IBM z/VSE

Planning

Version 3 Release 1

SC33-8221-00

IBM z/VSE

Planning

Version 3 Release 1

SC33-8221-00

Note

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

First Edition (March 2005)

This edition applies to Version 3 Release 1 of IBM z/Virtual Storage Extended (z/VSE), Program Number 5609-ZVS, and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

```
IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany
```

You may also send your comments by FAX or via the Internet:

```
Internet: s390id@de.ibm.com
FAX (Germany): 07031-16-3456
FAX (other countries): (+49)+7031-16-3456
```

Contents

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. 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 of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Deutschland GmbH
Department 0790
Pascalstr. 100
70569 Stuttgart
Germany
```

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

Programming Interface Information

This manual also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of z/VSE. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming Interface Information

End of Programming Interface Information

Trademarks and Service Marks

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

ACF/VTAM	HiperSockets	Redbooks
Advanced Function Printing	IBM	RISC System/6000
Advanced Peer-to-Peer Networking	Language Environment	RMF
AFP	Magstar	RS/6000
AIX	Multiprise	SAA
APPN	MVS	SQL/DS
CICS	MVS/ESA	Systems Application Architecture
CICS/VSE	NetView	System/370
C/370	Operating System/2	System/390
DFSMS/VM	OS/2	S/390
DFSORT	OS/390	S/390 Parallel Enterprise Server
ECKD	Pennant	TotalStorage
Enterprise Storage Server	Personal System/2	Virtual Machine/Enterprise Systems Architecture
Enterprise Systems Architecture/390	PrintManager	VisualAge
Enterprise Systems/9000	Print Services Facility	VisualGen
ESA/390	Processor Resource/Systems Manager	VM/ESA
ES/9000	PR/SM	VSE/ESA
ESCON	PSF	VTAM
eServer	PS/2	zSeries
FICON	QMF	z/OS
FlashCopy	RACF	z/VM
GDDM	RAMAC	z/VSE

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

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

About This Book

z/VSE is the successor to IBM's VSE/ESA product. Many products and functions supported on z/VSE may continue to use VSE/ESA in their names.

z/VSE can execute in 31-bit mode only. It does not implement z/Architecture, and specifically does not implement 64-bit mode capabilities.

z/VSE is designed to exploit select features of IBM eServer zSeries hardware.

This manual describes the IBM z/VSE Version 3, Release 1, from a planning point of view.

Who Should Use This Book

The manual is intended for those who plan, administer, and maintain a z/VSE system. A knowledge of z/VSE and the z/VSE base programs is required.

How to Use This Book

The information in this manual is primarily meant to help you plan for installation and daily system use. Much of this involves *tailoring* your system to meet your particular data processing needs.

Where to Find More Information

This edition of the manual documents new and changed functions of z/VSE 3.1 that are related to this manual. The *z/VSE Release Guide* provides an overview of the changes and the functions that are new with z/VSE 3.1.

z/VSE Home Page

z/VSE has a home page on the World Wide Web, which offers up-to-date information about VSE-related products and services, new z/VSE functions, and other items of interest for VSE users.

You can find the z/VSE home page at:

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


Summary of Changes

- **z/VSE** is the successor to **VSE/ESA**. However, the names of many features and programs related to z/VSE remain unchanged (such as IBM Language Environment for VSE/ESA, IBM COBOL for VSE/ESA, or TCP/IP for VSE/ESA).
- Chapter 2, "Hardware Support," on page 3 has been updated and the latest z/VSE 3.1 hardware support added.
- Support for SCSI (Small Computer System Interface) disk devices has been added. See "Support for Fibre-Channel-Attached SCSI Disks" on page 10 and Chapter 9, "Using SCSI Disks With Your z/VSE System," on page 133.
- An introduction to Large Volume Support (LVS) has been added, including the VSE/VSAM support of large DASDs. See "Large Volume Support (LVS)" on page 12.
- Support for the 3592 J1A Tape Drive has been added, including support for WORM cartridges. See "Support for the IBM TotalStorage 3592 Model J1A Tape Drive" on page 22.
- An introduction to Virtual Tape support has been added. See "Virtual Tape Support" on page 33.
- "Predefined System Environments" on page 53 has been updated to include the large predefined Environment C.
- Section "z/VSE Base Programs" on page 39 provides details about z/VSE 3.1 base program changes.
- Section "z/VSE Optional Programs" on page 43 provides details about z/VSE 3.1 optional program changes.
- Chapter 5, "Installing z/VSE," on page 75 has been updated to reflect z/VSE 3.1 installation changes.
- Chapter 8, "Running z/VSE Under VM," on page 119 has been updated to reflect the latest changes when running z/VSE 3.1 under VM.
- Support for the OSA Express adapter has been added. See "OSA Express Support" on page 156.
- Support for HiperSockets has been added. See "HiperSockets Support" on page 160.
- Support for hardware encryption using a PCICA or similar card has been added. See "Using Hardware Cryptographic Support" on page 242.
- A description has been added of how to display cryptographic status information under z/VSE. See "Displaying Hardware Crypto Status Information Under z/VSE" on page 244.
- A description has been added of how to extend the security checking of the BSM using your own logic. See "BSM Post-Processing Installation Exit BSTXX01" on page 239.
- Appendix B, "z/VSE Disk Layouts (DOSRES, SYSWK1)," on page 291 has been updated to reflect the disk layouts for the DOSRES and SYSWK1 layouts for IBM 3380, 3390, FBA, and SCSI disks.
- Appendix F, "z/VSE Skeletons and REXX/VSE Procedures," on page 309 has been updated to reflect the latest skeleton changes.
- Because the IBM 9345 disk is no longer supported, the related information has been removed from this manual.

- Because the *IXFP/SnapShot for VSE/ESA* program is no longer supported, the related information has been removed from this manual.
- The information describing 31-bit addressing has been removed from this manual.

What is New With z/VSE 3.1?

For an overview of the items that have been introduced with z/VSE 3.1, refer to the *z/VSE Release Guide*, SC33-8220.

Chapter 1. How to Run z/VSE

How you run a z/VSE system depends on many aspects. One of the major aspects is the hardware you use. Another one is whether z/VSE is the only operating system that you want to run on a processor. This chapter is intended as an overview of the different ways of how you can run z/VSE.

You can run z/VSE:

- **Natively** as a single operating system installed on any zSeries (31-bit mode) or ESA/390 processor, as listed in "Processor Support" on page 4.
- As a **guest system under VM**.
- In logical partitioned (**LPAR**) mode under PR/SM.

Each method of running z/VSE has its advantages, some of which are listed below.

Advantages when Running z/VSE Natively

- Usually, it offers the best performance.
- It has the lowest hardware costs.
- No additional software is needed.
- Resources are shareable within z/VSE.

Advantages when Running z/VSE Natively in LPAR mode

- More than one z/VSE system on one processor possible.
- Partitions (LPARs) are independent from each other.
- Shared I/O units possible.

Depending on the processor type, **PR/SM** (Processor Resource/Systems Manager) provides a maximum of 30 logical partitions. Each logical partition can be used to run a single operating system such as z/VSE.

For more information about PR/SM and LPAR mode, refer to

Processor Resource/Systems Manager, Planning Guide, GA22-7236.

Advantages when Running z/VSE under VM

- More than one z/VSE system on one processor possible.
- Shared I/O units possible.
- Flexible communication between guest systems.
- Flexible use of terminals.
- Flexible configuration setup.
- Advanced data base integrity.
- Exploitation of expanded storage.
- Broad range of CMS applications.

For further details about running under VM, refer to these manuals:

- *z/VM Version 5 Release 1, Running Guest Operating Systems*, SC24-6115.

- *z/VM Version 4, Running Guest Operating Systems*, SC24-5997.

The z/VSE Turbo Dispatcher

The Turbo Dispatcher is the only dispatcher supported by z/VSE, and is always active.

The Turbo Dispatcher support is available for uniprocessors *and* multiprocessors. The Turbo Dispatcher can utilize multiprocessors by distributing the workload across several processors (CPUs), enabling them to work in parallel and thus increase the overall throughput of a z/VSE system.

Partition balancing allows the operator to specify a relative share of CPU time for any static partition and any dynamic class of a balanced group. This allows a more refined control of workload distribution.

Note: The z/VSE Turbo Dispatcher can **exploit up to 3-way** processors and in some cases up to 4-way processors (with the known workloads of today) and **tolerates up to 10-way** processors.

The Turbo Dispatcher provides:

- The same partition balancing support for a uniprocessor as described above for a multiprocessor environment.
- Planning information, by showing to what extent a workload running on the uniprocessor would be able to exploit the CPUs of a multiprocessor.

Chapter 2. Hardware Support

This chapter describes the hardware and related programs that are supported by z/VSE 3.1. These are the main sections of the chapter:

For a complete overview of the devices that are supported, refer to Appendix G, "Devices Supported," on page 319.

Processor Support

Processors Supported by z/VSE 3.1

These are the **uniprocessors** and **multiprocessors** of the IBM processor series that can be used with z/VSE 3.1:

- These IBM eServer[1] processors in **31-bit mode** (or equivalent):
  - **zSeries 890**
  - **zSeries 990**
  - **zSeries 800**
  - **zSeries 900**
- **IBM S/390 Parallel Enterprise Server - Generation 6 and Generation 5** processors (or equivalent).
- **IBM S/390 Multiprise 3000** processors (or equivalent).

The support includes all uniprocessor models of the processor series listed and ″n-way″ support for the multiprocessor models through the z/VSE Turbo Dispatcher.

Note that z/VSE can run as a guest system under z/VM Version 4 or higher, on all processors that are supported by z/VM Version 4 or higher.

Logical Partitions and LCSS Support

z/VSE 3.1 supports:

- **Up to 30 LPARs** (Logical Partitions) in which z/VSE can run.

  This support is available exclusively on IBM eServer zSeries 890 and 990 processors. A larger number of LPARs provides additional configuration flexibility. Support for up to 30 LPARs is available with VSE/ESA 2.6 and later.
- **Up to four Logical Channel Subsystems (LCSS)**.

  With multiple LCSSs, z890 and z990 processors can attach up to 1024 channels. However, any single instance of z/VSE remains limited to 256 channels. Additional LCSSs are available exclusively on IBM eServer zSeries 890 and 990 processors. The z890 supports up to 2 LCSS, and the z990 supports up to 4 LCSSs. Support for up to 4 LCSSs is available with VSE/ESA 2.6 and later.

Adapter Interruptions Support

Adapter interruptions are exclusive to z890 and z990 processors. High performance adapter interrupt handling, that was first introduced with HiperSockets, is also available for OSA-Express2 and OSA-Express in QDIO mode (CHPID type OSD), as well as FICON Express2 and FICON Express (CHPID type FCP). This zSeries advance provides a more efficient technique for handling I/O interrupts, and is designed to reduce path lengths and overhead in both the operating system and the adapter.

Adapter interruptions for CHPID=FCP are supported in z/VSE 3.1. Adapter interruptions for CHPID=OSD are supported with VSE/ESA 2.7 and later.

Spanned Channels

A z/VSE image in a single LPAR can use FICON Express2, FICON Express, OSA-Express2, and OSA-Express adapters defined across multiple LCSSs.

1. The IBM eServer brand consists of the established IBM e-business logo with the descriptive term ″server″ following it.

Spanned Channels also allow z/VSE to use HiperSockets that link LPARs or virtual servers using different Logical Channel Subsystems (LCSSs).

Processors Not Supported by z/VSE 3.1

These are the most recent processors that are **not** supported by z/VSE 3.1:
- IBM S/390 Multiprise 2000
- IBM S/390 Parallel Enterprise Server - Generation 3
- IBM S/390 Parallel Enterprise Server - Generation 4
- IBM S/390 Parallel Enterprise Server - all Rx1, Rx2, and Rx3 Models
- IBM ES/9000 Family - all 9021, 9121, and 9221 Models
- IBM S/390 Integrated Server
- IBM PC Server 500 System/390
- IBM RS/6000 with S/390 Server-on-Board feature.

Support for Hardware Crypto

z/VSE 3.1 provides *Hardware Encryption Assist support* (abbreviated to *Hardware Crypto support*), which requires either a:
- Crypto Express2 card or equivalent. The Crypto Express2 card is available on the z890 and z990 (or later) processors.
- PCI Cryptographic Accelerator (PCICA) card or equivalent. The PCICA card is available on *all* IBM eServer zSeries processors.

Both Crypto Express2 and PCICA cards provide encryption-assist support, and can help to increase the throughput in a TCP/IP network using SSL (Secure Sockets Layer). If your system runs under z/VM:
- The Crypto Express2 support requires *z/VM 5.1 or higher*.
- The PCICA support requires *z/VM 4.2 or higher*.

z/VSE 3.1 tests for the Crypto Express2 and PCICA feature at IPL time. The support for TCP/IP for VSE/ESA SSL transparently uses the Crypto Express2 or PCICA card, if available.

For further details about Hardware Crypto support, see "Using Hardware Cryptographic Support" on page 242.

Support for CP-Assisted Cryptographic Function (CPACF)

Cryptographic Function (CPACF) is available exclusively on each and every central processor (CP) within a z890 or z990 complex. It provides an "on processor" function that is designed to accelerate symmetric cryptographic operations. The function is delivered through a new set of zSeries architecture instructions.

CPACF is a no-charge hardware feature. z/VSE 3.1 tests for the CPACF feature at IPL-time.

The support for TCP/IP for VSE/ESA SSL transparently uses the CPACF feature, if available.

For further details about Hardware Crypto support, see "Using Hardware Cryptographic Support" on page 242.

Communication Adapter Support

Support for FICON Express2 and FICON Express

