification to elements of IBM-supplied function SYSMODs. USERMODs are similar to APAR fixes, but are supplied by the user and not by IBM. Examples of USERMODS include those in the following list:</p> <ul style="list-style-type: none"> ▶ A local fix to bypass a problem until an official IBM fix is available ▶ A user modification to add or alter functionality in CICS

3.6 z/OS delivery options

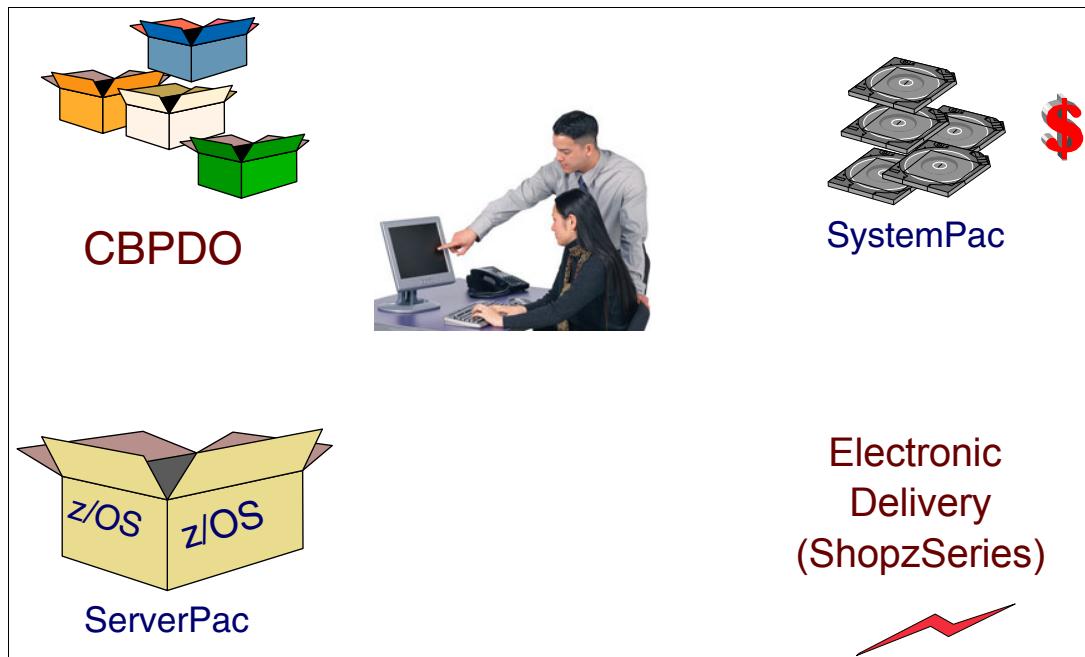


Figure 3-6 z/OS package options

z/OS package options for ordering

Several IBM packages are available for installing z/OS. Some packages are entitled with the product (as part of your z/OS license, at no additional charge). Other packages are available for an additional fee. This section describes each package:

ServerPac ServerPac is an entitled software delivery package consisting of products and service for which IBM has performed the SMP/E installation steps and some of the post-SMP/E installation steps. To install the package on your system and complete the installation of the software that it includes, you use the CustomPac Installation dialog. For ServerPac orders, service is integrated with product code.

Tip: Starting with z/OS V1R13, *ServerPac: Installing Your Order* is no longer sent in hardcopy format. Instead, the information is available in PDF format on a DVD that also includes the program directories.

CBPDO CB PDO is an entitled software delivery package consisting of uninstalled products and unintegrated service. There is no dialog program to help you install, as there is with ServerPac. You must use SMP/E to install the individual z/OS elements and features, and their service, before you can perform an IPL. Installation instructions are in the z/OS Program Directory.

SystemPac SystemPac is a software package, available for an additional fee and offered worldwide, that helps you install z/OS, subsystems (DB2, IMS, CICS, NCP, and WebSphere Application Server). SystemPac is tailored to your specifications; it is manufactured according to parameters and IODF definitions that you supply during order entry. The goal is to have the system tailored to your specifications and have products enabled according to your specified configuration.

As previously mentioned, parameters are collected by telephone. Using a printed questionnaire as a guide, you tell an IBM representative your responses. Upon completion, a printout showing all of the parameters and definitions you specified is sent to you for reference. The documentation *SystemPac Installation Guide* that is included with your order specifies the integrated service level applicable to your order.

Tip: Starting with z/OS V1R13, *SystemPac Installation Guide* is no longer sent in hardcopy format. Instead, the information is available in PDF format on a DVD that also includes the program directories.

ShopzSeries

ShopzSeries is an Internet application that you can use to order z/OS software products and service. Using ShopzSeries, you can order corrective and preventive service over the Internet, with delivery over the Internet or by tape.

Service with ShopzSeries reduces your research time and effort by using your uploaded SMP/E CSI to ensure that all applicable service, including reach-ahead service, for the installed FMIDs in the target zones is selected.

The following web address is the main page for ShopzSeries:

<https://www14.software.ibm.com/webapp/ShopzSeries/ShopzSeries.jsp>

Important: IBM provides entitled service ordering and service delivery capabilities for the z/OS platform products electronically using the Internet. ShopzSeries is the primary ordering and delivery method for software service on these platforms.

The IBM S/390 Service Update Facility (SUF), the Internet-based software service tool for ordering and receiving z/OS, z/OS.e, IBM z/VM®, and Virtual Storage Extended/Enterprise System Architecture (VSE/ESA) software service, was discontinued effective January 15, 2006.

3.7 Choosing an installation package

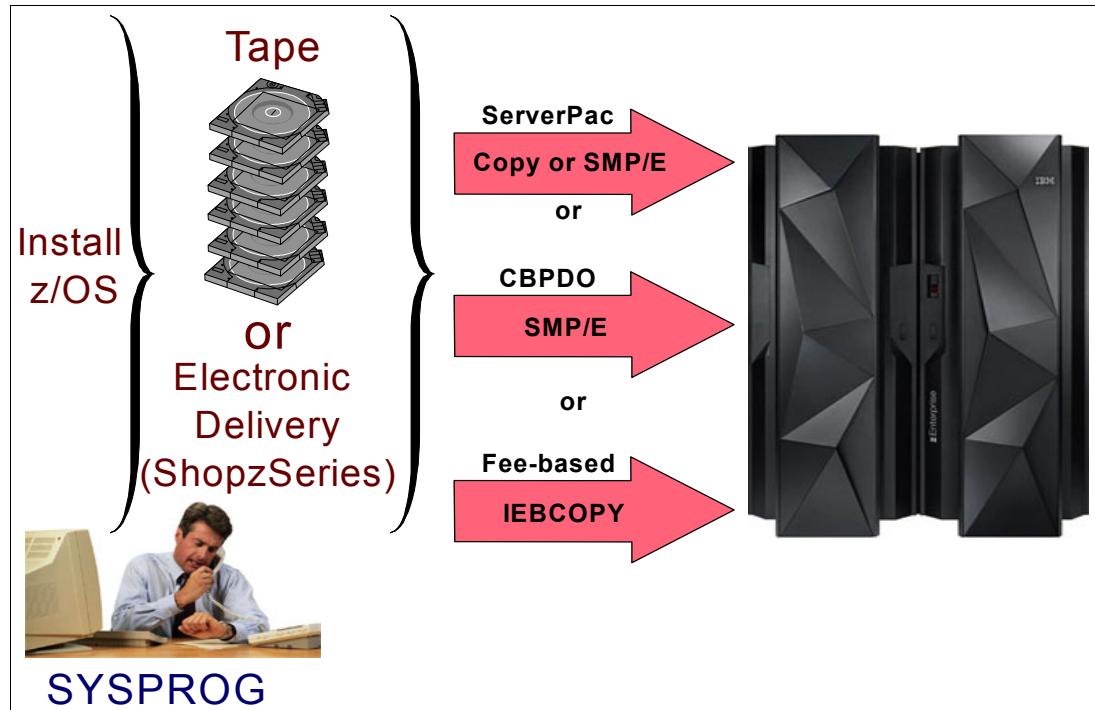


Figure 3-7 System programmer installing z/OS

Choosing an installation package

Before z/OS, large applications ran on an IBM Multiple Virtual Storage (MVS) OS that consisted of the Base Control Program (BCP), the Data Facility Storage Management Subsystem Data Facility Product (DFSMSdfp), and JES2 or JES3, plus a collection of other software products that the applications required, such as ISPF, TSO/E, and so on. You traditionally run these products at various release levels, using a “mix and match” approach.

With the introduction of z/OS, all of these products were integrated into a single product. You no longer order new levels of some products but not of others. Rather, you order and install an entire set of products integrated into one functionally rich OS. Only components for which you have a license are allowed to run. For z/OS components, this is documented in the IFAPRDxx member of SYS1.PARMLIB.

z/OS base elements and optional features

Because the base elements and optional features of z/OS, as described in 1.6, “z/OS V2R1 base elements” on page 10 and 1.7, “z/OS optional features” on page 17, are integrated into a single package with compatible service levels, you must install, with few exceptions, the entire z/OS product. You can install z/OS using one of several IBM packages. Two of these packages are available at no additional charge when you license z/OS:

- ▶ ServerPac
- ▶ CBPDO

Ordering ServerPac

ServerPac is an entitled software delivery package consisting of products and service for which IBM has performed the SMP/E installation steps and some of the post-SMP/E installation steps. To install the package on your system and complete the installation of the software that it includes, you use the CustomPac Installation dialog.

When you order a new system or a new release of z/OS, you also receive all of the new maintenance or service (including corrections) that is applicable to the release. ServerPac, when installed, replaces a complete z/OS system or subsystem. ServerPac is provided by system or subsystem type; that is, you get a separate ServerPac for each z/OS or subsystem.

Tape delivery

If you chose tape delivery when you ordered your ServerPac, you receive a box containing the ServerPac system tapes, the ServerPac documentation, and any publications or client code CDs for the products included in your order.

The ServerPac Installation dialog is used to install the ServerPac order, so the first step is to unload the dialog from tape. Then, you receive the order into the dialog, and use the dialog to tailor the ServerPac configuration to match your system environment. The dialog will use the tailored configuration to generate a customized installation job stream that will create your new target system from the system tapes.

Restriction: z/OS is no longer delivered on 3480, 3480C, and 3490E tapes. DVD is a new delivery medium available with z/OS V1R12. Like tape and the Internet, this method can be used to order z/OS and products that run on z/OS. Your order is placed and processed through ShopzSeries as a DVD (4.7 gigabyte (GB) single-sided, single-layered) media order. To upload your DVD order you need a DVD reader and a workstation that is network-attached to your z/OS host system.

New or existing user

The best installation method is usually the one that requires the least amount of work for you. The following methods are useful:

- ▶ If you are new to z/OS and never had a previous system, use ServerPac's full system replacement option.
- ▶ If you are new to z/OS, and you also want to install a ServerPac in the dump-by-data-set format, you can order and install the Customized Offerings Driver (COD) first. This software is a pre-built stand-alone driving system that you can use to prepare the installation of a CBPDO or a ServerPac if you do not have a driving system, or even if your driving system does not meet the minimum system requirements.

Electronic delivery using ShopzSeries

You can order a ServerPac using ShopzSeries or by contacting IBM or your IBM Business Partner. You select the products you want included in your ServerPac system or subsystem from the product catalog. The product catalog for ServerPac is refreshed monthly to include products that have been supported or withdrawn from ServerPac in the previous month.

When your order is ready for download, you will receive an email from IBM. When you log on to ShopzSeries to access your order, the download pages are dynamically built. You can either download your order directly to a host system using the ServerPac dialog, or download it to your workstation first using Download Director in ShopzSeries and then upload it to the host.

After you make your product selections, place your order as shown in Figure 3-8 on page 190. You receive an order confirmation in a few hours. After that, you must wait for the order to be built by IBM and placed on the download server. Depending on what you order, this might take up to 10 business days. After the order is ready to be downloaded, you will receive another note saying that it is ready. This second note will contain a link to the download page you need to use to get information for proceeding with the next step.

Fast Track: ShopzSeries is an Internet application that you can use to order z/OS software products and service. Using ShopzSeries, you can order corrective and preventive service over the Internet, with delivery over the Internet or by tape. Service with ShopzSeries reduces your research time and effort by using your uploaded SMP/E consolidated software inventory (CSI) to ensure that all applicable service, including reach-ahead service, for the installed FMIDs in the target zones is selected.

The following web address is the main page for ShopzSeries:

<http://www.ibm.com/software/shopzseries>

Fee-based options

Several fee-based options are available:

SystemPac

SystemPac offers the capability to build a system with integrated subsystems in either full volume dump/restore format or data set copy format. The full volume dump/restore format enables you to install z/OS without using the dialog.

Installation is done through pack restore using DFSMS Data Set Services (DFSMSdss). If the vendor product is selected in the order, installation can be done using IBM z/OS Fast Dump Restore (FDR).

SystemPac is designed for those who have limited skill or time to install or upgrade z/OS, but who want to install or upgrade to use z/OS functions in e-commerce or other areas.

SystemPac tailors z/OS to your environment, such as DASD layout, migration of MVS configuration program / I/O configuration program (MVSCP/IOCP) to IODF, and naming conventions, based on information provided to IBM. With this offering, selected non-IBM products can be integrated.

Entry Server Offering

The Entry Server Offering, only available in selected countries, is a packaged solution that includes hardware, software, installation services, maintenance, and financing to help clients obtain current technology.

3.8 ServerPac Internet delivery

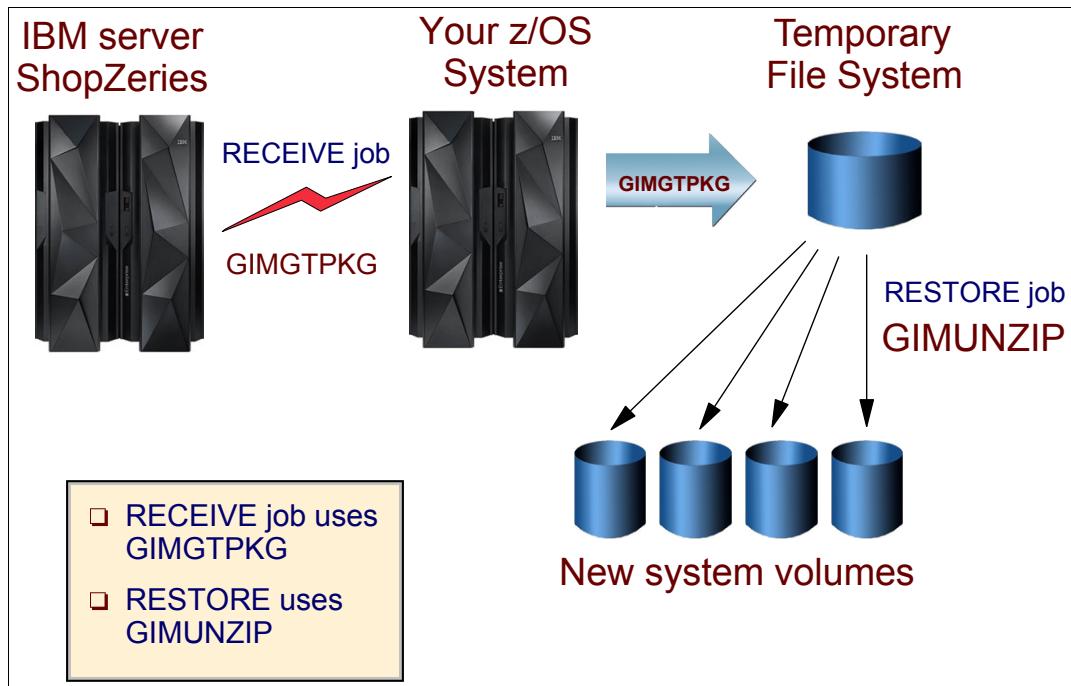


Figure 3-8 ServerPac internet delivery

ServerPac Internet delivery

To choose Internet delivery, you must place your order using ShopzSeries. As previously mentioned, when your order is ready for download, you will receive an email from IBM. When you log on to ShopzSeries to access your order, the download pages are dynamically built. You can either download your order directly to host using the ServerPac dialog or download it to your workstation first using Download Director in ShopzSeries, and then upload it to the host.

The dialog “Server” option will generate a RECEIVE job that uses the SMP/E GIMGTPKG utility to retrieve your order directly from the IBM server. GIMGTPKG will place the data in your Download file system. The dialog “File System” option will generate a RECEIVE job that will retrieve your order from the Download file system after you have uploaded it from your workstation, if you chose to download it to your workstation first.

Later, the dialog RESTORE job will use the SMP/E GIMUNZIP program to load your new target system’s volumes from the Download file system. When you have finished, you can delete the Download file system. However, consider dumping it first, just in case.

GIMGTPKG service routine

The GIMGTPKG service routine can be used to obtain GIMZIP packages from a remote File Transfer Protocol (FTP) server in an Internet Protocol network and store the packages on a local z/OS host. GIMGTPKG performs the functions of the **SMP/E RECEIVE FROMNETWORK TRANSFERONLY** command, but does so independently of SMP/E.

GIMGTPKG uses FTP to transport the files of a GIMZIP package from a remote FTP server to a local host, therefore providing the following advantages:

- ▶ Industry standard FTP protocol
- ▶ Secure transmission using the capabilities of the z/OS FTP client
- ▶ Ensured integrity of the transported files

GIMUNZIP service routine

SMP/E provides the GIMZIP and GIMUNZIP service routines to construct, and then later unwrap, network-transportable packages of software. This enables you to create your own packages of SMP/E installable software, and then distribute them in your own enterprise, or to other enterprises. Specifically, the GIMZIP service routine will accept partitioned or sequential data sets as input and create a network-transportable package as output.

The GIMUNZIP service routine is used to extract data sets, files, and directories from archive files in GIMZIP packages created by the GIMZIP service routine. These packages typically contain software and associated materials in the form of SYSMODs, RELFILE data sets, HOLDDATA, and other materials, such as documentation, samples, and text files. These GIMZIP packages can be transported through a network, processed by the GIMUNZIP service routine, and then processed by the SMP/E RECEIVE command.

More specifically, the GIMUNZIP service routine extracts data sets, files, and directories from the archive files that compose the GIMZIP package. An archive file consists of a portable image of a sequential, partitioned, or VSAM data set, or a file or directory in a UNIX file system, and the information needed to create that data set, file, or directory from the portable image.

The data set, file, or directory into which the archive file is to be extracted can already exist, or GIMUNZIP can create a new one of the appropriate type. New sequential and partitioned data sets created by GIMUNZIP are always catalogued.

3.9 System and installation requirements

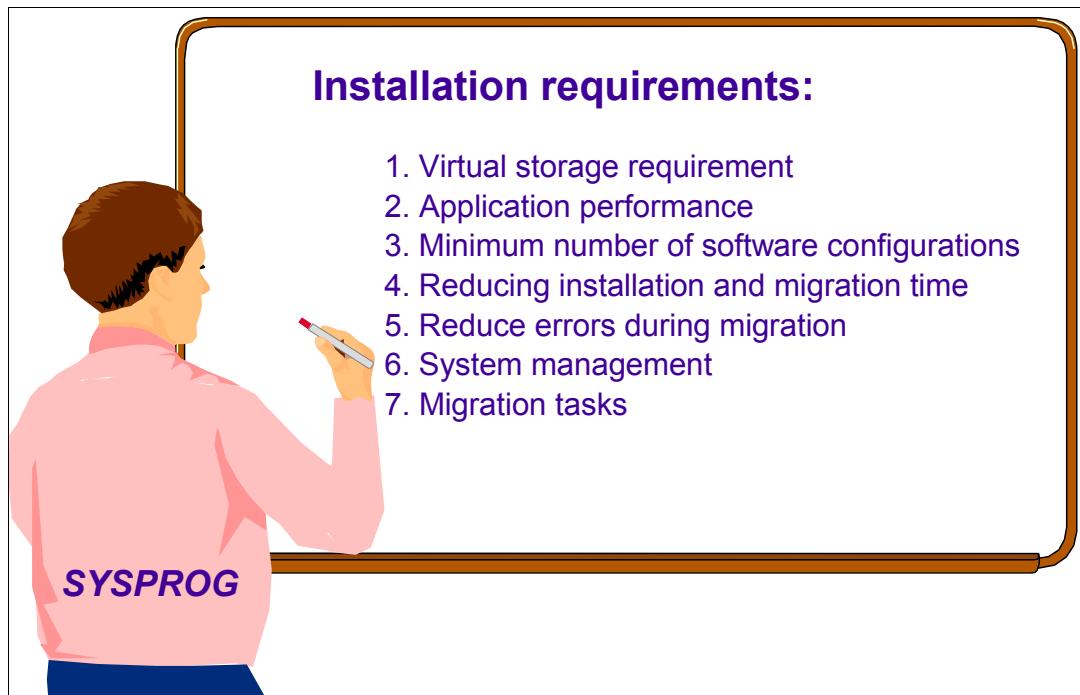


Figure 3-9 Determining the installation requirements

System and installation requirements

Having an installation plan helps you ensure that the software is able to meet your installation's functional requirements. Besides functionality, there are other issues to think about when planning to build a system:

- ▶ Hardware and software requirements, including non-IBM software compatibility
- ▶ Virtual storage mapping
- ▶ Application performance
- ▶ Building a minimum number of system software configurations
- ▶ Reducing installation and migration time
- ▶ Reducing the opportunities for error during migration
- ▶ Making it easier to manage the system after it is in production
- ▶ Minimizing migration actions for the people who use the system

How you choose to meet all of these requirements can have a significant effect on how much work is required to perform the tasks associated with each stage. Keep all of these additional requirements in mind when you are planning to build a new system.

3.10 Reviewing your current system

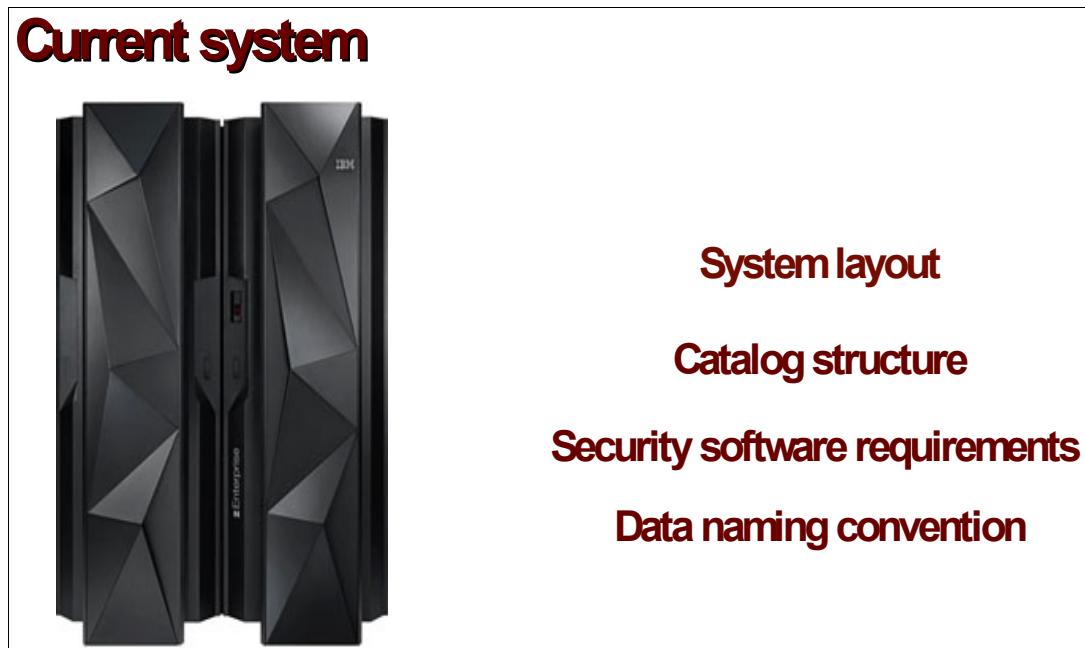


Figure 3-10 Reviewing your current system for migration

Reviewing your current system

More often than not, you will be planning to upgrade from your current z/OS system to the new release of z/OS. It is therefore very important to review the setup of your current environment while planning for the new system. Some areas to consider are listed here:

- ▶ **System layout**

When you install using ServerPac or dump-by-data set SystemPac, take advantage of the Recommended System Layout enhancement. This function takes into consideration the volume space available and the data set sizes in your order, and places the data sets accordingly. If you are a CBPDO customer, you might have to calculate available space for the data set types on target volumes to ensure that it is sufficient. Depending on the volume type, you might have to add another target volume.

- ▶ **Catalog structure**

A software upgrade uses your existing catalog structure. This includes your existing master catalog (with direct or indirect cataloging references) and user catalogs. In addition, software upgrade enables you to create new user catalogs as part of the installation process.

- ▶ **Data set naming conventions in your present environment**

Check the names of data sets on the volumes, because a data set name on a given volume must *not* be the same as the name of a data set that you plan to install on that volume. Use the View and Change option of Modify System Layout of the CustomPac Installation dialog to obtain a list of the data sets in your order.

- ▶ **Security software considerations**

Use the RACFDRV installation job as a sample of the security system definitions required so that you can perform the installation tasks.

Depending on your order, the system target and distribution libraries might exceed more than one DASD volume. Therefore, define your new system layout to be prepared for future installation and easy cloning of your system.

Important: See “Recommended data set placement” in *z/OS Planning for Installation*, GA32-0890, before defining where the following new data sets should exist:

- ▶ Target data sets
- ▶ Distribution libraries data sets
- ▶ Master catalog and user catalogs
- ▶ Dialog and order data sets

3.11 Driving system and target system

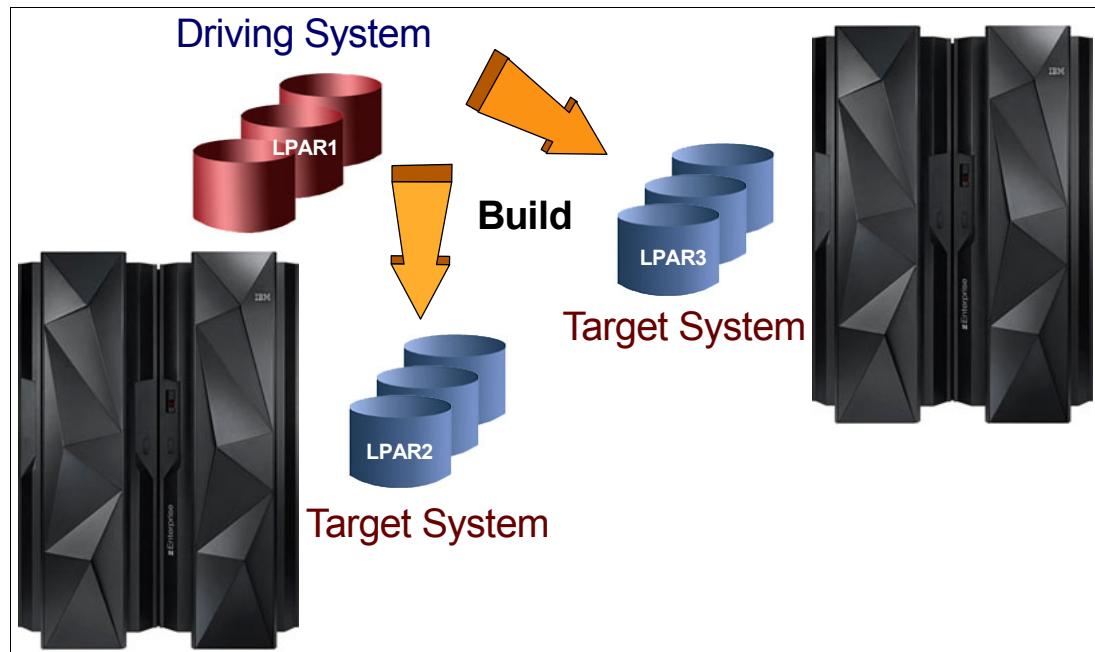


Figure 3-11 Using the driving system to create the new system

Driving system and target system

The *driving system* is the system image (both the hardware and software) that you use to install your new system image, which is the *target system*. The target system is the system software libraries and other data sets that you are installing. You log on to the driving system and run jobs there to create or update the target system. After the target system is built, it can be loaded on the same hardware (same LPAR or different LPAR on the same processor), or on different hardware from that used for the driving system.

Driving system requirements

The minimum driving system level for installing z/OS V2R1 is z/OS V1R12 (for installing z/OS V1R13, it is z/OS V1R11.)

To prepare the driving system *before* building the target system, you need to perform the following tasks:

- ▶ Identify the software requirements for the driving system according to the delivery package you are using, for example, ServerPac.
- ▶ Identify the hardware requirements for the driving system.

Preparing the target system

You are also required to prepare for the target system, as described here:

- ▶ Choose the software products to install and identify requisites.
- ▶ Order z/OS and related IBM products.
- ▶ Identify the hardware requirements for the target system.
- ▶ Identify the service needed for the target system.

Driving system requirements

Driving system requirements for installing z/OS by way of ServerPac or dump-by-data set SystemPac are listed here:

- ▶ An OS.
 - Use either of the following:
 - z/OS V1R12 or later.
 - The Customized Offerings Driver V3 (5751-COD). This entitled driving system is provided for those who either do not have an existing system to use as a driving system, or whose existing system does not meet the requirements of a driving system and choose to not upgrade their driving system.
 - ▶ A TSO/E session on the restarted system must be established using a locally-attached or network-attached terminal.
 - ▶ Appropriate authority by using the RACFDRV installation job as a sample of the security system definitions required so that you can perform the installation tasks.

Entitled packages ServerPac and CBPDO: Use ServerPac, using the full system replacement installation path. You will also need the COD V3 (5751-COD) as a driving system (COD is also entitled).

For more information about each of the requirements for both the driving and target system, see *z/OS Planning for Installation*, GA32-0890, for the z/OS release that you are installing.

Information: If you do not have a z/OS driving system, your SystemPac order contains the stand-alone versions of the following utility programs so that you can install your order:

- ▶ ICKDSF, to be used to initialize DASD, create volume tables of contents (VTOCs), and perform other utility functions during system installation
- ▶ DFSMSdss, to restore the volume from tape to DASD

The utilities are provided based on selections you make during local order entry. See “SystemPac Installation Guide” supplied with your order for details about running these utilities.

New installations with no driving system

As of February 22, 2011, the COD V3 (5751-COD) and PTF maintenance are no longer delivered on 3480, 3480C, and 3490E tapes. DVD is a new delivery medium available for this information.

3.12 z/OS installation using ServerPac

- ❑ HLQ for the CustomPac data sets
- ❑ CustomPac data sets placement
- ❑ Start the CustomPac dialogs

```
//EXTRACT JOB <JOB statement info goes here...>
//STEP01 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IDOC      DD DISP=SHR,DSN=SYS1.orderid.DOCLIB,LABEL=(06,SL),
//              VOL=SER=tapeser,UNIT=tapeunit
//ODOC      DD DSN=work.library.jcl,
//              DISP=(NEW,CATLG,DELETE),
//              VOL=SER=volser,UNIT=SYSALLDA,
//              SPACE=(9600,(240,30,20))
//*
//SYSIN     DD *
      COPY INDD=IDOC,OUTDD=ODOC
      S M=LOADRIM
/*

```

Figure 3-12 Installing the CustomPac dialog

z/OS installation using ServerPac

Your z/OS ServerPac order contains an ISPF dialog that you use to install z/OS. This dialog is called the CustomPac installation dialog because it is used to install all of the IBM CustomPac offering.

Installing the dialogs

To install the dialogs on your driving system for the first time, you must first copy the LOADRIM job to your driving system. Then use LOADRIM to load the rest of the dialog data sets onto your driving system's DASD.

Copying the LOADRIM job from tape

For orders delivered on tape, IBM provides the LOADRIM job as member LOADRIM in data set SYS1.*orderid*.DOCLIB, where *orderid* is the IBM-supplied number for your order. The SYS1.*orderid*.DOCLIB data set is the sixth file on the tape, which has a standard label (SL).

Copy LOADRIM from SYS1.*orderid*.DOCLIB. Create a job to copy the LOADRIM job to your driving system. Figure 3-12 shows a job that you can use.

Before beginning the installation, it is useful to perform these tasks:

- ▶ Review the contents of the ServerPac shipment that you received from IBM by checking the packing slip to verify that you have a complete set of installation tapes and documentation.
- ▶ Make sure your user ID has ALTER authority for the following high-level qualifiers:
 - CPAC
 - SYS1

- All product-specific high-level qualifiers for products that come with your package. You can find a listing of all qualifiers by using the A-ALIAS option of the dialog, or see Product Information in the appendix of *IBM ServerPac Using the Installation Dialog*, SA23-2278.
- System-specific aliases (SSA) named in the SSA panel. You will also need UPDATE access for the SSA qualifiers in the driving system's master catalog.
- ▶ During the job phase of the installation process, you might need RACF SPECIAL authority or equivalent if you use other security software.
- ▶ If you decided to use storage management subsystem (SMS) to manage data sets in your order, your user ID needs READ access to the FACILITY class profile STGADMIN.IGG.DIRCAT.

Installing the CustomPac dialogs

The first step in installing z/OS is to install the CustomPac dialogs from the RIM tape on your driving system. After they are installed, the dialogs do not have to be reinstalled with every order. They are auto-upgraded whenever you get a new order. Version checking starts the update of the dialogs during the CustomPac RECEIVE function.

Follow these steps to install the CustomPac dialogs:

1. Define the high-level qualifier (HLQ) for the CustomPac data sets (called *master dialog* data sets) pointing to a user catalog accessible by both the driving and the target systems; for example, SERVRPAC.
Because the dialogs are permanently installed at your installation, do not specify the IBM-supplied order number as the CustomPac qualifier.
2. If you intend to use SMS-managed dialog data sets, assign them to a management class that supports migration, unless SMS and Data Facility Storage Management Subsystem (DFSMS) Hierarchical Storage Manager (DFSMShsm) environments will be shared between driving and target systems.
3. As mentioned, for orders delivered on tape, IBM provides the LOADDRIM job as member LOADDRIM in data set SYS1.*orderid*.DOCLIB, where *orderid* is the IBM-supplied number for your order. The SYS1.*orderid*.DOCLIB is the sixth file on the tape, which has a standard label (SL).
4. The LOADDRIM job contains steps to:
 - a. Delete previous CustomPac dialog data sets.
 - b. Unload master dialog data sets.
 - c. Allocate an order inventory data set to contain control information for all sent ServerPac orders.

3.13 The RIM tape samples



Figure 3-13 Using the CustomPac dialogs

The RIM tape samples

The relational database management system (RDBMS) Interface Module (RIM) tape contains sample procedures, JCL, jobs, and command lists (CLISTS). They are in `SYS1.orderid.DOCLIB`. You can unload and modify these samples for your installation, as listed and described in Table 3-1.

Table 3-1 Useful samples from the RIM tape

Name	Description
LOADRIM	LOADRIM is the JCL to unload files from tape and set up the installation dialog. When you edit the LOADRIM sample JCL, you can choose the name of the master data sets, the unit name of your tape drives, and the volume serial number (VOLSER) of the DASD that receives the installation dialog's data sets.
SETUP	This is a sample LOGON procedure that includes the CustomPac dialog ISPF libraries.
CPPCSAMP	This sample CLIST can be used to set up the environment rather than modifying the LOGON procedure. CPPCSAMP uses library definitions (LIBDEFs), and is the preferred method to allocate the CustomPac libraries and start the dialog.
CPPINIT	With the CPPINIT CLIST, you can set up the environment from native TSO.
PRTDOC	This sample job prints the CustomPac Installation dialog reference manuals.

3.14 Starting the CustomPac dialogs

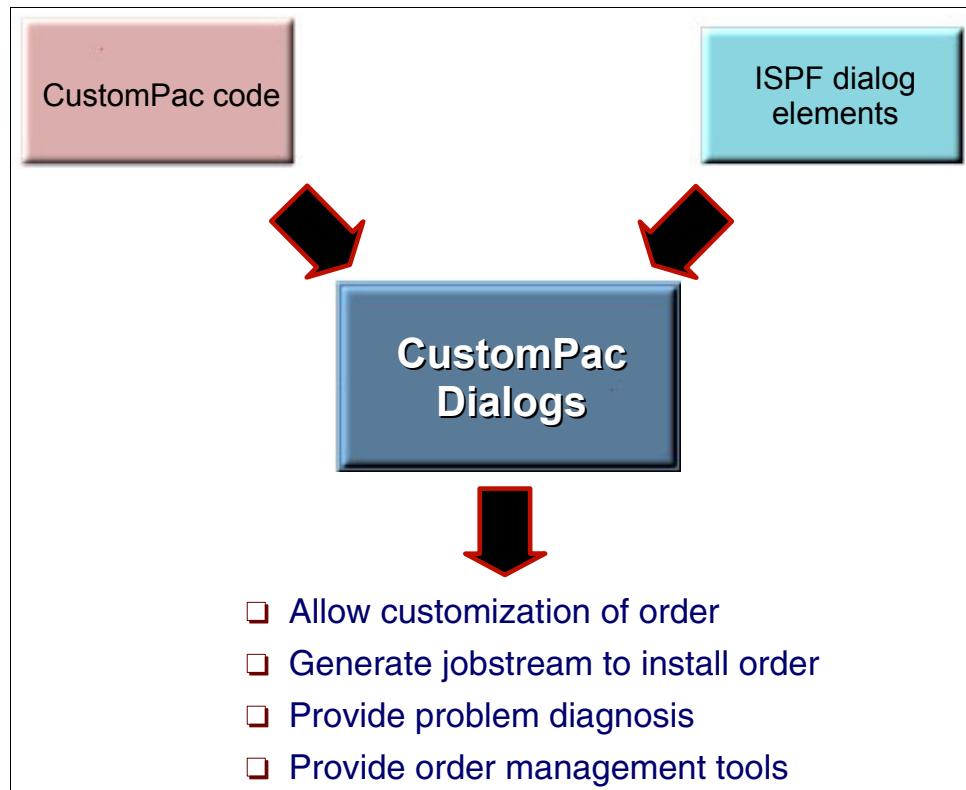


Figure 3-14 CustomPac dialogs

Starting the CustomPac dialogs

The CustomPac dialogs are now installed. To access the dialogs, follow these steps:

1. Customize the CPPCSAMP CLIST, changing the *custompac.qualifier* according to the qualifiers of the master dialogs data sets.
2. Save the customized CLIST.
3. Go to TSO/ISPF, option 6 and run the new CLIST, where DATA.SET.NAME is the partitioned data set where you saved the CLIST and CPPCSAMP is the member name where you saved the customized CPPCSAMP CLIST:
`exec 'DATA.SET.NAME(CPPCSAMP)'`

Figure 3-16 on page 203 shows the panel that you receive when you start the CustomPac dialogs.

Information: See *IBM ServerPac Using the Installation Dialog*, SA23-2278, for further considerations and a complete explanation of how to install your order with ServerPac.

3.15 Types of data sets and their volumes

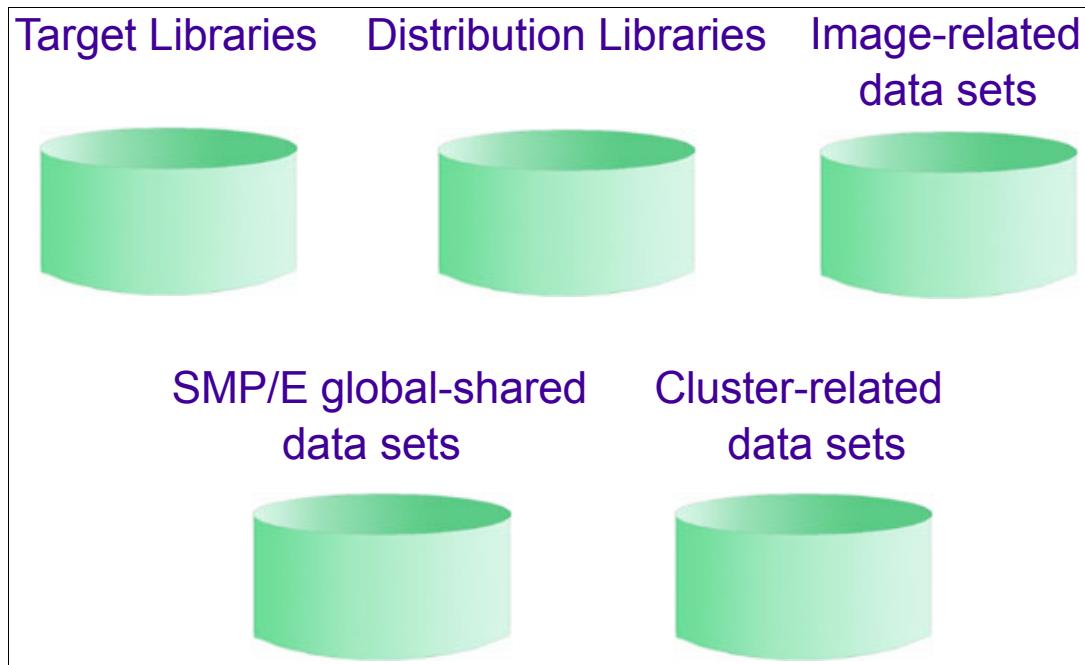


Figure 3-15 Installation data sets

Types of data sets and their volumes

An installation needs to determine which data sets to place on each volume based on data set type, *not* based on element, feature, or product. There are five types of data sets in the suggested data set layout. Each type is placed on a separate (logical) volume. The types of data sets and their volumes are listed here:

- ▶ SMP/E global-shared data sets, on a volume shared by all systems in the complex that need access to SMP/E global information

Tip: These data sets contain SMP/E global system information. For the sake of organization, and ease of backup and recovery, it is useful to keep them together on a volume shared by all systems that use SMP/E in your complex.

If you maintain multiple global zones for subsystems or vendors, the global zone described here should contain ZONEINDEX references to all other zones. This will assist you in cross-zone conditional requisite checking, without requiring any changes to your installation's maintenance procedures.

- ▶ Target libraries (TLIBs) for product sets, on the following volumes:
 - TLIB volume 1 (TVOL1)
 - TLIB volume 2 (TVOL2) through TLIB volume *n* (TVOL*n*)
 - Hierarchical file system (HFS) or IBM zSeries File System (zFS) target volume
 - Licensed product set volume (for licensed programs not installed with the z/OS product set)
 - Vendor product set target volume
 - Subsystem product set target volume

Information: TVOL1 through TVOLn can be shared with other systems (for Interactive Problem Control System (IPCS) or WLM migrations, for example). If TVOL1 does not contain enough space to hold all the data sets listed for your system, then the criterion for a split (between TVOL1A and TVOL1B, for instance) is that the IPCS and change migration libraries are kept together on the first volume (TVOL1A).

- ▶ Distribution libraries (DLIBs) for product sets, on the following volumes:
 - DLIB volumes for target volumes (which include TVOL1, TVOL2-n, HFS, and zFS)
 - DLIB volume for the licensed product sets
 - DLIB volume for the vendor product sets
 - DLIB volumes for the subsystem product sets

Tip: Place data sets on the DLIB volumes wherever they fit. There does not need to be a correlation between TVOL1 and the DLIB volume for TVOL1, or between TVOL2 and the DLIB volume for TVOL2, and so forth. It is possible, but not necessary, to SMS-manage the data sets.

- ▶ Image-related data sets, on the following volumes:
 - Page data sets volume 1
 - Page data sets volume 2 through n
 - HFS or zFS customization volume

Remember: These data sets contain unshareable system image information. Although it is preferable to put them on separate volumes, as described next, if DASD is scarce you can combine them but at the expense of performance, availability, or both.

- ▶ Cluster-related data sets, on the following volumes:
 - Master catalog volume (you can also choose to make this an image-related volume)
 - JES checkpoint volume
 - JES spool volume
 - Sysplex volume 1
 - Sysplex volume 2
 - Softcopy volumes

Tip: These are shareable data sets used in a multisystem environment. Use system symbolics in the names of cluster-related data sets for easier maintainability.

Although all cluster-related data sets can be combined on the same volume, it is usually preferable to separate certain data sets from others for performance or availability reasons. For example, the following data sets should usually not be placed on the same volume:

- ▶ Primary and secondary RACF databases
- ▶ JES spool and checkpoint data sets
- ▶ Primary and backup SMS data sets
- ▶ Primary and secondary couple data sets

3.16 Receiving the ServerPac order

```
CPPPPOLI ----- IBM Corporation ----- 23.00.41
OPTION ==> R_

CustomPac Order Management Menu

R RECEIVE      - Receive an Order
I INSTALL       - Install an Order
Order Number ==>          (Leave blank to list uninstalled orders)
D DISPLAY       - Select Orders to Display

Master dialog data set qualifiers: SERVILS1.MASTER

This dialog supports electronic delivery.

*****
* 5751-CS4, 5751-CS5, 5751-CS6, 5751-CS7 and 5751-CS9 *
* Copyright IBM Corp. 1988, 2010 *
*****
```

Figure 3-16 CustomPac primary menu

Receiving the ServerPac order

Before you use the dialog to install your order, perform the following tasks:

- ▶ Become familiar with the dialogs. Consult *IBM ServerPac Using the Installation Dialog*, SA23-2278, in particular the following sections:
 - “Features of the dialog Panels” describes the ISPF Edit settings used, the available primary and line commands, language setting, dynamic help, and diagnostic messages.
 - “Working With Your Order: An Overview of the dialog Activities” summarizes the steps to install a ServerPac order.
 - “Using the Installation Menu” explains the dialog functions provided.

After you start the dialog, the first thing to do is to receive the order by choosing option **R** in the primary menu panel, as shown in Figure 3-16. *Receiving the order* means that you copy the order from tape, or from an FTP server, or from a file system on the driving system to DASD. This unloads the control tables and installation jobs from the shipment medium to your DASD.

Information: HELP (PF1) is available on any panel. The HELP key is a useful online help facility that explains every panel function in detail. Certain panels have **PRIM** and **LINE** commands available. Using the HELP key enables you to obtain a description and example of how to use the commands.

3.17 Order Receive panel

```
CPPP610A ----- Receive an Order -----
COMMAND ==>

Receive the order from ==> T      F - File system
                           S - Server
                           T - Tape

Order Number            ==> RV210008

----- Order Dialog Data Set Allocation Information -----

Data Set Qualifiers    ==> SERVILS1.RV210008      (Must be unique)

Volume Serial           ==> TOTTS5      (Blank for SMS-managed data sets)
- or -
STORCLAS               ==>                  (Blank for non-SMS-managed data sets)

Dialog CLIST Record Format ==> FB      (FB or VB)

Press Enter to continue or End to cancel
```

Figure 3-17 Order Receive panel

Order Receive panel

When you select option **R**, you receive the panel shown in Figure 3-17. The following information is entered on the Order Receive panel:

Receive the order from Here you specify the source medium of your order. It is still possible to receive from tape, but today other media are supported. These are Internet (Server), or from a UNIX System Services file system. Enter F for File System, S for Server, or T for Tape to specify where the ServerPac order resides, as explained here.

File System indicates that you used the Store and Forward download method (download to workstation) and then uploaded the order to the host Download file system outside the dialog. The RECEIVE job generated by this option will retrieve the order directly from the file system and not attempt a download.

Server indicates that the ServerPac order is to be received from an FTP server. Tape indicates that the ServerPac order is to be received from tape.

Order Number Enter your order number, as it was supplied by IBM (two alphabetic characters followed by six numerics). In Figure 3-17, for example, the order number is RV210008.

Data Set Qualifiers This is the HLQ used to allocate the order installation data sets. It is best to include the order number as part of the qualifier.

Volume Serial Enter the VOLSER of the DASD volume that will receive the order data sets.

STORCLAS If you have an SMS-managed environment, enter a valid storage class for the order data sets.

Dialog CLIST Rec. Fm. Dialog CLISTS for the order are supplied in both fixed block (FB) and variable block (VB) formats. FB is the default.

Press Enter after you have filled out the order details. Depending on your choice in the *Receive order from* field, you will get different follow-on panels. Based on the selection in Figure 3-17 on page 204, the next panel looks as shown in Figure 3-18 on page 206.

3.18 Receive an order from tape

```
CPPP6101 ----- Receive an Order From Tape -----
COMMAND ==>

First System Tape Volume Serial ==> M008AA

Tape Unit ==> TAPE3592 (Generic or esoteric tape unit name)

Press Enter to continue or End to cancel
```

Figure 3-18 Receive order from tape panel

Receive an order from tape

If you chose **T** in the previous panel, you will get the receive order from tape panel, as shown in Figure 3-18. In this panel, enter the tape VOLSER of the first tape of your ServerPac delivery.

Important: Enter the VOLSER of the IBM-supplied tape that contains the order-related installation material. This is the tape labeled with the order number and VOLSER of the first system tape in the format MxxxAA, where xxx is the order number.

This is the RIM tape. Ensure that you enter the correct volume serial number, because there is no validation made through the CustomPac dialogs. You also must enter a generic or esoteric tape unit name. The default is 3590. You can obtain this information from your I/O configuration.

After pressing Enter you will be guided to the Edit JOB statement panel, shown in Figure 3-19 on page 207.

3.19 Edit JOB Statement panel

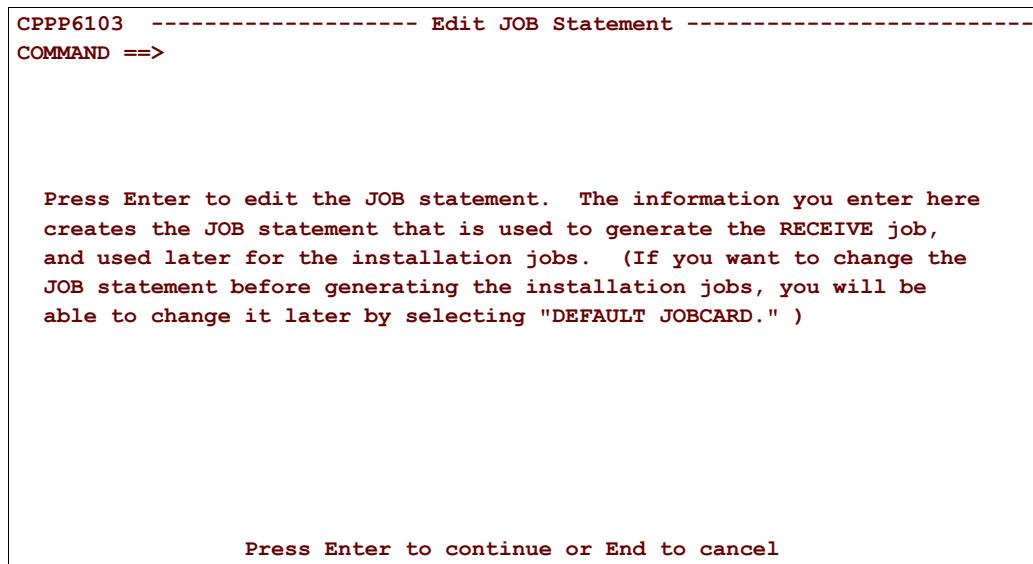


Figure 3-19 Edit JOB Statement panel

Edit JOB Statement panel

The Edit JOB Statement panel is an informational panel. After you press Enter you will see the ISPF editor with the job statements for the receive process, as shown in Figure 3-20 on page 208. At that point, you can make changes to the job statements to meet your requirements.

3.20 Panel with the job statement

```
CPPPEDIF - RV210008 ----- COLUMNS 000 000
COMMAND ==> SCROLL ==> PAGE
***** ***** Top of Data *****
000001 //JOBNAME JOB 'ACCOUNTING INFO','PROGRAMMER NAME',
000002 // CLASS=A,
000003 // MSGCLASS=H,
000004 // MSGLEVEL=(1,1),
000005 // NOTIFY=&SYSUID.,
000006 // USER=&SYSUID.,
000007 // TIME=NOLIMIT,
000008 // REGION=0M
000009 /**
000010 /** MODIFY THE JOB STATEMENT ABOVE AS REQUIRED FOR YOUR INSTALLATION'S
000011 /** REQUIREMENTS, AND PRESS END TO CONTINUE.
000012 /**
***** ***** Bottom of Data *****
```

Figure 3-20 Panel to edit the job statement

Edit job statement panel

After you perform that customization press the PF3 key, which leads you to Figure 3-21 on page 209.

Modify the job statement as required and press PF3. The modifications made to the job statement are typically your installation standards for a job card. The Edit RECEIVE Job panel is then displayed, as shown in Figure 3-21 on page 209.

3.21 Edit RECEIVE Job panel

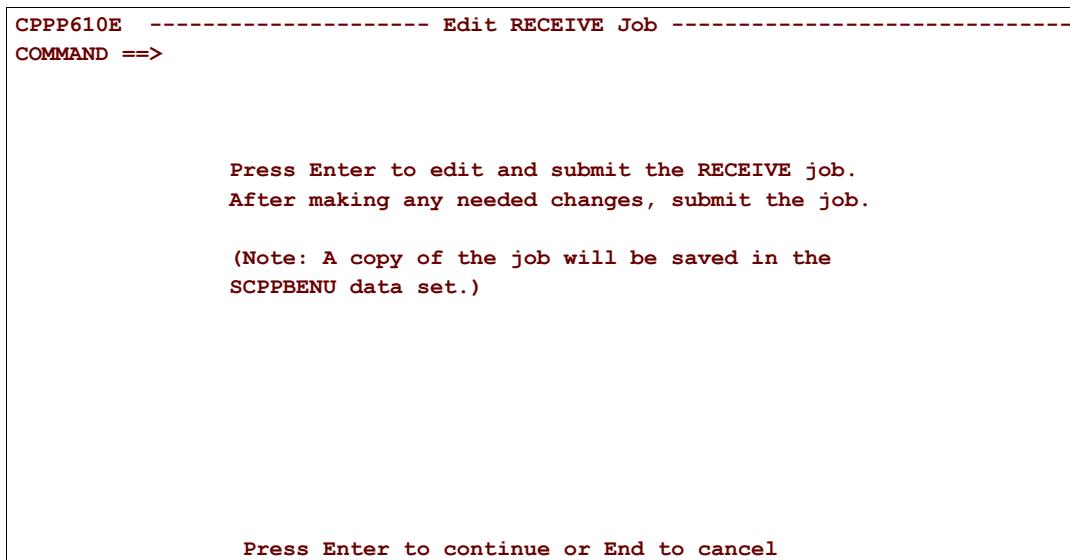


Figure 3-21 Edit RECEIVE Job panel

Edit RECEIVE job panel

After pressing Enter on this panel you receive the next ISPF editor panel, where the generated RECEIVE job is listed. At this point, you can review it and possibly make final changes to the job statements.

When you submit the job from the editor, it performs the following steps:

1. Verifies the status of the CustomPac data sets
2. Possibly deletes duplicate data sets
3. Allocates CustomPac order data sets (VSAM and non-VSAM)
4. Copies CustomPac data sets from the RIM tape to the specified DASD volume
5. Updates the existing order inventory for the next steps

If you run these steps successfully, you can proceed with the installation dialog.

Remember: Figure 3-22 only shows the first statements in the job. The JCL stream is very large. You need to modify this JCL to use your installation standards for creating JCL.

```
CPPPEDIF - RV210008 ----- COLUMNS 000 000
COMMAND ==>                               SCROLL ==> PAGE
***** ***** Top of Data *****
000001 //JOBNAME JOB 'ACCOUNTING INFO','PROGRAMMER NAME',
000002 //      CLASS=A,
000003 //      MSGCLASS=H,
000004 //      MSGLEVEL=(1,1),
000005 //      NOTIFY=&SYSUID,
000006 //      USER=&SYSUID,
000007 //      TIME=NOLIMIT,
000008 //      REGION=OM
000009 /**
000010 /*  MODIFY THE JOB STATEMENT ABOVE AS REQUIRED FOR YOUR INSTALLATION'S
000011 /*  REQUIREMENTS, AND PRESS END TO CONTINUE.
000012 /**
000013 /**
000014 /**
000015 /* * THIS JCL WAS GENERATED BY SKELETON CPPS6101 *
000016 /**
000017 /**
000018 /* Restart instructions:
000019 /**
000020 /* After determining the cause of the failure,
000021 /* and making any necessary corrections, restart
000022 /* this job using the following table as a guide.
000023 /**
000024 /* Failing      Restart      Note
000025 /* Step        Step
000026 /* -----      -----
000027 /* CHKINV      CHKINV
```

Figure 3-22 First statements of the JCL for the RECEIVE job.

Edit the job, if necessary, and save it. To submit the job, enter SUBMIT and then exit the dialog. The job loads your order from tape, FTP server, or file system to your DASD.

Important: You must exit the dialog for the RECEIVE job to run. You do not need to log off from TSO.

3.22 CustomPac Order Management Menu panel

```
CPPPPOLI ----- IBM Corporation ----- 23.00.41
OPTION ==> d

          CustomPac Order Management Menu

R  RECEIVE      - Receive an Order
I  INSTALL       - Install an Order
Order Number ==>           (Leave blank to list uninstalled orders)
D  DISPLAY       - Select Orders to Display

Master dialog data set qualifiers: SERVILS1.MASTER

This dialog supports electronic delivery.

*****
* 5751-CS4, 5751-CS5, 5751-CS6, 5751-CS7 and 5751-CS9  *
* Copyright IBM Corp. 1988, 2010                           *
*****
```

Figure 3-23 Installing a new order panel

Installing a new order

To begin installing an order, select option **I** on the Order Processing panel shown in Figure 3-23.

To go directly to a specific order, enter the order number in the blank field under “INSTALL” and press Enter. Processing proceeds directly to the Installation Menu for the order.

If you leave the order number field blank and press Enter, the dialog displays a list of orders from which to choose.

To list all of the uninstalled orders, select the **I** option, but leave the order number field blank. The dialog displays a list of orders for your selection, as shown in Figure 3-24 on page 212.

Restriction: Although z/OS V1R10 or higher supports extended address volumes (EAVs), the ServerPac dialog does not support EAV in the configuration during installation.

3.23 Selecting an order to install

ORDER LIST							Row 1 to 13 of 13	
							SCROLL ==>	PAGE
Primary Commands: (? SET Locate Find Next Previous OFile SORT VErbose)								
Line Commands: (Select Edit Delete Products Report Output)								
S	Order	System Name	SREL	Package	Build Date	Status	User	--- Last Change ---
								Date
	DO000203	Z038 SERV		2002/09/06	Started	RCONWAY		2002/09/13
	OS141002	P115 EXP		2003/02/05	Started	RC47		2003/02/19
	OS181055	Z038 SERV		2007/03/16	Started	RC47		2007/03/20
	OS181083	P115 EXP		2007/03/20	Started	RC47		2007/03/25
	OS181085	C150 EXP		2007/03/20	Started	RC47		2007/04/02
	OS181103	P115 EXP		2007/03/21	Started	RC47		2007/04/02
	OS190053	Z038 SERV		2007/10/01	Started	RC47		2007/11/09
	RL000006	Z038 SERV		2001/07/01	Started	RC47		2001/07/31
	RM000004	Z038 SERV		2001/12/01	Started	RC47		2002/01/11
	RO150026	Z038 SERV		2003/10/27	Started	RC47		2003/11/12
	RP160023	Z038 SERV		2004/06/22	Started	RC47		2004/07/21
	RQ170027	Z038 SERV		2005/06/13	Started	RC47		2005/07/12
S	RV210008	Z038 SERV		2010/06/16	Received	RC47		2011/06/01
***** Bottom of data *****								

Figure 3-24 Selecting an order to install panel

Selecting an order to install

On the order installation panel, you might have several already received or installed CustomPacs. Select the one that you just received to proceed with the installation process. If it is your first ServerPac installation, you can only select this order number.

Attention: The order number used in these examples is displayed in Figure 3-24 as RV210008.

Select **S** to go into the installation dialog.

Order status

The Order Installation panel indicates installable orders with either of the following status codes (in the Status column):

Received This status is set when the order can be selected on the order list panel. For orders delivered on tape, this is after the RECEIVE job has run. For orders that are downloaded, this is after the first phase of the download has been completed.

Started This status is set when the work configuration has been created.

3.24 Installation dialog

```
CPPPFLOW ----- Installation Options for Order ( RV210008 ) -----
OPTION ==>

Complete these options to install the order:

C   Create           Create the Work Configuration
*  Variables         Specify Values for Variables
*  Zones             Specify SMPTLIB and SMP/E Zone Names Information
*  Modify            Modify the System Layout
*  Alias              Specify Catalogs for High-Level Qualifiers
*  SSA                Specify Temporary Aliases (SSAs) for Catalogs
*  Installation       Create and Submit Installation Jobs

You can use Save any time after creating the work configuration:

*  Save               Save the Current Work Configuration
```

Figure 3-25 Installation dialog

Installation dialog

After selecting a ServerPac order to install, the main installation dialog panel is started, as shown in Figure 3-25. When this panel is shown for the first time during a ServerPac order installation, the only option available is:

- C Choosing option C on this panel enables you to select a configuration for merging an initial installation. If this is your first CustomPac installation, the Create Configuration panel will appear.

The other options are marked with an asterisk (*) and become available when the previous function has successfully finished. Now, select option C.

This panel is the starting point for all of the functions that are needed to install your order.

For an OS order, the Create option displays the Type of Installation panel shown in Figure 3-26 on page 214.

3.25 Choosing the installation type

```
CPPP6015 ----- CREATE Configuration ( RV210008 ) -----
OPTION ==> f

Select the Install type :

F - Full System Replacement installs a complete new IPL-able
     standalone system including all SMP/E-maintained libraries, SMP/E
     environment, operational data sets, and CustomPac sample data sets.
     The supplied operational data sets must be merged with or replaced
     by production operational data sets before the new system is used
     in production.

S - Software Upgrade installs only the SMP/E-maintained libraries,
     SMP/E zones, and CustomPac sample data sets. Operational data sets,
     including system control files (like LOGREC and VTAMLST), a security
     system database, and a master catalog must already exist. These
     existing operational data sets must be updated as required for new
     products and product changes before the first IPL.

For more information about Software Upgrade, enter ? in the option field
```

Figure 3-26 CREATE Configuration panel: Choosing the installation type

Choosing the installation type

When you use the dialog to select an order, its included configuration is added to the dialog. You must select and create a configuration to start the installation. Option F installs a complete z/OS system. It installs all of the data sets needed to perform an IPL, log on to the target system, and run a z/OS image to complete other installation and customization tasks, including the following items:

1. System software and related data sets, such as distribution, target, SMP/E libraries, and so on.
2. System data sets, such as page data sets, system control files, and master catalog.

Because IBM creates a working set of operational data sets for you, a full system replacement helps assure a successful first IPL.

Full system replacement

Choose Full System Replacement, option F. A full system replacement installs a complete z/OS system. It installs all data sets needed to perform an IPL, log on to the target system, and run a z/OS image for completing other installation and customization tasks. These data sets fall into two major categories:

- ▶ System software and related data sets (such as distribution and target libraries, SMP/E CSI data sets, and sample libraries).
- ▶ Operational data sets (such as page data sets, system control files, and a master catalog).

3.26 JES and SDSF element selection

```
CPPP6016 ----- JES Element Selection ( RV210008 ) -----
COMMAND ==>

Choose JES elements to be installed:

JES Elements to Install                      ==> JES2 (JES2, JES3, or BOTH)

Specify options for merging SMP/E target and DLIB zones:

Merge SDSF Zone into JES2 Zone                ==> N      (Y or N)

Merge SDSF/JES2 Zones into BCP Zone          ==> N      (Y or N)

Note: For more information, enter ? in the command field.
```

Figure 3-27 Selecting a JES and SDSF element

Selecting a JES and System Display and Search Facility for the configuration

Select at least one JES element for installation:

- ▶ JES2 and System Display and Search Facility (SDSF)
- ▶ JES3 and SDSF

If your installation requires both JES elements to be installed, you can select BOTH.

For each JES element that you select, specify whether the dialog is to merge the SMP/E zones of the JES element with the BCP zones. When you merge a JES element with the BCP zones, the SDSF zone is automatically merged with the BCP zones:

- ▶ If you specify the JES2 zone to be merged with the BCP zones, both JES2 and SDSF zones are merged with the BCP zones. Similarly, if you specify JES3 to be merged with the BCP zones, both JES3 and SDSF zones are merged with the BCP zones.
- ▶ If you specify both JES elements to be merged with the BCP zones, then the JES2, JES3, and SDSF zones are merged with the BCP zones.

Restriction: If you install both JES elements, you cannot merge only one of the JES elements with the BCP zones. The error message CPPP0601011E will appear to indicate that the action is not allowed.

Migration considerations

If you migrate JES2 or JES3 together with the rest of z/OS, it is preferable to specify Y on this panel to merge the selected JES elements into the BCP zone.

However, if you plan to stage your z/OS and JES migrations separately, do not merge zones. The dialog loads the JES element's zones, but does not merge them.

When you select a single JES, the dialog processes the data sets for only the JES that you select. The dialog does not create or load data sets for the JES you omit.

If you want to change any of these decisions later, you must re-create the configuration by selecting the Create (C) option, specifying the wanted changes, selecting a saved configuration for merge (if applicable), and issuing the **CR** (create) command.

3.27 CREATE Configuration panel

```
CPPP6011 ----- CREATE Configuration ( RV210008 ) ----- Row 1 to 7 of 7
COMMAND ==> CR                                     SCROLL ==> PAGE

Select Configuration

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

S Configuration           Comment
-----
* SERVILS1.RV210008        Always Selected for Order
-----
SERVILS1.DO000203.CONFIG   z/OS R1.4 FOR TRAINER
SERVILS1.OS181055.CONFIG   z/OS R8 FOR TRAINER
SERVILS1.RL000006.CONFIG   z/OS R1.2 FOR TRAINER
SERVILS1.RM000004.CONFIG   z/OS R1.3 FOR TRAINER
SERVILS1.RO150026.CONFIG   z/OS R1.5 FOR TRAINER
SERVILS1.RP160023.CONFIG   z/OS R6 FOR TRAINER
SERVILS1.RQ170027.CONFIG   z/OS R7 FOR TRAINER
*****
***** Bottom of data *****
```

Figure 3-28 CREATE Configuration panel

CREATE Configuration panel

Before you start the installation, you must select and create a configuration. On the CREATE Configuration panel, you can see the master configuration and, if available, other saved configurations. The included configuration is always automatically selected.

Any configurations you have saved are also displayed. Selecting a saved configuration (if you have one) will reduce the time needed to install the order. If you are using the dialog for the first time, simply enter the **CR** (create) primary command. Later in the installation, you will be able to save this configuration for use with subsequent ServerPac orders, or to preserve your work if you decide to re-create the configuration to change installation types or JES selections.

Information: See “Merging a Configuration with a Previous Order” in *IBM ServerPac Using the Installation Dialog*, SA23-2278, for further considerations.

Select configuration

When an order is sent to you, it is installed using control information stored in tables. These are generically referred to as the *shipped order configuration*. The configuration process copies this control information to WORK versions. You can merge variables and installation jobs that you have customized (and saved) for previous orders. From this panel you select the source configurations that will be used to create your WORK version.

You can choose to install ONLY the shipped order configuration or you can MERGE the shipped order configuration with a saved work configuration.

3.28 Installation Variables panel

CPPP6111 ----- Installation Variables (RV210008) - Row 1 to 20 of 153
 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			
	TARGET VOLSER	D	
==> HFS/ZFS INFO			
	INSTALL DIRECTORY	D	/Service
	JAVA PATH	D	/JavaPath
-----display of rest of panel not shown here-----			

Figure 3-29 Installation Variables panel

Installation Variables panel

After you select **CR** (create) as shown in Figure 3-28 on page 217, you then need to customize the Installation Options for Order to complete the install, as shown in Figure 3-25 on page 213. You need to go back through the install process, and when you get to the Installation Options for Order panel (Panelid CPPPFLOW) as shown in Figure 3-30, you see that you can now select Option V to specify values for variables.

After completing the C option, you will then be able to select the V (Variables) option the next time you enter this panel. After you complete each option, you can then select the next option.

```
CPPPFLOW ----- Installation Options for Order ( RV210008 ) -----
OPTION ==>

Complete these options to install the order:

C   Create           Create the Work Configuration
V   Variables      Specify Values for Variables
*   Zones            Specify SMPTLIB and SMP/E Zone Names Information
*   Modify           Modify the System Layout
*   Alias            Specify Catalogs for High-Level Qualifiers
*   SSA              Specify Temporary Aliases (SSAs) for Catalogs
*   Installation     Create and Submit Installation Jobs

You can use Save any time after creating the work configuration:

*   Save             Save the Current Work Configuration
```

Figure 3-30 Installation Options for Order panel

The Variable Selection List displays variables that are required to install your order. The list reflects the installation type you chose (full system replacement or software upgrade). Use this panel to set the values of these variables appropriately for your environment.

See the “Variables” appendix in *ServerPac: Installing Your Order* to see the specific variables for your order along with a brief description of each.

Verify the current contents and enter or change any values by typing over them in the Contents column if a value is either missing or invalid. Note that you cannot update variables with a status of C (customized).

It is useful to read “Defining Installation Variables” in *IBM ServerPac Using the Installation Dialog*, SA23-2278, before changing any installation variable values.

The variable for AUTH.LINKLIB can be an existing authorized library of your installation site. You can use the **VAR edit** command on some panels to change the installation variables later.

3.29 Define ZONE Information panel

```
CPPP6391 ----- Define SMPLIB and Zone Names ( RV      Enter required field
COMMAND ==>                                         SCROLL ==> PAGE

Primary Commands: (? CANCEL SAVE)
Line Commands: (elementS Fmids)
----- SMPLIB Information -----

Data set Prefix ==> SYS1.ZOS
SMS Managed ==> No (Yes/No)

Device Type ==>
Volume(s) ==>      ==>      ==>      ==>      ==>

----- Zone Names Information -----

      S   Nickname   DLIB Zone   Target Zone   SST
      -   -----      -----      -----      -----
          100    MVSD100    MVST100    MVS
          101    MVSD101    MVST101    MVS
          110    MVSD110    MVST110    MVS
          112    MVSD112    MVST112    MVS
***** Bottom of data *****
```

Figure 3-31 Define ZONE Information panel

Define ZONE Information panel

After you complete the V option, you will then be able to select the Z (Zones) option the next time you enter this panel, as shown in Figure 3-32. After you complete each option, you are able to select the next option.

```
CPPPFLOW ----- Installation Options for Order ( RV210008 ) -----
OPTION ==>

Complete these options to install the order:

      C   Create           Create the Work Configuration
      V   Variables        Specify Values for Variables
      Z   Zones            Specify SMPLIB and SMP/E Zone Names Information
      *   Modify           Modify the System Layout
      *   Alias            Specify Catalogs for High-Level Qualifiers
      *   SSA              Specify Temporary Aliases (SSAs) for Catalogs
      *   Installation     Create and Submit Installation Jobs

You can use Save any time after creating the work configuration:

      *   Save             Save the Current Work Configuration
```

Figure 3-32 Installation Options for Order panel

This brings you to the Define ZONE Information panel, where you can define your SMP/E zone configuration. This panel is displayed even if you do not plan to change the included zone names.

Choose zone names

Choose SMP/E zone names that are unique for your installation. Doing so will enable you to use SMP/E's cross-zone processing, such as SMP/E's reporting and management functions. If you want to rename the CSI data set, you can do so later in the Modify System Layout function. This panel enables you to change the names of zones in your order, but not the contents of zones.

After you enter the Z option, as shown in Figure 3-32 on page 220, the dialog displays the current DLIB and target zone names for your order, as shown in Figure 3-31 on page 220.

All DLIB zones for each system release (SREL) are physically defined in the same DLIB CSI data set. All target zones for each SREL are physically defined in the same target CSI data set.

Attention: For descriptions of the zones that came with your order, see "Zones Shipped with Your Order," in *ServerPac: IBM ServerPac Using the Installation Dialog*, SA23-2278.

Be aware that JES zones for a cleared JES element and JES zones that are to be merged with the BCP zones will not appear in the dialog, even though they are listed in *ServerPac: Installing Your Order*.

Changing the SMP/E zone names in your order

In the panel shown in Figure 3-31 on page 220, type over the DLIB and target zone names with the names that you want for your installation and press Enter. The dialog checks for duplicate zone names and, if one is found, displays a message (CPP0639006E DUPLICATE Zone Name) on the panel.

Ensure that the zone names you specify are unique for all environments in your installation. The new zone names should not appear in ZONEINDEX subentries in the global zone ZONE entries for any of your existing systems.

3.30 Modify System Layout Options panel

```
CPPP605T ----- Modify System Layout ( RV210008 ) -----
OPTION ==>

A  Create a Recommended System Layout (Automatically assign target and
    DLIB data sets to physical volumes by data set type)

C  View and change data sets by selected attributes
T  View and change device type table (DEVT)

D  Data Set Summary (SUMD)
M  Merged Data Set Summary (SUMD M)
S  Shipped and Merged Data Set Summary (SUMD S)
U  User Data Set Summary (SUMD U)

V  Physical Volume Summary (SUMP)
L  Logical Volume Summary (SUML)

P  Product, Feature and Element Summary

----- Session Control Options -----
K  Keep Changes made in this dialog session so far (SAVE)
B  Back Out changes from this dialog session (CANcel)
```

Figure 3-33 Modify System Layout Options panel

Modify System Layout Options panel

After you complete the Z option, you will then be able to select the M (Modify) option the next time you enter this panel; see Figure 3-34. After you complete each option, you are able to select the next option.

```
CPPPFLOW ----- Installation Options for Order ( RV210008 ) -----
OPTION ==>

Complete these options to install the order:

C  Create          Create the Work Configuration
V  Variables       Specify Values for Variables
Z  Zones           Specify SMPLIB and SMP/E Zone Names Information
M  Modify         Modify the System Layout
*  Alias           Specify Catalogs for High-Level Qualifiers
*  SSA             Specify Temporary Aliases (SSAs) for Catalogs
*  Installation   Create and Submit Installation Jobs

You can use Save any time after creating the work configuration:

*  Save            Save the Current Work Configuration
```

Figure 3-34 Installation Options for Order panel

Modify the System Layout panel

After you select the **M** option from Figure 3-34, the Modify the System layout panel shown in Figure 3-33 is displayed.

Defining the target system layout is one of the most important steps during order installation. During this part of the dialog, you create the data set layout for your new system. After you have modified this configuration, you can save it for merging with future ServerPac installations.

Modify your configuration in the following order:

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

Following this sequence can help you avoid situations in which volumes become over-allocated.

You can create the new data set layout in one of three ways using the option selections shown in Figure 3-33 on page 222:

- ▶ Option A. Create a Recommended System Layout is used to make the dialog automatically assign the target and DLIB data sets in the configuration to physical volumes. The dialog does not automatically assign any SMS-managed data sets in the configuration.
- ▶ Option C. View and Change is used to assign your order's data sets to volumes by displaying groups of data sets and using the **CHANGE PVOL** command to specify their placement on physical volumes.
- ▶ Other options (D, M, S, U, V, L, and P) are used to assign your order's data sets to logical volumes, and then assign those logical volumes to physical volumes (DASD).

Using the option Recommended System Layout provides a foundation for the ongoing growth and maintenance of your system. When you group your system's data sets by their content and importance to your installation, you help to minimize the complexity of future installations.

Read and use the section “Modifying the System Layout” in *IBM ServerPac Using the Installation Dialog*, SA23-2278. The *ServerPac: Installing Your Order* publication that is included with your order also contains all information relating to the products to be installed.

3.31 SUMMARY of Features/Elements panel

CPPP6051 ----- Modify System Layout (RV210008) - Row 1 to 19 of 135		
COMMAND ==>		SCROLL ==> PAGE
SUMMARY Of Features/Elements		
Primary Commands:(? SET Find Locate Next Previous SORT CANCEL SAVE DEVT SUMD SUML SUMP)		
Line Commands:(Dslist Select)		
S Feature/Element		Data sets
-	-----	-----
z/OS UNIX System Services - Intgd Call		2
d z/OS UNIX System Services Application Se		8
zSecure Base MUL		23
zSecure Visual MUL		2
Appl Wkload Mod Z/OS		7
AFP Download Plus		2
AFP FONT COLLECT 2 ENU		6
AFP TOOLBOX	1.01.0	24
AFP Upload		4
Base		5
BookManager Build ENU		43
BookManager Read ENU		33
BATCHPIPES		19
BCP Base		91
BDT		15
----- more Feature/Elements -----		

Figure 3-35 SUMMARY of Features/Elements panel

SUMMARY of Features/Elements panel

When you select Option P of the Modify System Layout Options panel, shown in Figure 3-33 on page 222, the SUMMARY of Features/Elements panel shown in Figure 3-35 is displayed.

This panel enables you to manually customize individual data sets, logical volumes, and physical volumes. The panel lists the products, features, and elements included in your ServerPac order, and the following CustomPac-specific data set groups:

- ▶ CustomPac SREL-specific SMP/E data sets
- ▶ CustomPac operational and sample data sets
- ▶ CustomPac JES2 data sets
- ▶ CustomPac JES3 data sets
- ▶ CustomPac SMP/E data sets

The panel shows primary and line commands that you can issue to perform the actions you need to customize volumes and data set names. For example, the **Select** command entered next to a product displays the Logical Volume By FEATURE/ELEMENT panel for the selected product, where you can use the line command **Assign** to assign all data set profiles for the selected logical volume to a different logical volume.

The **D** line command returns the Summary of Data Sets panel shown in Figure 3-36 on page 225.

3.32 Summary of data sets

CPPP6052 ----- Modify System Layout (RV210008) ---- Row 1 to 8 of 8 COMMAND ==> SCROLL ==> PAGE																																												
Summary Of FEATURE/ELEMENT : z/OS UNIX System Services Application Se																																												
Primary Commands:(? SET Locate Find Next Previous SORT CHange OFile OList FindComp)																																												
Line Commands:(Merge eXpand Conflict Unmerge Select)																																												
<table><thead><tr><th>S Data Set Name</th><th>X F</th><th>--- Data Set ---</th><th>Primary</th></tr><tr><th></th><th></th><th>Type RECFM LRECL</th><th>Tracks</th></tr></thead><tbody><tr><td>OMVS.ETC</td><td></td><td>ZFS</td><td>446</td></tr><tr><td>OMVS.VAR</td><td></td><td>ZFS</td><td>450</td></tr><tr><td>SYS1.AFOMHDRS</td><td></td><td>PDS FB</td><td>80 25</td></tr><tr><td>SYS1.AFOMOBJ</td><td></td><td>PDS FB</td><td>80 71</td></tr><tr><td>SYS1.AFOMMOD1</td><td></td><td>PDS U</td><td>0 1239</td></tr><tr><td>SYS1.SFOMHDRS</td><td></td><td>PDS FB</td><td>80 20</td></tr><tr><td>SYS1.SFOMOBJ</td><td></td><td>PDS FB</td><td>80 71</td></tr><tr><td>OMVS.ROOT</td><td></td><td>ZFS</td><td>40087</td></tr></tbody></table>					S Data Set Name	X F	--- Data Set ---	Primary			Type RECFM LRECL	Tracks	OMVS.ETC		ZFS	446	OMVS.VAR		ZFS	450	SYS1.AFOMHDRS		PDS FB	80 25	SYS1.AFOMOBJ		PDS FB	80 71	SYS1.AFOMMOD1		PDS U	0 1239	SYS1.SFOMHDRS		PDS FB	80 20	SYS1.SFOMOBJ		PDS FB	80 71	OMVS.ROOT		ZFS	40087
S Data Set Name	X F	--- Data Set ---	Primary																																									
		Type RECFM LRECL	Tracks																																									
OMVS.ETC		ZFS	446																																									
OMVS.VAR		ZFS	450																																									
SYS1.AFOMHDRS		PDS FB	80 25																																									
SYS1.AFOMOBJ		PDS FB	80 71																																									
SYS1.AFOMMOD1		PDS U	0 1239																																									
SYS1.SFOMHDRS		PDS FB	80 20																																									
SYS1.SFOMOBJ		PDS FB	80 71																																									
OMVS.ROOT		ZFS	40087																																									
***** Bottom of data*****																																												

Figure 3-36 Displaying data sets of a feature or element

Summary of data sets of a feature or element

Type D on the Feature/Element line in Figure 3-35 on page 224 to display the panel shown in Figure 3-36. This panel displays a summary of the data sets in the system; the current example shows the Feature/Element for z/OS UNIX System Services.

You can use the Summary of Data Sets panel to perform any of the following tasks:

- ▶ Merge or unmerge ServerPac-shipped data sets (you cannot merge or unmerge user-defined data sets).
- ▶ Modify the attributes of particular data sets, or modify their space information.
- ▶ Make global changes to multiple data sets.
- ▶ Write a list of the data sets in the ISPF LIST data set or a user-defined file.

The **Change** primary command enables you to make global changes to data set profiles. For example, you can change the HLQ for those product data sets. The line command **S** enables you to change data set name, logical volumes space, and BLKSIZE definitions for a specific data set profile.

3.33 Creating a Recommended System Layout

```
CPPP625B ----- Automatic Data Set Assignment ( RV210008 ) -----
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, 3390-9)
Model after Volume ==>            (For example, ZOSRES)
```

Figure 3-37 Automatic Data Set Assignment panel

Creating the Recommended System Layout

When you select option A from the panel shown in Figure 3-33 on page 222, the Recommended System Layout panel shown in Figure 3-37 is displayed.

This option causes the dialog to automatically assign the target and DLIB data sets in the configuration to physical volumes. The dialog does not automatically assign any SMS-managed data sets in the configuration.

Primary commands

The primary commands shown in Figure 3-37 are listed here:

- | | |
|------------------|--|
| A ALL | This setting causes all target and DLIB data sets in the configuration to be automatically assigned to physical volumes. |
| N NEW | This setting assigns only data sets that are new to the configuration. This option preserves the assignment of data sets that already exist on volumes in the saved configuration. |
| P PARTIAL | This setting automatically assigns new data sets, and also enables you to exclude particular volumes from the saved configuration, freeing their data sets for reassignment. |

The dialog does not automatically assign an order's operational data sets or any of the sample CustomPac data sets. You must place these data sets yourself.

Default Device Type

The Default Device Type field specifies the type of device to be used if the dialog creates more volumes for data set assignments. Enter a question mark (?) in the Default Device Type field and press Enter to see a list of other available devices. A configuration can include more than one device type.

Model after Volume

This field specifies a user-defined device type to be used for creating more volumes. This volume must either already be in the configuration, or you must have set the DYNAMIC DASD INFO installation variable to Y and the volume is online.

Installation considerations

Select ALL when you are installing a ServerPac order for the first time, or if you are not using a saved configuration as the basis of your new system. This approach creates a new configuration based only on the new order to be installed.

3.34 Summary of Physical Volumes panel

```

----- Modify System Layout ( RV210008 ) ---- Row 1 to 3 of 3
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 Existing RSVD Assignd Used Free
----- -----
MVSCAT          6027  3390-001           Y       0     0  8320  8320 21731
MVSDLB          6028  3390-001           Y       0     0 19389 19389 10662
MVSRES          6029  3390-001           Y       0     0 24236 24236  5815
***** Bottom of data *****

```

Figure 3-38 Summary of Physical Volumes panel

Summary of Physical Volumes panel

From Figure 3-33 on page 222, when you Enter V or SUMP on the command line, Figure 3-38 displays. This panel provides information about the physical volumes or SMS storage class names for a particular feature or element.

As sent by IBM, a new configuration consists of a target volume (MVSRES), a DLIB volume (MVSDLB), and a catalog volume (MVSCAT). Because MVSCAT contains only operational data sets, this volume is excluded from automatic assignments, and therefore is not shown in the panel display.

Use this panel to verify and change (if necessary) the attributes of the physical volumes. Before you proceed, however, note the following points:

- ▶ If there are warnings on this panel, do not try to resolve them until you have completed any data set merges, space allocation changes, or volume assignment changes that you intend to make.
 - ▶ If you have DYNAMIC DASD INFO set to No, you must use the **Select** line command to display the Display and Change Volume Attributes panel. On that panel you must specify the required information about each volume, and can choose to reserve space on the volume or tell the dialog not to initialize it.
 - ▶ If you have DYNAMIC DASD INFO set to Yes, you can use the **Select** line command to reserve space on the volume or tell the dialog not to initialize it.
 - ▶ These changes should be among the last tasks that you perform before exiting Modify System Layout (because other changes you make could force you to perform rework), but they must be completed before you attempt to run any jobs that install the data sets on your DASD volumes.

Line commands

The following line commands can be used.

- S | SELECT** This command enables you to select physical volumes and SMS storage classes for reassignment:

 - ▶ For a physical volume, the **S** line command displays a panel that enables you to specify a different physical volume, and to change the attributes of the current physical volume.

- ▶ For a storage class, the **S** line command displays a panel that enables you to specify a new storage class. To assign a STORCLAS in place of a physical volume, you must assign the logical volume to the STORCLAS.

D DLIST	This command displays the data sets for the selected physical volume or storage class. The Summary of Data Sets panel is displayed, enabling you to accomplish any of the following tasks:
	<ul style="list-style-type: none"> ▶ Merge or unmerge IBM-supplied data sets. ▶ Make global changes to groups of data sets. ▶ Modify the attributes of particular data sets or modify their space information. ▶ Write a list of the data sets to the ISPF list data set or a user-defined file.