FICON channels allow higher I/O bandwidth compared to ESCON channels.

FICON Express2 is a technology refresh from FICON Express, that offers 4 ports per feature and the potential for increased performance. FICON Express2 is exclusive to z890 and z990 processors.

z/VSE 3.1 supports CHPID type FC, FCV, and FCP. VSE/ESA 2.6 and later support CHPID type FC and FCV only.

Support for the OSA-Express Integrated Console Controller

The *Open Systems Adapter-Express Integrated Console Controller* (OSA-ICC) function supports the emulation of:
- TN3270E (RFC 2355)
- Non-SNA DFT 3270

3270 emulation for console-session connections is integrated into the zSeries 890/990 processors via a port on the OSA-Express 1000BASE-T Ethernet feature. This can help remove the need for external console controllers (such as the 2074, 3174). It also reduces costs and complexity. A new Channel Path Identifier (CHPID) of type **OSC** is introduced and is supported by z/VSE.

For practical information about how to use the OSA-ICC, refer to the redbook *OSA-Express Integrated Console Controller Implementation Guide*, SG24-6364.

Support for the OSA-Express2 10 Gigabit Ethernet (10 GbE)

OSA-Express2 represents a new generation of zSeries LAN adapters. It is available as an optional feature exclusively on z890 and z990 processors. OSA-Express2 can help satisfy the network bandwidth requirements of your applications demand.

Like the OSA-Express2 and OSA-Express features, the Open Systems Adapter-Express2 10 Gigabit Ethernet (abbreviated to *OSA-Express2 10 GbE* feature) only supports QDIO mode (CHPID type OSD) with full duplex operations, using one port to carry TCP/IP traffic.

Note that the OSA-Express2 10 GbE feature does not support auto-negotiation to any other speed. The OSA-Express2 10 GbE feature supports 64B/66B coding, whereas the OSA-Express GbE feature supports 8B/10B coding.

For z/VSE 3.1 and TCP/IP for VSE/ESA, the feature is compatible with OSA-Express2 GbE and OSA-Express. You can therefore use this feature without any change to your z/VSE or TCP/IP configurations.

Support for OSA-Express2 10GbE is available with VSE/ESA 2.6 and later.

Support for the OSA-Express2 Gigabit Ethernet (GbE)

OSA-Express2 is available as an optional feature exclusively on z890 and z990 processors. The Open Systems Adapter-Express2 Gigabit Ethernet (abbreviated to the *OSA-Express2 GbE*) feature supports QDIO mode (CHPID type OSD), using two ports to transmit TCP/IP traffic, and with full duplex operations only. For z/VSE 3.1 and TCP/IP for VSE/ESA, the feature is compatible with OSA-Express. You can therefore use this feature without any change to your z/VSE or TCP/IP configurations.

z/VSE 3.1 supports the capability of up to 640 TCP/IP stacks/connections per dedicated CHPID, or 640 total stacks across multiple LPARs using a shared or spanned CHPID on a single OSA-Express2 feature on the z890/z990 processors.

Support for OSA-Express2 GbE is available with VSE/ESA 2.6 and later.

Support for OSA-Express

OSA-Express is used for fast data transmission via TCP/IP. This support is provided via the OSA-Express adapter which is available on IBM eServer zSeries and IBM Parallel Enterprise Servers G5 and G6 processors.

z/VSE also supports *Adapter Interruption* for OSA-Express on zSeries processors. The OSA-Express adapter is based on the QDIO architecture. Originally the OSA-Express adapter generated a PCI interruption after data was placed into the QDIO input queue.

For further details about OSA-Express Support, see “OSA Express Support” on page 156.

Support for OSA-2

z/VSE 3.1 supports the S/390 *Open Systems Adapter (OSA-2)* feature, which is available on various IBM processors. It is an integrated hardware feature that lets its zSeries or S/390 host platform provide industry-standard connectivity to clients on the following types of area networks:

- Local area networks (LANs), or
- ATM-based networks (to be used in LAN-emulation mode only).

For details about OSA-2 support, see “OSA-2 Support” on page 163.

Support for HiperSockets

Using *HiperSockets*, TCP/IP traffic travels between LPARs and virtual servers at near-memory speed, rather than network speed.

The HiperSockets support is available on IBM eServer zSeries processors. If you run z/VSE under z/VM, you require z/VM 4.2 or later.

In addition to using native HiperSockets on IBM eServer zSeries processors, you can also use **virtual HiperSockets** support if you run z/VSE under z/VM on an IBM Multiprise 3000, or an IBM Parallel Enterprise Server G5 or G6. z/VM 4.2 or later is required in order to support virtual HiperSockets that can be used by guest operating systems such as z/VSE.

Support for HiperSockets, including HiperSockets *Spanned Channels* (that is, HiperSockets that connect LPARs using different Logical Channel Subsystems) is available with VSE/ESA 2.7 and later.

For further details about HiperSockets support, see "HiperSockets Support" on page 160.

Disk Device Support

This section describes the *disk devices* and related programs that are supported by z/VSE 3.1:

Notes:

1. The IBM 3380 Model A disk device (or an equivalent-sized VM minidisk) is not supported for the initial installation and FSU of z/VSE. However, this model can still be used as data devices.
2. z/VSE can be IPL'd with more than 1024 devices defined to the z/VSE system in, for example, the IOCDS or in the z/VM guest definition. However, z/VSE itself can only use the first 1024 detected devices.

Support for IBM TotalStorage ESS

The IBM TotalStorage Enterprise Storage Servers (ESS) are designed to offer scalability, performance, accessibility, security and reliability to support 24X7 operations.

In particular, the ESS Model 750 is designed to meet midrange price, capacity, and performance needs.

ESS offers FlashCopy, FlashCopy Version 2, Peer-to-Peer Remote Copy (PPRC), and PPRC Version 2 copy services.

The IBM TotalStorage ESS offers 3380 and 3390 (including 3390 'Large Volumes' with up to 32K cylinders) ECKD format. The ESS is designed to support a broad range of operating environments including IBM eServer zSeries, pSeries, iSeries, and xSeries as well as several non-IBM systems.

zSeries attachment is via ESCON, FICON, or FCP channels. S/390 attachment is via ESCON or FICON channels.

z/VSE 3.1 supports FCP attachment and SCSI disk format as well. Please Note: the ESS 750 does not offer FCP attached SCSI disks.

VSE/ESA 2.6 and later support ESCON and FICON attachment and ECKD disk formats.

Support for Fibre-Channel-Attached SCSI Disks

The objective of z/VSE Small Computer System Interface (SCSI) disk support is to offer clients more storage choices as well as help lower Total Cost of Ownership (TCO). Therefore, from z/VSE 3.1 onwards selected SCSI disks can be attached, in addition to Extended Count-Key-Data (ECKD) and Fixed Block Architecture (FBA) disks. Only SCSI disks that are contained within the IBM TotalStorage Enterprise Storage Server Model F20, 800, and 800 Turbo are qualified for this support.

The z/VSE SCSI-FCP support is designed to be transparent to z/VSE components and subsystems, vendor interfaces, and user-written programs. This support includes both operating system components and files containing user data.

The IBM eServer zSeries Fibre Channel Protocol (FCP) channels are designed to allow customers to attach industry-standard SCSI disk devices. A typical topology might consist of:
- One or more FCP channels.
- A switch (such as the IBM 2109).
- A controller, containing one or more ports and one or more SCSI devices.

FCP channels consist of FICON Express or FICON Express2 cards that have been personalized with the FCP feature.

For further details about z/VSE SCSI disk support, see Chapter 9, “Using SCSI Disks With Your z/VSE System,” on page 133.

Support for the IBM TotalStorage DS8000 Series

The IBM TotalStorage DS8000 series storage server is designed to provide robust, flexible, scalable, and cost-effective enterprise disk storage for mission-critical workloads.

The DS8100 is a dual 2-way system based on IBM’s POWER5 technology. It offers 16 to 128 GB of processor memory (disk cache). It offers 1.1 to 38.4 TB of RAID 5 or RAID 10 disk capacity in a single frame, and up to 115.2 TB with an expansion frame.

The DS8300 is a dual 4-way system based on IBM’s POWER5 technology. It offers 32 to 256 GB of processor memory (disk cache). It offers 1.1 to 38.4 TB of RAID 5 or RAID 10 disk capacity in a single frame, and up to 192 TB with two expansion frames.

The IBM TotalStorage DS8000 series:
- Squeezes more capacity into each enclosure than the previous generation of IBM storage units. The result is a disk system that occupies up to 20% less floor space than the ESS 800.
- Is designed to accommodate growth. The architecture is designed to scale up to 1 Petabyte of storage (where a Petabyte is equal to 1000 Terabytes, or a million Gigabytes).
- Offers 3380 and 3390 ECKD format.
- Is designed to support a broad range of operating environments IBM eServer zSeries, pSeries, iSeries, and xSeries, as well as several non-IBM systems.

zSeries attachments can be made via:
- ESCON channels.
- FICON channels.

- FCP channels.

S/390 attachments can be made via:
- ESCON channels.
- FICON channels.

From VSE/ESA 2.7 onwards, VSE supports ESCON and FICON attachments, and ECKD disk formats.

Support for the IBM TotalStorage DS6000 Series

The IBM TotalStorage DS6000 series is designed to deliver enterprise-class storage capabilities in a low-cost, space-efficient, and modular package.

The high performance and advanced functions found in enterprise disk storage devices are made available in a 19-inch rack-mountable package that is just 5.25 inches high. As your needs grow, you can add capacity with modular expansion units of the same size.

The DS6800 has a dual-active controller with an integrated PowerPC RAID controller. It offers 4 GB cache. Capacity ranges from 584 GB to 4.8 TB of RAID 5 or RAID 10 storage in a single unit. With optional DS6000 expansion enclosures, the maximum capacity is 67 TB.

The IBM TotalStorage DS6000 series:
- Offers 3380 and 3390 ECKD format.
- Is designed to support a broad range of operating environments: IBM eServer zSeries, pSeries, iSeries, and xSeries, as well as several non-IBM systems.

zSeries attachments can be made via:
- FICON channels.
- FCP channels.

S/390 attachments are made via FICON channels.

From VSE/ESA 2.7 onwards, VSE supports FICON attachments and ECKD disk formats.

FlashCopy 2 Support for IBM TotalStorage ESS

FlashCopy Version 2 (abbreviated to “FlashCopy 2”) includes the features of FlashCopy (including NOCOPY) *plus* extensions designed to improve capacity management and disk utilization. It offers functions which improve business efficiency and performance.

z/VSE 3.1 supports the following selected FlashCopy Version 2 functions:
- *Dataset Copy*, which offers a new level of granularity. It allows more efficient use of disk capacity. It can also reduce background completion times because FlashCopy no longer needs to be performed at the volume level when only a dataset copy is required. With Dataset Copy, the source and target volumes may be different sizes. It also allows the copied data to reside at a different location in the source and target volumes.

  Note: Because a VSAM “dataset” is a complex entity consisting of a catalog and one or more data extents, VSAM files are **not** candidates for the Dataset Copy function.

- *Elimination of the LSS Constraint*, which can help simplify administration and capacity planning for FlashCopy. Source and target volumes can now span logical subsystems within a storage server.
- *Multiple Relationship FlashCopy*, which offers new flexibility. It allows up to 12 target volumes to be created from one source volume in a single FlashCopy operation. Multiple target volumes can be used for testing, backup, and other applications.
- *Performance improvements* in FlashCopy 2, which are designed to reduce the time required to complete a FlashCopy establish command.

z/VSE 3.1 does **not** support these FlashCopy functions:
- Incremental FlashCopy.
- Consistency Group commands over a Remote Mirror link.
- Persistent FlashCopy Relationship.
- Inband Commands over a Remote Mirror link.

For details of how to use the FlashCopy 2 support, refer to the manual *z/VSE Administration*, SC33-8224.

FlashCopy Support for IBM TotalStorage ESS

IBM TotalStorage FlashCopy technology is a *point-in-time* copy capability that can be used to help reduce planned application outages caused by backups and other data copy activities. This support is part of z/VSE as shipped.

FlashCopy is designed to enable data to be copied in the background, then make both source and copied data available to users almost immediately. z/VSE 3.1 provides support for the NOCOPY option, which leads to greater efficiency in copying from disk to tape. The NOCOPY function can be used to copy most, or all, of the data directly from the source to tape *without* the need to first copy all of the physical data to an intermediate backup copy.

For a description of how to use the FlashCopy support, refer to the manual *z/VSE Administration*.

FlashCopy / Remote Mirroring Support for IBM TotalStorage DS8000/DS6000 Series

IBM TotalStorage DS8000 series and DS6000 series make no distinction between FlashCopy and FlashCopy Version 2 (“FlashCopy 2”). The features of FlashCopy 2 are included in a single FlashCopy Feature (PTC - Point in Time Copy) for DS8000 series and DS6000 series storage servers.

The IBM TotalStorage DS family also introduces new terminology. The ESS PPRC V2 capabilities are included in a new Remote Mirror and Copy Feature (RMC) for DS8000 series and DS6000 series storage servers.

VSE/ESA 2.7 and later support RMC Metro Mirror, Global Copy, and Global Mirror modes through the ICKDSF component.

Large Volume Support (LVS)

The large volume support (**LVS**) of the ESS is the ability to define custom-sized logical volumes (referred to as “custom volumes”) ranging in size from 1 to 32760 cylinders, for use by an operating system such as z/VSE. These large volumes help to:

- Reduce address constraints (especially since z/VSE only supports 3-digit volume addresses).
- Improve disk utilization by allowing you to consolidate multiple smaller disk volumes into a single large volume.

Custom volumes are reported to the z/VSE host system as:
- 3390-3 devices when their size is between 1 and 3339 cylinders.
- 3390-9 devices when their size is between 3340 and 32760 cylinders.

A large volume of the IBM TotalStorage Enterprise Storage Server (ESS) has the following characteristics:

```
z/VSE device type: ECKD
Number of cylinders: Up to 32760
Byte capacity per cylinder: 849960
Number of tracks per cylinder: 15
Byte capacity per track: 56664
```

The **VSE/VSAM** support is limited to the first **10017** cylinders which corresponds to the capacity of an IBM 3390 Model 9. This is a general restriction and applies to any volume larger than 10017 cylinders.

For details of the VSE/VSAM support of large DASDs, refer to the manual *VSE/VSAM User's Guide and Application Programming*, SC33-8246.

Other programs and functions of z/VSE support the full capacity of a Large Volume:
- Non-VSE/VSAM access methods SAM and DAM (including IDCAMS Backup and Restore).
- ICKDSF (Device Support Facilities) program.
- VSE/Fast Copy program.
- VSE/POWER, Librarian, and all other z/VSE-integrated applications.
- IPL commands DLA, DLF, and DPD.

Support for the IBM RAMAC Array Family

z/VSE supports the following devices of the RAMAC Array family.

IBM RAMAC Virtual Array

The IBM 9393 RAMAC Virtual Array (RVA) provides an effective storage capacity from 160 GB up to 420 GB and is attached through the IBM 9393 Disk Array Controller. It emulates both IBM 3380 and IBM 3390 disk devices.

Note: From z/VSE 3.1 onwards, the IXFP/SnapShot for VSE/ESA feature is no longer available.

IBM RAMAC Scalable Array

The IBM 9396 IBM RAMAC Scalable Array provides storage capacity up to 2.2 TB (terabytes) and is attached through the IBM 9396 IBM RAMAC Scalable Array Control Unit. The RAMAC Scalable Array is compatible with the IBM 3990 Models 3 and 6. The RAMAC Scalable Array uses 3380 and 3390 data formats and you define the Scalable Array as IBM 3390 or IBM 3380 disk device to z/VSE.

IBM RAMAC Electronic Array

The IBM 9397 IBM RAMAC Electronic Array is an electronic, external storage subsystem with a capacity of up to 4 GB. It can be upgraded to an IBM 9396 RAMAC Scalable Array. The IBM RAMAC Electronic Array is attached through the IBM RAMAC Electronic Array Controller. The IBM RAMAC Electronic Array supports up to 256 logical volumes in 3390 or 3380 data format and you define the Electronic Array as IBM 3390 or IBM 3380 disk device to z/VSE.

IBM RAMAC 3 Array Storage

The IBM RAMAC 3 provides storage capacity in the range of 45.4 GB to 726 GB and is attached either through the IBM 9390 Storage Control or through the IBM 3990 Model 6. The IBM RAMAC 3 Array Storage is to be defined to z/VSE as IBM 3390 or IBM 3380 disk device.

IBM RAMAC Array DASD/Subsystem

z/VSE supports the **IBM RAMAC Array Family** generation of storage devices. The IBM RAMAC Array family exploits the RAID-5 technology.

The IBM RAMAC Array Family is comprised of two rack-and-drawer single unit storage offerings:
- IBM RAMAC Array DASD (IBM 9391/IBM 9392)
- IBM RAMAC Array Subsystem (IBM 9394/IBM 9395)
- IBM RAMAC 2 Array DASD/Subsystem

Host connectivity options include parallel and/or ESCON attachment. ESCON configurations include 128 logical channel path addressing capability.

IBM RAMAC Array DASD: The IBM RAMAC Array DASD operates transparently to the operating system as an IBM 3390 Model 3 or IBM 3380 Model K disk device and attaches to the IBM 3990 Storage Control Unit Models 3 and 6. The disk devices take full advantage of the extended and advanced functions of the IBM 3990, such as DASD fast write and record cache.

IBM RAMAC Array Subsystem: The IBM RAMAC Array Subsystem can be used as either IBM 3390 Model 3 or IBM 3380 Model K disk device. It offers multilevel cache with aggregate cache sizes up to four gigabytes (GB). In addition, it offers:
- 9 to 90.8 GB of data storage in one rack

- Controller cache sizes ranging from 64MB to 2GB
- Drawer cache sizes ranging from 32MB to 2GB
- IBM 3380 or IBM 3390 disk device support
- Parallel and ESCON channel support

IBM RAMAC 2 Array DASD/Subsystem: Compared to the IBM RAMAC Array DASD/Subsystem, the IBM RAMAC 2 Array DASD/Subsystem can store from 11 to 180GB data storage in one rack. The IBM RAMAC 2 Array DASD, attached to an IBM 3990 Model 6, can be used in 3380 or 3390 track format and supports either 3380 Model K or 3390 Model 3 emulation.

Support for the IBM 3390 Model 9 Disk Device

For a detailed description of the various models of the IBM 3390 disk device family and their usage in a VSE environment, refer to:
Using IBM 3390 Direct Access Storage in a VSE Environment, GC26-4576.

For details of the VSE/VSAM support of the IBM 3390 Model 9 disk device, refer to the description of large DASD support contained in the manual *VSE/VSAM User's Guide and Application Programming*, SC33-8246.

The IBM 3390 Model 9 can be attached via an IBM 3990 control unit such as an IBM 3990 Model 6.

The IBM 3390 Model 9 disk device provides high capacity data storage, at lower performance, with a considerable savings in cost. It offers a three times higher capacity than the IBM 3390 Model 3. This is achieved by having the device rotate with 1/3 of the speed of the other models of the IBM 3390 which means, however, an increase in access time. The device should therefore not be used as storage media for performance critical data. Also, the system residence volumes (DOSRES and SYSWK1) should not be placed on an IBM 3390 Model 9.

The following list contains some types of data that are candidates for the IBM 3390 Model 9:

- Sequentially accessed data.
- Information currently stored on microfiche.
- Image data and indexes.
- Data whose retrieval demands fall below that of data stored on high performance disk devices, but whose demands are still above the level provided by tape or optical devices.
- Backup copies of data bases.
- Infrequently accessed information.
- Test copies of data bases.

The IBM 3390 Model 9 disk device operates in **3390 mode** only and has the following characteristics:

z/VSE device type:	**ECKD**
Track capacity in bytes:	**56664**
Number of cylinders:	**10017 (see Note 1)**
Number of tracks per cylinder:	**15**
Number of tracks per volume:	**150255 (see Note 2)**

Notes:
1. Since the number of cylinders exceeds 9999, 5 digits are required for representing the number of cylinders.
2. Since the number of tracks per volume exceeds 99999, 6 digits are required for representing the number of tracks.

VTOC Location

The Device Support Facilities program (ICKDSF) requires that the VTOC resides on the first 65535 tracks of each volume.

IPL Support

The IPL commands DLA, DLF, and DPD accept 5-digit decimal numbers for cylinder specification.

JCL Support

In the JCL // EXTENT statement you can specify a relative track number and the number of tracks. The number of digits available for both values has been increased from 5 to 6 to support the high track capacity of Model 9 for the access methods **SAM** and **DAM**.

IBM does not recommend to specify split cylinders for the Model 9. If the relative track number or the number of tracks is higher than 65535, split cylinders are not supported.

VSE/VSAM Support

VSE/VSAM provides full support for the IBM 3390 Model 9. For details, refer to the manual *VSE/VSAM User's Guide and Application Programming*, SC33-8246.

Non-VSE/VSAM Support

The non-VSE/VSAM access methods SAM and DAM can use the full capacity of the IBM 3390 Model 9.

VSE/Fast Copy Support

The VSE/Fast Copy program supports more than 65535 tracks per volume (6 digits needed instead of 5) and more than 9999 cylinders per volume (5 digits needed instead of 4).

Interactive Interface Support

The corresponding dialogs of the Interactive Interface support the IBM 3390 Model 9.

Support for the IBM 3990 Model 6 Control Unit

The IBM 3990 Model 6 is an enhancement of the IBM 3990 Model 3 control unit for attaching disk devices such as the IBM 3390 Models 3 and 9. Its main enhancements, compared to Model 3, are:
- Regular data format: this function offers improved performance for track updates.
- Quiesce/Resume function, CUIR (control unit initiated reconfiguration): this allows automatic quiescing and later resuming operation of devices and channel paths for maintenance application.
- 128 logical paths (with ESCON attachment).

In addition, the IBM 3990 Model 6 offers extended information about subsystem status which can be retrieved with the commands CACHE and ONLINE. Refer to the manual *z/VSE System Control Statements* under “CACHE” and “ONLINE” for details.

For details about the IBM 3990 Model 6 cache support refer also to “Cache Functions for IBM 3990 Models 3/6 Storage Control” on page 18.

Support for the IBM 3390 Model 3 Disk Device

The **IBM 3390 Model 3** offers a greater storage capacity than the IBM 3390 Model 1 and Model 2 and has the following characteristics:

- It has a track capacity of 56 664 bytes per track when operating in **3390 mode**. If you run it in 3390 mode, the device is ADDed as type **ECKD**.
- It can also operate in **3380 track compatibility mode** (TCM) and then has a track capacity of 47 476 bytes per track. If you run it in 3380 track compatibility mode, the device is ADDed as type 3380.
- It can be attached to a parallel as well as to an ESCON channel. If attached to an ESCON channel, you cannot run the IBM 3390 Model 3 in 3380 track compatibility mode.

After initial installation, you can use the *Hardware Configuration* dialog to ADD an IBM 3390 Model 3 to your system configuration. Select **3390-3** for an IBM 3390 Model 3 used in 3390 mode, and **3390-3TC** for an IBM 3390 Model 3 used in 3380 track compatibility mode.

The disk layout for the IBM 3390 Model 3 is the same as for the IBM 3390 Model 1 and Model 2, except for the amount of free space and the VTOC location at the end of the disk device, as shown in the following figure.

Refer also to Appendix B, “z/VSE Disk Layouts (DOSRES, SYSWK1),” on page 291 for the complete DOSRES and SYSWK1 layout as shipped for an IBM 3390 Model 1.

For predefined Environment B the values are as follows (if either journaling is not used, or the default journal files for CICS/VSE and the second CICS TS are not used):

Table 1. Unused Space on IBM 3390 System Disks

Model	Volume	Unused Space		End VTOC	
		Start Track	Number of Tracks	Start Track	Number of Tracks
1	DOSRES SYSWK1	11625 —	5055 —	16680 16680	15 15
2	DOSRES SYSWK1	11625 16680	21750 16695	33375 33375	15 15
3	DOSRES SYSWK1	11625 16680	38445 33390	50070 50070	15 15
9	DOSRES SYSWK1	11625 16680	138495 133440	150120 150120	15 15

Notes:
1. The IBM TotalStorage Enterprise Storage Server (ESS) controller allows disks to be defined that are larger than the IBM 3390 Model 9. The unused space will therefore be larger.
2. The values for unused space shown in the above table are based upon environment B.
3. For environment C, the free space on DOSRES is reduced by the page data set (1784 MB).

Support for the IBM 3990 Model 2 and 3 Storage Control Units

Through IBM 3990 Model 2 and Model 3 control units **with the ESCON feature**, you can attach IBM 3390 disk devices and also IBM 3380 (Models A, D, E, J, and K) disk devices to an **ESCON** channel of an ESA/390 processor. Thus, these disk devices can utilize the processor's ESCON channel support.

- Attaching IBM 3390 disk devices:

  IBM 3390 disk devices operating in 3390 mode are always added as **ECKD** devices during initial installation for nonsynchronous operation.

  An IBM 3390 disk device (Models 1, 2, and 3) can also operate in 3380 track compatibility mode. In this mode, the device type code is **3380** Attached to an ESCON channel, however, 3380 track compatibility mode is **not** supported.
- Attaching IBM 3380 disk devices:

  The standard device type code for the IBM 3380 disk device is **3380**. Attached to an ESCON channel through an IBM 3990 Model 2 or Model 3, however, IBM 3380 disk devices are sensed and added during initial installation as **ECKD** devices for nonsynchronous operation to exploit the full performance capacity.

Cache Functions for IBM 3990 Models 3/6 Storage Control

z/VSE supports the IBM 3990-3/6 cache functions as described below. z/VSE guest systems under z/VM can use the cache support as described below for native systems.

Overview

The hardware functions of the IBM 3990-3/6 and their default settings are shown in Table 2.

Table 2. IBM 3990-3/6 Hardware Functions

Function	Default Setting
Read Only (Subsystem) Caching	ON
Basic Write Caching	ON
DASD Fast Write	OFF
Cache Fast Write	Enabled
Non-Volatile Storage (NVS)	OFF
Normal (LRU) Caching[1]	ON
Sequential Access Caching	Enabled
Bypass Cache	Enabled
Inhibit Cache Loading	Enabled

[1] LRU = Least Recently Used algorithm

Support for IBM 3990 Models 3 and 6 Storage Control Features

DASD Fast Write: This function causes WRITE I/O operations to appear as though they are completed instantaneously. Device-end status is presented with channel-end status as soon as the data is received by the control unit. The data **is saved** temporarily in nonvolatile storage until the asynchronous writing on DASD is completed. This eliminates the possibility of data loss, even if a power failure occurs.

Note: Nonvolatile storage (NVS) is random-access electronic storage with a battery backup power source. NVS maintains subsystem integrity for data not yet written to DASD, even in the event of a storage subsystem power failure.

Cache Fast Write: The function causes I/O operations to appear as though they are completed instantaneously. Device-end status is presented with channel-end status as soon as the data is transferred into the cache of a control unit. When cache space expires, the data is written onto a DASD. The data is **not saved** in nonvolatile storage, however. Thus this function should not be used for data that has to be saved.

Support for Dual Copy: This function allows you to maintain locally identical copies of a disk device on two different volumes in the same logical subsystem. The IBM 3990 internally synchronizes the volumes by writing all modified data on both volumes. Thus dual copy provides:
- Significantly enhanced data availability during a volume outage by automatically directing all accesses to the operational volume.
- Reduced host program overhead by eliminating the need for the host to issue duplicate write operations to maintain two identical volumes.
- Resynchronization of a duplex pair after repair.

Enhanced Fast Dual Copy: This function is the result of adding the benefits of DASD Fast Write to Dual Copy. It occurs automatically when both functions are present and enabled. Device-end status is presented with channel-end status as soon as the data is received by the control unit. The data **is saved** in nonvolatile storage, eliminating the possibility of data loss due to power failure, and both copies are written to DASD at a later time.

Activating the IBM 3990 Models 3 and 6 Cache Support

To activate caching, z/VSE provides the attention routine command **CACHE**. You must use the command to activate caching on the subsystem (IBM 3990-3/6) and on the unit (IBM 3380/3390) level. z/VSE provides no special programming support, but the command is accepted and passed to the IBM 3990-3/6 and processed as outlined below. Enter at the system console, for example:

- **`CACHE UNIT=cuu,ON`**

  This activates caching for an IBM 3380/3390 unit (Read Only Caching and Basic Write Caching).
- **`CACHE SUBSYS=cuu,ON`**

  This activates caching for an IBM 3990-3/6 subsystem at address cuu (Read Only Caching and Basic Write Caching). For cuu you can use any address of a device belonging to the subsystem.
- **`CACHE UNIT=cuu,FAST,ON`**

  This switches DASD Fast Write for an IBM 3380/3390 unit on. You can switch DASD Fast Write on at the unit level only.
- **`CACHE SUBSYS=cuu,FAST,ON`**

This switches Cache Fast Write for an IBM 3990-3/6 subsystem at address cuu on for user-provided software using the physical IOCS (PIOCS).

You can switch Cache Fast Write on at the subsystem level only.

- **CACHE SUBSYS=cuu,NVS,ON**

  This specifies that non-volatile storage has to be made available for an IBM 3990-3/6 subsystem which is a requirement for DASD Fast Write.

For a complete description of the CACHE command and its operands, refer to the manual *z/VSE System Control Statements* under “CACHE”.

Recommendations

You should be aware that the z/VSE basic caching support always works on a device level (not file level). Normal (LRU) caching manages automatically the cache contents in such a way that the overall benefit comes near to the optimum. To find out, you can use the parameters STATUS and REPORT of the CACHE command. When introducing caching, consider the following recommendations:

- Introduce caching for all ″important″ DASD volumes, especially those with:
  - Files which have a high read/write ratio and are accessed heavily.
  - Files which have a read hit ratio of 65% or better.
  - Highly used VSE/VSAM index files if data and index are on separate volumes.
  - Highly used VSE/VSAM catalogs.
  - Highly used libraries.

  Candidates for caching might also be the DASD volumes with the following z/VSE files:

  Lock File

  Label Area
- Start with the most important volumes and add additional ones in a controlled manner.
- If promising, consider a redistribution of your files.

Further Reading

If you plan to introduce IBM 3990-3/6 Storage Control at your z/VSE installation, you should also consult manuals like the following:

IBM 3990 Storage Control Introduction, GA32-0098

IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide, GA32-0100

IBM 3380 Direct Access Storage Introduction, GC26-4491

IBM 3380 Direct Access Storage in a VSE Environment, GC26-4494

Using IBM 3390 Direct Access Storage in a VSE Environment, GC26-4576

Display Station Support

2074 Console Support

The **2074** Console Support Controller eliminates the requirement for a non-SNA 3174 controller.

OSA-Express Integrated Console Controller (OSA-ICC) Support

The *Open Systems Adapter-Express Integrated Console Controller* (OSA-ICC) function supports the emulation of:
- TN3270E (RFC 2355)
- Non-SNA DFT 3270

3270 emulation for console-session connections is integrated into the zSeries 890/990 via a port on the OSA-Express 1000BASE-T Ethernet feature. This can help eliminate the need for external console controllers (such as the 2074, 3174). It also reduces costs and complexity. A Channel Path Identifier (CHPID) of type **OSC** is supported by z/VSE.

For practical information about how to use the OSA-ICC, refer to the redbook *OSA-Express Integrated Console Controller Implementation Guide*, SG24-6364.

Tape Device Support

This section describes these *tape devices* that are supported by z/VSE 3.1:

- "Support for the IBM TotalStorage 3592 Model J1A Tape Drive"
- "Support for the IBM TotalStorage 3590 Tape Subsystem" on page 24
- "Support for the IBM TotalStorage 3494 Tape Library" on page 27
- "Support for the IBM TotalStorage 3490E Tape Device" on page 29
- "Support for the Automatic Cartridge Loader" on page 31
- "Summary of VSE Support for IBM 3490 and IBM 3490E Tape Devices" on page 32

Related Section:

- "Virtual Tape Support" on page 33

Support for the IBM TotalStorage 3592 Model J1A Tape Drive

The IBM TotalStorage 3592 Enterprise Tape Drive J1A (3592) is designed to address the needs of applications for high capacity, fast access to data, and/or long term data retention.

The 3592 offers both 300 GB (up to 900 GB with 3:1 compression) and 60 GB cartridges. Either cartridge type is available in rewritable or WORM (Write Once Read Many) formats. WORM cartridges are designed to provide non-alterable, non-rewritable tape media for long term records retention. When the 3592 detects a WORM cartridge, the tape drive prevents overwrite or changes to existing customer data. New data can be appended to existing data until the capacity of the cartridge is reached.

A 3592 J1A tape drive can be used if it attached to an *IBM TotalStorage Enterprise Automated Tape Library 3494*. The 3592 J1A tape drive requires cartridges which are written in 512-track format.

Support for the 3592 WORM and 60 GB Media

From VSE/ESA 2.7.3 onwards, z/VSE supports the IBM TotalStorage Enterprise Tape Drive 3592 WORM (write-once-read-many) and 60 GB media.

Notes:

1. z/VSE supports the IBM 3592 WORM in standalone mode as well as through the VGS (VSE Guest Server) under z/VM.
2. No support is provided for use of the IBM 3592 WORM through the LCDD (Library Control Device Driver).

Description of WORM Data Cartridges:
The IBM TotalStorage Enterprise Tape Drive 3592 Model J1A **WORM** (Write-Once-Read-Many) data cartridges are designed to provide non-alterable, non-rewritable tape media for long-term records retention. When the IBM 3592 J1A Tape Drive detects a WORM cartridge, the tape drive prevents overwrite or changes to existing customer data. New data can be appended to existing data until the capacity of the cartridge is reached.

A short-length 60 GB re-writable media is also available and can be used with z/VSE.

Using 3592 WORM Tapes With z/VSE:
3592 devices are supported by VSE as follows:
- External device code: TPA
- Add statement: ADD cuu,TPA,5
- VSE Pub device type code: X'56'

To recognize a 3592 WORM tape and to check whether it is to READ/WRITE, z/VSE supports the following GETFLD service:

```
GETFLD FIELD = DEVPROP
(returns)
DVPTFLG2 DS  X     TAPE SUPPORTED DENSITIES
*        EQU X'80' RESERVED
DVPTWORM EQU X'40' TAPE IS A WORM CARTRIDGE
DVTPWOWR EQU X'20' TAPE IS A WORM READ/EXTENDABLE
```

For LIBR Backup/Restore, the following changes were implemented for the support of the 3592 WORM device:
- If the backup is performed to an IBM 3592 tape device with a WORM tape mounted, no end-of-backup (“EOB”) record is written onto the backup tape.
- Only two tape marks are written behind the Backup File, with this format:

```
              ...Backup File/TM/TM/
                                 |
   Tape positioned here -------|
```

3592 WORM devices to be used with VSE in a 3494 tape library:
The Library Media code is set to:

C’CST5’
For 3592 300GB.

C’CST6’
For 3592 WORM 300GB.

C’CST7’
For 3592 60GB.

C’CST8’
For 3592 WORM 60GB.

Changes to Macros for 3592 Devices

LBSERV macro calls for SQUERY, AQUERY and IQUERY requests will return in SERVL (“service list”) IJJLBSER, a media type IJJLTMED of either:
- CST5
- CST6
- CST7
- CST8

Note: When z/VSE is running under VM and the IBM 3592 J1x devices are not attached to VSE (and will be attached by VSE Guest Server, VGS, when a volume is mounted), the 3592 J1x cannot be correctly recognized. In this case, the 3592 J1x drives must be ADDed in the VSE startup procedure, as described in “IPL Statements Required for the 3592 Device.”

IPL Statements Required for the 3592 Device

To add the 3592 J1A drives to the startup procedure, you must specify an additional mode:

```
ADD cuu,TPA,5
```

You must also ADD a 3592 tape drive to z/VSE as `TPA,5` if the following are both true:

- A 3592 tape drive is to be used from a z/VSE system running under VM.
- The 3592 is not attached during the startup of the z/VSE operating system.

JCL ASSGN Statement Required for the 3592 Device

With the introduction of 3592 tape drive support, a device class **TPAT512** can be used with the ASSGN statement, where TPAT512 identifies a 3592 tape drive. The ASSGN statement has this format:

```
ASSGN SYSnnn device type code
```

In a mixed 3590/3592 tape drive environment (that is, there are different 3590/3592 models with different track characteristics, such as 128-track, 256-track, 384-track, and 512-track), you must specify ASSGN SYSnnn,TPAT512 to ensure that a 3592 tape drive with 512-track characteristic is selected.

You should be aware that a generic `ASSGN SYSnnn,TPA` statement will use this search order:

1. Eligible TPAT128 devices.
2. TPAT256 devices.
3. TPAT384 devices.
4. TPAT512 devices.

Using a 3592 Tape Drive in a 3494 Tape Library

The EXTRACT macro call **ID=ATLCUU** which requests a free 3592 tape drive in an automated tape library, will require this input:

- 3490E for the 3490E drive.
- TPA for the 3590 drive.
- EFMT1 for the 3592 drive.

You can use a 3592 tape drive in a 3494 tape library providing you install the APAR VM63325 on z/VM. The tape drive support is made through the VSE Guest Server (VGS). The VGS is supported by DFSMS/RMS.

Note: 3592 support is **not** available using the LCDD (Library Control Device Driver).

Support for the IBM TotalStorage 3590 Tape Subsystem

The IBM 3590 is an extension to IBM's family of 3480, 3490, and 3490E tape devices. It conforms to the Tape Products Architecture (TPA). z/VSE supports the IBM 3590 as device of type TPA.

The following models are available: A00, A14, B11, and B1A, where Model A00 is a control unit. The IBM 3590 is available either standalone or included in the IBM TotalStorage 3494 Tape Library.

The IBM 3590 has a cartridge capacity of up to 30 GB with compacted data, and 10 GB if the data is uncompacted.

Programming Aspects

The z/VSE device support for the IBM 3590 is similar to the IBM 3480/3490 device support, with the following exceptions that could affect user programs:

- TPA devices do not support READ BACKWARD CCWs.
- A CCW, READ PREVIOUS, is provided as a replacement for READ BACKWARD.

If you issue a READ BACKWARD CCW, z/VSE error recovery transforms it into a READ PREVIOUS CCW, but at the expense of performance degradation.

The standard HDR2 label has been changed to accommodate six digits, instead of the last 5 digits of the device serial number, which were moved into the HDR2 label record for other devices.

Applications waiting for device end or channel end on an IBM 3590 are posted at device end, to ensure proper error recovery. Note that rewind or positioning commands can cause excessive wait times, and thus degrade performance.

IBM 3590 Tape Modes

The TPA architecture allows many mode settings (those from the IBM 3480/3490 and in addition the WRITE formats 0-7):