Physical volumes

As shipped by IBM, a new configuration consists of a target volume (MVSRES), a DLIB volume (MVSDLB), and a catalog volume (MVSCAT). Because MVSCAT contains only operational data sets, this volume is excluded from automatic assignments:

MVSCAT	A VOLSER of the DASD on which SMP/E will allocate relative filenames (RELFIES) during RECEIVE processing if products are installed after the ServerPac installation jobs have been run. This value is required for the SMPTLIB DDDEF in the global zone.
MVSDLB	<p>DLIBs: With the ability to define very large volume sizes on certain hardware, it is possible that all your target libraries and DLIBs could fit on a single volume. If possible, use a large enough volume for your target libraries so that you will not need multiple SYSRES logical extension volumes.</p> <p>DLIBs for product sets have a suggested data set placement to be on the following volumes:</p> <ul style="list-style-type: none"> ▶ DLIB volumes for target volumes (which include TVOL1, TVOL2-n, HFS, and zFS) ▶ DLIB volume for the licensed product sets ▶ DLIB volume for the vendor product sets ▶ DLIB volumes for the subsystem product sets <p>Place data sets on the DLIB volumes wherever they fit. There does not need to be a correlation between TVOL1 and the DLIB volume for TVOL1, or between TVOL2 and the DLIB volume for TVOL2, and so forth. It is possible, but not necessary, to SMS-manage the data sets.</p>
MVSRES	<p>As shipped by IBM, a new configuration consists of a target volume (MVSRES). Some products include one or more data sets that should reside either on the first target volume or on one of the last target volumes in the configuration. For example, a product might include several data sets of element type DATA that should reside on the IPL volume, which is always the first target volume.</p> <p>During automatic assignment, the dialog recognizes such data sets and attempts to place them according to this requirement, rather than with other data sets of the same element type.</p>

3.35 Current Volume Configuration panel

```
CPPP625C ----- Automatic Data Set Assignment ( RV210008 ) Row 1 to 3 of 3
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
-  -----  -----  -----  -----  -----  -----  -----  -----
  MVSRES TARGET    T01    3390-27  73 %    85 %    0 %    0 %
  MVSDLB  DLIB     D01    3390-27  59 %    85 %    0 %    0 %
  MVSCAT  BOTH     B01    3390-27  25 %    85 %    0 %    0 %
***** Bottom of data *****
```

Figure 3-39 Automatic Data Set Assignment: Current Volume Configuration panel

Current Volume Configuration panel

As previously mentioned, as shipped by IBM, a new configuration consists of a target volume (MVSRES), a DLIB volume (MVSDLB), and a catalog volume (MVSCAT). MVSCAT contains only operational data sets. This panel is displayed by selecting option A in Figure 3-37 on page 226.

After you select a setting for automatic assignment the dialog displays the current volume configuration, as shown in Figure 3-39.

Command line options

From the command line, you can enter the following commands:

- | | |
|---------------|--|
| RESET | This command can be abbreviated as R . Use this command to reverse any changes you have made to the volumes. This panel is redisplayed as it appeared when you entered this panel for the first time. |
| CREATE | This command can be abbreviated as CR . Use this command to accept the volume information as it is displayed on this panel. This panel is redisplayed after you have confirmed the data set assignment. |

Line command options

In the Current Volume Configuration panel, use line command **S** to select volumes for changes. Use line command **L** to list the data sets currently assigned to a volume. Use line command **I** to insert more volumes into the work configuration.

In this panel, you use line commands in the **S** column for the following purposes:

- | | |
|-----------------|--|
| S SELECT | This selects a volume for an action. If you select a volume for changes, see "Display and change volume attributes panel" on page 232. |
| I INSERT | This adds a volume to this list. |
| L LIST | This lists the data sets on the volume. |
| M MOVE | This moves a volume. Use with BEFORE or AFTER : |
- ▶ **AFTER** places selected volumes after this entry in the list.
 - ▶ **BEFORE** places selected volume before this entry in the list.

Information: When you select the ALL or PARTIAL settings, the following line commands are also available on this panel:

- ▶ **M, A, B:** Move a volume to a different position in the list. To move a volume after another, enter **Move** for the volume to be moved, and **After** for the volume that is to precede it. To move a volume before another, enter **Move** and **Before**.
- ▶ You can move a volume only among other volumes of the same type. For example, you can move a target volume before or after other target volumes, but not into a group of DLIB volumes or BOTH volumes.
- ▶ To move a volume into a group of volumes of a different type, you must first change the volume's type to the other type (target, DLIB, or BOTH) through line command **S**. You can then move the volume in the group of volumes with that type. Moving a volume will cause the dialog to adjust the volume sequence numbers accordingly.

X EXCLUDE This excludes a volume from this list.

If you plan to rename these volumes, select the volumes now through line command **S** and rename them as needed. Later, when the new configuration is created, it is more difficult to rename these volumes. When you select **S** the panel shown in Figure 3-40 on page 232 is shown.

Tip: You can further limit the dialog's use of a volume capacity to less than the default 85% by reserving additional space. For example, if you want the volume to have 25% free space at the end of automatic assignment, you can accomplish this by reserving 10% of the volume's space (334 cylinders for a 3390-3) for the ALL or PARTIAL settings or 15% for the NEW setting.

3.36 Display and change volume attributes panel

```
CPPP625D ----- Automatic Assignment - Attributes ( RV210008 ) -----
COMMAND ==>

Display and Change Volume Attributes

Volume Serial      ==> MVSRES    (Always required)

Volume Type        : TARGET     (Target, DLIB or Both)
Device Type        ==> 3390-001 (For example, 3390-9)

Reserved Space    ==> 0         (Cylinders)

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

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

Figure 3-40 Display and change volume attributes panel

Display and Change Volume Attributes panel

If you select a volume for changes from Figure 3-39 on page 230, the dialog displays the panel shown in Figure 3-40, enabling you to change the volume and save your changes.

Display and Change Volume Attributes panel

This panel enables you to change the attributes of the selected volume, as explained here:

Volume Serial Enter the name of a volume. Use a name that is not already used in the configuration.

Volume Type Enter one of the following values: TARGET, DLIB, or BOTH.

Device Type Enter the physical volume's device type. By default, this value is set to 3390-3.

Reserved Space Enter the amount of space to be reserved on the volume in cylinders (for example, for data set expansions).

Initialize Volume This indicates whether the volume is to be initialized.

Use caution in resetting volumes from **Initialize Volume** NO to **Initialize Volume** YES.

Doing so causes the volume to be initialized by the installation jobs, and any existing data on the volume is lost. Volumes for which **Initialize Volume** is set to NO must be online, even if **DYNAMIC DASD INFO** is set to NO. To save your changes to the volume, press Enter. You are returned to the Current Volume Configuration display.

3.37 Define alias-to-catalog relationships

```
CPPP6021 ----- ALIAS to CATALOG ( RV210008 ) ----- Row 1 from 87
COMMAND ==>                                         SCROLL ==> PAGE

Define CATALOG Dataset Names

Primary Commands: (? SET Locate Find Next Previous SORT CANcel SAVE)
Line Commands: (Delete Insert Repeat)

      S   Alias     STA Target System Catalog Data Set Name          Type
      -   ----- -----
        @PL       CATALOG.TOTICF1.VTOTCAT
        ABJ      CATALOG.TOTICF1.VTOTCAT
        AIO      CATALOG.TOTICF1.VTOTCAT
        AOP      CATALOG.TOTICF1.VTOTCAT
        AOX      CATALOG.TOTICF1.VTOTCAT
        APK      M MCAT.ZOSR1B.Z1BCAT          MCAT
        ASM      M MCAT.ZOSR1B.Z1BCAT          MCAT
        ATX      CATALOG.TOTICF1.VTOTCAT
        AUP      CATALOG.TOTICF1.VTOTCAT
        AUZ      M MCAT.ZOSR1B.Z1BCAT          MCAT
        .....
***** Bottom of data *****
```

Figure 3-41 Define CATALOG Data Set Names panel

Define alias-to-catalog relationships

From the Installation Menu, enter A, Define Alias to Catalog Relationships, as shown in Figure 3-42.

```
CPPPFLOW ----- Installation Options for Order ( RV210008 ) -----
OPTION ==>

Complete these options to install the order:

      C   Create           Create the Work Configuration
      V   Variables        Specify Values for Variables
      Z   Zones            Specify SMPLIB and SMP/E Zone Names Information
      M   Modify            Modify the System Layout
      A   Alias             Specify Catalogs for High-Level Qualifiers
      *   SSA               Specify Temporary Aliases (SSAs) for Catalogs
      *   Installation     Create and Submit Installation Jobs

You can use Save any time after creating the work configuration:

      *   Save              Save the Current Work Configuration
```

Figure 3-42 Installation Options for Order panel

Use the ALIAS to CATALOG panel to specify which HLQ you want to be associated with a catalog. An M in the STA column indicates that this alias name must be associated with a master catalog. If you find several question marks (???????) in the TARGET System Catalog DSName field, it indicates that there is no catalog defined yet. This function enables you also to insert additional user-defined alias names and catalogs.

Tip: Use this panel to define a catalog data set name for each of the aliases in your order.

Figure 3-43 displays how the ALIAS to CATALOG panel first appears. Enter catalog names under the Target System Catalog Data Set Name field, as shown in Figure 3-41 on page 233. The catalog names are initially shown as ??????? because they are not yet defined.

You can specify the catalog name with which an alias is to be associated by typing over the question marks. The Type column specifies MCAT if the alias is associated with the master catalog. If present in your order, the SYS1 alias is automatically associated with the master catalog.

CPPP6021 ----- ALIAS to CATALOG (RV210008) ----- Row 1 from 87					
COMMAND ==> SCROLL ==> PAGE					
Define CATALOG Dataset Names					
Primary Commands:(? SET Locate Find Next Previous SORT CANcel SAVE)					
Line Commands:(Delete Insert Repeat)					
S	Alias	STA	Target System Catalog	Data Set Name	Type
-	-	-	-	-	-
	@PL		?????.CATALOG		
	ABJ		?????.CATALOG		
	AIO		?????.CATALOG		
	AOP		?????.CATALOG		
	AOX		?????.CATALOG		
	SYS1	M	?MASTER.CATALOG		MCAT

Figure 3-43 ALIAS to CATALOG panel

STA column

The STA (status) column shows one of these values or a blank:

- Blank** This indicates that the HLQ can be associated with any catalog. Associate these HLQs with the master catalog or a user catalog.
- M** This indicates that the HLQ must be associated with the master catalog.
- U** This indicates that the HLQ is not associated with any of the data sets in the work configuration. You created the HLQ with the **R** or **I** line command on this panel. You can associate the HLQ with any catalog in the configuration.

Define an HLQ for catalogs

In this option, you specify the catalog data set name for each *alias*. The dialog and the installation process for ServerPac use the standard order of catalog search when defining and locating data sets. Therefore, there must be an alias in the target system master catalog for each high-level qualifier used for data sets that are to be cataloged in a user catalog. Also, there must be aliases in the driving system master catalog for the SSAs that you chose to use when installing the order.

To associate the master catalog with an alias, type over ?MASTER.CATALOG with the actual master catalog name. Then, to use the master catalog for another alias, enter the ?MCAT command. To use a catalog name that is already defined for another alias, enter a question mark (?), followed by the alias name (for example, ?SYS1).

Important: Before you begin to define the alias-to-catalog relationships and SSA-to-catalog relationships, read Chapters 9 and 10 in *IBM ServerPac Using the Installation Dialog*, SA23-2278, to become familiar with using SSAs and the catalog structure. Also, those chapters contain worksheet models you can use to plan:

- ▶ The catalog and alias names, and their relationships
- ▶ The catalog names and their associated SSAs

After specifying the alias-to-catalog relationship, you can select **SSA** on the Installation panel, which leads you to the SSA-to-Catalog panel shown in Figure 3-44 on page 236.

3.38 Define system-specific alias (SSA)

```
CPPP6031 ----- SSA to CATALOG ( RV210008 ) ----- Row 1 to 1 of 1
COMMAND ==> SCROLL ==> PAGE

CATALOG Selection List

Primary Commands: (? CANcel SAVE)
Line Commands: (Select)

          Allocate  Define
          |   |
S Catalog Name      SSA Name Type VOLume |   | Unit
----- -----
UCAT.CATALOG        MCAT           Y   Y
***** Bottom of data *****
```

Figure 3-44 Defining system-specific aliases

Define system-specific alias

From the Installation Menu, enter **SSA**, Specify Temporary Aliases (SSAs) for Catalogs, as shown in Figure 3-45. Figure 3-44 is then displayed.

```
CPPPFLOW ----- Installation Options for Order ( RV210008 ) -----
OPTION ==>

Complete these options to install the order:

C   Create          Create the Work Configuration
V   Variables       Specify Values for Variables
Z   Zones           Specify SMPLIB and SMP/E Zone Names Information
M   Modify          Modify the System Layout
A   Alias            Specify Catalogs for High-Level Qualifiers
SSA SSA          Specify Temporary Aliases (SSAs) for Catalogs
*   Installation    Create and Submit Installation Jobs

You can use Save any time after creating the work configuration:

*   Save             Save the Current Work Configuration
```

Figure 3-45 Installation Options for Order panel

Use the Catalog Selection List panel shown in Figure 3-44 to specify an SSA for each catalog. The SSAs you specify here are used to create alias entries for these catalogs in the driving system's master catalog.

Important: For a software upgrade installation, the target system's master catalog must already exist (the Allocate Catalog field is set to N and cannot be changed). User catalogs can be new or existing catalogs.

When you first display the Catalog Selection List panel, you might see a message warning you that you have not defined at least one SSA. If you try to exit this panel before completing the SSA definitions, the message reappears. Whether you receive this message or don't, ensure that the needed SSA definitions exist and that any SSA definitions inherited from a saved configuration are still valid.

The following line command is valid for this panel:

S This selects a catalog, enabling you to specify:

- SSA
- When an SSA is to be used for this catalog, the SSA name
- Whether the catalog is to be allocated
- When the catalog is to be allocated, the unit and volume
- When the catalog is to be allocated, its size

Enter line command **S** for a data set name on this panel and press Enter to display the next panel, which is described in Figure 3-47 on page 238.

After you define the SSAs, the dialog displays the Catalog Selection List panel again with the updated SSA information. It will now include the SSA Name, as shown in Figure 3-46.

----- SSA to CATALOG (RV210008) ----- Row 1 to 1 of 1																						
COMMAND ==>		SCROLL ==> PAGE																				
CPP9999977I UPDATE Request Cancelled																						
CATALOG Selection List																						
 Primary Commands:(? CANcel SAVE) Line Commands:(Select)																						
 <table><thead><tr><th>S Catalog Name</th><th>SSA Name</th><th>Type</th><th>VOLUME</th><th>Allocate</th><th>Define</th></tr></thead><tbody><tr><td>UCAT.CATALOG</td><td>SMPESSA</td><td>MCAT</td><td>MVSCAT</td><td>Y</td><td>Y 3390</td></tr><tr><td>*****</td><td>*****</td><td>*****</td><td>*****</td><td>*****</td><td>*****</td></tr></tbody></table> Bottom of data *****					S Catalog Name	SSA Name	Type	VOLUME	Allocate	Define	UCAT.CATALOG	SMPESSA	MCAT	MVSCAT	Y	Y 3390	*****	*****	*****	*****	*****	*****
S Catalog Name	SSA Name	Type	VOLUME	Allocate	Define																	
UCAT.CATALOG	SMPESSA	MCAT	MVSCAT	Y	Y 3390																	
*****	*****	*****	*****	*****	*****																	

Figure 3-46 SSA Name now displayed in the panel

When you are satisfied with the results, return to the Installation Menu panel Figure 3-25 on page 213.

3.39 Define SSA and CATALOG Data panel

```
CPPP6033 ----- SSA to CATALOG ( RV210008 ) -----
COMMAND ==>

Define SSA and CATALOG Data

Catalog : UCAT.CATALOG
Type    : MCAT

Define SSA      ==> Y      (Y or N)
Allocate Catalog ==> Y      (Y or N)

SSA Name        ==> SMPPESSA

Catalog Volume   ==> MVSCAT (? For List of Available Vols)

If allocating the catalog, the following information is required:

Primary Space   ==> 12      (1-999 Cylinders)
Secondary Space  ==> 12      (1-999 Cylinders)
```

Figure 3-47 Define SSA and CATALOG Data

Define SSA and CATALOG Data panel

The process you use to define SSAs depends on which of the following types your installation is:

- ▶ Full system replacement
- ▶ Software upgrade

See the section that applies to your installation in *IBM ServerPac Using the Installation Dialog*, SA23-2278, as listed here:

- ▶ “Defining SSA and catalog data for full system replacement”
- ▶ “Defining SSA data for software upgrade”

The panel shown in Figure 3-47 is displayed when you use full system replacement, option F on the panel shown in Figure 3-26 on page 214.

The panel fields are as follows:

Catalog Name of the catalog.

Type Type of catalog. MCAT indicates a master catalog; UCAT indicates a user catalog.

Define the following fields:

Define SSA Set this value to Y (yes) to define a new SSA in the driving system's master catalog. Set this value to N (no) if the SSA is already defined in the driving system's master catalog. If you set the Allocate Catalog field to Y, you must set the Define SSA field to Y.

Allocate Catalog This specifies whether to allocate the catalog. Set this value to Y (yes) if the catalog does not yet exist on the target system, and is to be allocated. Set this value to N (no) if the catalog already exists on the target system; it cannot be allocated again. All catalog names must be unique.

SSA Name	This specifies the system-specific alias to be used to locate target system data sets that will be cataloged in the specified catalog. The name you choose must not have an existing alias entry in the driving system's master catalog, and cannot be the same as the high-level qualifier of any data set cataloged in the driving system's master catalog.
Catalog Volume	<p>This specifies the VOLSER of the DASD on which the catalog is to reside. For a pop-up list of volumes that exist in the work configuration, enter a question mark (?). Select a volume by entering S before the wanted volume serial. Note that you are not required to choose a volume from the list.</p> <p>Any VOLSER can be specified; if it does not exist, it is added to the configuration. If the volume you specified already exists in the work configuration, the unit you specify here must match the unit value you specified previously. Otherwise, a message is displayed to prompt you to resolve the mismatch.</p>
Space	This specifies the primary and secondary space to be allocated to the catalog. See "Determining Catalog Size" in <i>z/OS DFSMS Managing Catalogs</i> , SC26-7409, for information about how much space to allocate for catalogs. If you did not specify that the catalog is to be allocated (you set the Allocate Catalog field to N), do not enter values for these fields.

This is the end of the customization steps for the ServerPac. You are now ready to run the supplied installation jobs. From the Installation Menu, enter I and the panel in Figure 3-48 on page 240 will be displayed.

3.40 Job Selection List panel

```
CPPP6121 ----- Installation JOBs ( RV210008 ) --- Row 1 to 19 of 334
COMMAND ==>                                     SCROLL ==> PAGE

JOB Selection List                               SS$( EXCLUDE )

Primary Commands:(? SET Locate Find Next Previous GENskel Ofile OList SUMmary
                     SS$ VARedit)
Line Commands:(Backup Delete Edit Insert Log Output Select SS-block Vars)

S      Description          STEP     MC STAtus    RC
---  -----
SRC DEFAULT JOBCARD

==> INSTALLATION JOBS
DOC RUNNING INSTALLATION JOBS
DOC INSTALLATION SETUP
JOB INITIALIZE REQUIRED DASD                  OFFLIMIT 00
JOB INITIALIZE SMS VOLUME                     INITSMS 00
DOC DEFINE CATALOGS AND RESTORE
JOB RACF PROFILES ON DRIVING SYSTEM           RACFDRV 00
JOB DEFINE CATALOGS                          DEFCAT   00
JOB DEFINE SYSTEM-SPECIFIC ALIASES            DEFSSA   00
JOB ALLOCATE AND CATALOG DS                 ALLOCDS 00
JOB RESTORE DATA SETS                        RESTORE 04
JOB UPDATE DEFAULT DATA SET NAMES            CPPUPDT 04
DOC Z/OS 1.12 UNIX APPL SERV
DOC DEFINE NEW SMP/E ENVIRONMENT
JOB UPDATE NEW SMP/E DDDEFS                  UPDDDD 04

==> POST-INSTALLATION - FROM DRIVING
```

Figure 3-48 Job Selection List panel

Job Selection List panel

From the Installation Menu enter I, Create and Submit Installation Jobs, as shown in Figure 3-48. Figure 3-49 is then displayed.

```
CPPPFLOW ----- Installation Options for Order ( RV210008 ) -----
OPTION ==>

Complete these options to install the order:

C  Create          Create the Work Configuration
V  Variables       Specify Values for Variables
Z  Zones           Specify SMPTLIB and SMP/E Zone Names Information
M  Modify          Modify the System Layout
A  Alias           Specify Catalogs for High-Level Qualifiers
SSA SSA           Specify Temporary Aliases (SSAs) for Catalogs
I  Installation    Create and Submit Installation Jobs

You can use Save any time after creating the work configuration:

*  Save             Save the Current Work Configuration
```

Figure 3-49 Installation Options for Order panel

There are three types of components shown on the Installation Jobs panel:

SRC	Source data such as parameter lists
DOC	Documentation
JOB	Executable JCL

The installation steps are grouped into the following sections:

- ▶ Package-specific installation
- ▶ Product-specific installation
- ▶ Post-installation
- ▶ Additional post-installation
- ▶ Customization section
- ▶ Installation verification section
- ▶ Cleanup jobs
- ▶ Migration section
- ▶ Customer-specific customization

When you enter the Installation Jobs panel for the first time, the installation jobs have still not been generated. All installation jobs are generated using ISPF tailoring services. Use the **GENSKEL** command on the command line to tailor all of the installation jobs at one time, as shown in Figure 3-50 on page 242.

When **GENSKEL** completes, the dialog saves the jobs in a backup data set. For more information about the commands you can use on this panel, see *IBM ServerPac Using the Installation Dialog*, SA23-2278.

Information: At this stage of the installation process, your primary source of information is *ServerPac: Installing Your Order*.

3.41 GENERATE File Tailored Installation Jobs panel

```
CPPP6126 ----- Installation JOBS ( RV210008 ) -----
COMMAND ==>

GENERATE File Tailored Installation Jobs

This function generates a BATCH job which will file tailor
ALL Installation Jobs in one pass, and save the jobs to the
BACKUP dataset.

If a job already exists in the backup dataset

REPLACE Job ==> Y (Y or N)

Note: After submitting the GENSKEL job, you must exit
      the dialog to release GENSKEL processing. Also,
      to avoid dataset contention, you may not invoke
      the dialogs until the GENSKEL job has completed.
```

Figure 3-50 GENERATE File Tailored Installation Jobs panel

GENERATE File Tailored Installation Jobs panel

When you enter the **GENSKEL** command, the panel shown in Figure 3-50 is displayed.

The **GENSKEL** command submits a batch job, which generates all of the installation jobs. Each job is stored in the SCPPBENU data set that is provided through the ServerPac RECEIVE process.

For z/OS orders, **GENSKEL** processing can take as much as 30 minutes or longer to complete. Subsystem orders might need only several minutes to complete.

The installation jobs should be submitted in sequence. Always read the DOC section before you select and submit the related jobs. All installation steps and jobs are also described in *ServerPac: Installing Your Order*.

File-tailored jobs might already exist in the SCPPBENU data set. Set the Replace Job field to Y to replace jobs; set the field to N to preserve them.

You must tailor each of the installation jobs before you submit them. You can use the **GENSKEL** command to tailor installation jobs in the background. **GENSKEL** generates the installation jobs for your order, and stores them in the backup data set (SCPPBENU). Having the installation jobs stored in SCPPBENU enables you to review them, if wanted, after the order is installed.

File-tailored jobs might already exist in the SCPPBENU data set. Specify whether **GENSKEL** is to replace these jobs or preserve them. Set the Replace Job field to Y to replace jobs; set the field to N to preserve them.

Press Enter to generate the **GENSKEL** job. Then, submit the job with the **SUBMIT** command. Exit the dialogs until the **GENSKEL** job completes. You will see the JCL shown in Figure 3-51 on page 243.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
ISREDDE2 ROGERS.CPPTEMP1.SCPPWORK Columns 00001 00072
Command ==> Scroll ==> PAGE
***** ***** Top of Data *****
000001 //SERVRP1 JOB (999,POK),'PAUL ROGERS',
000002 // CLASS=A,
000003 // MSGCLASS=T,
000004 // MSGLEVEL=(1,1),
000005 // NOTIFY=ROGERS,
000006 // TIME=NOLIMIT,
000007 // REGION=0M
000008 /*JOBPARM S=SC47,L=9999
000009 /*
000010 /* MODIFY THE JOB STATEMENT ABOVE AS REQUIRED FOR YOUR INSTALLATION'S
000011 /* REQUIREMENTS, AND PRESS END TO CONTINUE.
000012 /*
000013 /*
000014 //JOBLIB DD DSN=SERVILS1.RV210008.SCPPLOAD,
000015 // DISP=SHR

```

Figure 3-51 First part of the file tailoring JCL you need to submit

For your convenience, the Job Selection List panel displays only the installation jobs that apply to your order's installation type (full system replacement or software upgrade). To view the entire set of jobs generated by GENSKEL, including jobs that are not needed for your installation type, check the SCPPBENU data set. (The GENSKEL output lists all of the generated jobs in the SYSPRINT output data set.) To access these jobs, enter line command **B** on the Installation Jobs display.

Important: After submitting the GENSKEL job, you must exit the dialog to release GENSKEL processing. Also, to avoid data set contention, you cannot start the dialogs until the GENSKEL job has completed.

Return back into the dialog

You reenter the dialog and select Option I, as shown in Figure 3-52.

```

CPPPFLOW ----- Installation Options for Order ( RV210008 ) -----
OPTION ==>

Complete these options to install the order:

C   Create           Create the Work Configuration
V   Variables        Specify Values for Variables
Z   Zones            Specify SMPLIB and SMP/E Zone Names Information
M   Modify           Modify the System Layout
A   Alias            Specify Catalogs for High-Level Qualifiers
SSA SSA             Specify Temporary Aliases (SSAs) for Catalogs
I   Installation    Create and Submit Installation Jobs

You can use Save any time after creating the work configuration:

*   Save              Save the Current Work Configuration

```

Figure 3-52 Installation Options for Order panel

3.42 List of installation jobs

```
CPPP6121 ----- Installation JOBs ( RV210008 ) --- Row 1 to 19 of 334
COMMAND ==>                                         SCROLL ==> PAGE

JOB Selection List                                     SS$( EXCLUDE )

Primary Commands: (? SET Locate Find Next Previous GENskel OFile OList SUMmary
                     SS$ VARedit)
Line Commands: (Backup Delete Edit Insert Log Output Select SS-block Vars)

S      Description          STEP     MC STAtus    RC
----- -----
SRC DEFAULT JOBCARD

==> INSTALLATION JOBS
DOC RUNNING INSTALLATION JOBS
DOC INSTALLATION SETUP
JOB INITIALIZE REQUIRED DASD           OFFLINIT 00
JOB INITIALIZE SMS VOLUME             INITSMS  00
DOC DEFINE CATALOGS AND RESTORE
JOB RACF PROFILES ON DRIVING SYSTEM   RACFDRV  00
JOB DEFINE CATALOGS                  DEFCAT   00
JOB DEFINE SYSTEM-SPECIFIC ALIASES    DEFSSA   00
JOB ALLOCATE AND CATALOG DS         ALLOCDS  00
JOB RESTORE DATA SETS                RESTORE  04
JOB UPDATE DEFAULT DATA SET NAMES   CPPUPDT  04
DOC z/OS 1.12 UNIX APPL SERV
DOC DEFINE NEW SMP/E ENVIRONMENT
JOB UPDATE NEW SMP/E DDDEFS          UPDDDD  04

==> POST-INSTALLATION - FROM DRIVING
```

Figure 3-53 Installation jobs

Installation jobs

At this stage of the installation process, your primary source of information is *ServerPac: Installing Your Order*. When you have finished running the installation jobs and verification programs (IVPs) described in *ServerPac: Installing Your Order*, you have completed the installation.

Figure 3-53 displays the jobs and documentation that are required to install your order, based on the installation type that you chose earlier in the installation (full system replacement or software upgrade).

The “Jobs used to install the ServerPac” table in *ServerPac: Installing Your Order* lists all of the jobs and indicates the installation type to which they apply (full system replacement or software upgrade).

The job selection list includes the maximum return code MC of the STEP, the status of the STEP, and the actual return code of the STEP. The RC field shows SYST for a system abend.

3.43 Displaying the processing log

CPPP6125 ----- Installation JOBs (RV210008) ---- Row 1 to 11 of 11									
COMMAND ==>				SCROLL ==>					
PAGE									
Processing LOG									
PRIM Cmds: (? SET Locate First Next Previous SORT)									
LINE Cmds: (Output)									
S	STEPname	JOB name	job ID	RC	UserID	DATE stamp			
-	-----	-----	-----	-----	-----	-----			
	ALLOCDS	SERVRP93	JOB24592	ROGERS	11/06/23 23:18:21				
	CALLLINK	SERVRP97	JOB24817	ROGERS	11/06/24 20:13:10				
	CPPUPDT	SERVRP95	JOB24814	ROGERS	11/06/24 19:54:06				
	DEFSSA	SERVRP92	JOB24591	ROGERS	11/06/23 22:37:21				
	DELSSA	SERVRP9B	JOB24838	ROGERS	11/06/24 21:11:57				
	RECATDS	SERVRP99	JOB24822	ROGERS	11/06/24 20:42:03				
	RESTORE	SERVRP94	JOB24678	ROGERS	11/06/24 08:24:37				
	UPDDDD	SERVRP96	JOB24816	ROGERS	11/06/24 20:02:38				
	UPDDDUV	SERVRP98	JOB24819	ROGERS	11/06/24 20:28:14				
	VERIFY	SERVRP9	JOB24589	ROGERS	11/06/23 22:35:09				

Figure 3-54 Processing Log Panel: SUMMARY command

Displaying the processing log

If you enter the **SUMMARY** primary command at the Job Selection List shown in Figure 3-53 on page 244, the Processing Log panel is displayed, as shown in Figure 3-54. This panel lists the jobs that were submitted from the job stream and their respective return codes. If output logging is active, you can browse the job output.

This panel lists the jobs that were submitted from the job stream and their respective return codes. If output logging is active, you can browse the job output. After a job's completion, the job output can be seen using the **Output** line command.

The job copying data sets to SystemPac Volumes (RESTORE) can run for a long time, depending on the number of products your ServerPac order contains. You should have two tape drives and all of the tape cartridges shipped with your order available before you start the RESTORE job.

Post-installation and customization is product-dependent and installation-dependent, and should be related to your specific requirements.

After the installation jobs have completed, you should be able to perform an IPL and test your new z/OS system.

3.44 SAVE Configuration panel

```
CPPP6041 ----- SAVE Configuration ( RV210008 ) -----
COMMAND ==>

Specify SAVE Library

Enter the High Level Qualifier of the Library to which the
Order Configuration will be Saved

==> SERVILS1.RV210008.CONFIG

The default qualifier used is 'OrderHLQ'.
You may enter a Comment to identify the Configuration. This
is recommended if you use a qualifier other than the default.

==> FIRST SETUP

MASTER HLQ is : SERVILS1.MASTER
```

Figure 3-55 SAVE Configuration panel

SAVE Configuration panel

After you install a ServerPac order, you can use the Save Used Configuration function of the dialog to save your work configuration. This can help you save time in installing subsequent ServerPac orders. Rather than manually reentering all of the data required for each new order, you can merge the saved configuration with the new order and avoid much of the data entry.

Specify the HLQ for the configuration. The configuration data set is appended with either of the following low-level qualifiers:

- ▶ SCPPSENU for skeleton libraries
- ▶ SCPPTENU for table libraries

If the libraries do not exist, the dialog prompts you to confirm that the libraries can be allocated. If the libraries already exist, a new panel is displayed to confirm the deletion of the old libraries.



z/OS storage concepts

This chapter describes the following basic IBM z/OS storage concepts:

- ▶ Virtual storage and address spaces
- ▶ How processor storage is managed by z/OS
- ▶ How virtual storage is managed by z/OS
- ▶ Address space map for 31-bit and 64-bit
- ▶ System address spaces
- ▶ Dynamic address translation
- ▶ Residence Mode and Addressing Mode
- ▶ Subsystem definitions
- ▶ Multiprogramming and multitask
- ▶ Module object and load module
- ▶ Memory hierarchies

For more detailed information about virtual storage concepts, see *ABCs of z/OS System ABCs of z/OS System Programming Volume 10*, SG24-6990.

4.1 Processor storage overview

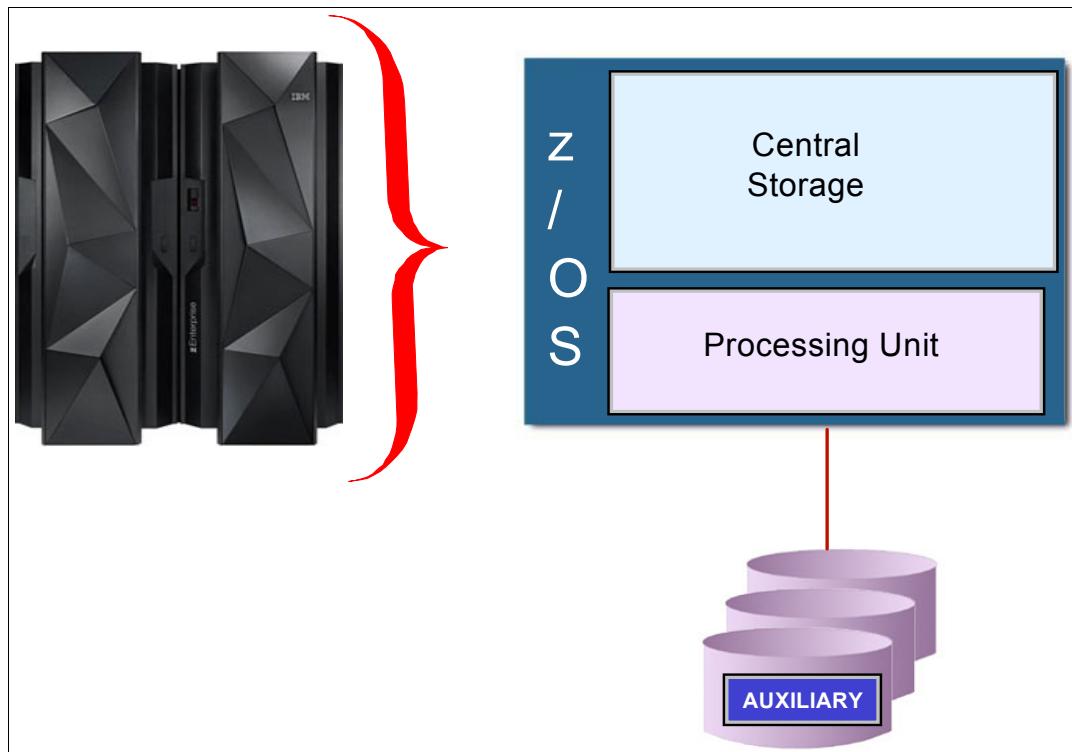


Figure 4-1 Overview of processor storage

Processor storage overview

Logically, a system consists of main storage and one or more processors. The system administers the use of processor storage and directs the movement of virtual storage pages between auxiliary storage slots and real frames in blocks of 4096 bytes. It makes all addressable virtual storage in each address space appear as main storage. Only the virtual pages necessary for program execution are kept in main storage.

The remainders exist on auxiliary storage. The system employs the auxiliary storage manager to perform the actual paging input/output (I/O) necessary to transfer pages in and out of main storage. The system also provides direct access storage device (DASD) allocation and management for paging space on auxiliary storage.

The system assigns real frames upon request from pools of available frames, thereby associating virtual addresses with main storage addresses. Frames are repossessed when freed by a user, when a user is swapped-out, or when needed to replenish the available pool. Although a virtual page occupies a real frame, the page is considered pageable unless it is fixed by the FIX option of the PGSER macro, a PGFIX macro, a PGFIXA macro, or obtained from a fixed subpool.

The system also allocates virtual equals real (V=R) regions upon request by those programs that cannot tolerate dynamic relocation. Such a region is allocated contiguously from a predefined area of main storage, and it is nonpageable.

Information: The terms *processor storage*, *main storage*, *real storage*, and *memory* always refer to *physical memory*, which is the one that resides in the processor cage.

The technology of main storage has the following characteristics:

- ▶ The processor is directly accessible (for programs and data).
- ▶ It is volatile, fast, and expensive, when compared with magnetic storage (DASD or tape).

Information: Storage is viewed as a long horizontal string of bits. For most operations, accesses to storage proceed in a left-to-right sequence. The string of bits is subdivided into units of eight bits. An eight-bit unit is called a *byte*, which is the basic building block of all information formats.

IBM /Architecture

In IBM z/Architecture, there is no expanded storage, because the 31-bit real address limitation is relieved to a 64-bit real address (up to 16 exabytes (EB) of real addresses). However, the largest mainframe (the IBM zEnterprise 196, or z196) has a total memory capacity of up to 3 terabytes (TB).

The system initialization process begins when the system operator selects the LOAD function at the system console. This causes an initial program load (IPL), which is equivalent to a boot in other platforms. z/OS locates all of the usable main storage that is online and available in the logical partition (LPAR) that has undergone IPL, creating a virtual storage environment for the building of various system areas. z/OS uses main storage to map the virtual storage, which implies allocating and using auxiliary storage.

Main storage

Main storage (sometimes referred to as *central storage*), provides the system with a volatile processor that is directly addressable, with fast access for the electronic storage of data.

Because of the volatile property of main storage, modern mainframes have an integrated battery feature (IBF) that keeps main storage running so operators can perform normal shutdown procedures in power failure situations.

Both data and programs must be loaded into main storage (from magnetic devices, such as disks and tapes) before they can be processed by the processors. The maximum main storage size per LPAR is restricted by hardware and the z/OS and system architecture.

Auxiliary storage

Auxiliary storage consists of z/OS paging data sets (files) located in DASD. Note that DASD in mainframe terminology is referred to as a *hard disk drive* on other platforms. DASD is a non-volatile magnetic memory made of iron oxide. Paging data sets are used to implement virtual storage, which contain the paged-out portions of all virtual storage address spaces.

In addition, output to virtual I/O (VIO) devices can also be stored in the paging data sets. The concept of address spaces is described in 4.2, “Virtual storage concepts” on page 250. The concept of VIO is described in 4.10, “Auxiliary storage manager” on page 265.

Processing unit

Figure 4-1 on page 248 depicts a processing unit (PU). The PU is the hardware in charge of executing instructions located in main storage. The PU contains the sequencing and processing facilities for instruction execution, interruption action, timing functions, IPLs, and other machine-related functions. PUs and main storage are packed in units known as *books* in the CEC cage in a z9 machine.

4.2 Virtual storage concepts

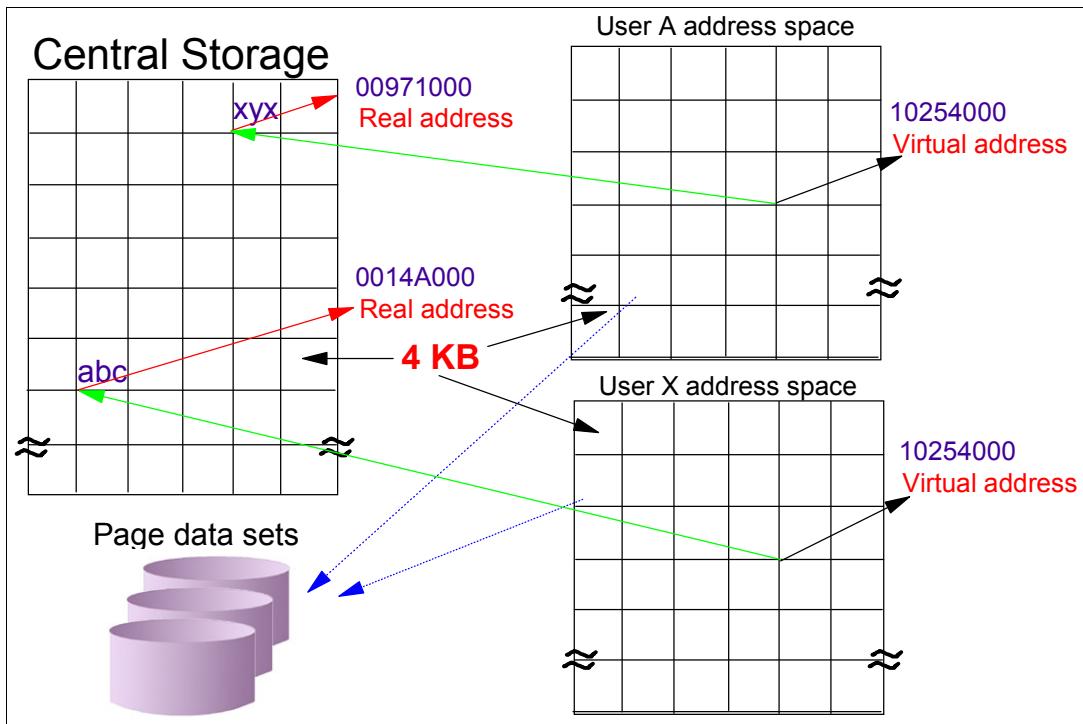


Figure 4-2 Virtual storage, auxiliary storage

Virtual storage concepts

Virtual storage is an illusion created by z/Architecture together with z/OS, such that the program seems to have more main storage than it really has. Each user or program gets an address space, and each address space contains the same range of storage addresses. Only those portions of the address space that are needed at any point in time are actually loaded into main storage. z/OS keeps the inactive pieces of address spaces in auxiliary storage.

z/OS manages address spaces in units of various sizes:

- ▶ Page address spaces are divided into 4 kilobyte (KB) units of virtual storage called *pages*.
- ▶ Segment address spaces are divided into 1 megabyte (MB) units called *segments*. A segment is a block of sequential virtual addresses spanning megabytes, beginning at a 1 MB boundary. A 2-gigabyte (GB) address space, for example, consists of 2048 segments.
- ▶ A virtual address, accordingly, is divided into four principal fields:
 - Bits 0-32 are called the region index (RX).
 - Bits 33-43 are called the segment index (SX).
 - Bits 44-51 are called the page index (PX).
 - Bits 52-63 are called the byte index (BX).
- ▶ An address (called virtual) as referred to by a program is an identifier of a required piece of information in main storage. This enables the size of an address space (all virtual addresses available to a program) to exceed the main storage size.
- ▶ All main storage references are made in terms of virtual storage address.

- ▶ A hardware mechanism called dynamic address translation (DAT) is employed to perform a mapping between the virtual storage address and its physical location in main storage.
- ▶ When a requested address is not in main storage (that is, there are more virtual addresses than bytes in main storage), an interruption is signaled and the required data is brought into memory.