```
'00' to '0F' (buffered)
'20' to '2F' (unbuffered)
```

For the IBM 3590, only the following WRITE formats are valid:

```
0  (use device default)
1  (3590 cartridge format)
7  (use media default)
```

When using WRITE format 0, the following modes apply for an IBM 3590:

Table 3. IBM 3590 Tape Modes (WRITE Format 0)

Tape Action	Uncompacted	Compacted
Buffered Write	00	08 (default)
Tape Write Immediate	20	28

Mode 08 is the default mode, which is valid if the following two conditions exist:

- The IBM 3590 drives have been added without a mode specification.
- The ASSGN does not specify a mode value.

For performance reasons, mode 08 should be used.

IPL and JCL Support

The IBM 3590 will be properly recognized during IPL, if it is operational at the time of interrogation. If the device is not ready, the IBM 3590 must be added as follows:

```
ADD  cuu,TPA[,mode]
```

JCL allows you to assign IBM 3590 tape units in two ways:

- By specifying the cuu directly:
  ```
  ASSGN SYSxxx,cuu
  ```
- By specifying the device type TPA with which the device (cuu) was added during IPL:
  ```
  ASSGN SYSxxx,TPA
  ```

All additional parameters that can be specified for existing tape devices are valid for the IBM 3590 as well.

JCL also allows LIBSERV MOUNT commands for IBM 3590 tape units in tape libraries (such as the IBM 3494). If several tape-device types are contained in a library, JCL chooses a tape unit that is able to process the specified tape volumes.

Data compaction is handled as specified during IPL in the ADD statement, or as set with the JCL ASSGN statement. Volume, header and trailer labels are not compacted.

Support for Tape Initialization

Tapes (cartridges) for the IBM 3590 may have to be initialized. The Initialize Tape Utility (INTTP) checks whether the device requires the mounted tape to be formatted, and if so, does the formatting before any volume labels are written to tape.

For IBM 3590 tapes only, the INTTP accepts a utility control statement requesting only the formatting of a tape without writing volume labels.

The normal tape initialization processing, that is, writing volume labels on magnetic tape for standard label checking, is extended for IBM 3590 tapes. If the tape requires formatted write, and formatting is supported, INTTP:

- Formats and then rewinds the tape.
- Writes the specified Volume 1 label in uncompacted mode.
- Writes a dummy header label and a tape mark after this label.
- Rewinds or rewinds-unloads the tape, as specified in the parameter of the utility control statement.

If you assign an IBM 3590 to INTTP, you must not assign tape units of any other device type. If you do, the utility job is canceled. For different tape device types, submit separate jobs.

To initialize IBM 3590 tapes without writing standard volume labels, use the utility control statement without any control parameters. For example:

```
// JOB
// ASSGN SYS000,1A1   (1)
// ASSGN SYS001,UA    (2)
// EXEC INTTP         (3)
// INTT               (4)
/*
/&
```

Explanation of the control statements:

1. Assign SYS000 to the IBM 3590 on 1A1.
2. Unassign next sequential unit.
3. Start tape utility INTTP.
4. This statement is accepted without further parameters for the IBM 3590 only and is interpreted as:
   a. Perform formatting write.
   b. Rewind the tape after formatting completed.
   c. Do not write labels to tape.

To write volume labels on tape for standard label checking, use the same parameters as for other tape devices.

Interactive Interface Support

The Interactive Interface supports the IBM 3590 as TPA device. The following dialogs have been extended:

- Configure Hardware (device and mode selection).
- Backup/Restore.

If you select a mode which is not supported by the physically attached model, the dialog does not raise an error. An error is raised during the first I/O operation to the tape unit.

Support for the IBM TotalStorage 3494 Tape Library

The IBM TotalStorage 3494 Tape Library is an intermediate automated tape library dataserver. It automates the retrieval, storage, and control of IBM 3490E, IBM 3590, and IBM 3592 cartridge tapes.

The 3494 Tape Library is designed to provide reliable, granular, scalable tape automation. It addresses data retention and business continuity requirements. It supports 3490E, 3590 and 3592 tape drives with high capacity or short length rewritable media, as well as 3592 WORM media. A single 3494 Tape Library can be shared across multiple platforms, including IBM eServer zSeries, pSeries, iSeries, and xSeries, as well as various non-IBM platforms.

With the support of the IBM 3590 and IBM 3592, the IBM 3494 may contain devices of more than one device type. Whenever a *cuu* is specified on an LBSERV macro request, the device type used is the one which was specified during IPL.

z/VSE programming support for the IBM 3494 is available for:
- Native z/VSE systems.
- z/VSE systems running as guest systems under z/VM.

z/VSE supports:
- The IBM 3494 natively
- LCDD
- VGS (VSE Guest Server)

These three possibilities of the z/VSE 3494 support are now described.

Native 3494 Support

The native 3494 support is available from z/VSE 3.1 onwards. It allows the 3494 Tape Library to be supported through the S/390 channel command interface, mainly using the Perform Subsystem Function (PSF) and Perform Library Function (PLF) commands.

z/VSE tape library support includes support for the IBM TotalStorage 3592 tape drives. Existing LCDD (native z/VSE) and VGS (where z/VSE is configured as a VM guest) 3494 Tape Library support is based on the XPCC/APPC communication protocol. The native 3494 support that is available from z/VSE 3.1 onwards is easier to use and more extendable than the 3494 Tape Library support.

For compatibility reasons, the LCDD and VGS interfaces are unchanged and can still be used along with the native 3494 support available from z/VSE 3.1 onwards. However, the LCDD support is not enhanced. For example, LCDD support does not include 3592 tape drives.

For further details of native 3494 support, refer to the chapter describing how to implement 3494 Tape Library Support in the manual *z/VSE Administration*, SC33-8224.

3494 Virtual Tape Server (VTS) Support

The IBM TotalStorage 3494 Virtual Tape Server (referred to as “3494 VTS”) support is available from z/VSE 3.1 onwards. It is designed to help reduce operating costs and improve the overall performance of the tape library processing.

To help improve performance, the 3494 VTS subsystem writes virtual 3490E volumes to a tape volume cache on a high-speed RAID disk buffer. It then stacks the virtual volumes on high-capacity 3592 or 3590 cartridges.

The 3494 VTS can help reduce batch processing time, reduce TCO, and management cost. A single 3494 VTS can be shared across multiple platforms, including IBM eServer zSeries, pSeries, and xSeries, as well as various non-IBM platforms.

LCDD Support

The z/VSE LCDD support uses the LAN attachment to the IBM 3494 for library control of the IBM 3494 by one or more z/VSE host(s). A z/VSE IBM 3494 library control device driver (LCDD) runs as a VTAM application in a partition of its own. The library control commands are transmitted on the LAN as packets to the library manager, using LU6.2 advanced program-to-program communication (APPC). The channel attachment to the IBM 3494 control unit provides the tape data path.

The z/VSE API is the main programming interface for controlling the library through the LCDD. In addition, the LCDD supports a full set of interactive functions for operations use.

Both native and guest z/VSE systems with attached IBM 3494 devices may exploit the LAN library control connection and use the LCDD support application.

Note: LCDD does **not** support 3592 devices.

VGS (VSE Guest Server) Support

The capabilities of the VM component DFSMS/VM RMS extend the IBM 3494 support to z/VSE guests running under z/VM. The function set supported through the z/VSE API for guest systems includes scratch pool exploitation, inventory management operations, and additional information retrieval.

The DFSMS/VM RMS component provides library control through the 3490E host channel attachment of the IBM 3494 which is also the tape data path.

DFSMS/VM provides a z/VSE Guest Server (VGS) virtual machine to facilitate the z/VSE-guest usage of CMS interfaces. The z/VSE API for the IBM 3494 is the main programming interface for controlling the library through VGS. In a guest environment, operator commands for controlling the IBM 3494 interactively are available through DFSMS/VM RMS.

Support Functions For the IBM 3494

The z/VSE support for the IBM 3494 consists of the following major functions:

- z/VSE recognizes 3490E tape units of an IBM 3494 as tape devices of type **3490E**. The same modes as for the IBM 3490E apply:

  00 Buffered write

  08 Buffered write with compaction (IDRC)

  20 Tape write immediate mode

  28 Tape write immediate with compaction (IDRC)

  The 3590 and 3592 devices are recognized as tape devices of type **TPA**.
- Support of Parameter ATL in IPL SYS Command

  The ATL (automatic tape library) parameter is required to activate the tape library support for the IBM 3494 during startup.
- LIBSERV Command/Statement

The LIBSERV command/statement is provided for specifying all information for mounting, maintaining, and detaching tape volumes on an IBM 3494. Refer to the manual *z/VSE System Control Statements* under "LIBSERV" for a description of the LIBSERV command/statement.

- LBSERV Macro

  The LBSERV macro provides the application programming interface (API) which allows to write applications that can access volumes of the IBM 3494 tape library. Refer to the manual *z/VSE System Macros Reference* under "LBSERV Macro" for the syntax and a description of the LBSERV macro and consult the manual *z/VSE System Macros User's Guide* under "Example for a LBSERV MOUNT Request" for programming examples.

Support for the IBM TotalStorage 3490E Tape Device

z/VSE supports IBM 3490E tape (cartridge) devices. The IBM 3490E with its control unit Models A10/20 (including B20/B40), D41/42, and C10/C11/C22 is a double-density follow-on device to the IBM 3480/3490 tape subsystem. The models of the IBM 3490E have a standard cartridge capacity of up to 1200MB/400MB (with or without compaction through the IDRC feature). The cartridge capacity with the IDRC feature installed is dependent on the compaction ratio. All models of the IBM 3490E can read IBM 3480/3490 cartridges but not vice versa.

Note: IDRC means Improved Data Recording Capability.

The IBM 3494, introduced under "Support for the IBM TotalStorage 3494 Tape Library" on page 27, is treated like an IBM 3490E and includes models C1A and C2A of the IBM 3490E.

Considerations for Adding an IBM 3490E/3490/3480 Tape Device

At IPL time, the IBM 3490E, IBM 3490, and IBM 3480 are treated differently which is shown in the following table. The table also applies when defining an IBM 3490E, IBM 3490, or IBM 3480 tape device during initial installation.

Table 4. Defining/Adding IBM 3480/3490 Tape Devices

Tape Device	To be defined or added as 3490E	To be defined or added as 3490	To be defined or added as 3480
IBM 3480			x
IBM 3480 with IDRC feature		x	
IBM 3490			x
IBM 3490 with IDRC feature		x	
IBM 3490E	x		

If a tape unit is operational at IPL time, it is sensed and is recognized correctly whether it is installed with or without the IDRC feature. If a tape unit is **not** operational at IPL time, you have to use one of the following commands to set the correct IDRC capability in the device list:

```
ADD cuu,3480,mode
ADD cuu,3490,mode
ADD cuu,3490E,mode
```

The following mode settings can be used for all IBM 3490E, IBM 3490, and IBM 3480 tape devices with the IDRC feature installed:

00 Buffered write
08 Buffered write with compaction (IDRC)

20 Tape write immediate mode
28 Tape write immediate with compaction (IDRC)

If no mode is specified in the **ADD** command, the following default values are used:
00 For an IBM 3480 or an IBM 3490
08 For an IBM 3490E

Note: The IBM 3480 and IBM 3490 Model D31/D32 have the IDRC feature as an **optional** feature only. The other models of the IBM 3490 and IBM 3490E include the IDRC feature as a **standard** feature.

3480-XF and 3480-2 XF Recording Format Compatibility

When using these tape devices, you must observe the following:

An IBM 3490E uses the **3480-2 XF** recording format. This format is different from the format of the IBM 3480 and IBM 3490 tape subsystems. They use the **3480-XF** recording format.

Note: The 3480-2 XF recording format is an enhanced capability format with 36-track bidirectional recording (for IBM 3490E tape devices) compared to 18-track recording format 3480-XF (for IBM 3480/3490 tape devices).

- Cartridges created on an IBM 3490E can **only** be read on IBM 3490E tape devices.
- IBM 3490E tape devices can read cartridges created on an IBM 3480 or IBM 3490 tape device.
- All IBM 3490 can be attached to an ESCON channel if they have the 4.5MB/sec ESCON feature with a 2MB buffer or if they have the 9MB/sec ESCON feature with an 8MB buffer installed.

For the system support of the automatic cartridge loader of the IBM 3490E, IBM 3490, and IBM 3480 tape devices, see the topics on “Support for the Automatic Cartridge Loader” on page 31 and “Summary of VSE Support for IBM 3490 and IBM 3490E Tape Devices” on page 32.

Access to a Shared Tape Unit

You can share an IBM 3490E, 3490, or 3480 tape unit, for example, between two CPUs. Only one of these CPUs, however, can “own” the device at a particular

time. Two operator commands, **ONLINE** and **OFFLINE**, allow controlled access to the shared device. Figure 1 is a simplified example:
Whenever the operator issues an ONLINE command for an IBM 3480, 3490 or

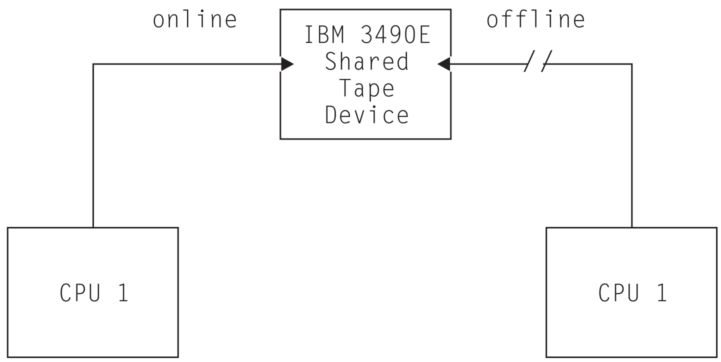


Figure 1. Access to a Shared Tape Unit

3490E tape (cartridge) device, the tape device is being dedicated to that CPU. However, if the tape device is already in use by another CPU, a message indicates this status. No other CPU can access this tape device until the operator releases the device with the OFFLINE command. Under “ONLINE” and “OFFLINE”, the manual *z/VSE System Control Statements* describes the ONLINE and OFFLINE commands in detail.

Tape Label Display

The **VOL1 label** of a labeled tape cartridge device (like the IBM 3480, IBM 3490, and IBM 3490E) is displayed on the tape device’s message display, when the device shows a READY or VOLUME CHANGE interrupt.

Support for the Automatic Cartridge Loader

z/VSE supports the automatic cartridge loader of IBM 3592, IBM 3590, IBM 3490E, IBM 3490, and IBM 3480 tape devices. This forces the automatic processing of more than one cartridge mounted on the same tape drive (address) before switching to an alternate tape drive (address). Refer also to “Characteristics of IBM 3490/3490E Tape Devices” on page 32 for more information.

Summary of VSE Support for IBM 3490 and IBM 3490E Tape Devices

Following is a summary of the VSE support for IBM 3490 and IBM 3490E tape (cartridge) devices.

Characteristics of IBM 3490/3490E Tape Devices

Table 5. Characteristics of IBM 3490/3490E Tape Devices

	IBM 3490 A01/A02 (see Note 1) B02/B04	**IBM 3490 D31/D32**	**IBM 3490E A10/A20 (see Note 1) B20/B40**	**IBM 3490E D41/D42 C10/C11/C22 C1A/C2A (see Note 3)**
Number of tracks	18	18	36	36
IDRC	standard	optional	standard	standard
IDRC disable switch	YES	YES	not applicable	not applicable
Automatic Cartridge Loader	standard	optional	standard	(see Note 4)
Performance Enhancement Function (see Note 2)	not applicable	not applicable	standard	optional
Buffer Size (MB)	2	2	8	2
ESCON Adapter	optional	optional	optional	optional
Parallel Adapter	optional	standard	optional	standard

Notes:

1. The IBM 3490E Models A01/A02 and A10/A20 do not include channel attachment as standard equipment. Channel adapter type and quantity must be specified at initial order.
2. This function uses an 8MB buffer and an increased speed compactor. An instantaneous data rate of up to 9MB/sec is provided on ESCON channels.
3. Models C1A and C2A of the IBM 3490E are included in the IBM 3494. For the z/VSE support of the IBM 3494 refer to "Support for the IBM TotalStorage 3494 Tape Library" on page 27.
4. For the models C11, C22, C1A, and C2A, it is standard and for the models D41/D42 it is optional. For the model C10 it is not applicable.

Virtual Tape Support

A virtual tape is intended to be transparent to applications and to provide you with the ability to read from or write to a virtual tape in the same way as if it were a physical tape. For technical and performance reasons, the full range of the capabilities of a physical tape has not been implemented in z/VSE.

In z/VSE, a virtual tape is a file (or dataset) containing a tape image. A virtual tape can be:

- A VSE/VSAM **ESDS** file on the z/VSE host side.
- A **remote** file on the server side; for example, a Linux, UNIX, or Windows file. To access such a remote virtual tape, a **TCP/IP** connection is required between z/VSE and the remote system.

The Virtual Tape Support is available for tape devices with a device type code of **3480**, **3490**, or **3490E**.

For details of how to use virtual tapes, refer to the *z/VSE Administration*, SC33-8224.

Channel-Related Support

Dynamic I/O Handling

z/VSE provides for improved I/O handling through the dynamic channel subsystem which is a capability of zSeries (31-bit) and ESA/390 processors. The channel subsystem supports:

- Dynamic path selection (DPS)
- Dynamic path reconnection (DPR)
- Up to 256 channels
- Up to 8 paths to a single device.

z/VSE supports DPS for all devices with multiple paths and DPR for all devices which support DPR.

With dynamic I/O handling, the traditional device specification via cuu gets a new meaning. cuu no longer specifies channel and unit address but the device number as generated via IOCP (Input/Output Configuration Program) for the device. In z/VSE, a device number can be between 0000 and 0FFF.

In such an environment, channels are transparent to z/VSE. They are important to the dynamic channel subsystem only and are defined via IOCP for which details are given under "z/VSE Input/Output Configuration Program (IOCP)" on page 80.

FICON Support

FICON is an abbreviation for (Native FIber CONnection) channel support. This enables these devices, for example, to be accessed via S/390 or zSeries:

- FICON-attached devices such as tapes (IBM 3590–A60, for example).
- Disks (IBM Enterprise Storage Server, for example).

ESCON Support

ESCON Channel Support

Some **IBM 3174 controllers** have ESCON channel interfaces for SNA and non-SNA control units.

ESCON Channel-to-Channel Support

Through VTAM, z/VSE supports the ESCON CTC connection between two S/390 processors.

ESCON Multiple Image Facility (EMIF)

ESCON Multiple Image Facility allows for a more flexible utilization of resources, such as ESCON channel sharing between PR/SM logical partitions, less channel connections, and common usage of channels and peripheral devices.

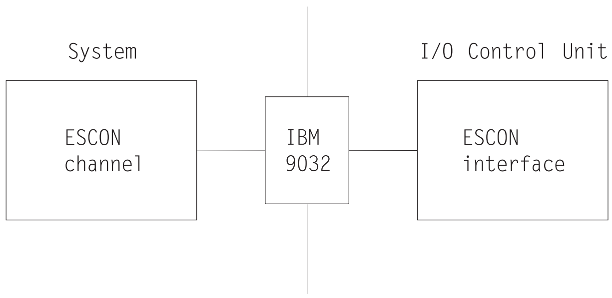
IBM Enterprise Systems Connection Director (ESCD) Support

z/VSE supports the following Enterprise Systems Connection Director (ESCD) devices:

- **IBM 9032**
- **IBM 9033**

The ESCD devices provide dynamic ESCON connections. The various models of the IBM 9032 support up to 124 *ports*, where port means an attachment for ESCON channels or control units. The IBM 9033 supports a minimum of 8 and a maximum of 16 ports. The ports can be Light Emitting Diode (LED) or Extended Distance Feature (XDF) or both.

When controlled from z/VSE, the device is recognized during IPL. The Error Recovery Procedure (ERP) handles all errors reported by solicited and unsolicited interrupts of that device. The following example illustrates the connection between a processor and the I/O control unit via an ESCD device:



When you run z/VSE under z/VM, you can make use of ESCON manager functions. In a native z/VSE system, ESCON directors have to be controlled through their operator console.

IBM 9035 ESCON Channel Protocol Converter

z/VSE supports the IBM 9035 ESCON Channel Protocol Converter. It lets you attach an ESCON control unit to parallel channels.

Miscellaneous Hardware Related Items

Assembling and Cataloging IOCS Modules

This requirement varies from installation to installation and depends on the kind of application programs used. It may arise if your migration to a new release involves also the installation of a new I/O device other than disk or tape.

Perform this step only if the release you get:

1. Includes the support of a new unit-record device (a printer or optical mark reader, for example).
2. Your system's configuration includes this new device.
3. Your system does not include a compatible I/O module.

IBM supplies IOCS (Input/Output Control System) modules needed by the z/VSE supported compilers. Since these modules may be linked also to user-written programs, there may be no need for you to catalog IOCS modules of your own. For a list of IOCS modules shipped with z/VSE, see the *Program Directory*.

Figure 2 on page 36 shows a job stream example for assembling and cataloging IOCS modules for the **IBM 3525** punch device.