Virtual storage terminology

In z/OS, virtual storage is implemented (transparently to the program) using the following concepts:

- | | |
|----------------|--|
| Segment | Program address space is divided into segments of 1 MB addresses in size. |
| Page | A segment is divided into pages, which are blocks of 4 KB addresses in size. |
| Frame | Main storage is divided into frames, which are blocks of 4 KB in size. |
| Slots | Auxiliary storage page data sets are formatted in slots, which are blocks of 4 KB in size. |
| DAT | DAT is implemented by hardware and by software throughout page tables and segment tables. |

A z/OS program accesses addresses located in virtual storage. Only pages of the program currently active need to be in a main storage frame at processing time. The inactive pages are held in auxiliary storage.

Page data sets

Page data sets contain the paged-out portions of all virtual storage address spaces. In addition, output to VIO devices can be stored in the paging data sets. Before the first IPL, an installation must allocate sufficient space on paging data sets to back up the following virtual storage areas:

- ▶ Primary storage for the pageable portions of the common area

A *common area* is a data-only section that can be shared by multiple address spaces. z/OS is structured around address spaces, which are ranges of addresses in virtual storage. Each user of z/OS gets an address space containing the same range of storage addresses. The use of address spaces in z/OS supports isolation of private areas in different address spaces for system security, yet also allows for inter-address space sharing of programs and data through a common area accessible to every address space.
- ▶ Secondary storage for duplicate copies of the pageable common area

The system uses the system's page data sets to keep track of auxiliary storage slots. Specifically:

 - Slots for virtual storage pages that are not in main storage frames
 - Slots for pages that do not occupy frames but, because the frame's contents have not been changed, the slots are still valid.
- ▶ Space for all address spaces that are, or were, swapped out

Swapping is the process of transferring all of the pages of an address space between main storage and auxiliary storage. A swapped-in address space is active, having pages in main storage frames and pages in auxiliary storage slots. A swapped-out address space is inactive; the address space resides on auxiliary storage and cannot execute until it is swapped in.
- ▶ VIO data sets that are backed by auxiliary storage

VIO is designed to reduce the need for the processor to transfer data between DASD and main storage. In this way, all three speed up the execution of your programs.

4.3 Frames, slots, and pages

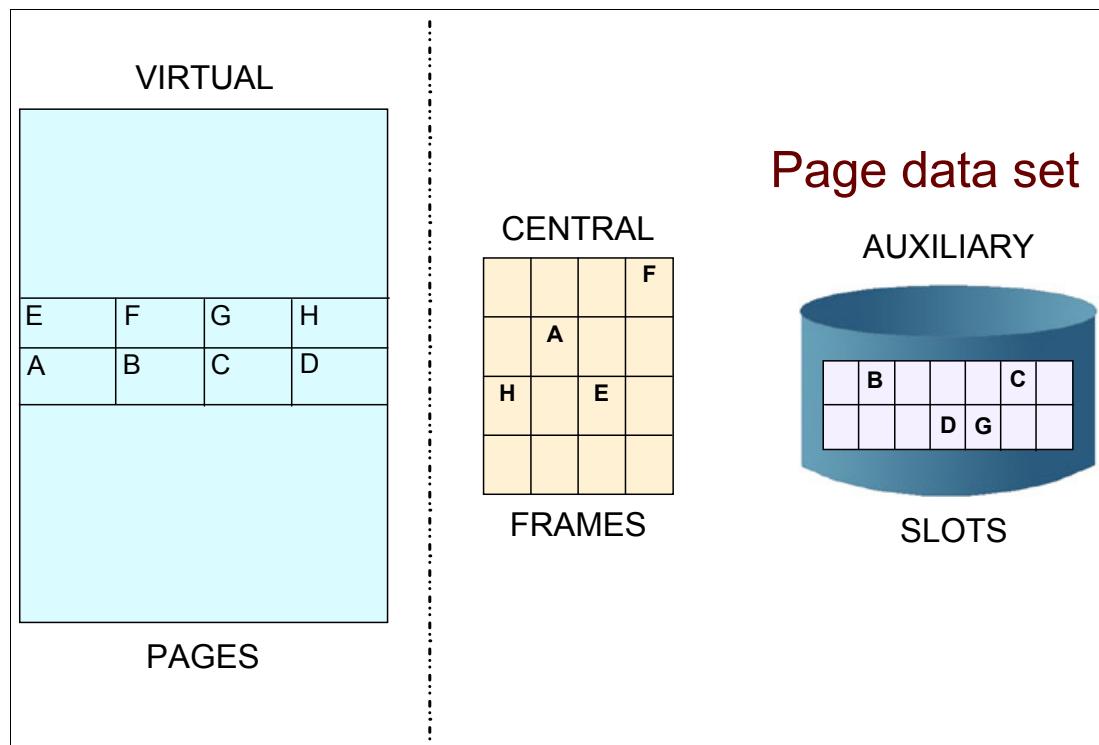


Figure 4-3 Storage frames, slots, and pages

Frames, slots, and pages

When a program is selected, z/OS brings it into virtual storage (enabling it to use a range of virtual addresses) and divides it into pages of 4000 addresses. Then, z/OS transfers the pages of the program into main storage for execution, and out to auxiliary storage when not needed when main storage is under contention.

Actually, not all pages of a program are necessarily in main storage at one time. To the programmer, the entire program appears to occupy a contiguous space of addresses in main storage at all times. The pages that are in main storage, however, do not necessarily occupy contiguous space.

Pages to auxiliary storage (paging)

To understand how paging works, assume that DAT encounters an invalid page table entry during address translation, indicating that a page is required that is not in a main storage frame. To resolve this page fault, the system must bring the page in from auxiliary storage.

First, however, it must locate an available main storage frame. If none is available, the request must be saved and an assigned frame freed. To free a frame, the system copies its contents to auxiliary storage, and marks its corresponding page table entry as invalid. This operation is called a page-out.

The parts of a program executing in virtual storage must be moved between real and auxiliary storage. To enable this, z/OS breaks the storage into blocks of 4000:

- ▶ A block of 4 KB of main storage is a *frame*.
- ▶ A block of 4 KB addresses in virtual storage is a *page*. A virtual storage is backed by:
 - Main storage
 - Auxiliary storage
- ▶ A block of storage on an auxiliary device is a *slot*.

Frames, pages, and slots

Frames, pages, and slots are all the same size (4 KB). An active virtual storage page resides in a main storage frame. A virtual storage page that becomes inactive resides in an auxiliary storage slot (in a paging data set). Figure 4-3 on page 252 shows the relationship of pages, frames, and slots.

In Figure 4-3 on page 252, z/OS is performing paging for a program running in virtual storage. The lettered boxes represent parts of the program. In this simplified view, program parts A, E, F, and H are active and running in main storage frames. Program parts B, C, D, and G are inactive and have been moved to auxiliary storage slots. All of the program parts, however, reside in virtual storage and have virtual storage addresses.

Storage management

z/OS tries to keep an adequate supply of available main storage frames on hand. When a program refers to a page that is not in main storage, z/OS uses a main storage page frame from a supply of available frames. See 4.10, “Auxiliary storage manager” on page 265.

When this supply becomes low, z/OS uses page stealing to replenish it. It takes a frame assigned to an active user and makes it available for other work. The decision to steal a particular page is based on the activity history of each page currently residing in a main storage frame. Pages that have not been active for a relatively long time are good candidates for page stealing.

Paging algorithms

z/OS uses a sophisticated paging algorithm to efficiently manage virtual storage based on which pages were most recently used. An unreferenced interval count indicates how long it has been since a program referenced the page. At regular intervals, the system checks the reference bit for each page frame. If the reference bit is off (the frame has not been referenced), the system adds to the frame's unreferenced interval count. It adds the number of seconds since this address space last had the reference count checked.

If the reference bit is on, the frame has been referenced, and the system turns it off and sets the unreferenced interval count for the frame to zero. Frames with the highest unreferenced interval counts are the ones most likely to be stolen.

4.4 z/Architecture address space

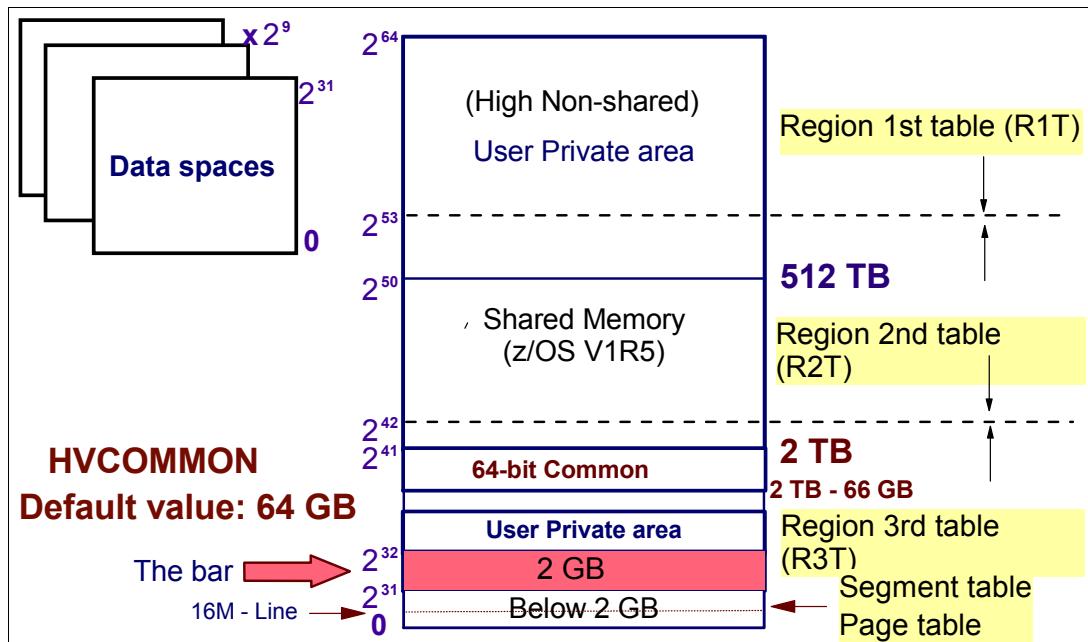


Figure 4-4 z/Architecture address space

z/Architecture address space

The two GB address in the address space is marked by a virtual line called *the bar*. The bar separates storage below the two GB address, called *below the bar* from storage above the two GB address, called *above the 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*.

Your installation can set a limit on the use of the address space above the bar for a single address space. The limit is called the MEMLIMIT in the SMFPRMxx parmlib member.

Note: IBM System/370 (S/370) defined storage addresses as 24 bits in length, which meant that the highest accessible address was 16,777,215 bytes (or 224-1 bytes). The use of 24-bit addressability allowed IBM Multiple Virtual Storage (MVS)/370, the operating system (OS) at that time, to allot to each user an address space of 16 MB. Over the years, as MVS/370 gained more functions and was asked to handle more complex applications, even access to 16 MB of virtual storage fell short of user needs.

With the release of the System/370-Extended Architecture (XA) in 1983, IBM extended the addressability of the architecture to 31 bits. With 31-bit addressing, the OS (now called MVS/XA) increased the addressability of virtual storage from 16 MB to 2 GB. MVS/XA provided an address space for users that was 128 times larger than the address space provided by MVS/370. The 16 MB address became the dividing point between the two architectures, and is commonly called the *line* or *boundary*.

Virtual storage and 64-bit addressability

A program running in an address space can reference all of the storage associated with that address space. A program's ability to reference all of the storage associated with an address space is called *addressability*.

z/Architecture and z/OS operating systems

The new architecture did not require clients to change existing application programs. To maintain compatibility for existing programs, MVS/XA remained compatible for programs originally designed to run with 24-bit addressing on MVS/370, but enabled application developers to write new programs to use the 31-bit technology.

To preserve compatibility between the different addressing schemes, MVS/XA did not use the high-order bit of the address (Bit 0) for addressing. Instead, MVS/XA reserved this bit to indicate how many bits would be used to resolve an address: 31-bit addressing (Bit 0 on) or 24-bit addressing (Bit 0 off).

zSeries mainframes and 64 bit addressing

With the release of IBM zSeries mainframes in 2000, IBM further extended the addressability of the architecture to 64 bits. With 64-bit addressing, the potential size of a z/OS address space expands to a size so vast that you need new terms to describe it, as shown in Figure 4-4 on page 254.

Each address space, called a *64-bit address space*, is 16 EB in size. An *exabyte* is slightly more than one billion gigabytes. The new address space has logically 264 addresses. It is 8 billion times the size of the former 2 GB address space, or 18,446,744,073,709,600,000 bytes.

A program running on z/OS and the zSeries mainframe can run with 24-, 31-, or 64-bit addressing (and can switch among these if needed). To address the high virtual storage available with the 64-bit architecture, the program uses 64-bit-specific instructions. Although the architecture introduces unique 64-bit exploitation instructions, the program can use both 31-bit and 64-bit instructions, as needed.

Region tables and segment tables

In a 16 EB address space with 64-bit virtual storage addressing, there are three additional levels of translation tables called *region tables*:

- ▶ Region third table (R3T)
- ▶ Region second table (R2T)
- ▶ Region first table (R1T)

The region tables are 16 KB in length, and there are 2048 entries per table. Each region has 2 GB.

Segment tables and page table formats remain the same as for virtual addresses below the bar. When translating a 64-bit virtual address, after the system has identified the corresponding 2 GB region entry that points to the segment table, the process is the same as that described previously.

4.5 The address space concept

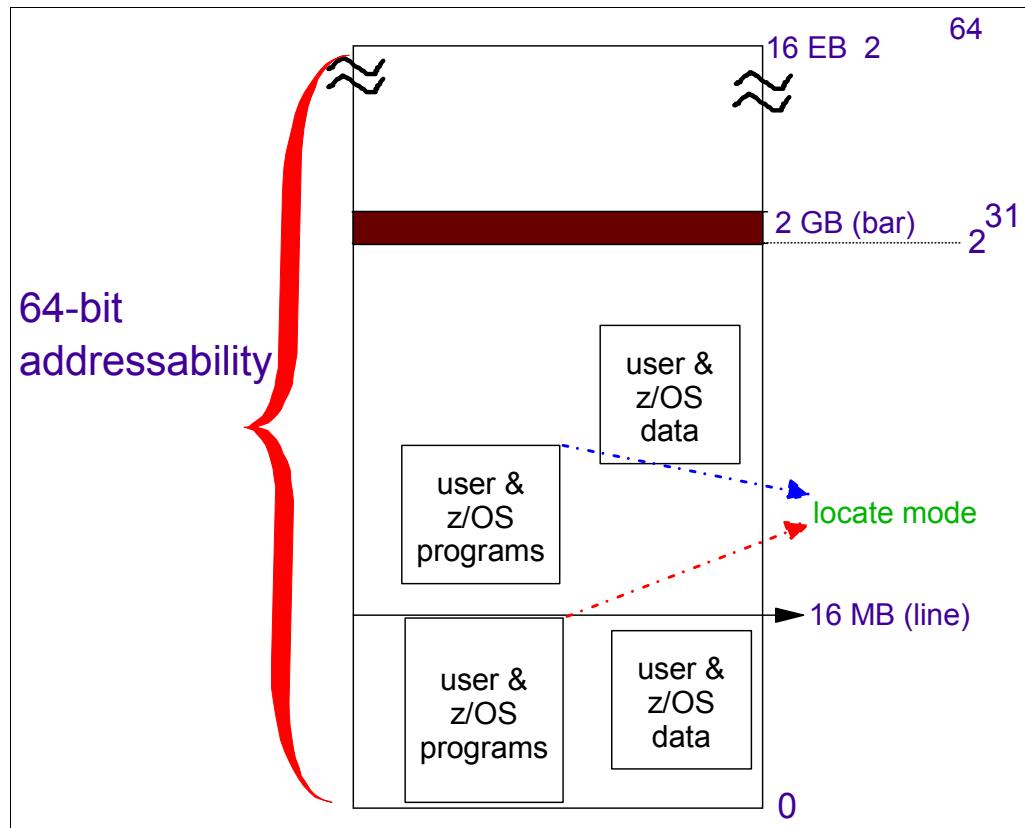


Figure 4-5 Address space concept

Address space concept

The range of virtual addresses that the operating system assigns to a user or separately running program is called an *address space*. This is the area of contiguous virtual addresses available for executing instructions and storing data. The range of virtual addresses in an address space starts at zero and can extend to the highest address permitted by the OS architecture.

Previous IBM architectures

S/370 was the first IBM architecture to use virtual storage and address space concepts. The address space size is decided by the length of the fields that keeps such addresses. Because it maps all of the available addresses, an address space includes system code and data, and user code and data. Therefore, not all of the mapped addresses are available for user code and data. The S/370 architecture used 24 bits for addressing. So, the highest accessible address in the MVS/370 was 16 MB, which was also the address space size.

With MVS/XA, the XA architecture extended to 31 bits for addressing, and the address space size went from 16 MB to 2 GB, which is 128 times bigger. The 16 MB address became the division point between the two architectures, and as mentioned is commonly called the line.

For compatibility, programs running in MVS/370 should run in MVS/XA, and new programs should be able to use the new technology. So, the high-order bit of the address (4 bytes) is *not* used for addressing, but rather to indicate to the hardware how many bits are used to solve an address: 31 bits (bit 32 on) or 24 bits (bit 32 off).

However, the use of multiple virtual address spaces in z/OS provides special benefits. Virtual addressing permits an addressing range that is greater than the main storage capabilities of the system. The use of multiple virtual address spaces provides this virtual addressing capability to each job in the system by assigning each job its own separate virtual address space. The potentially large number of address spaces provides the system with a large virtual addressing capacity.

z/OS z/Architecture

With z/OS, the z/Architecture extended to 64 bits and the address space size went from 2 GB to 16 Exabytes, which is 8 billion times bigger. As mentioned, the area above 2 GB address is called the bar. The addresses above the bar are used for data only.

4.6 Addressing mode and residence mode

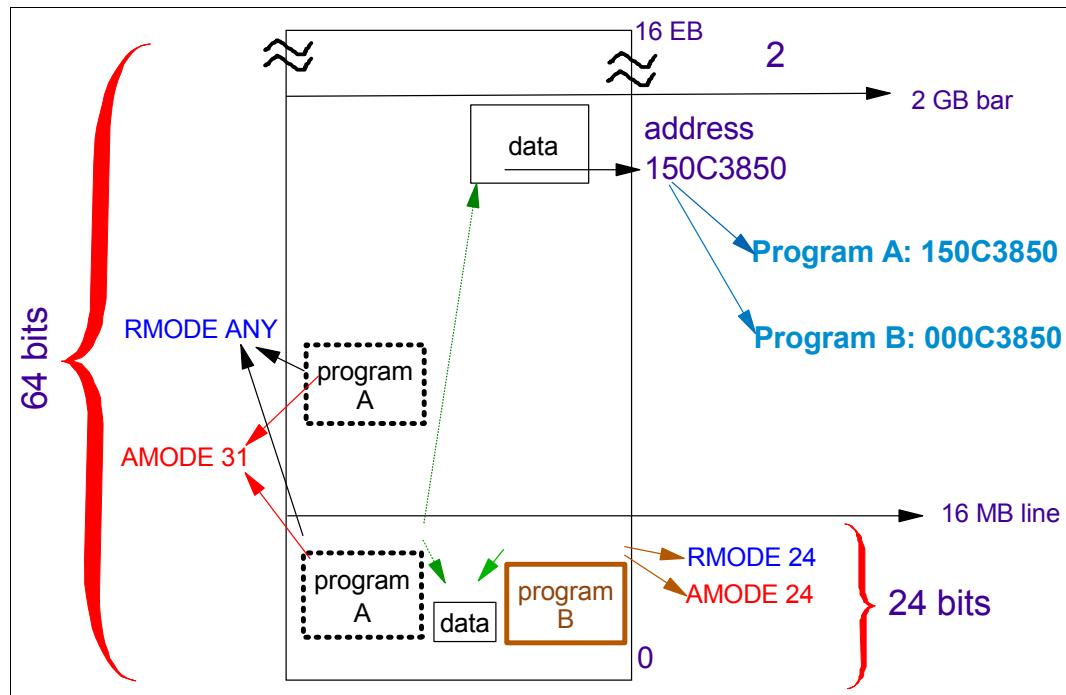


Figure 4-6 Addressing mode and residence mode

Addressing mode and residence mode

MVS/XA introduced the concept of *addressing mode* (AMODE). AMODE is a program attribute to indicate which hardware addressing mode should be active to solve an address; that is, how many bits are to be used for solving and dealing with addresses:

- ▶ AMODE=24 indicates that the program can address up to 16 M virtual addresses.
- ▶ AMODE=31 indicates that the program can address up to 2 G virtual addresses.
- ▶ AMODE=64 indicates that the program can address up to 16 Exa virtual addresses (only in z/Architecture).

Residency mode

The concept of *residency mode* (RMODE) is used to indicate where a program is to be placed in the virtual storage (by z/OS program management) when the system loads it from DASD:

- ▶ RMODE=24 indicates that the module *must* reside below the 16 MB virtual storage line. Among the reasons for RMODE24 is that the program is AMODE24, the program has control blocks that must reside below the line.
- ▶ RMODE=ANY indicates that the module can reside anywhere in virtual storage, but preferentially above the 16 MB virtual storage line. Because of this, such an RMODE is also called RMODE 31. Note that in z/OS there is no RMODE=64, because the virtual storage above 2 GB is not suitable for programs, only data.

Addressing mode

AMODE is a program attribute that can be specified (or defaulted) for each control section (CSECT), load module, and load module alias. AMODE states the addressing mode that is expected to be in effect when the program is entered. AMODE can have one of the following values:

- ▶ AMODE 24. The program is designed to receive control in 24-bit addressing mode.
- ▶ AMODE 31. The program is designed to receive control in 31-bit addressing mode.
- ▶ AMODE ANY. The program is designed to receive control in either 24-bit or 31-bit addressing mode.

Note: AMODE and RMODE are load module attributes assigned when load modules are created by the Binder program and are placed in load module's directory entry in a partitioned data set.

AMODE and RMODE combinations at execution time

At execution time, there are only three valid AMODE/RMODE combinations:

- ▶ AMODE 24, RMODE 24, which is the default
- ▶ AMODE 31, RMODE 24
- ▶ AMODE 31, RMODE ANY

The ATTACH, ATTACHX, LINK, LINKX, XCTL, and XCTLX macros give the invoked module control in the AMODE previously specified. However, specifying a particular AMODE does not guarantee that a module that gets control by other means will receive control in that AMODE. For example, an AMODE 24 module can issue a BALR to an AMODE 31, RMODE 24 module. The AMODE 31 module will get control in 24-bit addressing mode.

4.7 Storage managers

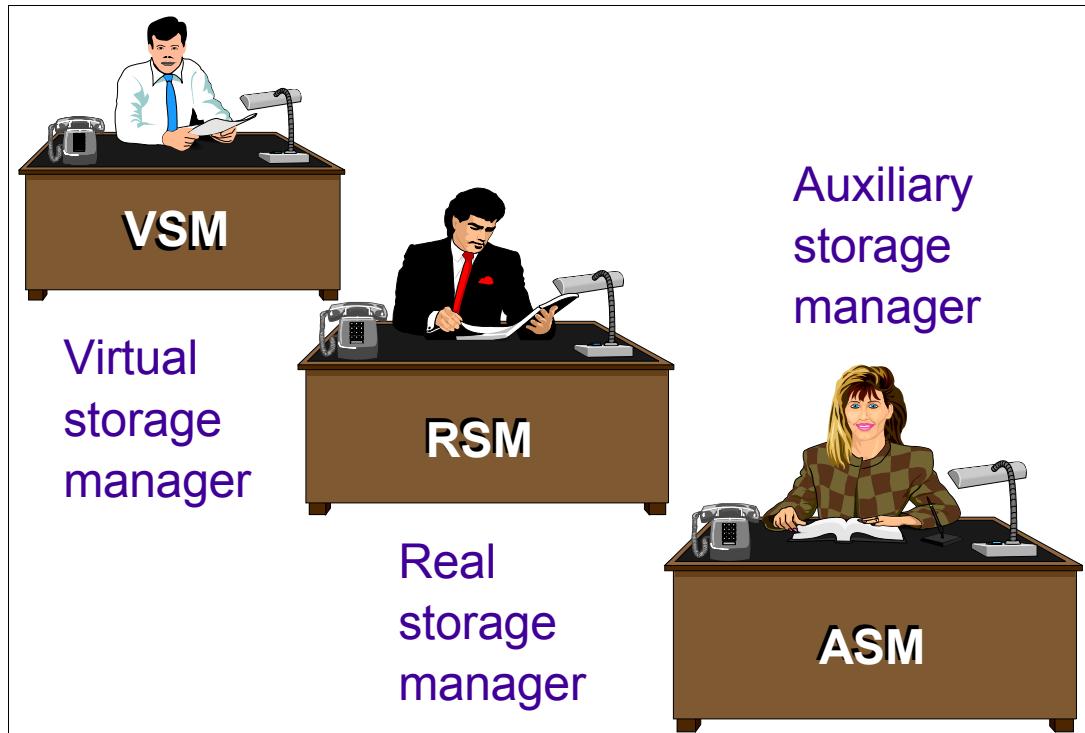


Figure 4-7 The storage component managers

Storage managers

In a z/OS system, storage is managed by the following z/OS components:

- ▶ Virtual storage manager (VSM)
- ▶ Real storage manager (RSM)
- ▶ Auxiliary storage manager (ASM)

Virtual storage manager

Virtual storage is managed by the VSM. The main function of VSM is to control the use of virtual storage addresses by programs. Each installation can use virtual storage parameters (at data set SYS1.PARMLIB) to specify how certain virtual storage areas are to be allocated to programs. These parameters have an effect on main storage use and overall system performance.

VSM keeps track of the map of virtual storage for each address space. In so doing, it sees an address space as a collection of 256 subpools, which are logically related areas of virtual storage identified by the numbers 0 to 255. Being logically related means the storage areas in a subpool share characteristics:

- ▶ Share a storage protect key
- ▶ Whether they are fetch protected, pageable, or swappable
- ▶ Where they must reside in virtual storage (above or below 16 MB)
- ▶ Whether they can be shared by more than one task

Real storage manager

RSM keeps track of the contents of main storage. It manages the paging activities, such as page-in, page-out, page stealing, and helps with swapping an address space in or out. RSM also performs page fixing, which is marking pages as unavailable for stealing.

Auxiliary storage manager

ASM controls the use of page data sets and the implicit paging I/O operation. As a system programmer, you are responsible for the size and the performance of the page data sets. The ASM uses the system's page data sets to keep track of auxiliary storage slots:

- ▶ Slots for virtual storage pages that are not in main storage frames
- ▶ Slots for pages that do not occupy frames but, because the frame's contents have not been changed, the slots are still valid

When a page-in or page-out is required, ASM works with RSM to locate the proper main storage frames and auxiliary storage slots.

4.8 Virtual storage manager

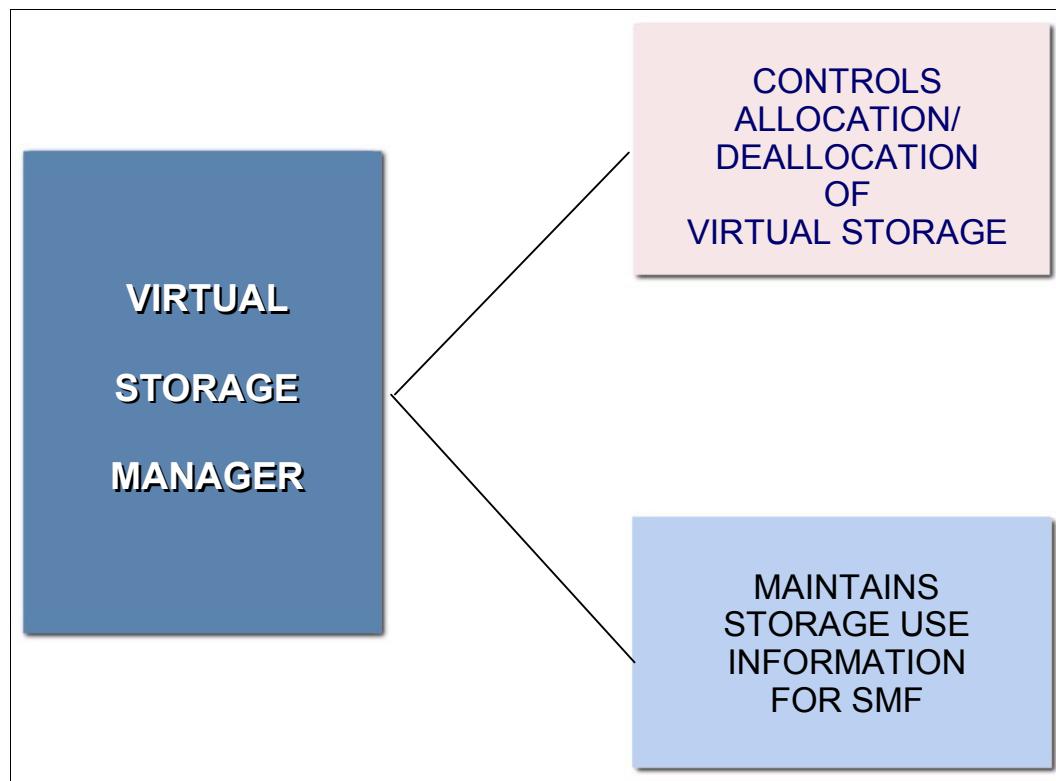


Figure 4-8 Virtual storage manager

Virtual storage manager

VSM is the z/OS component that manages virtual storage. Its main function is to control the use of virtual storage addresses. Virtual storage enables you to write large programs without the need for complex overlay structures.

The existence of an address space does not imply that all virtual addresses are automatically available to programs. Virtual storage addresses must be requested by programs through the use of the GETMAIN or STORAGE OBTAIN macros and returned to the VSM using the FREEMAIN or STORAGE RELEASE macro. The following list describes VSM functions:

- ▶ Allocate and release blocks of virtual storage on request by programs.
- ▶ Ensure that main storage frames exist for naturally fixed pages, such as system queue area (SQA), local SQA (LSQA), common service area (CSA).
- ▶ Maintain storage use information by generating IBM System Management Facility (SMF) records.
- ▶ Associate a storage protection key with each virtual storage block requested. This function is provided through the GETMAIN and STORAGE macros. See *ABCs of z/OS System Programming Volume 10*, SG24-6990, for more information about storage protection.

VSM also provides services that are especially useful when determining available storage, coding recovery procedures, or specifying areas to be included in a dump, as listed here:

- ▶ VSMREGN macro. List the starting address and the size of the private area regions associated with a given program (task).
- ▶ VSMLOC macro. Verify that a given area has been allocated through a GETMAIN or STORAGE macro.
- ▶ VSMLIST macro. List the ranges of virtual storage allocated in a specified area.

4.9 Real storage manager

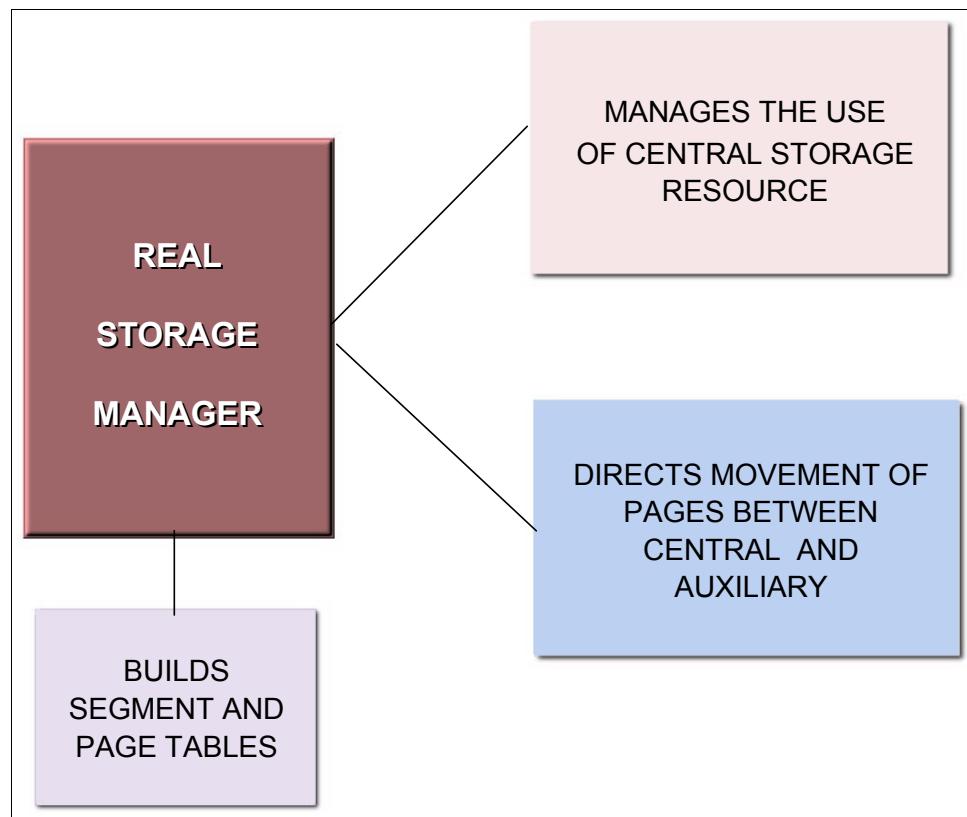


Figure 4-9 Real storage manager

Real storage manager

RSM is the z/OS component that controls the usage of main storage frames. RSM acts together with ASM to support the virtual storage concept, and with VSM to ensure that a page on which GETMAIN has been run is backed up in a main storage frame. Furthermore, RSM establishes many services to other components and application programs to manipulate the status of pages and frames. RSM functions include the following tasks:

- ▶ Allocate frames for pages located in slots during page-in operations.
- ▶ Allocate main storage for page fixing. Fixing a page in a frame means that the page will never be stolen from main storage even if there is a lack of frames. The reason justifying such functionality is availability, not performance.
- ▶ Allocate frames to satisfy GETMAIN requests for SQA, LSQA, and CSA (pages naturally fixed).
- ▶ Build segment and page tables. These tables are used to translate a virtual address to a real address.
- ▶ Work with z/OS Workload Manager (WLM) in page swapping, page stealing, and the unreference interval count (UIC) calculation process.

4.10 Auxiliary storage manager

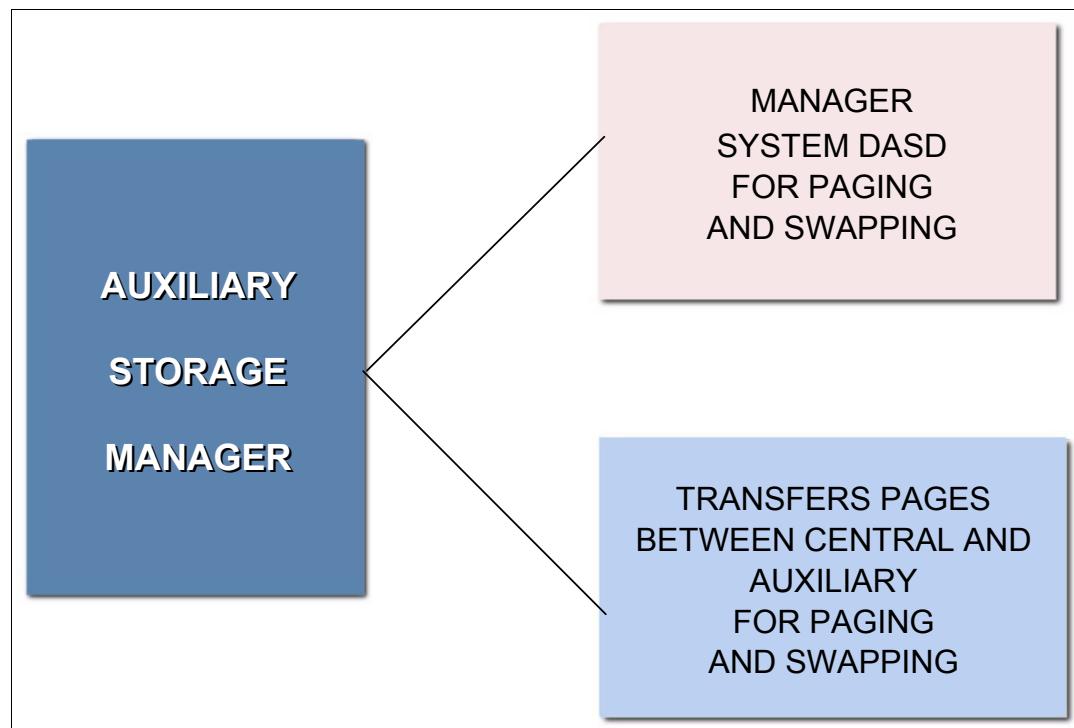


Figure 4-10 Auxiliary Storage Manager

Auxiliary storage manager

ASM is a z/OS component responsible for transferring virtual pages between central frames and auxiliary storage slots (page data sets). This is done as either a paging operation (one page at time), or as a physical swapping operation (an address space, all pages at a time). ASM manages the transfer by initiating the I/O and by maintaining tables to reflect the current status of the slots. This status includes the location of each page in each slots.

To page efficiently and expediently, ASM divides the pages into classes, namely pageable link pack area (PLPA), common, and local. There is at least one page data set for each class. Contention is reduced when these classes of pages are placed on different physical devices. In addition, output to VIO devices can be stored in local paging data sets. VIO are user temporary data sets allocated in central and auxiliary storage. Page data sets are created and formatted by the system programmer through the **DEFINE IDCAMS** command.

ASM attempts to maximize page I/O efficiency by incorporating a set of algorithms to distribute the I/O load evenly (through the local page data sets). In addition, every effort is made to keep the system operable in situations where a shortage of a specific type of slots exists. ASM selects a local page data set for page-out from its available page data sets. ASM selects these data sets in a circular order in each type of data set, subject to the availability of free space and the device response time.

If the address space is physically swapped out directly from main storage to auxiliary storage, ASM reads and writes these working set pages in parallel. Since z/OS V1R8, there is no more physical swapping. See 4.11, “Paging and swapping” on page 266 for more information.

4.11 Paging and swapping

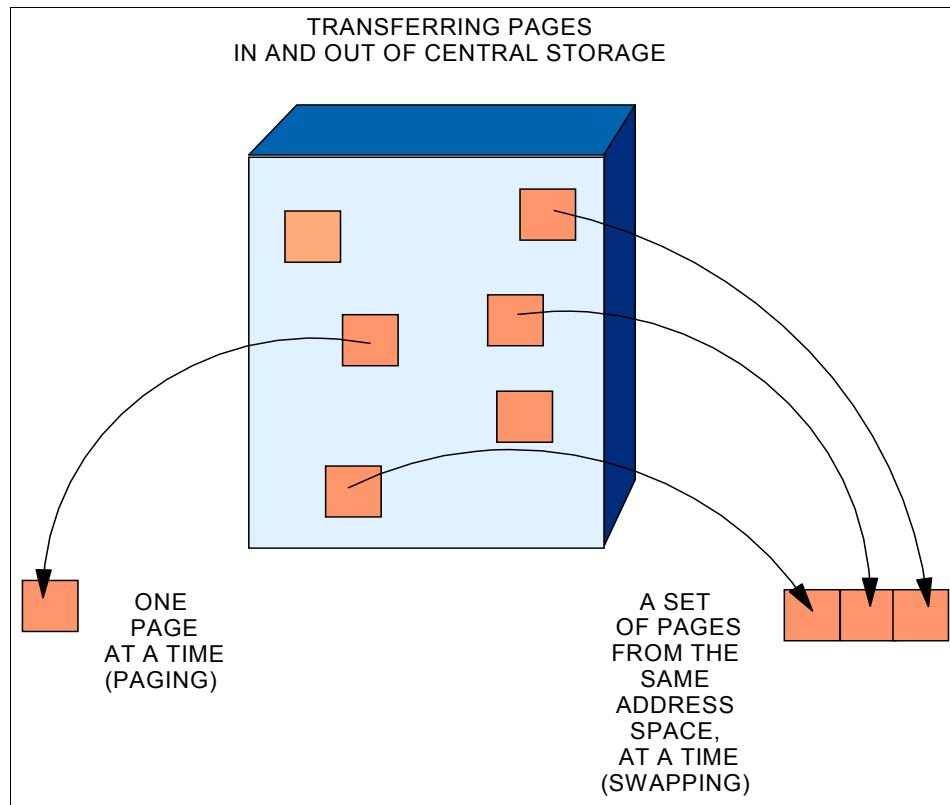


Figure 4-11 Paging and swapping concepts

Paging and swapping

Paging is the movement of pages between main storage frames and auxiliary storage slots.

There are two types of paging operations:

- ▶ Page-in, which flows from a slot to a frame. It is caused by a page fault. A page fault is an interrupt caused by the hardware in charge of translating a virtual address into a real address. The page fault happens because the page is not currently mapped to a frame. RSM gains control and, through ASM, provides a page-in operation to retrieve the page from auxiliary storage.
- ▶ Page-out, which flows from a frame to a slot. It is caused when a changed page needs to be stolen from main storage because this memory is under contention. RSM calls ASM to schedule the paging I/O necessary to send these pages to auxiliary storage.

Swapping and the working set

As mentioned, swapping is the process of transferring all of the pages of an address space between main storage and auxiliary storage. A swapped-in address space is active, having pages in main storage frames and pages in auxiliary storage slots. A swapped-out address space is inactive; the address space resides on auxiliary storage and cannot execute until it is swapped in.

Swapping is the primary function used by WLM (a z/OS component in charge of performance) to exercise control over the distribution of resources and system throughput.

One reason for swapping is, for example, pageable storage shortages. There are two types of swapping:

Physical swapping Transferring all pages in an address space between main storage and auxiliary storage.

A physical swapped-in address space is an active one (its programs can be executed) that has pages in main storage and pages in auxiliary storage.

A physical swapped-out address space is an inactive one having all pages in auxiliary storage, so it cannot execute its programs until it is swapped-in.

Logical swapping To reduce the processor and channel subsystem overhead involved during a physical swap needing to access auxiliary storage, WLM performs logical swaps where possible.

In a logical swap, LSQA fixed frames and recently referenced frames are kept in main storage (in contrast to physical swaps, where these frames are moved to auxiliary storage). Address spaces that are swapped for wait state conditions are the best candidates for logical swaps. Since z/OS 1.8, all swaps are logical.

4.12 Auxiliary page data sets

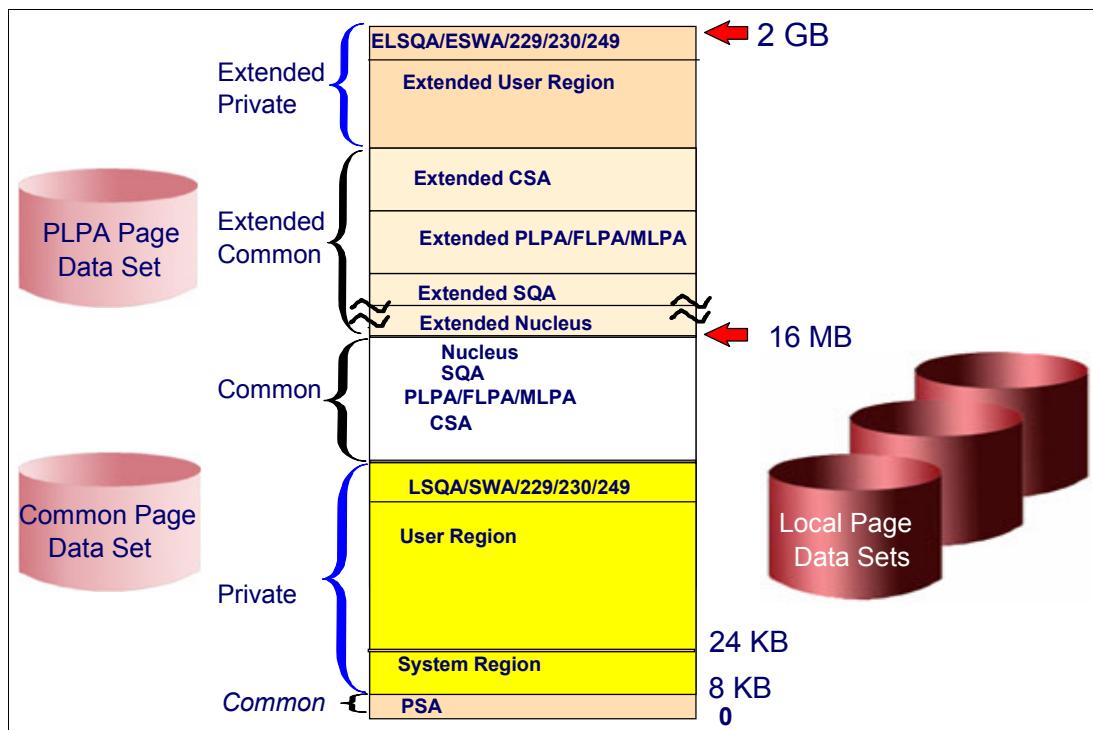


Figure 4-12 Paging to auxiliary data sets

Auxiliary page data sets

Auxiliary page data sets are formatted in slots. They should contain pages that for various reasons are not to stay in main storage frames. To page efficiently and expediently, ASM divides the pages of the system into classes, namely PLPA, common, and local. Contention is reduced when these classes of pages are placed on different physical devices. Multiple local page data sets are preferable.

Although the system requires only one local page data set, performance improvement can be obtained when local page data sets are distributed across more than one device, even though other devices might be large enough to hold the entire amount of necessary page space. The PLPA and common page data sets are both required data sets, and there can be only one of each. Spillage back and forth between the PLPA and common page data sets is permissible, but, in the interest of performance, only spilling from PLPA to common is to be permitted. The page data set classes are as follows:

PLPA page data set This unique and required page data set contains pageable link pack area pages. See “Link pack area” on page 282 for more information about this topic.

Common page data set This unique and required page data set contains the CSA non-fixed virtual pages of the system common area. See “Common service area” on page 280 for more information about this topic.

Local page data sets These contain the private area pages of all address space pages, data spaces, and any VIO data sets. To better distribute the paging activity load on different volumes and controllers, you can have several local page data sets.

Tip: Peaks in main storage demand can occur during system operation, resulting in heavy use of local page data set slots. To address this situation, local page data sets can be dynamically added to and deleted from the paging configuration without restarting the system.

4.13 System z Flash Express



Figure 4-13 System z Flash Express

System z Flash Express

System z Flash Express, an optional priced feature available with the zEC12 and zBC12, is a Peripheral Component Interconnect Express (PCIe) I/O adapter with NAND flash solid state drives.

Flash memory is a non-volatile computer storage technology. It was introduced on the market decades ago. Flash memory is commonly used today in memory cards, Universal Serial Bus (USB) flash drives, solid-state drives (SSDs), and similar products for general storage and transfer of data.

An SSD, sometimes called a *solid-state disk* or *electronic disk*, is a data storage device that uses integrated circuit assemblies as memory to store data persistently. SSD technology uses electronic interfaces compatible with traditional block I/O hard disk drives (HDDs). SSDs do not employ any moving mechanical components.

This characteristic distinguishes them from traditional magnetic disks, such as HDDs, which are electromechanical devices that contain spinning disks and movable read/write heads. With no seek time or rotational delays, SSDs can deliver substantially better I/O performance than HDDs. Flash SSDs demonstrate latencies that are 10 - 50 times lower than the fastest HDDs, often enabling dramatically improved I/O response times.

The Flash Express adapters are ordered in pairs, and each pair provides up to 1.4 terabytes (TB) of usable storage. The maximum of four pairs installed provides up to 5.6 TB of storage. Flash Express is a faster paging device when compared to auxiliary storage on traditional hard disk drives. Flash Express is supported by z/OS V1R13 and above, with the z/OS V1R13 RSM Enablement Offering web deliverable installed. It is fully supported by z/OS V2R1.

4.14 Storage-class memory on System z Flash Express

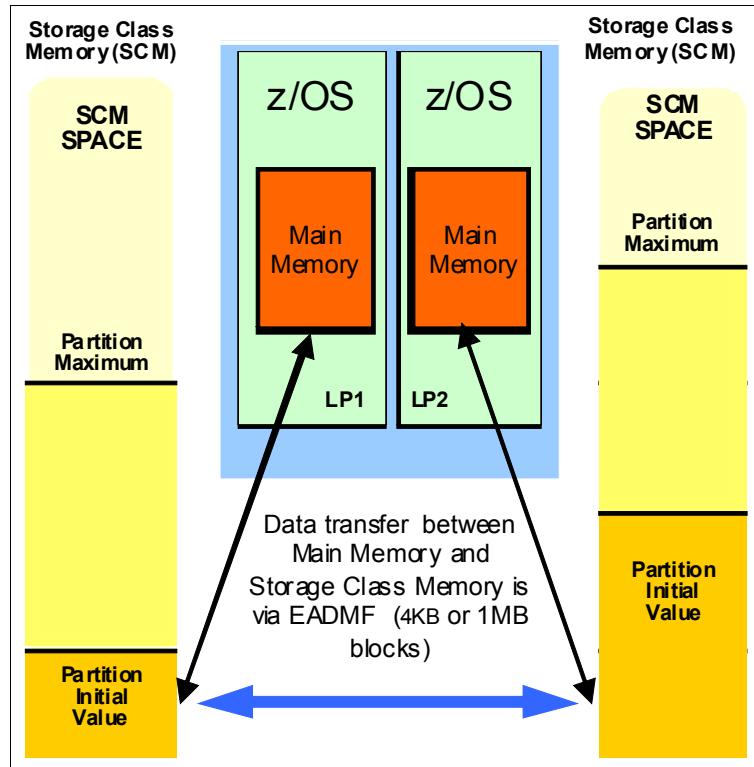


Figure 4-14 Paging to auxiliary data sets

Storage-class memory on System z Flash Express

The storage provided by these adapters is called storage-class memory (SCM). When enabled, SCM is used by z/OS for paging (4K and larger pages), caching of PLPA pages, and for staging of IBM System Storage® SAN Volume Controller (SVC) dumps.

You can use the Flash Express allocation windows on the Support Element (SE) or Hardware Management Console (HMC) to define the initial and maximum amount of Flash Express available to an LPAR. The maximum memory that is allocated to an LPAR can be dynamically changed. On z/OS, this process can also be done by using an operator command. Flash memory can also be configured offline to an LPAR.

Flash Express is used by the ASM with paging data sets to satisfy page-out and page-in requests received from the RSM. It supports 4 KB and 1 MB page sizes. ASM determines where to write a page based on space availability, data characteristics, and performance metrics. ASM still requires definition of a PLPA, common, and at least one local paging data set. VIO pages are only written to DASD because persistence is needed for warm starts.

A new PAGESCM keyword in IEASY\$xx member defines the minimum amount of flash to be reserved for paging. Value can be specified in units of MB, GB, or TB. NONE indicates that the system does not use flash for paging. ALL (default) indicates all flash that is defined to the partition is available for paging.

4.15 31-bit address space map

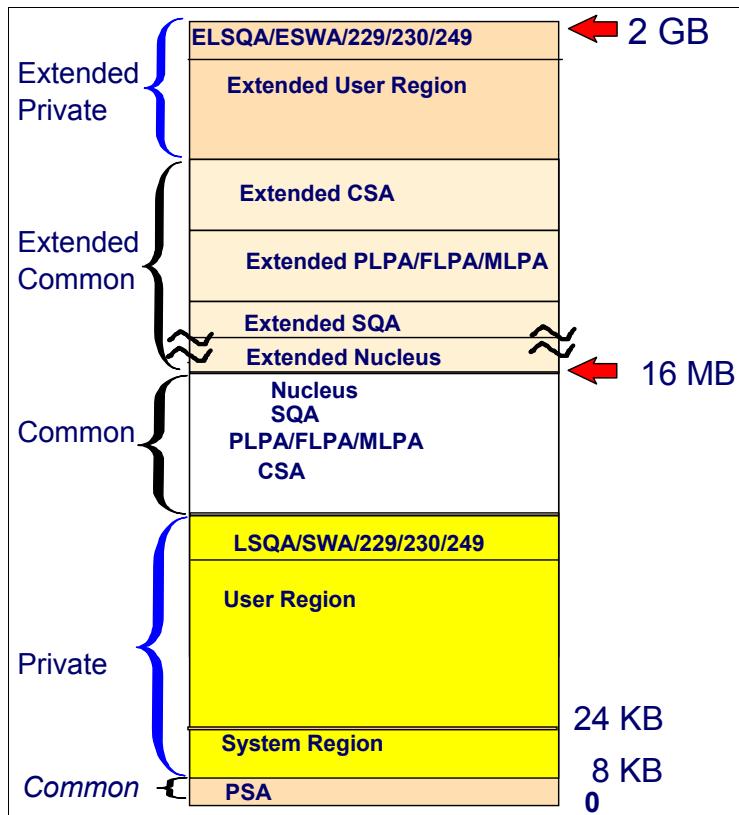


Figure 4-15 31-bit address space map

31-bit address space map

Figure 4-15 shows the layout of the storage areas for an address space in virtual storage. Note that most of the system areas exist both below and above 16 MB, providing an environment that can support 24-bit, 31-bit, and 64-bit addressing. However, each area and its counterpart above 16 MB can be thought of as a single logical area in virtual storage.

Since the introduction of XA architecture, the address space size is 2 GB addresses because the fields keeping the virtual addresses are 31 bits in size. (See Figure 4-23 on page 288 for information about the 64-bit address space map.)

A 2 GB virtual storage address space is provided for:

- ▶ The master scheduler address space
- ▶ IBM Job Entry Subsystem (JES)
- ▶ Other system component address spaces, such as allocation, system trace, SMF, and dumping services
- ▶ Each user, batch or Time Sharing Option Extensions (TSO/E)

User address spaces

The system uses a portion of each virtual address space. Each virtual address space consists of:

- ▶ The common area below 16 MB
- ▶ The private area below 16 MB
- ▶ The extended common area above 16 MB
- ▶ The extended private area above 16 MB

Map of 31-bit address space

The virtual address space, shown in Figure 4-15 on page 272, is divided into areas (sets of addresses) according to their use. Virtual storage allocated in each address space is divided between system requirements and user requirements. z/OS itself requires space from each of the basic areas. Each virtual address space consists of:

- ▶ The common area below 16 MB addresses
- ▶ The private area below 16 MB addresses
- ▶ The extended common area above 16 MB addresses
- ▶ The extended private area above 16 MB addresses

For more information about common areas and private areas, see 4.16, “The common virtual storage area” on page 274.

Most of z/OS areas exist both below and above 16 MB, providing an environment that can support both 24-bit and 31-bit addressing. However, each area and its counterpart above 16 MB can be thought of as a single logical area in virtual storage.

Private area of storage

The portion of the user's private area in each virtual address space that is available to the user's problem programs is called the *user region*. The private area consists of LSQA and scheduler work area (SWA) in subpools 229, 230, and 249.

There are two types of user regions: virtual (or V=V) and real (or V=R). Virtual and real regions are mutually exclusive; private areas can be assigned to V=R or V=V, but not to both. It is the installation's responsibility to determine the type of region to which jobs are assigned.

Usually, V=R should be assigned to regions containing jobs that cannot run in the V=V environment, or that are not readily adaptable to it. Programs that require a one-to-one mapping from virtual to main storage, such as program control interruption-driven channel programs, are candidates for real regions.

Extended private area of storage

The extended private area of storage is an area above the 16 MB line that is in the 24-bit addressing area. The extended private area contains extended LSQA (ELSPA) and extended SWA (ESWA) in subpools 229, 230, and 249, and is above the 16 MB line in the 31-bit addressing area.

4.16 The common virtual storage area

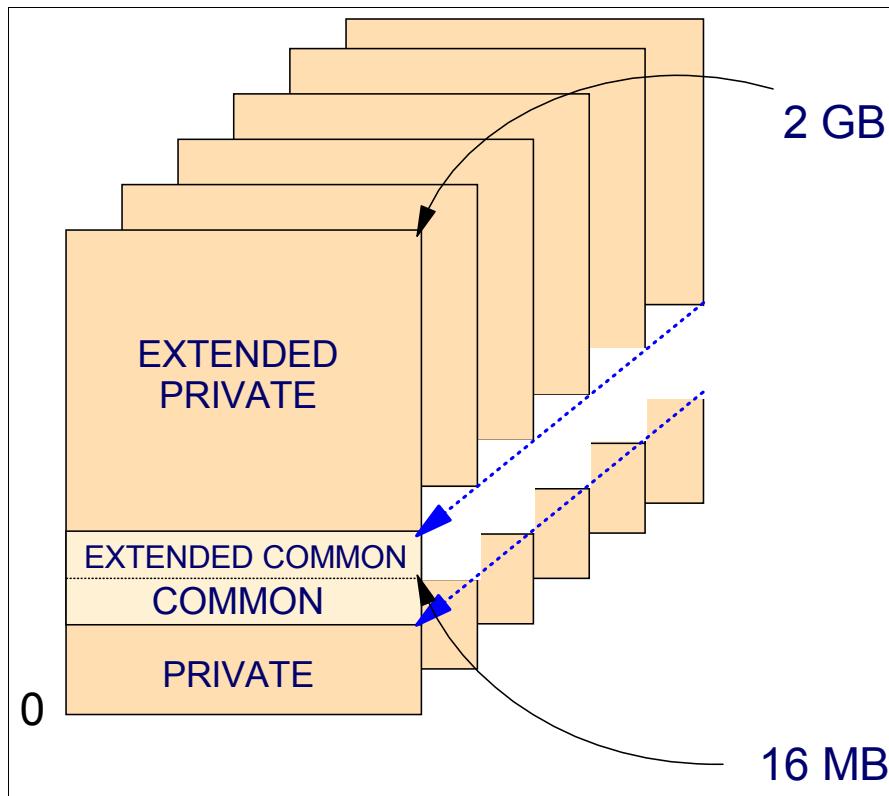


Figure 4-16 Common storage area

The common virtual storage area

The z/OS implementation of virtual storage is to have one address space per set of related programs (like a job step). The advantage of this design is isolation; any error is contained in one address space and cannot be propagated to another address space. Also, because the number of address spaces can be large, the number of virtual addresses to be used by programs is enormous.

However, such a design poses a problem: The need for communication between programs from different address spaces. To solve that problem, the common area was introduced. All address spaces in a z/OS system image share a virtual storage area known as the *common area*. That means that all address spaces programs in this z/OS access the same common data and the same common programs, with the same virtual address.

Common area below the 16 MB line

Each storage area in the common area (below 16 MB) has a counterpart in the extended common area (above 16 MB), except the prefixed save area (PSA). The CSA and SQA sizes are settled during the IPL, according to system initialization parameters in the SYS1.PARMLIB (IEASYSxx) system data set. The common area contains system control programs and control blocks.

The following storage areas are located in the common area:

- ▶ **Prefixed storage area**

This area is often referred to as *low core*. The PSA is a common area of virtual storage from address zero through 8191 in every address space. There is one unique PSA for every processor installed in a system.

The PSA maps architecturally fixed hardware and software storage locations for the processor. Because there is a unique PSA for each processor, from the view of a program running on z/OS, the contents of the PSA can change any time the program is dispatched on a different processor. This feature is unique to the PSA area, and is accomplished through a unique DAT manipulation technique called *prefixing*.

- ▶ **Common service area**

This portion of common area storage (addressable by all address spaces) is available to all applications. The CSA is often used to contain data frequently accessed by multiple address spaces. The size of the CSA area is established at system initialization time (IPL), and cannot change when the operating system is active.

- ▶ **Pageable link pack area, fixed link pack area, and modified link pack area**

This area contains the link pack areas, which are the PLPA, fixed link pack area (FLPA), and modified link pack area (MLPA), contain system-level programs that are often run by multiple address spaces. For this reason, the link pack areas reside in the common area, which is addressable by every address space, therefore eliminating the need for each address space to have its own copy of the program. This storage area is below the 16 MB boundary, and is therefore addressable by programs running in 24-bit mode.

- ▶ **System queue area**

This area contains system-level (key 0) data accessed by multiple address spaces. The SQA area is not pageable (fixed), which means that it resides in main storage until it is freed by the requesting program. The size of the SQA area is predefined by the installation, and cannot change when the OS is active. Yet it has the unique ability to “overflow” into the CSA area as long as there is unused CSA storage that can be converted to SQA.

- ▶ **Nucleus, which is fixed and non-swappable**

This is a key 0, read-only area of common storage that contains OS control programs.

4.17 z/OS nucleus

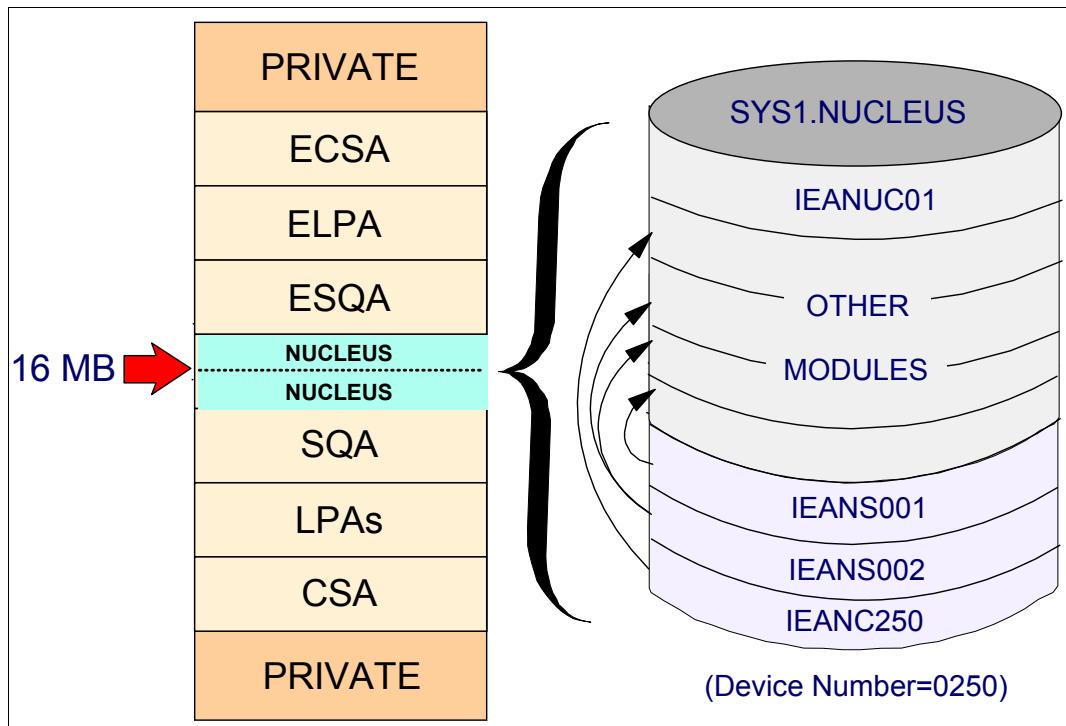


Figure 4-17 z/OS nucleus

z/OS nucleus

After the IPL and when the system is loaded, control is passed to IEAIPL00 which prepares an environment suitable for starting the programs and modules that make up the operating system, as follows:

1. It clears main storage to zeros before defining storage areas for the master scheduler.
2. It locates the SYS1.NUCLEUS data set on the SYSRES volume and loads a series of programs.

The nucleus in the common area contains the z/OS nucleus programs (kernel) and extensions to the nucleus that are initialized during IPL processing. The nucleus contains the most important z/OS programs. The nucleus RMODE24 programs reside below the 16 MB line. The nucleus RMODE31 programs reside above the 16 MB line.

Nucleus area

The nucleus area contains the nucleus load module, and extensions to the nucleus that are initialized during IPL processing. The nucleus includes a base and an architectural extension. Specify the correct architectural extension with the ARCHLVL statement in the LOADxx member of SYS1.PARMLIB for your system to run in either IBM Enterprise Systems Architecture/390 (IBM ESA/390) mode or z/Architecture mode.

Nucleus program modules

The program modules that are to be added to the nucleus area must reside as members in the SYS1.NUCLEUS data set. During the IPL process, the operator points to the device number (0250, in the example shown in Figure 4-17) of the volume containing that data set, causing the copy of the programs to be stored in memory.

The system programmer can add or delete modules from the nucleus by simply specifying the members on INCLUDE or EXCLUDE statements in the data set SYS1.PARMLIB, at member NUCLSTxx. The nucleus is always fixed in main storage (no pages from the nucleus can be stolen to page data sets slots).

Information: A set of tables, called nucleus module lists (NMLs), are used to identify the members in SYS1.NUCLEUS that are to be loaded into the DAT-on nucleus region. NMLs can be installed as part of an IBM product, a vendor product, or a customer user modification. Each NML contains a list of the SYS1.NUCLEUS members that are part of the same product or user modification. The NMLs are load modules that also reside in SYS1.NUCLEUS.

The NML must have a module name (CSECT name) in the form of IEANY*nnn*:

Y Can be either of S or C

S Stands for IBM provided NML

C Stands for customer-provided NML

nnn Is a 3-digit decimal number from 001 through 256

IEANxxxx example

Certain products have modules that must be incorporated into the z/OS nucleus, such as the IBM Information Management System (IBM IMS) Type 2 SVC.

Attention: The SYS1.NUCLEUS must not have secondary extents. z/OS cannot recognize secondary extents.

To incorporate the IMS Type 2 SVC into the z/OS nucleus, perform one of the following tasks:

- ▶ Bind the Type 2 SVC with the z/OS nucleus. You can bind the Type 2 SVC with the z/OS nucleus using one of the two following steps.
 - a. Invoking the Binder utility through a batch job
 - b. Creating and then performing a RECEIVE and APPLY for an SMP/E USERMOD
- ▶ Load the Type 2 SVC from SYS1.NUCLEUS using the Nucleus Module Loader facilities.
 - a. Create an NML that contains the list of IMS SVCs that you want loaded into the z/OS nucleus. IMS uses the IEANS001 NML.
 - b. Assemble and bind the Type 2 SVC into SYS1.NUCLEUS.

4.18 System queue area

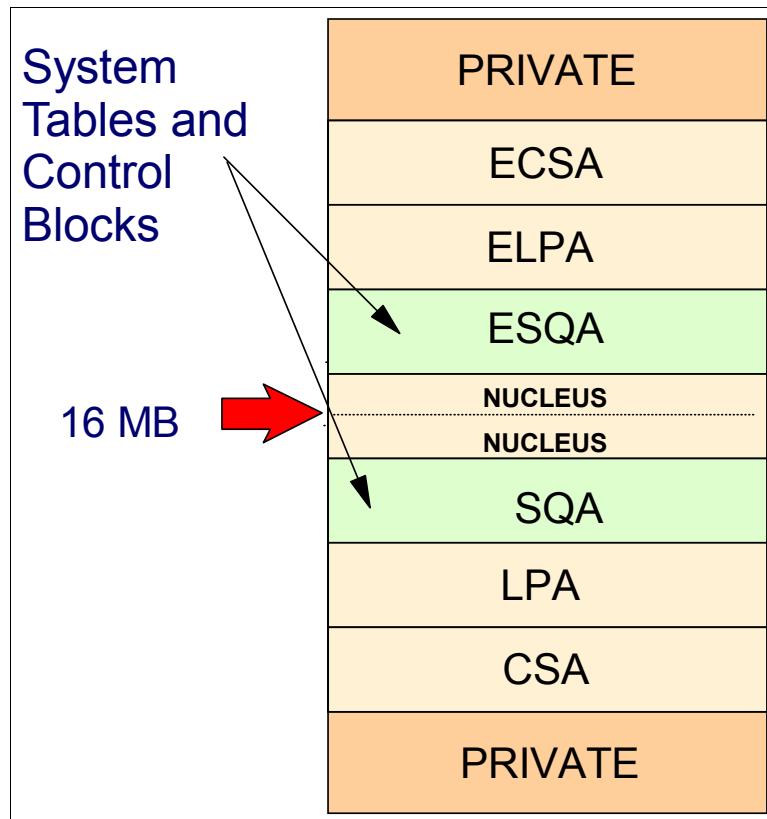


Figure 4-18 System queue area

System queue area

The system queue area (SQA) is a GETMAIN-able common area containing control blocks used by z/OS to manage transaction workloads and the use of system resources. It is a kind of virtual storage reserved area for future GETMAINs or FREEMAINs issued by authorized z/OS and non-z/OS programs. The number of active address spaces (which depends on the workload executed in the system) affects the system's use of SQA.

SQA is allocated directly below the nucleus, as shown in Figure 4-18. Extended SQA (ESQA) is allocated directly above the extended nucleus. Both allocations occur at IPL time. The size of SQA can be specified through the SQA parameter in the IEASYSxx parmlib member or like any IEASYSxx parmlib parameter, through the operator in a z/OS console at IPL:

SQA=(a,b)

This parameter specifies the sizes of the virtual SQA and ESQA. The subparameter **a** specifies the size of the SQA, located below 16 MB. The subparameter **b** specifies the size of the ESQA, located above 16 MB. These values are added to the system's minimum SQA of eight 64 KB blocks (or 512 KB) and minimum ESQA of approximately 8 MB. Both the SQA and ESQA are fixed in main storage as they are used.

Requests for SQA

When a GETMAIN for SQA/ESQA cannot be fulfilled because SQA/ESQA is full, then the SQA/ESQA overflows to CSA/extended CSA (ECSA). When SQA/ESQA pages are in use (GETMAINed), they are fixed in main storage.

Ensuring the appropriate size of SQA/ESQA and CSA/ECSA is critical to the long-term operation of z/OS. If an occupancy threshold is crossed, z/OS takes the following actions, trying to avoid an unprogrammed IPL:

- ▶ Message IRA100E displays in a z/OS console.
- ▶ No new address spaces are created.
- ▶ No new jobs are selected by initiators.

Note: The system also reserves additional SQA and ESQA storage for the I/O configuration. The amount of SQA and ESQA depends on the number of devices and control units installed.

Because the system adds these amounts to the SQA and ESQA blocks specified on the SQA parameter in IEASYSxx, the actual amounts of SQA and ESQA allocated might be more than you specified. Also, MVS will not round the lowest address of SQA down, or the uppermost address of ESQA up, to cause these areas to start or end on a 64 KB, page, or segment boundary.

4.19 Common service area

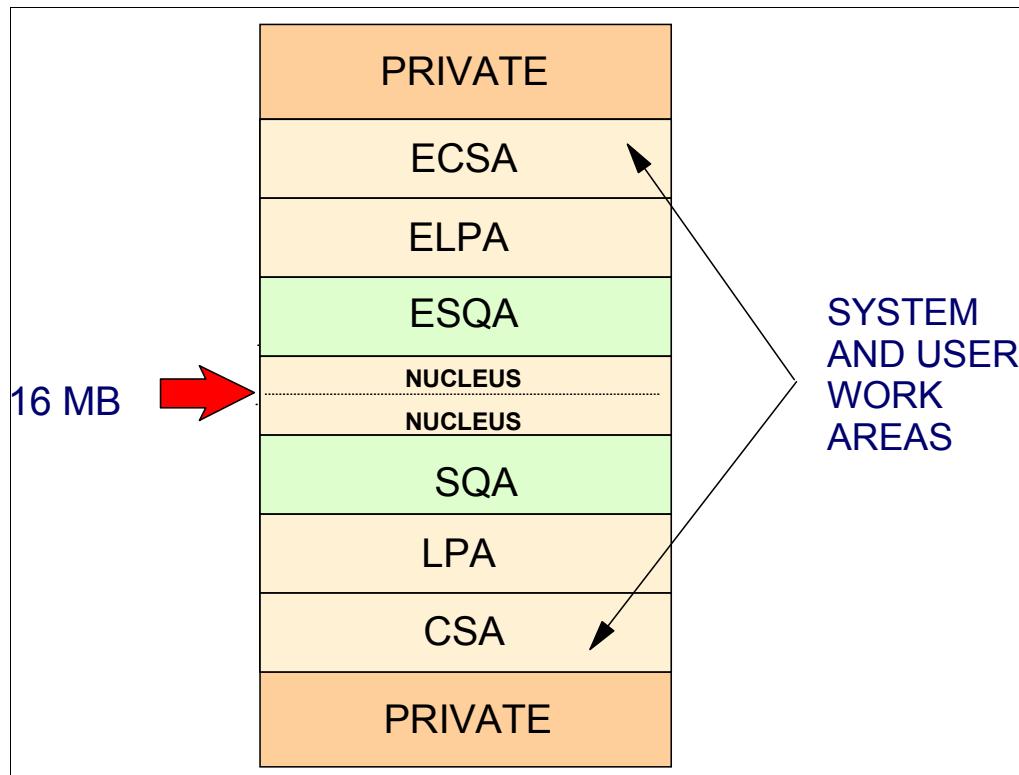


Figure 4-19 Common service area

Common service area

The CSA is a GETMAIN-able common area containing control blocks used by subsystem programs, such as JES2, Data Facility Storage Management Subsystem (DFSMS), and IBM Resource Access Control Facility (RACF), and access methods, such as Virtual Storage Access Method (VSAM). It is a sort of virtual storage-reserved area for future GETMAINs/FREEMAINs issued by such programs.

CSA/ECSA normally contains data referenced by several system address spaces, enabling address spaces to communicate by referencing the same piece of CSA data. In a sense, CSA/ECSA looks like SQA/ESQA.

CSA is allocated directly below the MLPA. ECSA is allocated directly above the extended MLPA, as shown in Figure 4-15 on page 272. If the virtual SQA/ESQA space is full, z/OS allocates additional SQA/ESQA space from the CSA/ECSA.

The size of the CSA/ECSA can be specified through the CSA parameter in the IEASYSxx member of SYS1.PARMLIB, or like any IEASYSxx parmlib parameter, through the operator in a z/OS console at IPL. The CSA contains pageable and fixed data areas that are addressable by all active virtual storage address spaces.

Important: If the size allocated for ESQA is too small or is used up quickly, the system attempts to steal space from ECSA. When both ESQA and ECSA are used up, the system allocates space from SQA and CSA below 16 MB. The allocation of this storage can eventually lead to a system failure. Ensuring the appropriate size of ESQA and ECSA storage is critical to the long-term operation of the system.

CSA/ECSA thresholds

Be aware that the following conditions can be responsible for a shortage of SQA/CSA:

- ▶ There has been storage growth beyond the previous normal range.
- ▶ Allocation of SQA or CSA is inadequate.
- ▶ The current thresholds at which the IRA100E or IRA101E messages are issued are too high for your installation.

4.20 Link pack area

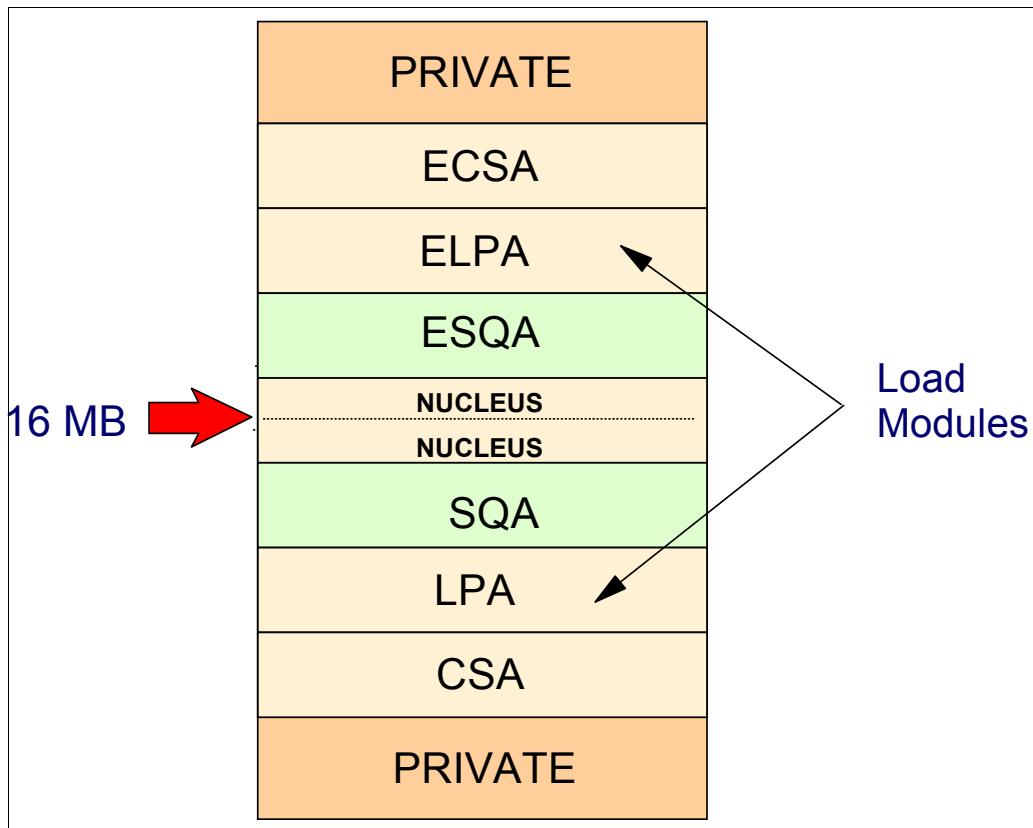


Figure 4-20 Link pack area

Link pack area

The LPA and ELPA contain programs that are preloaded at IPL time in the common area, from the SYS1.LPALIB data set. These programs can be certain z/OS SVC routines, access methods code, other read-only z/OS programs (the ones not modified along its execution), and any read-only reentrantable user programs selected by an installation.

Because such code is in the common area, all of these single-copy programs can be run in any address space. Their copy is not self-modifying (reentrant), so the same copy of the module can be used by any number of tasks in any number of address spaces at the same time. This reduces the demand for main storage, and lowers the program fetch overhead.

System libraries for the link pack area

The LPA is part of an address space's common area storage, and is divided into pageable, fixed, and modified sections:

- ▶ Libraries specified in SYS1.LPALIB, the LPALSTxx, or PROGxx parmlib members are loaded into PLPA. These libraries contain modules for read-only system programs, along with any read-only reentrantable user programs selected by an installation that can be shared among users of the system.
- ▶ IEAFIXxx members specify the modules loaded into FLPA. This area is to be used only for modules that significantly increase performance when they are fixed rather than pageable. The best candidates for the FLPA are modules that are infrequently used, but are needed for fast response. Modules placed in FLPA are always in main storage.

- IEALPAxx members specify the modules loaded into MLPA. The MLPA is used to contain reenterable routines from authorized program facility (APF)-authorized libraries that are to be part of the pageable extension to the LPA during the current IPL. Note that the MLPA exists only for the duration of an IPL.

LPA/ELPA specifications

The LPA/ELPA size depends on the number of modules loaded in it. When modules are added to LPA, the growth in LPA can cause the common area to cross one or more segment (1 MB) boundaries. This reduces the available private area for all address spaces, even for those address spaces not using the load modules added to LPA.

All modules placed in LPA are assumed to be APF-authorized. Being APF-authorized means that a program can invoke any SVC routine that accesses protected system and private areas. Although LPA boundaries cannot be changed after IPL, it is possible to dynamically include new load modules to LPA without an IPL. In this case, z/OS issues a GETMAIN from CSA/ECSA, and uses such virtual storage area to load the load module.

The RMODE attribute of the program (load module) decides its location (LPA or ELPA). The ELPA is built above 16 MB.

Pages from modules (programs) placed anywhere in LPA are always in virtual storage. Pages from modules placed in FLPA are also always in main storage. Whether modules that are in LPA, but outside FLPA, are in main storage depends on how often they are used by all of the programs of the system, and on how much main storage is available. The more often an LPA module is used, and the more main storage that is available on the system, the more likely it is that the pages containing the copy of the module will be in main storage at any given time.

Each address space uses the same common area. Portions of the common area are paged in and out as the demands of the system change, and as new user jobs (batch or time-shared) start and old ones terminate.

4.21 31-bit address space private area

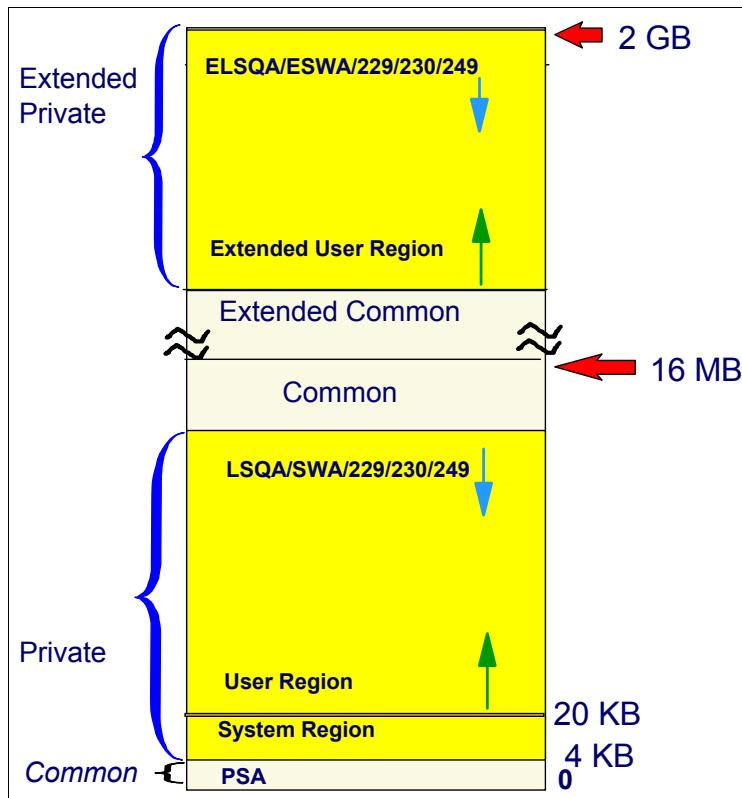


Figure 4-21 The user's private area (user region)

31-bit address space private area

The portion of the user's private area in each virtual address space that is available to the user's problem programs is called the *user region*. The use of address spaces enables z/OS to maintain the distinction between the programs and data belonging to each address space. The private areas in one user's address space are isolated from the private areas in other address spaces, and this address space isolation provides much of the OS's security.

Private areas

There are two private areas: below the 16 MB line is the private (PVT), and above the 16 MB line is the extended private (EPVT). Their size is the complement of the common area's size. The virtual addresses in the private area is unique to the programs running in such areas.

The private area is formed by the following areas:

- ▶ Subpools 229, 230, and 249

This area enables private storage to be obtained in the requestor's storage protect key. The area is used for control blocks that can be obtained only by authorized programs (such as z/OS) having appropriate storage protect keys.

A subpool is a virtual storage area with the same properties regarding storage key, pageable or fixed, private or common, fetch protected or not, and so on. When a program GETMAINs virtual storage addresses, it must indicate the subpool number.

- ▶ Local system queue area

This area contains tables and control blocks queues associated with the address space. LSQA is intermixed with SWA and subpools 229, 230, and 249 downward from the bottom of the CSA into the unallocated portion of the private area, as needed.

ELSPA is also intermixed, but it is allocated downward from 2 GB into the unallocated portion of the extended private area, as needed. LSQA does not take space below the top of the highest storage currently allocated to the user region.

- ▶ Scheduler work area

SWA contains control blocks that exist from task initiation to task termination. It includes control blocks and tables created during job control language (JCL) interpretation.

- ▶ A 16 KB system region area

- ▶ User region

This region is used for running user program applications (loaded at subpools 251/252) and storing user program data (subpools from 0 - 127).

When a module is loaded into the private area for an address space, the region available for other components is reduced by the amount of storage used for the module. The amount of private virtual storage that a job can use for subpools 251/252 and from 0 - 127 (the low addresses of the two private areas) can be limited through the REGION keyword on the JOB or EXEC JCL statements. Also, the region size can be controlled and overridden through the SMF exit IEFUSI. A value equal to 0 KB or 0 MB gives the job all private storage available.

4.22 Data spaces and IBM Hiperspace

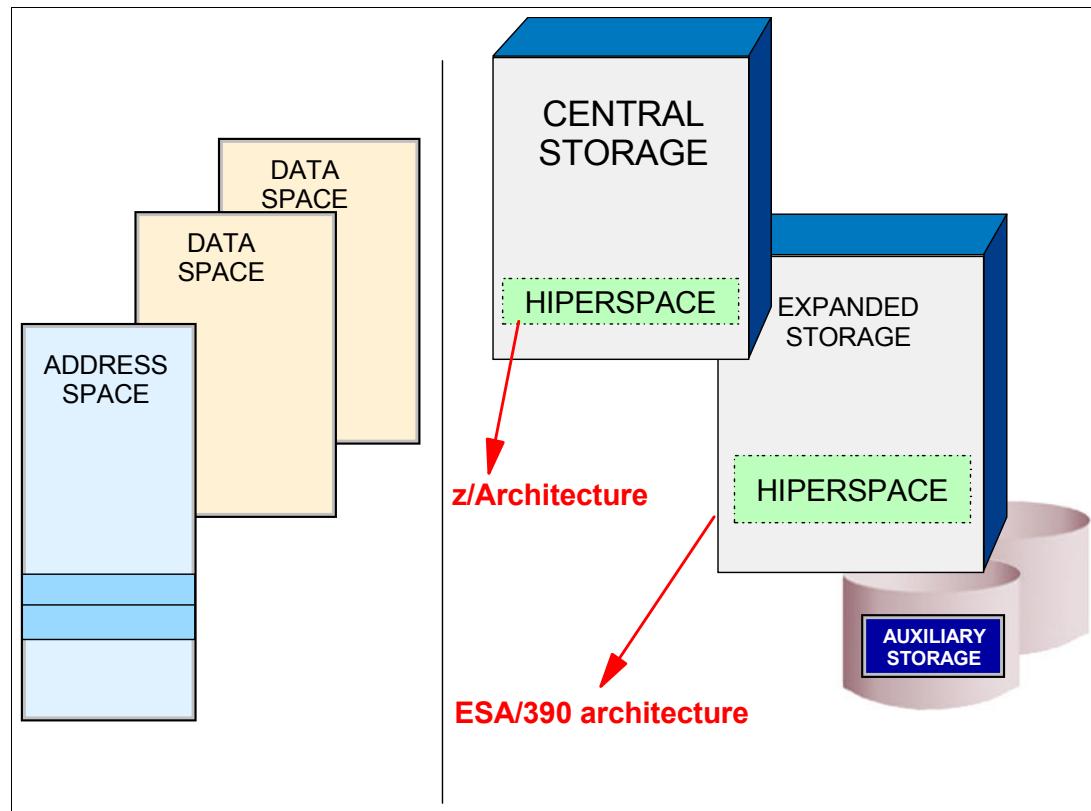


Figure 4-22 Data space management

Data spaces and IBM Hiperspace

IBM ESA/370 and MVS/ESA had improvements to relieve virtual storage constraint, bringing horizontal growth to virtual storage with data spaces (a new type of z/OS address space) and IBM Hiperspace™ (a new type of service).

The growth of processing, storage, and I/O capabilities led to a virtual storage constraint. The upward growth in virtual storage (address space size) was limited by the architecture (hardware and software).

The main difference between data spaces and Hiperspace is the way a program references data. A program references data in a data space directly, in much the same way that it references data in an address space. It addresses the data by the byte, manipulating, comparing, and performing arithmetic operations. The program uses the same instructions (such as load, compare, add, and move character) that it uses to access data in its own address space.

To reference the data in a data space, the program must be in the Address Space Control (ASC) mode called access register (AR) mode. Pointers that associate the data space with the program must be in place, and the contents of ARs that the instructions use must identify the specific data space.

Data spaces

A *data space* is a type of virtual storage space with a range up to 2 GB of contiguous virtual storage. The virtual storage map of a data space is quite different than an address space. The entire 2 GB is available for user data, and does not contain specific areas.

A data space can hold only data (operands accessed by instructions located in address spaces). It does not contain z/OS control blocks or programs in execution. Program code does not execute in a data space, although a program can reside in a data space as data (to be executed, however, it needs to be copied to an address space). A program can refer to data in a data space at bit level, as it does in a work file.

Hiperspace

High performance data access, known as *Hiperspace*, is a kind of data space created with the same RSM services used to create a data space. It provides the applications an opportunity to use expanded storage as a substitute to I/O operations. Hiperspace differs from data spaces in the following ways:

- ▶ Main storage is never used to back the virtual pages in Hiperspace, where pages are located in expanded or auxiliary storage.
- ▶ Data can be retrieved and stored between a Hiperspace and a data space only using MVS services. This avoids the complex programming required when accessing data in a data space.
- ▶ Data is addressed and referred to as a 4 KB block.

Although z/OS does not support Expanded Storage when running under the z/Architecture, Hiperspace continues to operate in a compatible manner. Under z/Architecture, Hiperspace is mapped in main storage (rather than expanded) and auxiliary storage.

Programs can use data spaces and Hiperspace as described here:

- ▶ To obtain more virtual storage than a single address space gives a user.
- ▶ To isolate data from other tasks (programs) in the address space. Data in an address space is accessible to all programs executing in that address space. You might want to move data to a database or Hiperspace for security or integrity reasons. You can restrict access to data in those spaces to one or several units of work.
- ▶ To share data among programs that are executing in the same address space, or different address spaces. Rather than keeping the shared data in common areas, you can create a database or Hiperspace for the data that you want your programs to share. Use this space as a way to separate your data logically by its own particular use.
- ▶ To provide an area in which to map a data-in virtual object.

4.23 64-bit address space map

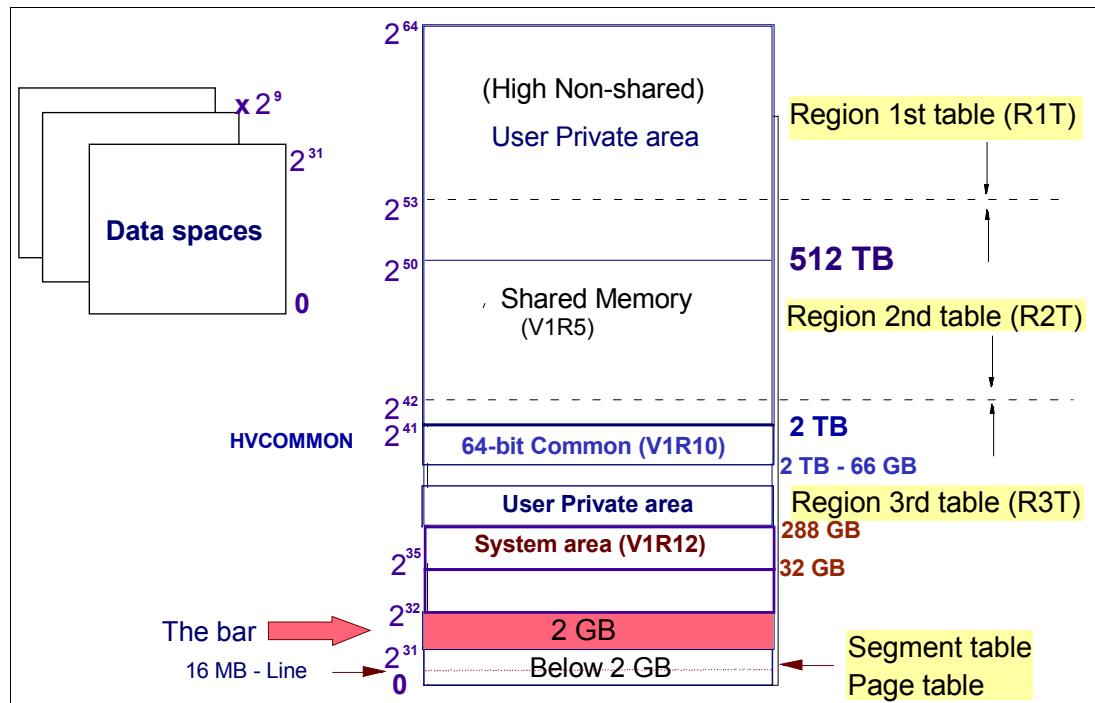


Figure 4-23 64-bit address space map

64-bit address space map

As previously mentioned, z/Architecture broke the 2 GB (31-bit) main storage limit and the 2 GB (31-bit) address limit, and moved the limit to 16 EB (64-bit). The maximum of a z/OS address space is 16 EB addresses, which makes the new address space 8 billion times the size of the former 2 GB address space. However, any new created address space in z/OS is initialized with 2 GB addresses (as it was previously), but with the potential to go beyond.

For compatibility, the layout of the virtual storage areas for an address space is the same under 2 GB. The area that separates the virtual storage area below the 2 GB address from the user private area is called the *bar*, as shown in Figure 4-23, and is 2 GB addresses thick. In a 64-bit virtual storage environment, the terms *above the bar* and *below the bar* are used to identify the areas between 2^{31} and $2^{64}-1$, and 0 and $2^{31}-1$, respectively.

For example, an address in the range 0 - 7FFFFFFF is below the bar. An address in the range FFFFFFFF - FFFFFFFF_FFFFFFFF is above the bar. This is basically an alteration to the 2 GB 31-bit terminology that related “below the line” to 24-bit storage, and “above the line” to 31-bit addresses.

The 64-bit address space map differs from the 31-bit address space map in the following ways:

- | | |
|---|--|
| 0 to 2^{31} | The layout is the same; see Figure 4-15 on page 272. |
| 2^{31} to 2^{32} | From 2 GB - 4 GB is considered the <i>bar</i> . Below the bar can be addressed with a 31-bit address. Above the bar requires a 64-bit address. |
| $2^{32} - 2^{35}$ | Reserved area addressable by the Java Virtual Machine (JVM) using 32-bit pointer compression. |

2**35 - 2**41	The low non-shared area starts at 4 GB and goes to 2**41. A portion of this storage is designed to be used for system storage as an equivalent to LSQA below the 2 GB bar. This system area will not be copied during the fork() process when RSM copies the parent storage to the child address space.
	Memory objects are allocated in the system area that starts at X'8_00000000' - 32 GB and ends at X'28_00000000' - 288 GB using the IARV64 macro with REQUEST=GETSTOR,LOCALSYSARES=YES.
2**41 - 2**50	The Shared Area starts at 2**41 and goes to 2**50 or higher if requested (up to 2**53).
2**50 - 2**64	The high non-shared area starts at 2**50 or wherever the shared area ends and goes to 2**64.

Attention: The area above the bar is designed to keep data (such as IBM DB2 buffer pool and IBM WebSphere data), and not to load modules (programs). There is no RMODE64 as a load module attribute. However, such programs running below the bar might request virtual storage above the bar and access it. To access such an address, the program must be AMODE64.

To allocate and release virtual storage above 2 GB, a program must use the services provided in the IARV64 macro. The GETMAIN, FREEMAN, STORAGE, and CPOOL macros do not allocate storage above the 2 GB address, nor do callable cell pool services.

Region tables

In a 16 EB address space with 64-bit virtual storage addressing, there are three additional levels of translation tables, called *region tables*:

- ▶ Region third table (R3T)
- ▶ Region second table (R2T)
- ▶ Region first table (R1T)

The region tables are 16 KB in length, and there are 2048 entries per table. Each region has 2 GB.

Page and segment tables

Segment tables and page table formats remain the same as for virtual addresses below the bar. When translating a 64-bit virtual address, after the system has identified the corresponding 2 GB region entry that points to the segment table, the process is the same as that described previously.

User private area

This area above the bar is intended for application data; no programs run above the bar. No system information or system control blocks exist above the bar, either. Currently there is no common area above the bar.

The *user private area*, as shown in Figure 4-23 on page 288, includes:

- ▶ Low private. The private area below the line.
- ▶ Extended private. The private area above the line.
- ▶ Low non-shared. The private area just above the bar.
- ▶ High non-shared. The private area above the Shared Area.

As users allocate private storage above the bar, it will first be allocated from the low non-shared area. Similarly, as the shared area is allocated, it will be allocated from the bottom up. This is done to enable applications to have both private and shared memory above the bar, and avoid additional machine cycles to perform dynamic address translation (DAT).

For virtual storage above the bar, a new JCL keyword (MEMLIMIT) is introduced on the JOB and EXEC JCL statements. For virtual storage above the bar, there is no practical limit to the amount of virtual address range that an address space can request. However, there are practical limits to the main storage and auxiliary storage needed to back the request. Therefore, a limit is placed on the amount of usable virtual storage above the bar that an address space can use at any one time.

Important: MEMLIMIT controls the amount of usable storage above the 2 GB line. Also, there is an exit IEFUSI that does the same.

4.24 Size and number notation in bytes

Symbol	Power of 2	Decimal Value
Kilo (K)	2^{10}	1024
Mega (M)	2^{20}	1,048,576
Giga (G)	2^{30}	1,073,741,824
Tera (T)	2^{40}	1,099,511,627,776
Peta (P)	2^{50}	1,125,899,906,842,624
Exa (E)	2^{60}	1,152,921,504,606,846,976

Figure 4-24 Size and number notation in bytes

Size and number notation in bytes

The introduction of 64-bit virtual addresses presents a new order of magnitude of numbers, so this section covers the names and raw sizes of these numbers.

In bytes, the letters K, M, G, T, P, and E denote the multipliers 2^{10} , 2^{20} , 2^{30} , 2^{40} , 2^{50} , and 2^{60} , respectively. The letters are from the International System of Units (SI) unit of measurement prefixes and their associated decimal multipliers, and stand for Kilo (10^3), Mega (10^6), Giga (10^9), Tera (10^{12}), Peta (10^{15}), and Exa (10^{18}). Note that Exa is 1 followed by 18 zeroes.

- Kilobyte** A kilobyte is 2^{10} bytes, or 1,024 bytes.
- Megabyte** A megabyte is 2^{20} bytes, or 1,048,576 bytes.
- Gigabyte** A gigabyte is 2^{30} bytes, or 1,073,741,824 bytes.
- Terabyte** A terabyte is 2^{40} bytes, or 1,099,511,627,776 bytes.
- Petabyte** A petabyte is 2^{50} bytes, or 1,125,899,906,842,624 bytes.
- Exabyte** An exabyte is 2^{60} bytes, or 1,152,921,504,606,846,976 bytes.

In z/Architecture, these multiplier letters do not have the SI decimal meaning, but rather represent the power of 2 closest to the corresponding power of 10. Figure 4-24 shows the names and the decimal values of these multipliers.

4.25 Segment tables and page tables in 31-bit addressing

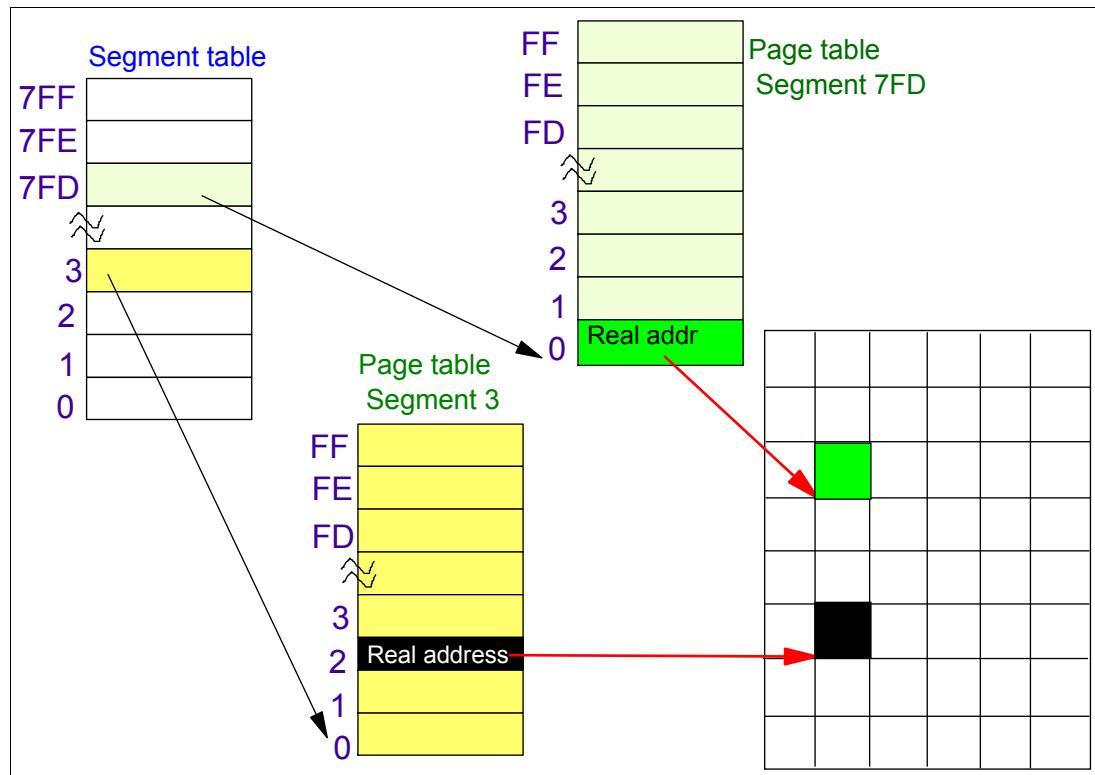


Figure 4-25 31-bit virtual address and dynamic address translation

Virtual addressing

Main storage is viewed as a long sequence of bits. The sequence of bits is subdivided into units of 8 bits, called a *byte*. Each byte location in storage is identified by a unique integer starting with zero (0), called an *address*. Addresses are either 31-bit or 64-bit integer values.

An *address space* is a sequence of virtual addresses that is associated with virtual storage. The addresses are usually contiguous, but they do not need to be. A *page* is 4096 bytes, and is the minimum size of an address space. A program of fewer than 4096 bytes fits into a single page. All of the addresses used in a program are set up assuming that the program is loaded into main storage starting at location 0. In reality, it is not, but this assumption makes decoding the virtual address somewhat easier.

If the program size increases beyond 4096 bytes (the size of a single page), another page is allocated. It does not matter whether the newly allocated page is adjacent to the first page or a hundred pages away from it. The IBM z/Transaction Processing Facility (z/TPF) system decodes the addresses in the same way.

Segment tables and page tables

DAT is the hardware in charge of translating a virtual address during a storage reference into the corresponding real address, using translation tables (segment tables and page tables) prepared by the z/OS component RSM.

Each address space has its own *segment tables* and *page tables*. Each segment table entry has a pointer to the correlated page table. There are pointers only for those page tables having pages with GETMAINed addresses. A page table is allocated when the first page on that segment is allocated. There is a maximum of 2048 page tables.

To make this translation easier, the virtual address space is partitioned into segments, each one of 1 MB addresses. Therefore, each 2 GB address space has 2048 segments. Each segment has 256 4 KB pages. Given a virtual address, DAT finds the following information contained in the virtual address:

- ▶ Segment index (number of the segment) in the first 11 bits of the address, up to 2047 = b'111 1111 1111'= X'7FF'.
- ▶ Page index (number of the page in that segment) in the next 8 bits, up to 255 = b'1111 1111'= X'FF'.
- ▶ Byte index (displacement in the page) in the last 12 bits, up to 4095 = b'111111111111'= X'FFF'.

For more information about this topic, see Figure 4-15 on page 272.

Segment number

The segment number is mapped with an entry into a *segment table* (one entry per segment), with 2048 entries. Each entry is identified from 0 to 2047. The entry 0 refers to segment 0, the entry 1 refers to segment 1, and so on.

Each address space has one segment table. Each entry in a segment table points to a *page table* that maps each page of a segment into an entry table. The system uses a page table with 256 entries.

Because each page is 4 KB, each address space segment has 256 pages. Each entry identifies each page in that segment. Therefore, entry 0 refers to the first page of the segment, entry 1 refers to the second page in the same segment, and so on. The page table entry has the real address of the frame mapping the page, or an invalid bit, when this mapping does not happen. This invalid bit causes a program interrupt known as a *page fault*. When a page fault occurs, the contents of the page are in a auxiliary storage slot on DASD.

4.26 Program status word format

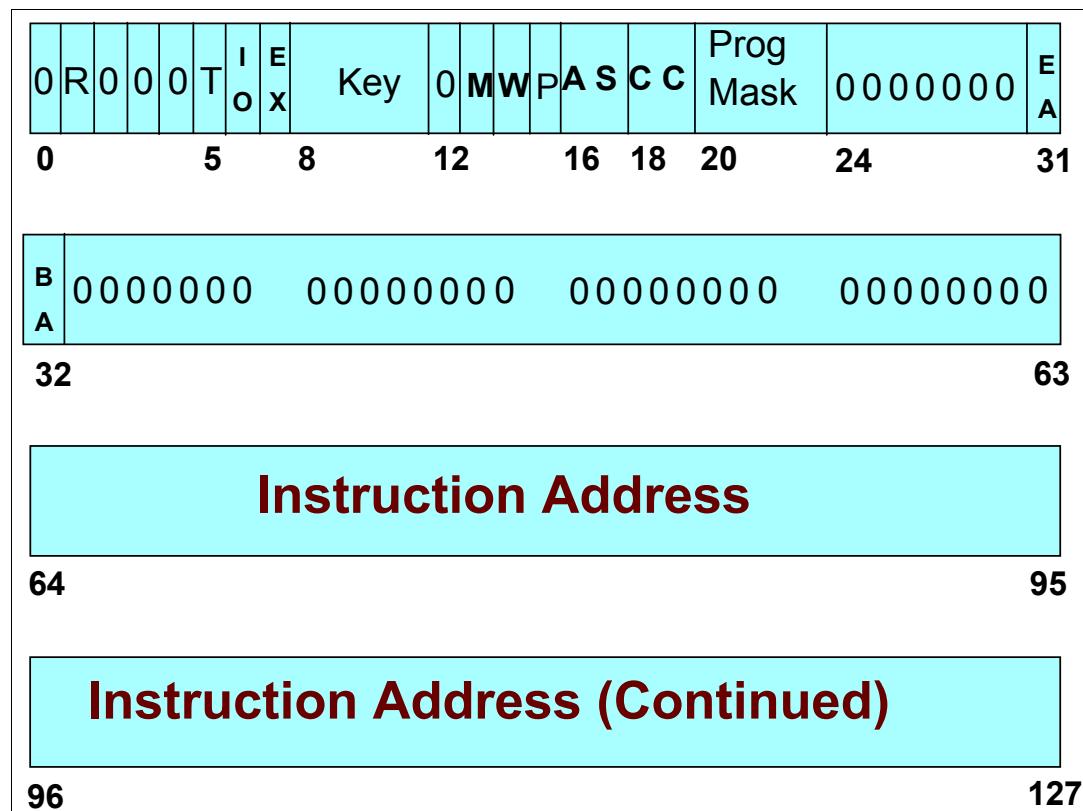


Figure 4-26 Program status word format

Program status word

The current program status word (PSW) in the central processor unit (CPU) contains information required for the execution of the currently active program. The PSW is 128 bits in length and includes the instruction address, condition code, and other control fields. In general, the PSW is used to control instruction sequencing, and to hold and indicate much of the status of the CPU in relation to the program currently being executed.

Additional control and status information is contained in control registers and permanently assigned storage locations. The status of the CPU can be changed by loading a new PSW or part of a PSW.

PER mask - R (bit 1)

Bit 1 controls whether the CPU is enabled for interrupts associated with program-event recording (PER). When the bit is zero, no PER event can cause an interruption. When the bit is one, interruptions are permitted, subject to the PER-event-mask bits in control register 9.

DAT mode - T (bit 5)

Bit 5 controls whether implicit dynamic address translation of logical and instruction addresses used to access storage takes place. When the bit is zero, DAT is off, and logical and instruction addresses are treated as real addresses. When the bit is one, DAT is on, and the dynamic-address-translation mechanism is invoked.

I/O mask - IO (bit 6)

Bit 6 controls whether the CPU is enabled for I/O interruptions. When the bit is zero, an I/O interruption cannot occur. When the bit is one, I/O interruptions are subject to the I/O-interruption subclass-mask bits in control register 6. When an I/O-interruption subclass-mask bit is zero, an I/O interruption for that I/O-interruption subclass cannot occur. When the I/O-interruption subclass-mask bit is one, an I/O interruption for that I/O-interruption subclass can occur.

External mask - EX (bit 7)

Bit 7 controls whether the CPU is enabled for interruption by conditions included in the external class. When the bit is zero, an external interruption cannot occur. When the bit is one, an external interruption is subject to the corresponding external subclass-mask bits in control register 0:

- ▶ When the subclass-mask bit is zero, conditions associated with the subclass cannot cause an interruption
- ▶ When the subclass-mask bit is one, an interruption in that subclass can occur.

PSW key (bits 8-11)

Bits 8-11 form the access key for storage references by the CPU. If the reference is subject to key-controlled protection, the PSW key is matched with a storage key when information is stored, or when information is fetched from a location that is protected against fetching. However, for one of the operands of each of MOVE TO PRIMARY, MOVE TO SECONDARY, MOVE WITH KEY, MOVE WITH SOURCE KEY, and MOVE WITH DESTINATION KEY, an access key specified as an operand is used rather than the PSW key.

Processor-check mask - M (bit 13)

Bit 13 controls whether the CPU is enabled for interruption by processor-check conditions. When the bit is zero, a processor-check interruption cannot occur. When the bit is one, processor-check interruptions due to system damage and instruction-processing damage are permitted, but interruptions due to other processor-check-subclass conditions are subject to the subclass-mask bits in control register 14.

Wait state - W (bit 14)

When bit 14 is one, the CPU is waiting; that is, no instructions are processed by the CPU, but interruptions can occur. When bit 14 is zero, instruction fetching and execution occur in the normal manner. The wait indicator is on when the bit is one. When in wait state, the only way of getting out of that state is through an Interruption, or by an IPL (a z/OS boot).

Certain bits, when off in the current PSW, place the CPU in a disabled state; the CPU does not accept Interrupts. So when z/OS, for any error reason (software or hardware) decides to stop a CPU, it sets the PSW to the Disable and Wait state, forcing an IPL to restore the CPU back to the running state.

Problem state - P (bit 15)

When bit 15 is one, the CPU is in the problem state. When bit 15 is zero, the CPU is in the supervisor state. In the supervisor state, *all* instructions are valid.

In the problem state, only those instructions are valid that provide meaningful information to the problem program and that cannot affect system integrity; such instructions are called *unprivileged instructions*.

The instructions that are never valid in the problem state are called *privileged instructions*. When a CPU in the problem state attempts to execute a privileged instruction, a privileged-operation exception is recognized. Another group of instructions, called *semi-privileged instructions*, are executed by a CPU in the problem state only if specific authority tests are met; otherwise, a privileged-operation exception or a special-operation exception is recognized.

Address-space control -AS (bits 16-17)

Bits 16 and 17, in conjunction with PSW bit 5, control the translation mode.

Condition code - CC (bits 18-19)

Bits 18 and 19 are the two bits of the condition code. The condition code is set to 0, 1, 2, or 3, depending on the result obtained in executing certain instructions. Most arithmetic and logical operations, and various other operations, set the condition code. The instruction BRANCH ON CONDITION can specify any selection of the condition-code values as a criterion for branching.

The part of the CPU that executes instructions is called the arithmetic logic unit (ALU). The ALU has internally four bits that are set by certain instructions. At the end of these instructions, this 4-bit configuration is mapped into bits 18 and 19 of the current PSW.

As an example, the instruction COMPARE establishes a comparison between two operands. The result of the comparison is placed in the CC of the current PSW, as follows:

- ▶ If CC=00, then the operands are equal.
- ▶ If CC=01, then first operand is lower.
- ▶ If CC=10, then first operand is greater.

To test the contents of a CC (set by a previous instruction), use the BRANCH ON CONDITION (BC) instruction. It contains an address of another instruction (branch address) to be executed, depending on the comparison of the CC and a mask M. The instruction address in the current PSW is replaced by the branch address if the condition code has one of the values specified by M; otherwise, normal instruction sequencing proceeds with the normal updated instruction address. The following list includes the types of codes:

- ▶ Condition code (bits 18, 19 PSW).
- ▶ Return code. A code associated with how a program ended.
- ▶ Completion code. A code associated with how a task ended.
- ▶ Reason code. A code passed in the GPR 15 detailing how a task ended.

Program Mask (bits 20-23)

During the execution of an arithmetic instruction, the CPU might find an unusual (or error) condition, such as overflows, loss of significance, or underflow. In such cases, the CPU generates a program interrupt.

When this interrupt is treated by z/OS, usually the current task is abnormally ended (ABEND). However, in certain situations, programmers do not want an ABEND, so by using the instruction SET PROGRAM MASK, they can mask such interrupts by setting various program mask bits to OFF. Each bit is associated with one type of condition:

- ▶ Fixed point overflow (bit 20)
- ▶ Decimal overflow (bit 21)
- ▶ Exponent underflow (bit 22)
- ▶ Significance (bit 23)

The active program is informed about these events through the condition code posted by the instruction where the events described happened.

The contents of the CPU can be totally changed by two events:

- ▶ Loading a new PSW from storage along an interruption
- ▶ Executing the instruction LPSW, which copies 128 bits from memory to the current PSW

Extended addressing mode - EA, BA (bits 31-32)

The combination of bits 31 and 32 identify the addressing mode (24, 31, or 64) of the running program. Bit 31 controls the size of effective addresses and effective-address generation in conjunction with bit 32, the basic-addressing-mode bit. When bit 31 is zero, the addressing mode is controlled by bit 32. When bits 31 and 32 are both one, 64-bit addressing is specified.

4.27 31-bit virtual address

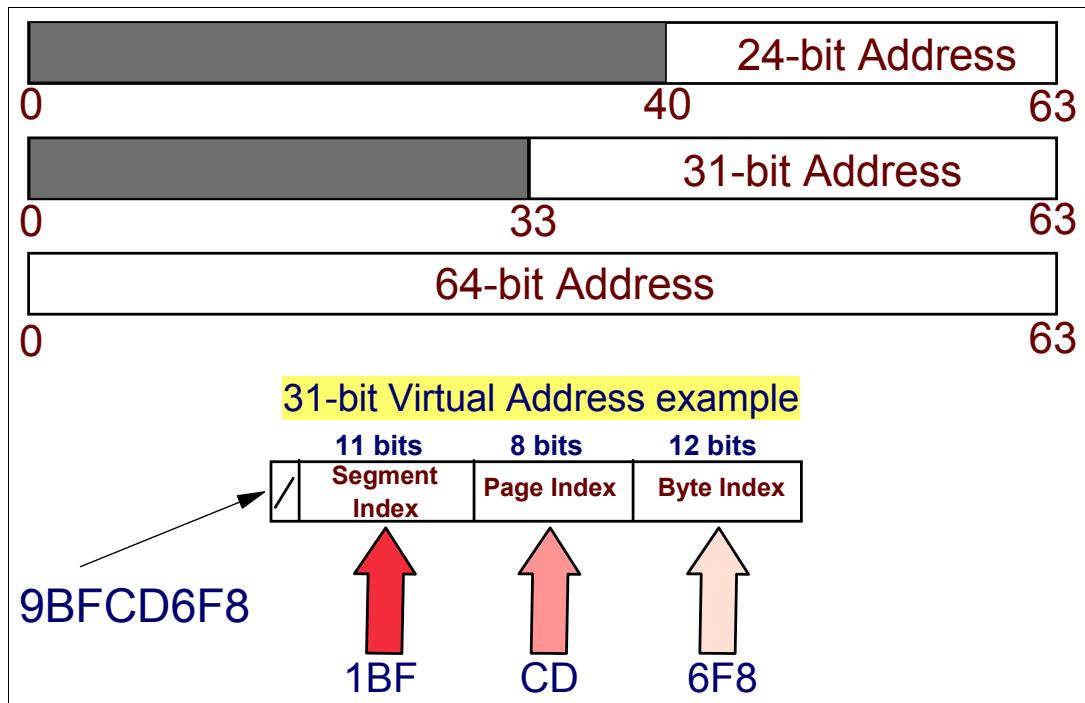


Figure 4-27 Translating 31-bit virtual addresses

Virtual address

A *virtual address* identifies a location in virtual storage. When a virtual address is used for an access to main storage, it is translated by means of DAT to one of the following types of address:

- A real address, which is then further converted by prefixing to an absolute address
- Directly to an absolute address

Address size

An *address size* refers to the maximum number of significant bits that can represent an address. Three sizes of addresses are provided: 24-bit, 31-bit, and 64-bit:

- A 24-bit address can accommodate a maximum of 16,777,216 bytes (16 MB).
- A 31-bit address can address 2,147,483,648 bytes (2 GB).
- A 64-bit address can address 18,446,744,073,709,551,616 bytes (16 EB).

Information: A 24-bit or 31-bit *virtual address* is expanded to 64 bits by appending 40 zeros (24-bit) or 33 zeros (31-bit) on the left before it is translated by means of the DAT process.

A 24-bit or 31-bit *real address* is similarly expanded to 64 bits before it is transformed by prefixing.

A 24-bit or 31-bit *absolute address* is expanded to 64 bits before main storage is accessed. Therefore, the 24-bit address always designates a location in the first 16 MB block of the 16 EB storage addressable by a 64-bit address, and the 31-bit address always designates a location in the first 2 GB block.

31-bit virtual addresses

A virtual address contains the following sections:

- ▶ Bit 33 - 43 identifies the segment number (segment index).
- ▶ Bit 44 - 51 identifies the page number in that segment (page index).
- ▶ Bit 52 - 63 indicates the displacement of the data in that page (byte index).

To translate a 31-bit virtual address (2 GB) into a real address, DAT uses the following bits:

- ▶ Bits 33 - 43 as an index in the segment table to find the entry that points to the page table address of that segment
- ▶ Bits 44 - 51 as an index in the page table entry that has the frame real address, or that has the invalid bit turned on
- ▶ Bits 52 - 63 as the displacement of the data from the beginning of the frame, to be added to the frame real address to get the real address

Information: The size of effective addresses is controlled by bits 31 and 32 of the PSW, the extended-addressing mode bit (bit 31) and the basic-addressing-mode bit (bit 32), respectively. When bits 31 and 32 are both zero, the CPU is in the 24-bit addressing mode, and 24-bit operand and instruction-effective addresses are specified.

When bit 31 is zero and bit 32 is one, the CPU is in the 31-bit addressing mode, and 31-bit operand and instruction-effective addresses are specified.

When bits 31 and 32 are both one, the CPU is in the 64-bit addressing mode, and 64-bit operand and instruction-effective addresses are specified. See Figure 4-26 on page 294 and “Extended addressing mode - EA, BA (bits 31-32)” on page 297 for more information about this topic.

4.28 64-bit virtual address translation

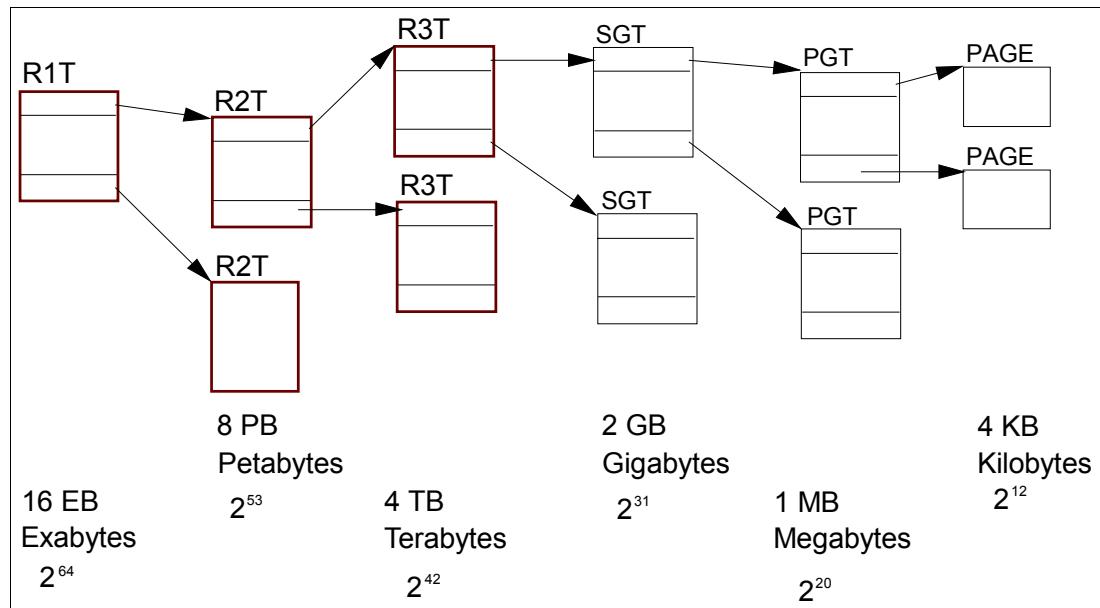


Figure 4-28 64-bit dynamic address translation

64-bit virtual address translation

In a 16 EB address space with 64-bit virtual storage addressing, there are three additional levels of translation tables, called *region tables*. They are known as the region third table (R3T), the region second table (R2T), and the region first table (R1T). The region tables are 16 KB in length, and there are 2048 entries per table. Each region has 2 bytes. Figure 4-28 illustrates the table hierarchy and sizes.

Important: Consider the following points:

- ▶ When the first GETMAIN is obtained above the bar, RSM creates the R3T. The R3T table has 2048 segment table pointers, and it provides addressability to the low 4 TB addresses.
- ▶ When virtual storage addresses greater than 4 TB are GETMAINed, an R2T is created. An R2T has 2048 R3T table pointers, and it provides addressability to 8 PB addresses.
- ▶ When virtual storage greater than 8 PB is GETMAINed, an R1T is created. The R1T has 2048 R2T table pointers, and it provides addressability to 16 EB.

Segment table and page tables

Segment table (SGT) and page table (PGT) formats remain the same as for virtual addresses below the bar. When translating a 64-bit virtual address, and after you have identified the corresponding 2 GB region entry that points to SGT, the process is the same as described previously.

RSM only creates the additional levels of region tables when it is necessary to back virtual storage that is mapped. The region tables are not built until a translation exception occurs.

For example, if an application requests 60 PB of virtual storage, then the necessary R2T, R3T, SGT, and PGTs are only created if they are needed to back a referenced page.

Up to five lookup tables can be needed by DAT to accomplish translation, but the translation only starts from the table that provides translation for the highest usable virtual address in the address space.

4.29 Translating a 64-bit virtual address

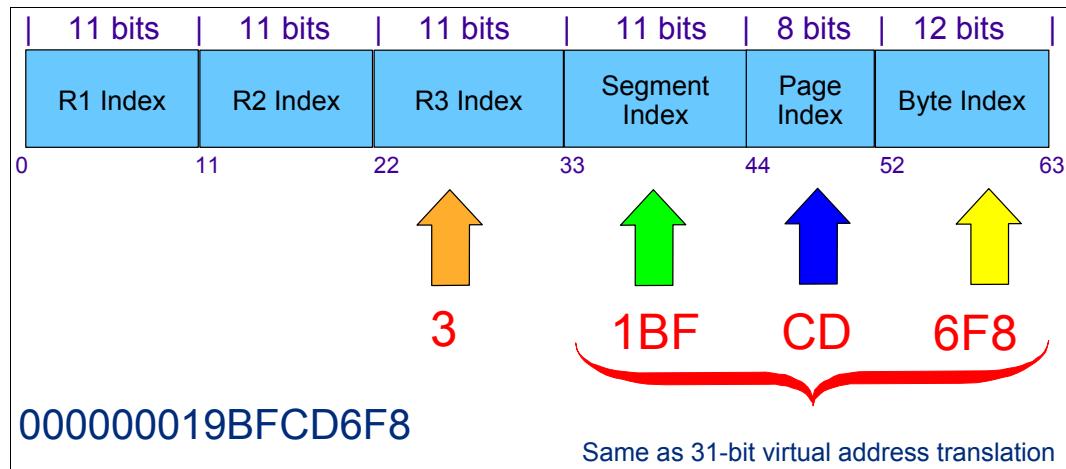


Figure 4-29 Translating a virtual address

Translating a 64-bit virtual address

Up to three additional levels of DAT tables, called *region tables*, are used for translating 64-bit virtual addresses.

To translate a 64-bit virtual address into a real address, DAT uses:

- Bits 0 - 10 are the first region index into the R1T table.
- Bits 11 - 21 are the second region index into the R2T table.
- Bits 22 - 32 are the third region index into the R3T table.
- Bits 33 - 43 are the segment index into the SGT.
- Bits 44 - 51 are the page index into the PGT.
- Bits 52 - 63 indicate the data displacement into the page itself.

With 64-bit virtual addressing, there are now three more 11-bit region indexes to the three region tables, as illustrated in Figure 4-29.

RSM only creates the additional levels of region tables when it is necessary to back storage that is mapped.

4.30 z/OS 1 MB and 2 GB Large Pages

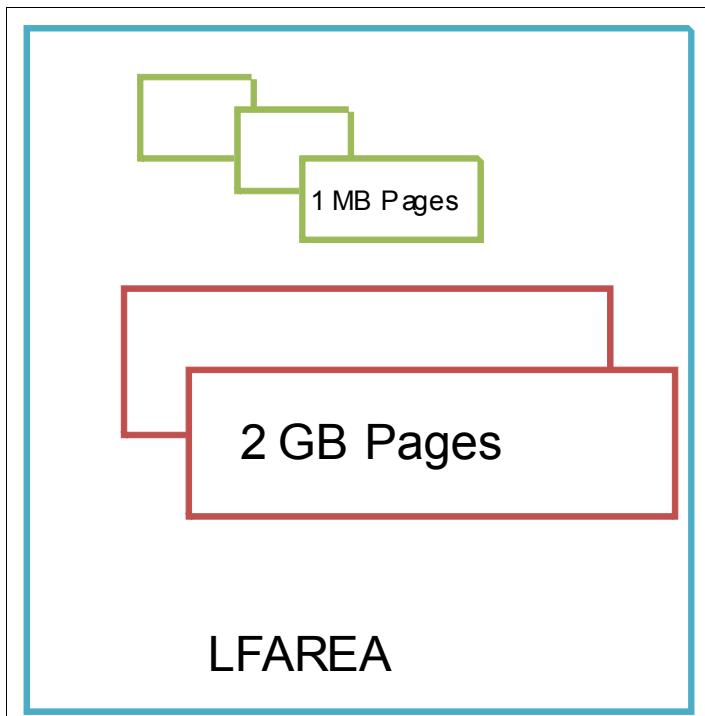


Figure 4-30 1 MB and 2 GB large pages

1 MB and 2 GB Large Pages

z/OS V1R9 with a z10 or later System z server supports 1 MB fixed pages.

z/OS V1R13 or later with the RSM Enablement Offering for z/OS R13, an IBM zEnterprise EC12 (zEC12) or IBM zEnterprise BC12 (zBC12), and the Flash Express feature supports pageable large (1 MB) pages.

z/OS V1R13 or later with the RSM Enablement Offering for z/OS R13, a zEC12 or zBC12, and authorized program analysis report (APAR) OA40967 provides support for 2 GB fixed pages.

The IEASYSxx LFAREA parameter specifies the amount of real storage to be made available for 1 MB and 2 GB large pages. All 1 MB and 2 GB pages are backed by contiguous 4 KB real storage frames, and are allocated from real storage as specified by the LFAREA parameter.

If the system becomes constrained by a lack of sufficient 4 KB pages to handle workload demand, it can use free 1 MB large pages to back 4 KB page requests, enabling the system to react dynamically to changing system storage frame requirements. Virtual storage for large pages are allocated from above the 2 GB bar.

The RSM Enablement Offering for z/OS R13 and a description of the new functions can be found on the following website:

<http://www-03.ibm.com/systems/z/os/zos/tools/downloads/>

Details on the specification of the LFAREA keyword are documented in the *MVS Initialization and Tuning Reference*, SA23-1380.

4.31 System initialization (IPL process)

- ❑ Initialization process at first IPL
- ❑ Starting z/OS (IPL process)
 - Types of IPLs as follows:
 - Cold Start
 - Quick Start
 - Warm Start
- ❑ System initialization process
 - Starting system address spaces
 - Starting other address spaces

Figure 4-31 System initialization (IPL process)

Initialization process at first IPL

At the first IPL after system installation, the system automatically loads the PLPA from the LPALST concatenation. The page data sets for this IPL are those specified in the IEASYS00 parmlib member, plus any specified by the operator.

After the first IPL, you must run IFCDIP00 to initialize the LOGREC data set. This routine must also be run whenever the LOGREC data set is reallocated.

Types of IPL

There are several types of IPLs:

- | | |
|--------------------|---|
| Cold start | This refers to any IPL that loads (or reloads) the PLPA, but does not preserve VIO data set pages. The first IPL after system installation is always a cold start, because the PLPA is initially loaded. Subsequent IPLs are cold starts when the PLPA is reloaded, either to alter its contents or to restore its contents if they were destroyed. |
| Quick start | This refers to any IPL that does not reload the PLPA and does not preserve VIO data set pages. The system resets the page and segment tables to match the last-created PLPA. |
| Warm start | This refers to any IPL that does not reload the PLPA, but preserves journaled VIO data set pages. |

System initialization

To tailor the system's storage parameters, you need a general understanding of the system initialization (IPL) and storage initialization processes.

The system initialization process prepares the system control program (z/OS) and its environment to perform work for the installation. The process essentially consists of the following tasks:

- ▶ System and storage initialization (virtual, real, and auxiliary), including the creation of the common area and the system component address spaces
- ▶ Master scheduler initialization and subsystem initialization

When the system is initialized and the JES is active, the installation can submit jobs and start other address spaces through the START, LOGON, or MOUNT commands.

Also, z/OS creates system component address spaces. z/OS establishes an address space for the master scheduler (*MASTER* address space), and other system address spaces for various subsystems and system components.

Important: Several z/OS components do not need a specific address space. System component address spaces include:

- ▶ Program call/authorization for cross-memory communications
- ▶ System trace
- ▶ Global resource serialization
- ▶ Dumping services

4.32 z/OS address spaces

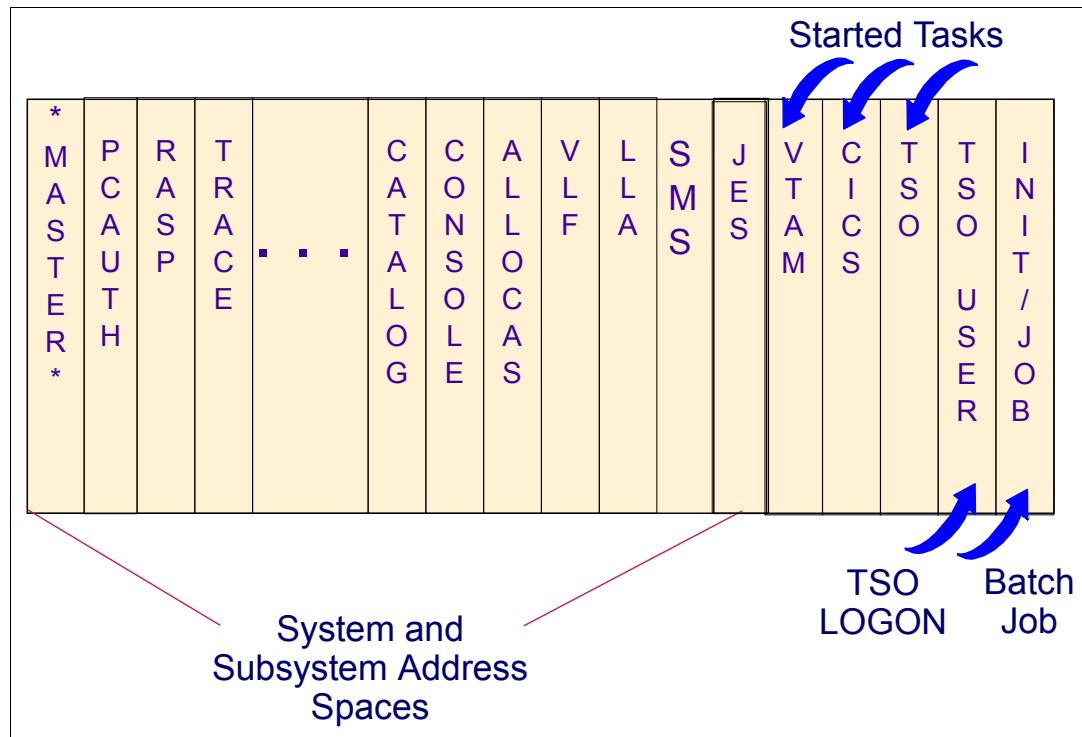


Figure 4-32 System and subsystem address spaces in z/OS

z/OS address spaces

When you start z/OS, master scheduler initialization routines initialize system services, such as the system log and communications task, and start the master scheduler address space (*MASTER*). Each address space created has a number associated to it, known as the address space identifier (ASID). Because the master scheduler is the first address space created in the system, it becomes address space number one (ASID=1). Other system address spaces are then started during the initialization process of z/OS.

Next, subsystem address spaces are started. The master scheduler starts the job entry subsystem (JES2 or JES3). JES is the primary job entry subsystem. Then other defined subsystems are started. All subsystems are defined in SYS1.PARMLIB, member IEFSSNxx. These subsystems are secondary subsystems. See “Subsystem definitions” on page 308 for more information about these subsystems.

System component address spaces

In addition to initializing system areas, MVS establishes system component address spaces. MVS establishes an address space for the master scheduler (the master scheduler address space) and other system address spaces for various subsystems and system components.

Several of the component address spaces are shown in Figure 4-33.

MASTER	Master address space
ABARS, ABARxxxx 1 to 15	DFSMShsm secondary address spaces to perform aggregate backup or aggregate recovery processing.
ALLOCAS	Allocation services and data areas
ANTMAIN	Concurrent copy support
APPC	APPC/MVS component
ASCH	APPC/MVS scheduling
CATALOG	Catalog functions. Also known as CAS (catalog address space).
BPXOINIT	z/OS UNIX System Services
CONSOLE	Communications task
DFM	Distributed File Manager/MVS
DFMCAS	Distributed File Manager/MVS
DLF	Data lookaside facility
DUMPSRV	Dumping services
HSM	DFSMShsm
HZSPROC	IBM Health Checker for z/OS
FTPSERVE	FTP server(s); can be user-specified names.
GDEDFM	For each Distributed File Manager/MVS user conversation that is active, an address space named GDEDFM is created.
GRS	Global resource serialization
IEFSCHAS	Scheduler address space
IOSAS	I/O supervisor, ESCON, I/O recovery
IXGLOGR	System logger
JES2	JES2
JES2AUX	JES2 additional support
JES2MON	JES2 address space monitor
JES3	JES3
JES3AUX	JES3 additional support
JES3DLOG	JES3 hardcopy log (DLOG)
JESXCF	JES common coupling services address space
LLA	Link list
NFS	DFSMS/MVS Network File System address space
OAM	DFSMSdfp Object Access Method
OMVS	z/OS UNIX System Services
PCAUTH	Cross-memory support
PORTMAP	Portmapper function
RASP	Real storage manager (includes advanced address space facilities support)
RMM	DFSMSrmm
RRS	Resource recovery services (RRS)
SMF	System management facilities
SOM	SOMobjects
SMS	Storage management subsystem
SMSPDSE1	Optional restartable PDSE address space. If the SMSPDSE1 address space is started, SMSPDSE manages PDSEs in the LINKLST concatenation and SMSPDSE1 manages all other PDSEs.
SMSVSAM	VSAM record level sharing
TCPIP	TCP/IP for MVS
TRACE	System trace
VLF	Virtual lookaside facility
XCFAS	Cross system coupling facility
VTAM	VTAM
WLM	Workload management

Figure 4-33 System component address spaces

4.33 Subsystem definitions

SYS1.PARMLIB: IEFSSNxx

```
SUBSYS      SUBNAME (subname)
            [CONSNAME (consname) ]
            [INITRTN (initrtn)
            [INITPARM (initparm) ] ]
            [PRIMARY ({NO|YES} )
            [START ({YES|NO} ) ] ]
```

SUBSYS SUBNAME(JES2) PRIMARY(YES)

SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN)
INITPARM('ID=60,PROMPT=YES')

SYS1.PARMLIB: IEASYSxx SSN=xx

Figure 4-34 Subsystem definitions in SYS1.PARMLIB

Subsystem definitions

A *subsystem* is a service provider that performs one function or many functions, but does nothing until it is requested. Although the term “subsystem” is used in other ways, in this document a subsystem must be the master subsystem, or be defined to MVS using the IEFSSNxx parmlib member. You can use either the keyword format or positional format of the IEFSSNxx parmlib member. It is preferable to use the keyword format, because it enables you to define and dynamically manage your subsystems.

Subsystem initialization

Subsystem initialization is the process of readying a subsystem for use in the system. The IEFSSNxx members of SYS1.PARMLIB contain the definitions for the primary subsystems (such as JES2 or JES3) and the secondary subsystems (such as storage management subsystem (SMS), IBM CICS, DB2, and so on). For detailed information about the data contained in IEFSSNxx members for secondary systems, see the installation manual for the specific subsystem.

IEFSSNxx parmlib member

The IEFSSNxx parmlib member enables you to specify the following items:

- ▶ The subsystem initialization routine to be given control during master scheduler initialization
- ▶ The input parameter string to be passed to the subsystem initialization routine
- ▶ A primary subsystem name and whether you want it started automatically

Important: The order in which the subsystems are initialized depends on the order in which they are defined in the IEFSSNxx parmlib member on the SSN parameter. Unless you are starting the SMS, start the primary subsystem (JES) first.

Note: The SMS is the *only* subsystem that can be defined before the primary subsystem.

Types of subsystems

There are two types of subsystems:

- ▶ The primary subsystem, which is the job entry subsystem that MVS uses to complete work. It can be either JES2 or JES3.
- ▶ Secondary subsystems, which provide functions as needed by IBM products, vendor products, or the installation.

MVS communicates with subsystems through the subsystem interface (SSI). Various subsystems require the services of the primary subsystem in their initialization routines. Problems can occur if subsystems that use the subsystem affinity service in their initialization routines are initialized before the primary subsystem. If you are starting SMS, specify its record before you specify the primary subsystem record.

Tip: In general, it is good practice to make the subsystem name the same as the name of the member of SYS1.PROCLIB used to start the subsystem. If the name does not match, you can receive error messages when you start the subsystem.

IEASYSxx parmlib member

The SSN parameter in IEASYSxx identifies the IEFSSNxx member that the system is to use to initialize the subsystems, as follows:

```
SSN=          {aa          }  
             { (aa,bb,...) }
```

The two-character identifier, represented by *aa* (or *bb*, and so on), is appended to IEFSSN to identify IEFSSNxx members of Parmlib. If the SSN parameter is not specified, the system uses the IEFSSN00 Parmlib member.

The order in which the subsystems are defined on the SSN parameter is the order in which they are initialized. For example, a specification of SSN=(13,Z5) causes those subsystems defined in the IEFSSN13 Parmlib member to be initialized first, followed by those subsystems defined in the IEFSSNZ5 Parmlib member.

Information: If you specify duplicate subsystem names in IEFSSNxx parmlib members, the system issues message IEFJ003I to the SYSLOG, the master console, and consoles that monitor routing code 10 messages.

SYS1.PARMLIB is described in detail in *ABCs of z/OS System Programming Volume 2*, SG24-6982.

4.34 Multiprogramming and multiprocessing

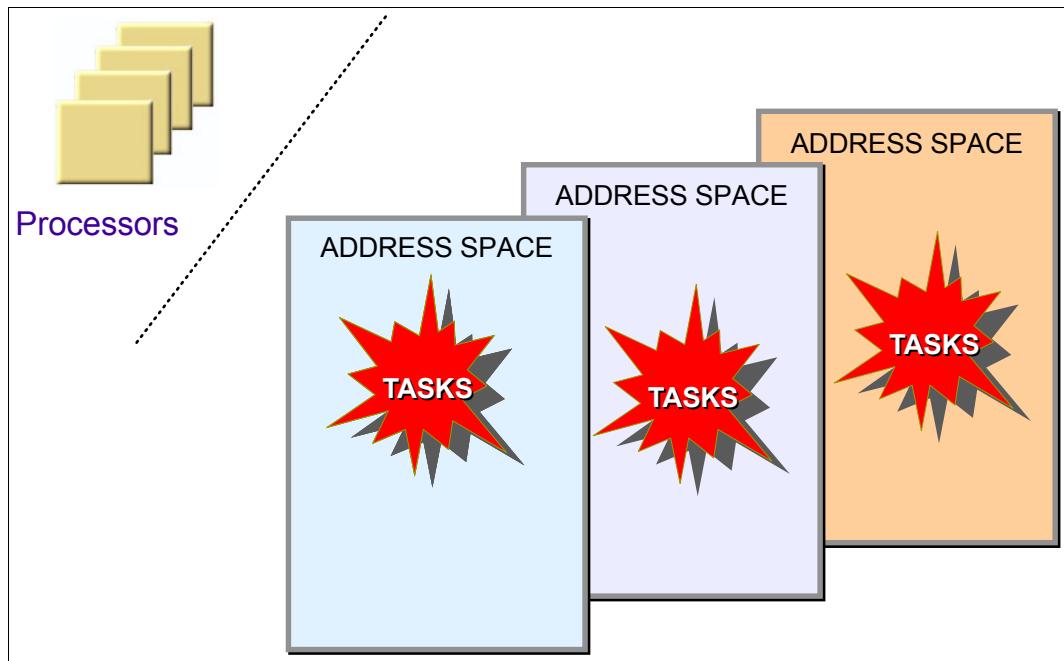


Figure 4-35 Multiprogramming and multiprocessing

Concept of a task in z/OS

The control program creates one *task* in the address space as a result of initiating execution of the job step (the job step task). You can create additional tasks in your program. However, if you do not, the job step task is the only task in the address space being executed. The benefits of a multiprogramming environment are still available even with only one task in the job step. Work is still being performed for other address spaces when your task is waiting for an event, such as an input operation, to occur.

The benefit of creating additional tasks in the job step is that more tasks are competing for control. When a wait condition occurs in one of your tasks, it is not necessarily a task from another address space that gets control; instead, it can be one of your tasks, a portion of your job.

Only one processor at a time can execute the same task. However, several processors executing different tasks can execute the same program (just as people driving different cars going to different places can share the same street).

Multiprogramming

Multiprogramming means that many tasks can be in a system at the same time, with each task running programs in its own address space (or sometimes in the same address space). In a single processor system (such as a city with only one car), only one of these tasks can be active at a time.

However, the active task can lose control of the processor at any time (for example, because of I/O requests that place the task in a wait state, meaning it is not a candidate to get the processor). The OS then selects which task will get control next, based on a number called *dispatching priority*.

Multiprocessing

Multiprocessing is a logical expansion of multiprogramming. Multiprocessing refers to the execution of more than one task simultaneously on more than one processor (which is like a city with several cars). All processors operate under a single copy of the OS, and share the same memory.

Keep the following points in mind:

- ▶ Each processor has a current PSW, its own set of registers, and assigned storage locations.
- ▶ When a single processor shares main storage with other processors, then all of them are controlled by a single OS copy. This is called a *tightly coupled multiprocessing complex*.
- ▶ When a single processor shares a common workload with others, but does not share main storage, this is called a *loosely coupled multiprocessing complex*.

4.35 Program compile, link-edit, and execution

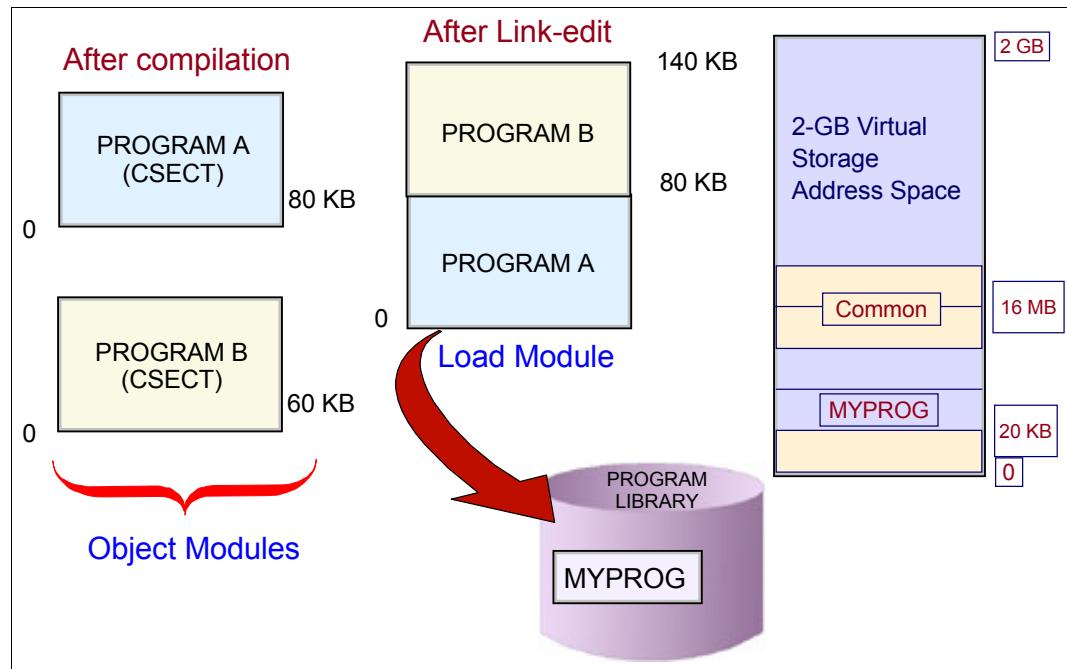


Figure 4-36 Executing a program after compiling and link-editing

Program compile, link-edit, and execution

A z/OS system can appear to be one big block of code that drives your processor. Actually, z/OS is a complex system composed of many different smaller blocks of code. Each of those smaller blocks of code perform a specific (specialized) function in the system.

Each module of symbolic language code is first assembled or compiled by one of the language translators or the assembler. The input to a language translator is a *source module*. The output from a language translator is an *object module*, made of CSECTs.

The Binder is a z/OS processing program that accepts object modules, control statements, and options as input. It combines these object modules, according to the requirements defined by the control statements and options, into a single output load module (executable code). This module can be stored in a partitioned data set (PDS) or PDS extended (PDSE) program library, and loaded into storage for execution by the z/OS component program management loader. A load module can also be an input to the Binder.

Each system function is composed of one or more load modules. This is also true for an installation application. In a z/OS environment, a load module represents the basic unit of machine-readable executable code. Load modules are created by combining one or more object modules and processing them with a link-edit utility. The link-editing of modules is a process that resolves external references and addresses. The functions on your system, therefore, are one or more object modules that have been combined and link-edited.

4.36 Library lookaside

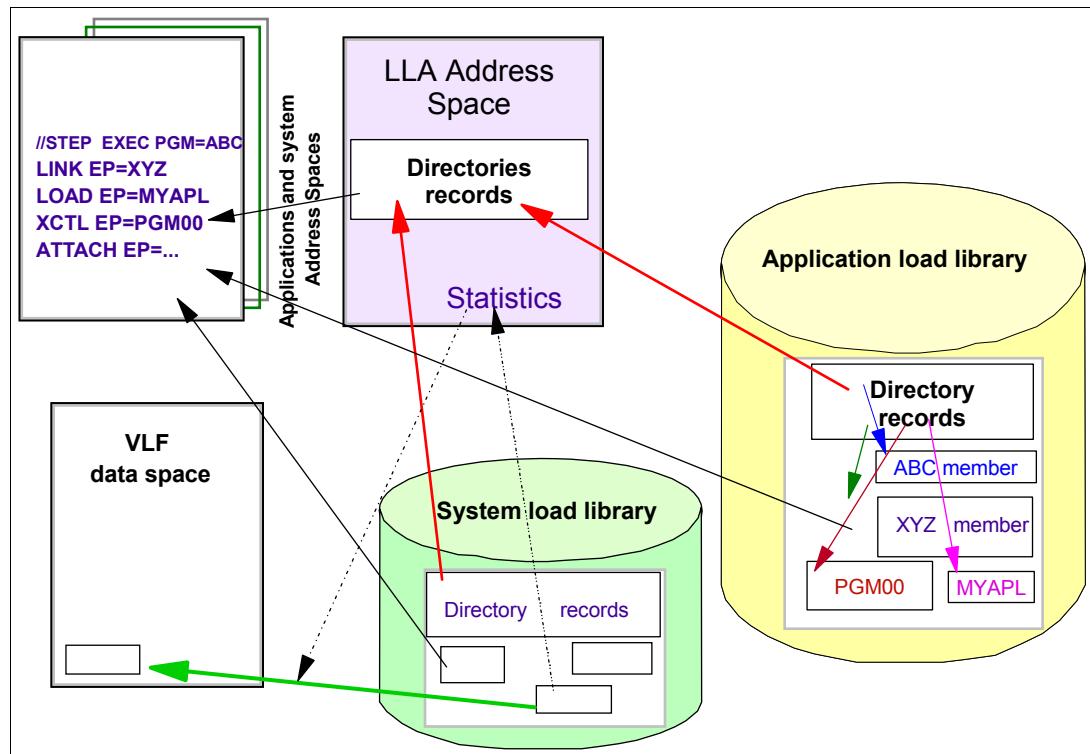


Figure 4-37 Library Lookaside

Library lookaside

Library lookaside (LLA) is a z/OS function that improves system performance by reducing the amount of I/O needed to locate and fetch load modules from DASD storage (PDS data sets). A PDS is stored only on a DASD. It is divided into sequentially organized members, each described by one or more directory entries. Each member has a unique name, 1 to 8 characters long, stored in a directory that is part of the data set. The records of a given member are written or retrieved sequentially.

The main advantage of using a PDS is that, without searching the entire data set, you can retrieve any individual member after the data set is opened. For example, in a program library that is always a PDS, each member is a separate program or subroutine. The individual members can be added or deleted as required. When a member is deleted, the member name is removed from the directory, but the space used by the member cannot be reused until the data set is reorganized (compressed using the IEBCOPY utility).

LLA services run in an LLA address space, which is a started task. They improve module fetch performance in the following ways:

- ▶ LLA maintains, in an LLA address space, copies of the PDS directories. To fetch a module, the system first searches the directory for the load module location in the PDS data set. The system can quickly search the LLA copy of a directory in virtual storage rather than using costly I/O to search the directories on DASD.

- ▶ LLA places copies (staging) of selected load modules in a virtual lookaside facility (VLF) data space (when the LLA class is defined to VLF). VLF is another z/OS component in charge of keeping load modules and specific data in virtual storage, to avoid I/O operations. See “Virtual lookaside facility” on page 315 for more information about this topic.

A data space is like an address space, but it only contains data, not instructions. The system can quickly fetch modules from virtual storage, rather than using slower I/O to fetch the modules from DASD.

- ▶ LLA determines which modules, if staged, provide the most benefit to module fetch performance. LLA evaluates modules as candidates for staging based on statistics that LLA collects about the members of the PDS data sets that it manages, such as module size, fetch count, and the time required to fetch a particular module.

LLA benefits

The benefits of LLA apply only to load modules that are retrieved through the system functions LINK, LOAD, ATTACH, XCTL, and XCTL. Directory entries for the primary system library (SYS1.LINKLIB), load modules libraries concatenated to it as declared in the LNKLSTxx member of SYS1.PARMLIB.

Additional production libraries named in SYS1.PARMLIB(CSVLLAxx) are read into the private area of the LLA address space (AS) during its initialization. Subsequent searches for programs in these libraries begin with the directories in LLA, and not in the directories on DASD.

You obtain the most benefit from LLA when you have both LLA and VLF functioning together. This can be achieved by defining the LLA class to VLF and starting VLF, so the most active modules from LLA-managed libraries are staged into the DCSVLLA VLF data space.

4.37 Virtual lookaside facility

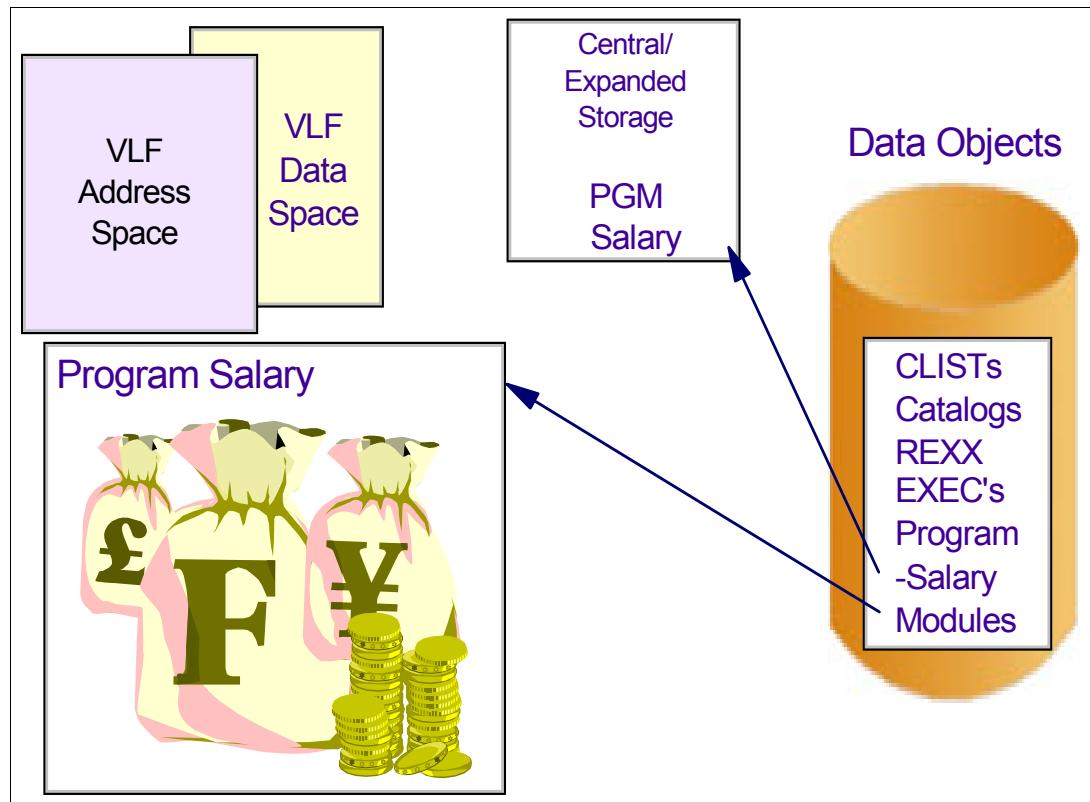


Figure 4-38 Virtual lookaside facility

Virtual lookaside facility

VLF is a z/OS component that improves the response time of applications that must retrieve a set of objects (data or code for many users). VLF creates and manages a data space to store an application's most frequently used objects. When the application makes a request for an object, VLF checks its data space to see if it is there. If the object is present, VLF can rapidly retrieve it without requesting I/O to DASD.

To take advantage of VLF, an application must identify the objects that it needs. Objects should be small to moderate in size, named according to the VLF naming convention, and associated with an installation-defined class of objects.

z/OS components and products, such as LLA (load modules), TSO/E (Restructured Extended Executor (REXX) and command list (CLIST) procedures), control address space (CAS, for user catalog entries), and RACF (RACF database entries), use VLF as an alternate way to access their objects. Because VLF uses virtual storage for its data spaces, there are main storage performance considerations that each installation must weigh when planning for VLF.

Important: VLF is intended for use with major applications. Because VLF runs as a started task that the operator can stop or cancel, it cannot take the place of any existing means of accessing data on DASD.

Any application that uses VLF must also be able to run without it.

4.38 Memory hierarchy

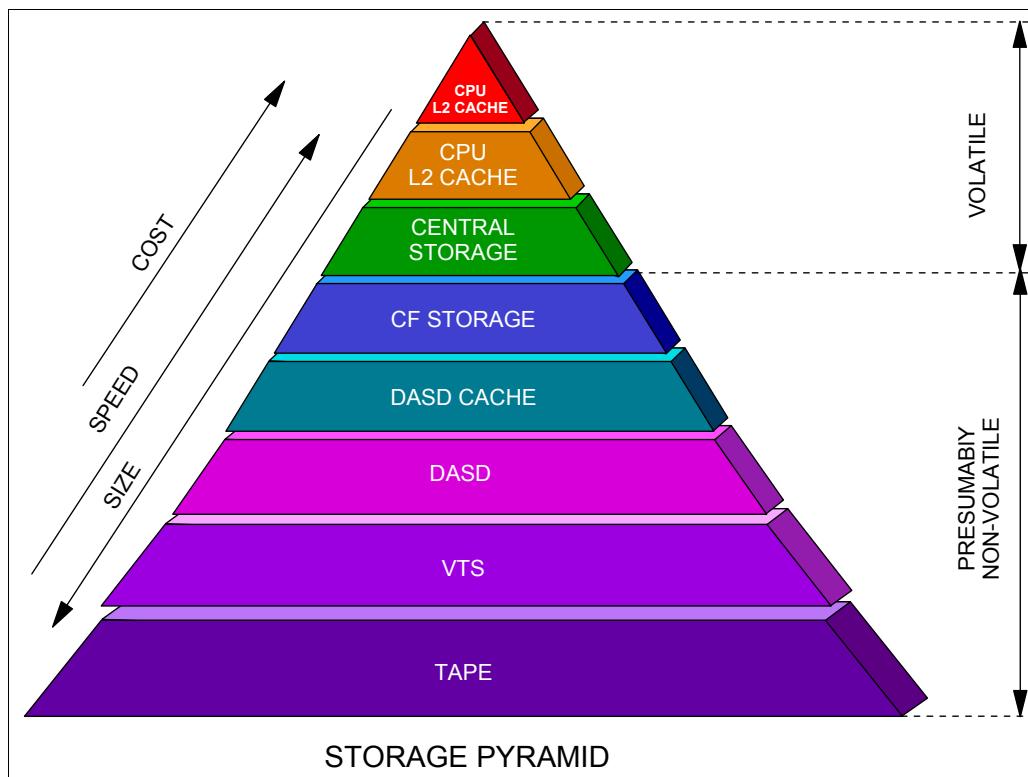


Figure 4-39 Memory hierarchy

Memory hierarchy

The memory hierarchy shown in Figure 4-39 illustrates the different layers of memory along a computational process. The higher the layer of memory, then the faster, more expensive, and smaller it is. The top layers are directly accessed by the processor.

The problem is how to distribute data and programs across memory, with regard to their importance to the business and their frequency of use. There are several considerations to remember:

- ▶ Which data (and programs) need to migrate to a lower layer when the occupied layer is full or close to full? In this case there are two algorithms that can be used:
 - A least recently used (LRU) algorithm keeps the most referenced data or programs in storage. This assumes, for commercial processing, that if data or code was referenced in the past, it will likely be referenced in the future, as well. LRU is good for random access, such as in an online environment.
 - A sequential algorithm is used for sequential access. After data is accessed in a high layer (such as main storage), it can be demoted to the DASD cache because there are no chances for a revisit.

- ▶ What do you do with data that was updated in a volatile layer? In this case, there are also two algorithms that can be used:
 - Store-in, where the updated data is kept in the volatile layer (main storage, for example) without being copied to the non-volatile lower layer (the DASD cache, for example). This algorithm favors performance but demands a log to keep write integrity. Examples of store-in include the L2 cache and the DB2 buffer pool in virtual storage (main storage).
 - Store-through, where the updated data is immediately (synchronously) copied to the volatile lower layer (generally demanding an I/O operation to the DASD cache). This algorithm favors write integrity, but does not help performance. Examples of store-through include the L1 cache and the VSAM buffer pool in virtual storage (main storage).
- ▶ How can integrity problems be solved when the lower layers are shared between different copies of z/OS systems? Such data sharing helps continuous availability (24x7), but a coupling facility (CF) is needed.



ShopzSeries

ShopzSeries is an Internet application that you can use to order z/OS software products and service. ShopzSeries provides entitled service ordering and service delivery capabilities for the IBM z/OS platform products electronically using the Internet. ShopzSeries is the primary ordering and delivery method for software service for ordering and receiving z/OS. Using ShopzSeries, you can order corrective and preventive service over the Internet, with delivery over the Internet or by tape.

Service with ShopzSeries reduces your research time and effort by using your uploaded System Modification Program/Extended (SMP/E) consolidated software inventory (CSI) to ensure that all applicable service, including reach-ahead service, for the installed function modification identifiers (FMIDs) in the target zones is selected. You can find more information on the ShopzSeries web address:

<http://www.ibm.com/software/shopzseries>

SMP/E Internet Service Retrieval

Obtaining software service over the Internet was improved in z/OS V1R7 (and in the SMP/E V3R4 product and later releases) with the introduction of SMP/E Internet Service Retrieval (ISR). Without this function, ordering and obtaining service over the Internet through ShopzSeries involves several steps:

1. Running an SMP/E job to create an inventory file
2. Initiating a service order transaction on ShopzSeries
3. Uploading the inventory file
4. Waiting for notification that the service package is available
5. Accessing the package on ShopzSeries
6. Running an SMP/E job to download and process the service package

SMP/E ISR consolidates these steps into one. You use a new form of the RECEIVE command to run an SMP/E job to place a service order, wait for the IBM server to fulfill the order, download the service package (which contains program temporary fixes (PTFs) and HOLDDATA), and process its contents, all in one step. With SMP/E ISR, you can request service on demand, and even automate the service delivery process. For example, you can schedule an SMP/E job to run once a week, or even every night, to order and download the latest HOLDDATA and critical PTF service, and have these service updates available exactly when you want.

5.1 IBM software ShopzSeries

The screenshot shows the IBM ShopzSeries start menu. At the top, there's a navigation bar with links for Home, Business solutions, IT services, Products, Support & downloads, and My IBM. A search bar is also present. Below the navigation bar, a welcome message for Redelf Janßen is displayed. The main content area features a title 'ShopzSeries' and a brief introduction: 'Welcome to ShopzSeries, IBM's productivity tool for planning and ordering zSeries software. With ShopzSeries you can:'. It lists several bullet points: 'order tailored product packages,' 'order tailored service packages,' 'review your software licenses, and' 'plan for future upgrades.' To the left is a sidebar with links for ShopzSeries, Product catalog, Help, News, Feedback, and Customer service. Below the sidebar is a 'Related links' section with links to various IBM products like zSeries home, Operating systems, and Enhanced HOLDDATA. On the right, there's a 'My ShopzSeries' section with links for Sign in and Register, and a note about requesting access. A banner at the bottom right says 'zSeries software for the on demand world' and 'Make on demand a reality'. At the bottom of the page, there are links for E-mail this page, Print this page, About IBM, Privacy, Contact, and Terms of use.

Figure 5-1 ShopzSeries start menu

IBM software ShopzSeries

IBM ShopzSeries is a web-based application, where you as a registered client can perform several tasks related to IBM System z software ordering and tracking.

In this section you will learn how you can place z/OS software orders using IBM ShopzSeries. You will also learn how you can handle the CustomPac receive process with a downloaded ServerPac.

You can reach the start page by entering the following link:

<http://www.ibm.com/software/shopzseries>

You must sign in using your IBM ID and password.

5.2 ShopzSeries order process

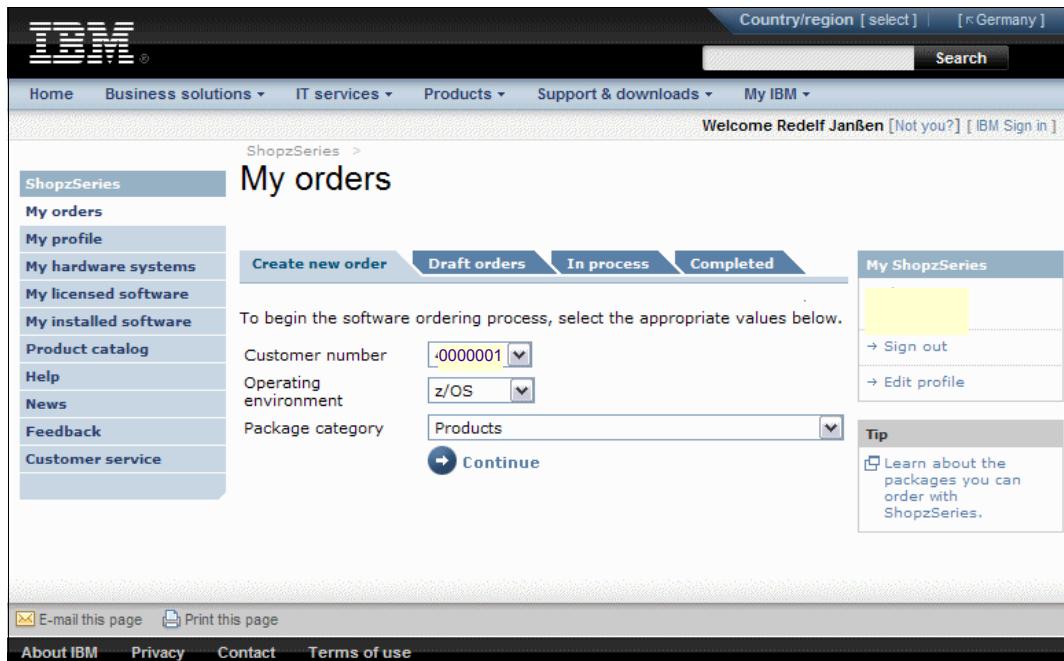


Figure 5-2 ShopzSeries order process menu

ShopzSeries order process

After you are logged on to the ShopzSeries application, click **My Orders** on the left side of the web page and the window shown in Figure 5-2 will be displayed.

Placing your order

You must place your order using ShopzSeries to choose Internet delivery. When your order is ready for download, you will receive an email from IBM. When you log on to ShopzSeries to access your order, the download pages are dynamically built. You can either download your order directly to the host using the ServerPac dialog, or download it to your workstation first using Download Director in ShopzSeries and then upload it to the host.

The dialog “Server” option generates a RECEIVE job that uses the SMP/E GIMGTPKG utility to retrieve your order directly from the IBM server. GIMGTPKG will place the data in your Download file system. The dialog “File System” option will generate a RECEIVE job that will retrieve your order from the Download file system after you have uploaded it from your workstation, if you chose to download it to your workstation first.

Later, the dialog RESTORE job will use the SMP/E GIMUNZIP program to load your new target system’s volumes from the Download file system. When you have finished, you can delete the Download file system. However, consider dumping it first, just in case.

Information for your order

When you create a new order, you must enter the following information into the window:

- ▶ Your Customer number
- ▶ Operating environment (in this case, z/OS)
- ▶ Package category (in this case, Products)

After entering this information, press **Continue** to proceed.

5.3 Specify order basics step 1 of 8

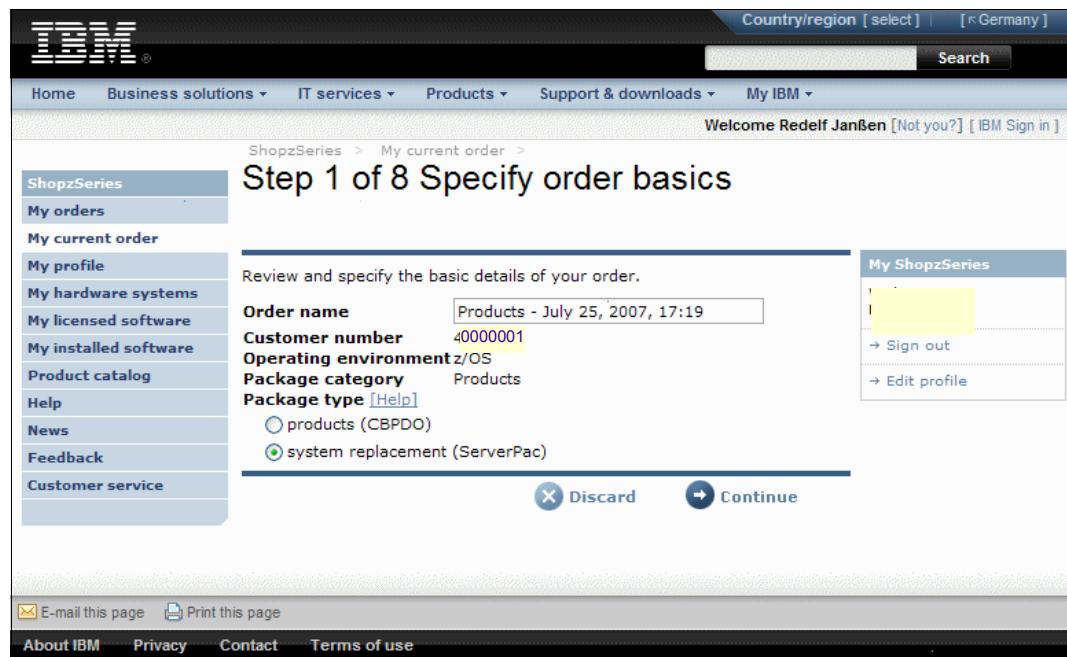


Figure 5-3 Specify order basics menu

Specify order basics step 1 of 8

On this panel, you must choose the package type that you want to order. You can either select products using Custom-Built Product Delivery Option (CB PDO) or a ServerPac for system replacement:

1. For this installation process, we selected **system replacement**.
2. Then, we pressed **Continue**.

5.4 Select hardware systems step 2 of 8

The screenshot shows the IBM ShopzSeries interface. At the top, there's a navigation bar with links for Home, Business solutions, IT services, Products, Support & downloads, and My IBM. A search bar is also present. On the right, it says 'Welcome Redelf Janßen [Not you?] [IBM Sign in]'. Below the navigation, a left sidebar lists 'ShopzSeries', 'My orders', 'My current order', 'My profile', 'My hardware systems' (which is highlighted in blue), 'My licensed software', 'My installed software', 'Product catalog', 'Help', 'News', 'Feedback', and 'Customer service'. The main content area has a title 'Step 2 of 8 Select hardware systems' and a subtitle 'Products - July 25, 2007, 17:19'. It asks to 'Select the system(s) on which you plan to run the software you are ordering.' and provides a tip about saving work if the session times out. A table titled 'Hardware systems for customer number 0000001' lists several systems with columns for Aggregation, Select, Nickname, Designated machine / Description, Type-model, Serial no., and System no. The system '2094022991E' is selected, indicated by a checked checkbox in the 'Select' column.

Aggregation	Select	Nickname	Designated machine / Description		
			Type-model	Serial no.	System no.
None	<input type="checkbox"/>	NODESIG	NODESIG / no description available	?	?
	<input type="checkbox"/>	XCDCOM1	XCDCOM1 / no description available	?	?
	<input type="checkbox"/>	XCDCOM1	XCDCOM1 / no description available	?	?
	<input checked="" type="checkbox"/>	2094022991E	2094022991E / no description available	?	?
	<input type="checkbox"/>	3084QC8	3084QC8 / no description available	?	?
	<input type="checkbox"/>	30900271061	30900271061 / no description available		

My ShopzSeries

- Sign out
- Edit profile

Tip

Save your order periodically to avoid losing work if your session times out.

Tip

→ Assign nicknames to your hardware systems.

Tip

Report inaccuracies in your license information.

Figure 5-4 Select hardware systems menu

Select hardware systems step 2 of 8

In this step, you find the installed base of your zSeries hardware systems:

1. You must select one or more systems where the appropriate software licenses exist, by selecting the box (or boxes) under **Select**.
2. Then press **Continue** (not shown here) to proceed to the next step.

5.5 Report installed software step 3 of 8

The screenshot shows a web-based application for managing software reports. At the top, there's a navigation bar with links for Home, Business solutions, IT services, Products, Support & downloads, and My IBM. A search bar is also present. Below the navigation, a welcome message for 'Redelf Janßen' is shown, along with links for 'Not you?' and 'IBM Sign in'. The main content area has a title 'Step 3 of 8 Report installed software' and a subtitle 'Products - July 25, 2007, 17:19'. On the left, a sidebar menu includes 'ShopzSeries', 'My orders', 'My current order' (which is selected), 'My profile', 'My hardware systems', 'My licensed software', 'My installed software' (selected), 'Product catalog', 'Help', 'News', 'Feedback', and 'Customer service'. The central panel contains a descriptive text about ServerPac orders and two radio button options: 'Do not use a report for this order' (selected) and 'Upload a new report for this order'. It also features input fields for 'Report name' and 'File name' with a 'Browse...' button. At the bottom of this panel are buttons for 'Previous', 'Save', 'Discard', and 'Continue'. To the right, there's a 'My ShopzSeries' sidebar with 'Sign out' and 'Edit profile' options, and a 'Tip' section with a link to learn how to create installed software reports.

Figure 5-5 Report installed software menu

Report installed software step 3 of 8

If you order a ServerPac electronically, you might decide to upload a report that contains the installed base of your software that runs under z/OS. This report is used to verify your actual product status, software upgrade paths, and missing prerequisites:

1. If you use this option, the order process will consider this and give you the current releases of all of your installed components. (In this case, we did not use this option.)
2. To proceed, press **Continue**.

5.6 Shop for products step 4 of 8

The screenshot shows the IBM ShopzSeries interface. At the top, there's a navigation bar with links for Home, Business solutions, IT services, Products, Support & downloads, and My IBM. A search bar is also present. Below the navigation, a welcome message says "Welcome Redelf Janßen [Not you?] [IBM Sign in]". On the left, a sidebar titled "ShopzSeries" contains links for My orders, My current order, My profile, My hardware systems, My licensed software, My installed software, Product catalog, Help, News, Feedback, and Customer service. The main content area is titled "Step 4 of 8 Shop for products" and shows the date "Products - July 25, 2007, 17:19". It includes a note about selecting a product catalog view, a "Products in order: 0" section, and a "Catalog view" section where users can choose a group (MVS - z/OS Operating System), language (English (US)), and filter (Show all products). There's also a tip about viewing a legend. At the bottom, there are buttons for Previous, Save, Discard, and Continue, along with links for E-mail this page and Print this page, and footer links for About IBM, Privacy, Contact, and Terms of use.

Figure 5-6 Shop for products menu

Shop for products step 4 of 8

In step 4 of the order process, you can select the products that you need. To order a z/OS ServerPac, you must select valid values from the three menus:

1. Group. Enter the kind of operating system (OS) that you want to order.
2. Language. Select from the list of available languages for the selected product.
3. Filter. Show all available products from the selected operating system, or select only a subset.
4. After you make your selection, press **Show catalog** to see the components in detail. The list is shown in 5.7, “Shop catalog” on page 326.

5.7 Shop catalog

The screenshot shows the IBM ShopzSeries interface. At the top, there's a navigation bar with links for Home, Business solutions, IT services, Products, Support & downloads, and My IBM. A search bar is also present. Below the navigation, it says "Welcome Redlef Janßen [Not you?] [IBM Sign in]". The main content area is titled "Step 4 of 8 Shop for products" and "Products - July 25, 2007, 17:19". On the left, a sidebar lists "ShopzSeries", "My orders", "My current order", "My profile", "My hardware systems", "My licensed software", "My installed software", "Product catalog", "Help", "News", "Feedback", and "Customer service". The "Product catalog" link is highlighted. The main content area has a heading "Select a view of the product catalog to add products. Repeat the process with other views, and then continue to the next step to review the complete contents of your order." It shows a "Products in order: 0" section with dropdowns for Group (MVS - z/OS Operating System), Language (English (US)), and Filter (Show all products). Below this is a button "Show catalog". Under "MVS: z/OS Operating System", there's a list of products:

Product	Price	Language
[S694-A01] z/OS V1 Base [FMIDs]	1.08.00	English (US)
[S655-P97] Encryption Facil Encrypt Ser [FMIDs]	1.02.00	English (US)
[S655-P97] Encryption Facil dss Encrypt [FMIDs]	1.02.00	English (US)
[S694-A01] z/OS V1 Alternate Base [FMIDs]	1.08.00	English (US)
[S694-A01] z/OS V1 BDT FTF [FMIDs]	1.08.00	English (US)
[S694-A01] z/OS V1 BDT SNA NJE [FMIDs]	1.08.00	English (US)

On the right, there's a "My ShopzSeries" sidebar with "Sign out" and "Edit profile" options, and a "Tip" section with "View a legend for this page".

Figure 5-7 Shop catalog menu

Shop catalog

Product orders contain software products that you can add to or replace on your hardware systems. ShopzSeries enables you to order a variety of product orders for each operating environment.

z/OS products

The CBPDO product adds or upgrades individual products on an existing system. These orders initially contain no products, and you shop for products from a catalog. Your z/OS license entitles you to order this package.

A system replacement (ServerPac) replaces an entire existing system, or subsystem. These orders are automatically primed with the current releases of all of the products installed on your system, making it easy to order an updated system replacement. Your z/OS license entitles you to order this package.

On the window, as shown in Figure 5-7, you see all 67 products that belong to the z/OS operating system. Depending on your license, you can select the appropriate components. In this case, select the z/OS base, which contains all features, as shown in 1.6, “z/OS V2R1 base elements” on page 10.

You can continue shopping by pressing **Continue** (not shown in this example).

5.8 Specify order contents step 5 of 8

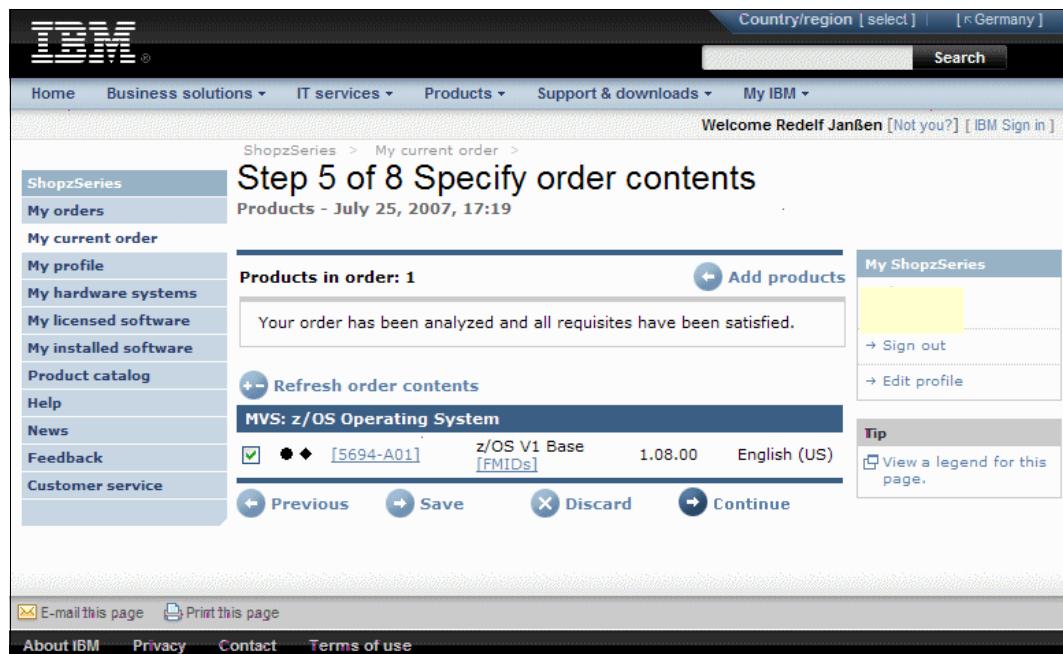


Figure 5-8 Specify order contents menu

Specify order contents step 5 of 8

The process of creating an order is broken up into a series of small steps. Various categories of orders have various steps:

- ▶ Service orders
- ▶ Service subscription orders
- ▶ Automated delivery certificates
- ▶ Product orders
- ▶ Driving systems
- ▶ Customized service
- ▶ Customized products

While you are working on an order, you can navigate to other features of ShopzSeries and quickly return to your order using the **My current order** link in the left navigation area. Save your order periodically by using the **Save** button on the bottom of each step. That way, if you leave ShopzSeries or your session times out, you will be able to find and continue working on your saved draft orders on the Draft orders page.

Continue with the next steps in specifying order contents:

1. As a result of step 4, your order has been verified.
2. In step 5, you can make changes to the contents, or keep them as they were selected.
3. Press **Continue** to proceed to step 6.

5.9 Select new licenses step 6 of 8

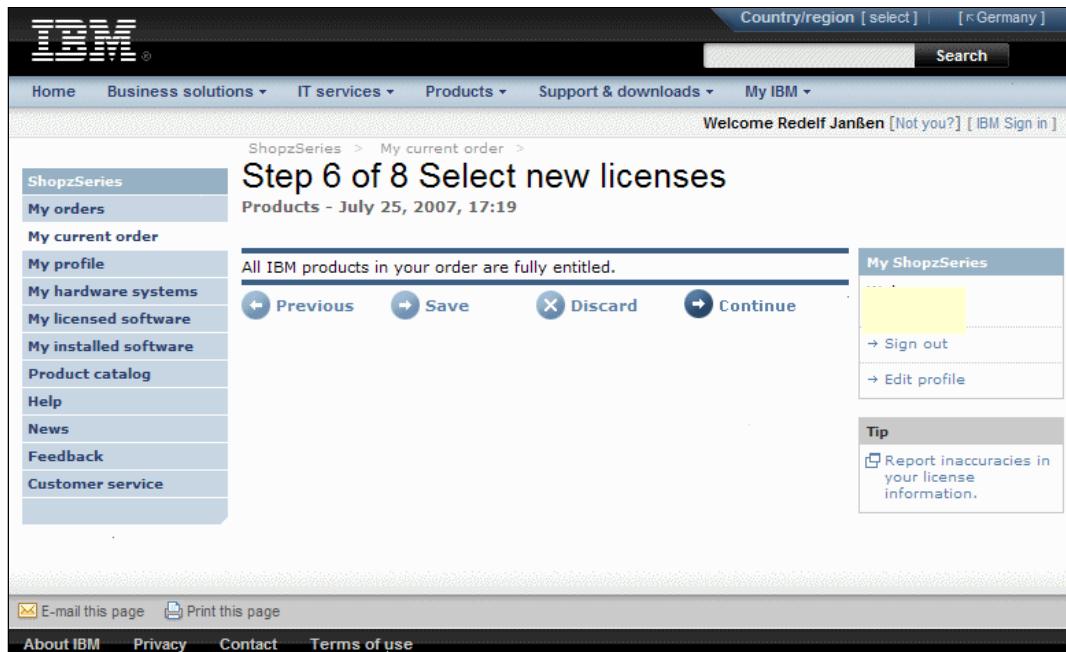


Figure 5-9 Select new licenses menu

Select new licenses step 6 of 8

ShopzSeries gives you access to an online snapshot of your zSeries software licenses. Three types of reports are available:

- ▶ Software license overview
- ▶ Software license details
- ▶ Version chessboards

You can save these reports to files for later review, printing, and so on.

Restriction: See the Customer Service page for information about how to resolve discrepancies in your licensing records. Software license reports are not available in all countries.

Continue with the next steps in specifying order contents:

1. Step 6 asks for the selection of new licenses.
2. In this case, everything was in place, so **Continue** was pressed.

5.10 Specify delivery options step 7 of 8

The screenshot shows the IBM ShopzSeries website interface. At the top, there's a navigation bar with links for Home, Business solutions, IT services, Products, Support & downloads, My IBM, and a search bar. Below the navigation is a welcome message for 'Redelf Janßen'. The main content area is titled 'Step 7 of 8 Specify delivery options' and shows the date 'Products - July 25, 2007, 17:19'. On the left, a sidebar menu lists 'ShopzSeries', 'My orders', 'My current order', 'My profile', 'My hardware systems', 'My licensed software', 'My installed software', 'Product catalog', 'Help', 'News', 'Feedback', and 'Customer service'. The right side features a 'My ShopzSeries' section with 'Sign out' and 'Edit profile' links. A 'Tip' box contains the text: 'Consult with your tax advisor as your selection of software delivery 'preferred media' may effect taxability for sales and use tax purposes.' The central form area has sections for 'Delivery media' (with a dropdown set to 'Internet' and a 'Go' button) and 'Shipping address'. There are also checkboxes for 'Do not ship any physical materials, I require 100% Internet delivery.'

Figure 5-10 Specify delivery options menu

Specify delivery options step 7 of 8

Continue with the next steps in specifying order contents:

1. After you submit an order, you can review it by clicking its name in the In process section of the "My orders" page. (If all processing for the order has already completed, however, you can find it in the Completed section.)
2. You can also track your order's progress. If you selected Internet delivery, you can download it from ShopzSeries.
3. Most ShopzSeries orders are eligible for Internet delivery. If your order is eligible, you will be able to select Internet as the preferred delivery media for your order.
4. If you prefer the delivery by cartridge, you can enter the type of cartridge here (possible types are 3480, 3490, 3590, and 3592).

z/OS host requirements

Internet Delivery requires z/OS V1R3 (5694-A01) or higher. Internet Delivery might also require an appropriate version of SMP/E, depending on the type of package and your selected download method.

For Internet Delivery of CBPDO or service using the SMP/E RECEIVE FROMNETWORK function, you need IBM SMP/E for z/OS and OS/390 V3.1 (5655-G44), including SMP/E PTF UR53608 (SMP/E level 31.12)

For Internet Delivery of ServerPac, which uses the SMP/E GIMGTPKG service routine, you need IBM SMP/E for z/OS and OS/390 V3.3 (5655-G44) or higher.

Internet delivery also requires:

- A download file system

Your order is provided in a compressed format and is saved in a download file system. The size of this file system should be approximately twice the compressed size of your order, to accommodate the order and workspace to process it.

- ▶ Firewall configuration

If your enterprise requires specific commands to enable the download of your order using File Transfer Protocol (FTP) through a local firewall, you must identify these commands for later use in the ServerPac dialog or RECEIVE FROMNETWORK job (RFNJOB), which manages the download of your order.

- ▶ The correct dialog level for ServerPac orders

If you are using a dialog with a Package Version earlier than 17.00.00, you must upgrade the dialog to this level or later. To determine whether you have the correct dialog level, look for this text at the bottom of the main panel, CPPPPOLI:

This dialog supports electronic delivery.

If your dialog is not at the minimum level, follow the migration scenarios and steps described in ServerPac: Using the Installation dialog.

SMP/E

SMP/E is an element in z/OS, or you can order the current SMP/E product as a no-charge product (5655-G44), entitled to z/OS clients. You can even download SMP/E V3.3 from the Internet, but if you do, also order it separately to ensure that your software profile is updated and that you are registered to receive service.

This package includes the function required for Internet delivery, and is also intended to provide the SMP/E network capabilities in situations where you do not have the required level of SMP/E installed. Download SMP/E V3.3.

ICSF

Internet Delivery also requires either ICSF (active) or SMP/E V3.4: Integrated Cryptographic Service Facility (ICSF).

ICSF is a component of base element Cryptographic Services, and is used by SMP/E V3 to calculate Secure Hash Algorithm (SHA)-1 hash values. These hash values are calculated by the GIMGTPKG service routine or the RECEIVE FROMNETWORK function for files in a GIMZIP package to verify the integrity of the data in the package. You must have ICSF configured and active.

Next steps

In step 7, you are asked to specify your delivery medium of choice:

1. In this case, Internet Delivery was chosen to demonstrate the process of ordering and receiving through this medium.
2. Press **Continue** to proceed to the last step of the order process.

5.11 Review and submit order step 8 of 8

The screenshot shows the 'Step 8 of 8 Review and submit' page of the IBM ShopzSeries website. The top navigation bar includes links for Home, Business solutions, IT services, Products, Support & downloads, My IBM, and a search bar. A welcome message for 'Redefl Janßen' is displayed. The main content area is titled 'Step 8 of 8 Review and submit' and shows the date 'Products - July 25, 2007, 17:19'. It contains a summary message: 'Your order is ready to be submitted for fulfillment. Please review it before submitting it.' Below this, a note states: 'All electronically enabled publications for this order will be provided via the Internet.' On the left, a sidebar menu lists 'ShopzSeries', 'My orders', 'My current order', 'My profile', 'My hardware systems', 'My licensed software', 'My installed software', 'Product catalog', 'Help', 'News', 'Feedback', and 'Customer service'. On the right, a 'My ShopzSeries' sidebar offers 'Sign out' and 'Edit profile' options. The central content is organized into sections: 'Order basics' (listing Order name, Date created, Last modified, Customer number, Operating environment, Package category, and Package type), 'Hardware systems' (listing Aggregation: None and a specific item 2094022991E), 'Installed software' (listing None), and 'Order contents' (listing MVS: z/OS Operating System, 5694-A01, z/OS V1 Base, 1.08.00, English (US)).

Figure 5-11 Review and submit order menu

Review and submit order step 8 of 8

The final step of every product order is to review all of your selections from the previous steps:

1. As you review your order information, you can jump back to any step in which you want to change your choices.
2. After you finish verifying your order, go to the bottom of the web page (not shown on Figure 5-11) and click **Submit** to submit your z/OS order for processing by IBM.
If the order does not include any new licenses, it is submitted directly for fulfillment. However, if you did request new licenses, then it is submitted for pricing and you will be notified when your pricing is ready.
3. A few days after submitting your order, you will receive an email from IBM indicating that the ServerPac has been produced and is ready for download.

5.12 In process orders

The screenshot shows the IBM ShopzSeries 'My orders' page. The left sidebar has links for 'My orders', 'My current order', 'My profile', 'My hardware systems', 'My licensed software', 'My installed software', 'Product catalog', 'Help', 'News', 'Feedback', and 'Customer service'. The main content area has a message 'Your survey response has been received. Thank you.' Below it are tabs for 'Create new order', 'Draft orders', 'In process' (which is selected), and 'Completed'. A note says 'To review or process an order, click on its name. To track it, click on its status.' with a 'Refresh order status' link. A table titled 'In process orders' lists one item: 'U00404081 - Products - July 25, 2007, 17:19' with 'Customer number: 0000001'. The 'Status' column shows 'Physical Submitted Internet Manufacturing'. On the right, a sidebar has 'My ShopzSeries' with 'Sign out' and 'Edit profile' options. At the bottom are links for 'E-mail this page', 'Print this page', 'About IBM', 'Privacy', 'Contact', and 'Terms of use'.

Figure 5-12 In process orders menu

In process orders

Orders placed through ShopzSeries can also be tracked through delivery directly from ShopzSeries. To track your orders, go to the In process section of the "My orders" page. The current status of each order is displayed. If there is additional tracking information for an order, the status can appear as a link that can be followed to view the additional information.

For service orders, when you review the order by clicking its name, you can also see a Manufacturing order number and a Manufacturing status, which are returned by the service manufacturing system. This information can help in the unlikely event that problems occur with your order. For example, if the service cannot be sent through the Internet, or if the service you ordered cannot be found, the Manufacturing status field will inform you of the situation.

The menu as shown in Figure 5-12 displays an overview of your current orders. There can be draft orders, orders in process, or completed orders. If you have placed several orders, you can see the status of each one.

On the status field shown here, you can see that the order submitted is in the process of being manufactured.

5.13 Finished order

The screenshot shows the 'My orders' page of the IBM ShopzSeries website. At the top, there's a navigation bar with links for Home, Business solutions, IT services, Products, Support & downloads, My IBM, and a search bar. Below that is a secondary navigation bar with links for ShopzSeries, My orders, My profile, My hardware systems, My licensed software, My installed software, Product catalog, Help, News, Feedback, and Customer service. The main content area has a title 'My orders' and a breadcrumb trail 'ShopzSeries >'. It features tabs for 'Create new order', 'Draft orders', 'In process', and 'Completed'. The 'In process' tab is selected, showing a table with one row. The table columns are 'Select', 'Order reference number - Order name', and 'Status'. The row contains a checkbox, the text 'U00404081 - Products - July 25, 2007, 17:19', and a status row with 'Physical', 'None', 'Internet', and a 'Download' link. Below the table is a button labeled 'Delete selected orders'. On the right side, there's a sidebar titled 'My ShopzSeries' with options for 'Sign out' and 'Edit profile'. At the bottom of the page are links for 'E-mail this page', 'Print this page', and navigation links for About IBM, Privacy, Contact, and Terms of use.

Figure 5-13 Finished order menu

Finished order

While you are working on an order, you can navigate to other features of ShopzSeries and quickly return to your order by using the **My current order** link in the left navigation area. However, save your order periodically by using the **Save** button on the bottom of each step. That way, if you leave ShopzSeries or your session times out, you will be able to find and continue working on your saved draft orders on the “Draft orders” page.

After you submit an order, you can review it by clicking its name in the In process section of the “My orders” page. (If all processing for the order has already completed, you can find it in the Completed section.)

Orders placed through ShopzSeries can also be tracked through delivery directly from ShopzSeries. To track your orders, go to the In process section of the “My orders” page. The current status of each order is displayed. If there is additional tracking information for an order, the status can appear as a link that can be followed to view the additional information.

For service orders, when you review the order by clicking its name, you might also see a Manufacturing order number and a Manufacturing status, which are returned by the service manufacturing system. This information can help in the unlikely event that problems occur with your order. For example, if the service was not able to be sent through the Internet, or if the service you ordered cannot be found, the Manufacturing status field will inform you of the situation.

5.14 Download order information

The screenshot shows the IBM ShopzSeries download order information page. At the top, there's a navigation bar with links for Home, Business solutions, IT services, Products, Support & downloads, and My IBM. A search bar is also present. Below the navigation, a welcome message says "Welcome Redelf Janßen [Not you?] [IBM Sign in]". The main content area has a title "Download U00404081 - Products - July 25, 2007, 17:19". It lists several download links: "Order Packing List" (0.007 MB), "Read Me: A checklist for preparing to install your order" (0.011 MB), "Installation Documentation" (View or Download to your workstation), "ServerPac Material (Host)" (Download directly to host), "ServerPac Material (Workstation)" (Download to your workstation), and "Product Publications" (View or download unlicensed publications for your order). On the left, a sidebar menu includes "ShopzSeries", "My orders", "My current order" (which is selected and highlighted in yellow), "Download", "My profile", "My hardware systems", "My licensed software", "My installed software", "Product catalog", "Help", "News", "Feedback", and "Customer service". On the right, a "My ShopzSeries" sidebar offers "Sign out" and "Edit profile". At the bottom, there are links for "E-mail this page" and "Print this page", along with "About IBM", "Privacy", "Contact", and "Terms of use".

Figure 5-14 Download order information menu

Download order information

When your order is ready to be downloaded, you will receive an email notification with a link that will take you directly to your order for immediate downloading. You can also download your orders from the In process section of the “My orders” page by clicking the **Download** link for the order you want to download.

After you reach the “Download” page for your order, click the links to view or download the various components of your order.

5.15 Download instructions

The screenshot shows a web browser displaying the IBM ShopzSeries website. The top navigation bar includes links for Home, Business solutions, IT services, Products, Support & downloads, My IBM, and a search bar. A welcome message for 'Redelf Janßen' is shown. The main content area is titled 'Download U00404081 - Products - July 25, 2007, 17:19'. It provides instructions for downloading ServerPac material directly to a host, mentioning 'Read Me: Instructions for download directly to host' and two dialog options: 'Dialog Option 1: Install a new ServerPac Dialog that supports Internet delivery' and 'Dialog Option 2: Migrate existing ServerPac Dialog to support Internet delivery, if not done previously'. On the left, a sidebar lists 'ShopzSeries' and 'My orders' under 'My current order', along with links for My profile, My hardware systems, My licensed software, My installed software, Product catalog, Help, News, Feedback, and Customer service. On the right, a 'My ShopzSeries' sidebar offers 'Sign out' and 'Edit profile' options. At the bottom, there are links for E-mail this page, Print this page, and standard footer links for About IBM, Privacy, Contact, and Terms of use.

Figure 5-15 Download instructions menu

Downloading z/OS order directly to a host

For ServerPac orders, you will use the ServerPac dialog to generate the RECEIVE job, which will start the SMP/E GIMGTPKG service routine to download your order from the IBM server to the download file system on your host. You will need to provide the information found on the download page to the dialog for the RECEIVE job to be generated.

For CBPDO orders, you will use the provided customized job control language (JCL) job (RFNJOB). This job starts the SMP/E RECEIVE FROMNETWORK command to download your order using Transmission Control Protocol/Internet Protocol (TCP/IP) FTP from an IBM server to your SMPNTS file system, and then performs the SMP/E RECEIVE for the order. For most service orders, you can also choose to download the order directly to your host by using a sample batch FTP JCL job that is supplied with your order.

CustomPac logon information

Customized offerings, or CustomPacs, are IBM Global Services offerings that require special contracts, as follows:

- ▶ ProductPac. Delivers products which are built according to the SMP/E CSI that you provide.
- ▶ FunctionPac. Delivers a group of z/OS-related products in a new SMP/E zone as a snap-on.
- ▶ SystemPac. A system migration package that helps you plan and install IBM and independent software vendor (ISV) products and subsystems in a single package with up-front customization and subsequent maintenance to maintain your system over time.



z/OS maintenance concepts

Software management is a key discipline that can help you to achieve high availability (HA) and continuous operation in your IBM z/OS environment. It can also assist in lowering the cost of installing, testing, operating, and maintaining your systems.

This chapter provides information about the following topics:

- ▶ Aspects of software management
- ▶ Software management tasks
- ▶ z/OS software management cycle
- ▶ How current your software should be

6.1 Aspects of software management

- Why should you manage software?
- How current should you be with your software?
- An approach for keeping your environment current
- Installation strategy
- Implementation strategy
- Concurrent maintenance

Figure 6-1 Aspects of software management

Aspects of software management

There are different aspects of software management. Although this book is focused on the z/OS platform, many of the software management concepts presented here are generally valid for other platforms, as well.

Why you should manage software

Managing software is important because it concerns which techniques you use to implement changes in your operating system software environment. In a typical information technology (IT) environment, it is likely that there will be a continuous need for changes based on, for example, the following factors:

- ▶ Implementation of new functions
- ▶ Support for new hardware
- ▶ Software maintenance
- ▶ Implementation of new software releases

How current your software should be

This aspect of software management involves making decisions concerning how up-to-date you need your software to be. See 6.4, “How current your should software be” on page 343 for more information about this topic.

An approach for keeping your environment current

There are different approaches and techniques for keeping your z/OS environment current. Although you could keep your system as it was when installed and only update it when errors occur, this is not a recommended approach.

Instead, you can use common tools, such as these to keep your environment current:

- ▶ ServerPac and SystemPac
- ▶ Custom-Built Product Delivery Option (CBPDO)
- ▶ IBM ServicePac®

The tools are described in more detail in Chapter 5, “ShopzSeries” on page 319.

Installation strategy

Before making a change, you must decide which upgrade method to use:

- ▶ System replacement
- ▶ System upgrade

The method you choose also depends on different considerations:

- ▶ Complexity of your environment
- ▶ Current service level
- ▶ Maturity of your system management processes (change management and problem management)
- ▶ Number of products (z/OS and vendor products)

Implementation strategy

Although installation is one important task, implementation is another. For example, for effective implementation, you must decide where to put data sets and how to handle IBM and vendor products, or how to prepare your system for the following cloning processes.

Concurrent maintenance

Under normal circumstances, you perform maintenance non-concurrently. Therefore, you install fixes and test them, then clone the environment and bring the actual version of the operating system or other software products into production.

However, there could be situations in which it is necessary to install fixes concurrently and activate the new modules using an IBM Multiple Virtual Storage (MVS) console command. Be aware that this method should be used only in *critical* situations.

6.2 Software management tasks

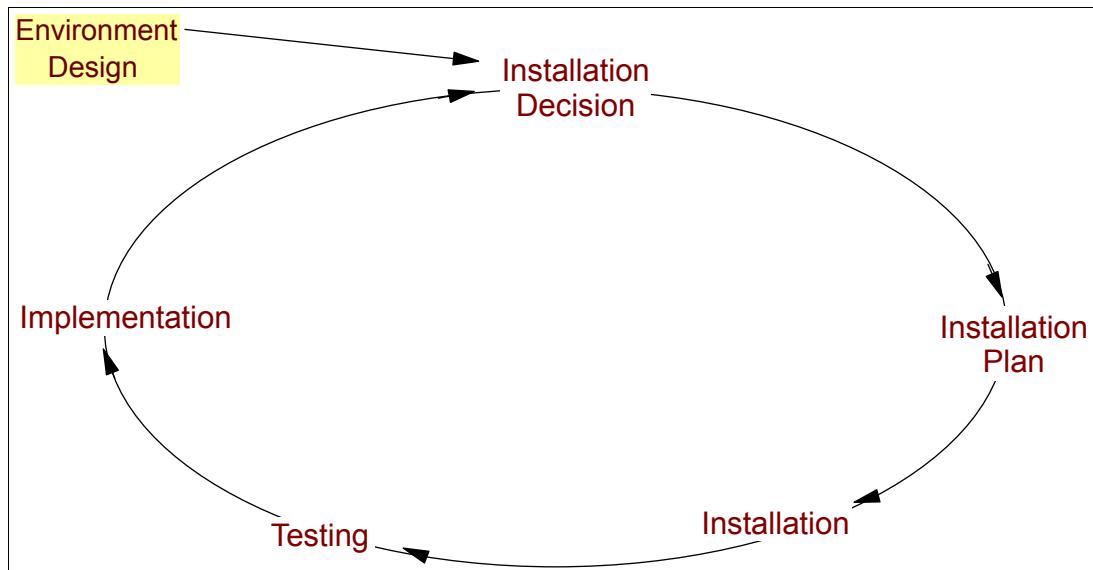


Figure 6-2 Cycle of software management tasks

Software management tasks

A *maintenance philosophy* involves numerous software management tasks, and it is an ongoing, cyclical process, as shown in Figure 6-2. For example, after finishing the implementation of a z/OS release, you might be asked to evaluate new functions that require program temporary fixes (PTFs).

Environment design

The environment design phase is the starting point for the cycle that follows. During this phase, you decide how to design and set up your environment for system maintenance. This phase includes the following tasks:

- ▶ Definition of installation-wide naming standards
- ▶ Logical design of the environment
- ▶ Use of shared system residences (SYSRES)
- ▶ Design of the catalog environment
- ▶ Physical design of the environment (for example, input/output (I/O) configuration)
- ▶ Cloning techniques

Installation decision

An installation decision is primarily a business decision, rather than a technical decision. It can involve, for example, the need for new function implementation, or the withdrawal of a z/OS release from service.

Installation plan

After the installation decision is made, you are responsible for the planning phase. Planning is important, because the resources you need for the subsequent installation, testing, and implementation will cost your organization money (perhaps in several areas):

- ▶ Hardware (CPU resources, DASD space)
- ▶ Software (for example, new licenses)
- ▶ Project support personnel

Planning requires good project management. Therefore, you will need to develop a plan that includes the following items:

- ▶ Key activities
- ▶ Responsibilities
- ▶ Timetable

Installation

In this case, installation is the process of bringing a new z/OS release to your direct access storage device (DASD) environment. As previously mentioned, you will normally perform an installation by using the ServerPac dialog technique, or by using CBPDO. The ServerPac installation technique is described in Chapter 5, “ShopzSeries” on page 319.

Testing

It is important to test new software components before introducing them into your production environment. Testing is necessary for quality assurance (QA). There are different test categories for verification and validation of a new environment. These categories can be:

- ▶ Unit tests
- ▶ Integration tests
- ▶ Function tests
- ▶ System tests
- ▶ Acceptance tests
- ▶ Regression tests
- ▶ Capacity tests
- ▶ Stress tests

Implementation

After completing the basic installation (for example, by using ServerPac), you still must perform many other implementation activities on your z/OS environment before you can propagate it to a test or production system. These activities, using the ServerPac custom dialog, are described in “Installing the CustomPac dialogs” on page 198.

6.3 The z/OS software management cycle

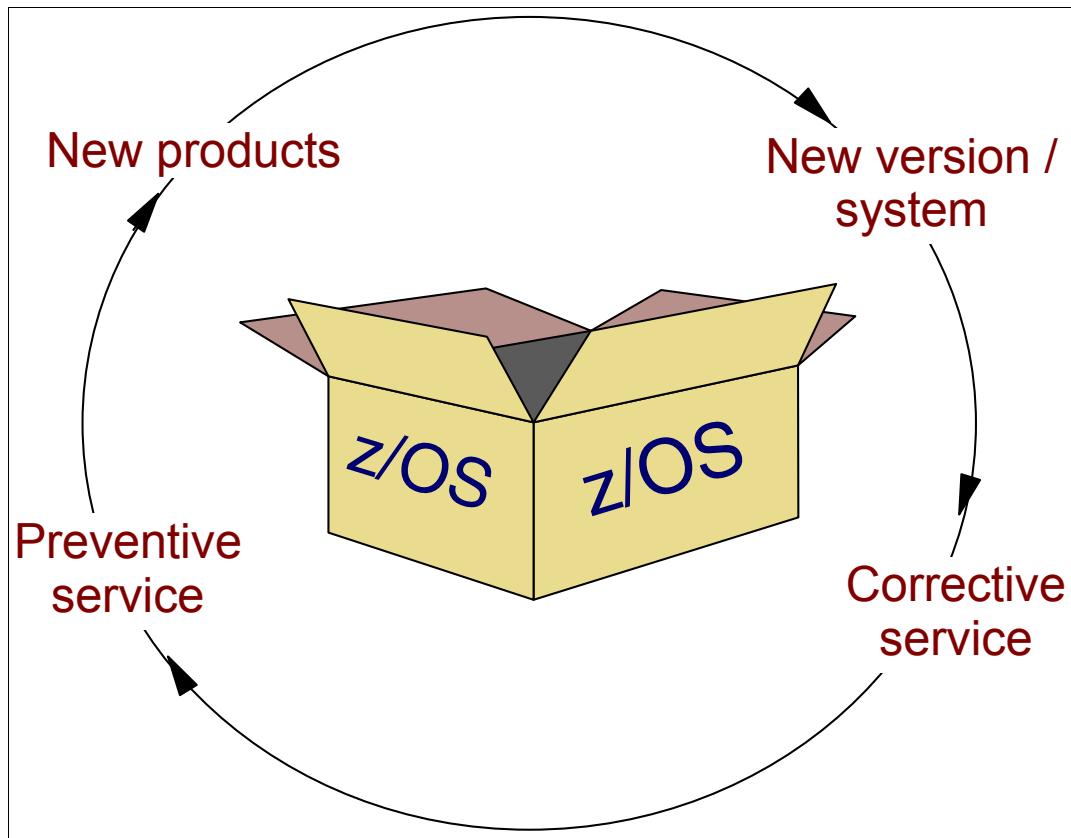


Figure 6-3 The software management cycle

The z/OS software management cycle

Figure 6-3 shows the software management cycle, which illustrates how software management works in a z/OS environment. From a software management perspective, it involves two major factors:

- The last software management activity you performed
- The software management strategy of your enterprise that is the base for your future activities

6.4 How current your software should be

- What is the risk of not changing software?**
- What is the risk of changing software?**
- What is the minimum risk point?**

Figure 6-4 Current level of software

How current your software should be

One challenge that is common to all enterprises, independent of IT infrastructure, is the risk raised by change. You might encounter a philosophy, such as “never change a running system.” However, when it comes to software maintenance, an IT organization must consider the following questions:

- ▶ How current should our z/OS environment be?
- ▶ What are our guidelines and procedures for z/OS maintenance levels?
- ▶ Should we install preventive maintenance, or fix problems as they occur?

The risk of not changing software

There are several issues that might occur if you do not keep your z/OS software stack current:

- ▶ You might not be able to implement new software functions.
- ▶ You cannot easily implement new hardware.
- ▶ New releases of software might not interact with other software due to incompatibilities or synchronization problems.
- ▶ Problems that are already resolved by IBM might be rediscovered in your environment and might result in unnecessary outages.

The risk of changing software

The closer you come to the most current level of your software, the greater is the risk of experiencing a failure or outage.

The minimum risk point

The point of minimum risk for installing new fixes to an already installed operating system (OS) release, or even to a new release, might be somewhere between being too far behind and too current.

Defining a point of minimum risk can be a challenge, so it might be helpful to first determine the requirements for your enterprise, and then compare these requirements to the level of software release that you are able to install and implement.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 346. Note that some of the documents referenced here might be available in softcopy only.

- ▶ *ABCs of z/OS System Programming Volume 2*, SG24-6982
- ▶ *ABCs of z/OS System Programming Volume 3*, SG24-6983
- ▶ *ABCs of z/OS System Programming: Volume 4*, SG24-6984
- ▶ *ABCs of z/OS System Programming: Volume 5*, SG24-6985
- ▶ *ABCs of z/OS System Programming Volume 6*, SG24-6986
- ▶ *ABCs of z/OS System Programming Volume 7*, SG24-6987
- ▶ *ABCs of z/OS System Programming Volume 8*, SG24-6988
- ▶ *ABCs of z/OS System Programming: Volume 9*, SG24-6989
- ▶ *ABCs of z/OS System Programming Volume 10*, SG24-6990
- ▶ *ABCs of z/OS System Programming Volume 11*, SG24-6327
- ▶ *ABCs of z/OS System Programming Volume 12*, SG24-7621
- ▶ *ABCs of z/OS System Programming Volume 13*, SG24-7717

Other publications

These publications are also relevant as further information sources:

- ▶ *z/OS Planning for Installation*, GA32-0890
- ▶ *z/OS MVS Initialization and Tuning Reference*, SA23-1380
- ▶ *z/OS MVS JCL Reference*, SA23-1385
- ▶ *z/OS MVS JCL User's Guide*, SA23-1386
- ▶ *z/OS Security Server RACF System Programmer's Guide*, SA23-2287
- ▶ *z/OS TSO/E Command Reference*, SA32-0975
- ▶ *z/OS TSO/E Customization*, SA32-0976
- ▶ *z/OS TSO/E Primer*, SA32-0984
- ▶ *z/OS TSO/E REXX User's Guide*, SA32-0982
- ▶ *z/OS TSO/E User's Guide*, SA32-0971
- ▶ *z/OS Using REXX and z/OS UNIX System Services*, SA23-2283
- ▶ *IBM ServerPac Using the Installation Dialog*, SA23-2278

- ▶ *OSA-Express Customer's Guide and Reference*, SA22-7935
- ▶ *Interactive System Productivity Facility Getting Started*, SC34-4440
- ▶ *z/OS ISPF Planning and Customizing*, GC19-3623
- ▶ *z/OS ISPF Dialog Developer's Guide*, SC19-3619
- ▶ *z/OS ISPF SCLM Project Manager's and Developer's Guide*, SC19-3625
- ▶ *z/OS Distributed File Services zFS Administration*, SC23-6887
- ▶ *z/OS DFSMS Managing Catalogs*, SC23-6853
- ▶ *z/OS ISPF Edit and Edit Macros*, SC19-3621
- ▶ *z/OS ISPF User's Guide Volume I*, SC19-3627
- ▶ *z/OS ISPF User's Guide Volume II*, SC19-3628

Online resources

These websites are also relevant as further information sources:

- ▶ The ShopzSeries web address:
<https://www14.software.ibm.com/webapp/ShopzSeries/ShopzSeries.jsp>
- ▶ The PMA website:
<http://www-03.ibm.com/systems/z/os/zos/features/smpe/pma/index.html>

How to get IBM Redbooks

You can search for, view, or download IBM Redbooks, IBM Redpapers, Technotes, draft publications, and Additional materials, and order hardcopy Redbooks, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads:

ibm.com/support

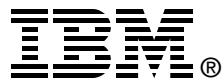
IBM Global Services:

ibm.com/services



ABCs of IBM z/OS System Programming Volume 1

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



ABCs of IBM z/OS System Programming

Volume 1



Get an introduction to z/OS and storage concepts

Learn about TSO/E, ISPF, JCL, and SDSF

Understand z/OS delivery and installation

The ABCs of IBM z/OS System Programming is a 13-volume collection that provides an introduction to the z/OS operating system and the hardware architecture. Whether you are a beginner or an experienced system programmer, the ABCs collection provides the information that you need to start your research into z/OS and related subjects.

Whether you want to become more familiar with z/OS in your current environment, or you are evaluating platforms to consolidate your online business applications, the ABCs collection will serve as a powerful technical tool.

Volume 1 provides an updated understanding of the software and IBM zSeries architecture, and explains how it is used together with the z/OS operating system. This includes the main components of z/OS needed to customize and install the z/OS operating system. This edition has been significantly updated and revised.

The other volumes contain the following content:

- ▶ Volume 2: z/OS implementation and daily maintenance, defining subsystems, IBM Job Entry Subsystem 2 (JES2) and JES3, link pack area (LPA), LNLST, authorized libraries, System Modification Program/Extended (SMP/E), IBM Language Environment
- ▶ Volume 3: Introduction to Data Facility Storage Management Subsystem (DFSMS), data set basics, storage management hardware and software, catalogs, and DFSMS Transactional Virtual Storage Access Method (VSAM), or DFMSMStvs
- ▶ Volume 4: z/OS Communications Server, Transmission Control Protocol/Internet Protocol (TCP/IP), and IBM Virtual Telecommunications Access Method (IBM VTAM)
- ▶ Volume 5: Base and IBM Parallel Sysplex, z/OS System Logger, Resource Recovery Services (RRS), Global Resource Serialization (GRS), z/OS system operations, z/OS Automatic Restart Manager (ARM), IBM Geographically Dispersed Parallel Sysplex (IBM GDPS)
- ▶ Volume 6: Introduction to security, IBM Resource Access Control Facility (IBM RACF), Digital certificates and public key infrastructure (PKI), Kerberos, cryptography and IBM eServer z990 integrated cryptography, zSeries firewall technologies, Lightweight Directory Access Protocol (LDAP), and Enterprise Identity Mapping (EIM)
- ▶ Volume 7: Printing in a z/OS environment, Infoprint Server, and Infoprint Central
- ▶ Volume 8: An introduction to z/OS problem diagnosis
- ▶ Volume 9: z/OS UNIX System Services
- ▶ Volume 10: Introduction to IBM z/Architecture, zSeries processor design, zSeries connectivity, LPAR concepts, HCD, and IBM DS8000
- ▶ Volume 11: Capacity planning, IBM Performance Management, z/OS Workload Manager (WLM), IBM Resource Management Facility (IBM RMF), and IBM System Management Facility (SMF)
- ▶ Volume 12: WLM
- ▶ Volume 13: JES2 and JES3 System Display and Search Facility (SDSF)

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, clients, and Business 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