```
* $$ JOB JNM=IOMOD,DISP=D,PRI=8,                                          C
* $$ NTFY=YES,                                                            C
* $$ LDEST=*,                                                             C
* $$ CLASS=0
* $$ PUN DISP=I,CLASS=0,PRI=9,PUN=SYSPCH,DEST=*
// JOB IOMOD
// LIBDEF *,SEARCH=(PRD1.MACLIB,PRD2.GEN1)
// OPTION DECK
// EXEC ASMA90,PARM='SIZE(MAX,ABOVE)'
     PUNCH '// JOB IOMOD    JOB TO ADD PUNCHED MEMBERS TO LIBRARY'
     PUNCH '// LIBDEF PHASE,SEARCH=IJSYSRS.SYSLIB'
     PUNCH '* **********************'
     PUNCH '* *'
     PUNCH '* * THIS JOB CATALOGS I/O MOD TO IJSYSRS.SYSLIB'
     PUNCH '* *'
     PUNCH '* **********************'
     PUNCH '// EXEC LIBR,PARM=''MSHP'''
     PUNCH 'AC S=IJSYSRS.SYSLIB'
     END
/*
// OPTION DECK
// EXEC ASMA90,PARM='SIZE(MAX,ABOVE)'
             PRINT NOGEN
             CDMOD CTLCHR=ASA,IOAREA2=YES,DEVICE=3525,TYPEFLE=OUTPUT,  C
                SEPASMB=YES
             END
/*
// EXEC ASMA90,PARM='SIZE(MAX,ABOVE)'
     PUNCH '/*'
     PUNCH '/&&'
     END
/*
/&
* $$ EOJ
```

Figure 2. Job Stream for Assembling and Cataloging IOCS Modules

Note: The statement

```
// EXEC ASMA90,PARM='SIZE(MAX,ABOVE)'
```

calls the High Level Assembler. Refer to Chapter 6, "Migrating From Earlier Releases," on page 91 for details about calling the High Level Assembler.

Devices Not Supported by z/VSE 3.1

These IBM devices are **not** supported by z/VSE 3.1:

Disks Devices

- 0671
- 3370
- 3375
- 3995
- 9332
- 9335
- 9336
- 9345.

Notes:

1. As a result of the withdrawal of the 9345 support, a Fast Service Upgrade from a 9345 device is not supported.
2. FBA devices can still be used with z/VSE 3.1 as “generic FBA” devices.

Tape Devices

- 2440
- 3422
- 3424
- 3430
- 9346
- 9347
- 9348

Network Communications Devices

- CETI
- SDLC ICA
- BSC ICA
- 2701
- 2703
- 3705
- 3720
- 3725

Other Devices

- 3540
- 3881
- 3886
- 3890
- 3895

Functionality Not Available With z/VSE 3.1

The following functionality is no longer supported by z/VSE 3.1:

- A Fast Service Upgrade (FSU) to z/VSE 3.1 from an IBM 9345 disk (since the IBM 9345 disk is no longer supported).
- The IXFP/SnapShot feature for VSE/ESA is no longer supported for the IBM 9393 RAMAC Virtual Array (RVA).
- VSE/VSAM for VM Version 6 (5686-081) is no longer available as an optional product.

Chapter 3. Base and Optional Programs

The *Program Directory*, which is shipped with z/VSE, provides additional details about base and optional programs such as the Modification Level, the Program ID, the Component ID and the Component Level Code (CLC).

z/VSE Base Programs

As shipped, z/VSE consists of a number of base programs as listed in Table 6 on page 40. Most of these programs are shipped on the z/VSE base tape, some of them on the Extended Base Tape as indicated in Table 6 and further explained below.

> **LE/VSE Fully Integrated Into VSE Central Functions!**
> From z/VSE 3.1 onwards, the LE/VSE (IBM Language Environment for VSE/ESA) base program is fully integrated into the VSE Central Functions.

The program number for ordering z/VSE (including all base programs) is **5609-ZVS**. Refer to "Shipment of z/VSE 3.1" on page 76 for additional details.

z/VSE 3.1 Base Program Changes

From z/VSE 3.1 onwards, LE/VSE is integrated into the VSE Central Functions.

Base Programs Shipped on the Extended Base Tape

Some of the z/VSE 3.1 base programs are shipped on the Extended Base Tape. These programs are in most cases not installed automatically as further explained below:

- **CICS/VSE 2.3**

  If you want to create a CICS coexistence environment, you must install CICS/VSE 2.3 separately using the dialog for optional program installation.
- **LE/VSE DBCS Locales**

  LE/VSE is shipped as part of VSE Central Functions on the z/VSE base tape, **except** for the *LE/VSE DBCS Locales* component. This component is shipped on the Extended Base Tape and needed if you use a language that requires the Double Byte Character Set. If you need this component, you must install it separately using the dialog for optional program installation. Refer also to "Considerations for LE/VSE and the VSE C Language Run-Time Support" on page 78.
- **DB2 Server for VSE 7.4**

  During initial installation you are asked whether you want install the *DB2 Server for VSE*. If you answer with YES, you must mount the Extended Base Tape for automatic installation. If you respond with NO or in case of an FSU, you can install it later using the dialog for optional program installation.

  The Extended Base Tape includes another DB2 program related to the *DB2 Server for VSE*: the **DataPropagator Relational Capture** option. If you want to use this program, you must install it separately using the dialog for optional program installation.

Base Programs

The z/VSE shipment includes an extra tape containing the **DB2 Server for VSE Help**. This component must be installed using the corresponding DB2 installation support. Refer also to “Use of Installation Member ARISIVAR.Z” on page 145.

Table 6. Base Programs Included in z/VSE 3.1

z/VSE Base Programs	Program Number	Version Release	Extended Base Tape	Comment
VSE Central Functions including:	5686–CF7	7.1		
VSE Advanced Functions	–	–		see Note 1
VSE/SP Unique Code	–	–		see Note 2
VSE/POWER	–	–		see Note 3
VSE/ICCF	–	–		see Note 4
VSE/VSAM	–	–		see Note 5
VSE/Fast Copy	–	–		see Note 6
REXX/VSE	–	–		see Note 7
VSE/OLTEP	–	–		see Note 8
OSA/SF	–	–		see Note 9
VSE Connectors	–	–		see Note 10
LE/VSE	–	–		see Note 11
VTAM	5686–065	4.2		see Note 12
CICS Transaction Server for VSE/ESA	5648–054	1.1.1		see Note 13
CICS/VSE			yes	see Note 14
TCP/IP for VSE/ESA	5686–A04	1.5		see Note 15
DB2 Server for VSE	5697–F42	7.4	yes	see Note 16
DataPropagator Relational Capture		7.4	yes	
DB2 Server for VSE Help		7.4	(*)	
High Level Assembler for VSE	5696–234	1.5		see Note 17
DITTO/ESA for VSE	5648–099	1.3		see Note 18
EREP	5656–260	3.5		see Note 19
ICKDSF	5747–DS2	1.17		see Note 20

Explanations to Table 6:

(*) The *DB2 Server for VSE Help* component is shipped on a separate tape (**not** the Extended Base Tape).

For details about the programs shipped on the Extended Base Tape refer to “z/VSE 3.1 Base Program Changes” on page 39.

Notes:

1. **VSE/Advanced Functions** provides basic system control and includes the supervisor and system programs such as the Librarian and the Linkage Editor.
2. **VSE/SP Unique Code** provides productivity support such as the Interactive Interface and workstation related support.
3. **VSE/POWER** (VSE/Priority Output Writers, Execution Processors and Input Readers) provides spooling support and networking control.
4. **VSE/ICCF** (VSE/Interactive Computing and Control Facility) provides interactive partition support and support for VSE/ICCF libraries.
5. **VSE/VSAM** (VSE/Virtual Storage Access Method) provides data management support.

6. **VSE/Fast Copy** is a utility program providing fast copy support for data on disk or tape devices.
7. **REXX/VSE** provides programming language support. Refer to “Support of REXX/VSE” on page 250 for details.
8. **VSE/OLTEP** (VSE/Online Test Executive Program) belongs to basic system control and is intended for use by IBM personnel.
9. **OSA/SF for VSE/ESA** (Open Systems Adapter Support Facility) provides support for customizing and managing Open Systems Adapters. Refer to “OSA-2 Support” on page 163 for further details.
10. **VSE Connectors** includes the VSE Connector Server, the VSE Connector Client, the VSE/VSAM CLI (Call Level Interface), and the VSE/VSAM Redirector Connector.
11. **LE/VSE** (IBM Language Environment for VSE/ESA) Version 1 Release 4 Modification Level 4 provides a language environment for high-level language applications. It supports COBOL, PL/I, and C. The VSE C Language Run-Time Support is a subset of it, and provides an environment to run applications written in the C programming language. Refer to “Considerations for LE/VSE and the VSE C Language Run-Time Support” on page 78 for further details. The *LE/VSE DBCS Locales* component is shipped on the Extended Base Tape.
12. **VTAM** (Virtual Telecommunications Access Method) provides display station and networking control support.
13. **CICS Transaction Server for VSE/ESA** provides enhanced (compared to CICS/VSE) transaction processing support.
14. **CICS/VSE** (Customer Information Control System/VSE) provides transaction processing support. It is shipped on the Extended Base Tape as a non-priced feature of the CICS Transaction Server for VSE/ESA.
15. **TCP/IP for VSE/ESA** provides a set of protocols by which computers can communicate with any other computer that runs an operating system with a TCP/IP implementation. Refer to “TCP/IP for VSE/ESA” on page 153 for further details.
16. **DB2 Server for VSE** provides advanced, relational database management and DB2 stored procedures support for z/VSE. The *DataPropagator Relational Capture* option of the DB2 Server for VSE is also available on the Extended Base Tape. An extra tape is provided including the *DB2 Server for VSE Help* component.
17. **High Level Assembler for VSE** provides enhanced programming language support at the assembler level replacing originally the DOS/VSE Assembler. Refer also to “Changing from DOS/VSE Assembler to High Level Assembler” on page 95 for migration details.
18. **DITTO/ESA for VSE** (Data Interfile Transfer, Testing and Operations Utility) provides support for moving data fast from one media to another.
19. **EREP** (Environmental Recording Editing and Printing) belongs to basic system control and provides diagnostics support.
20. **ICKDSF** (Device Support Facilities) belongs to basic system control and provides support for maintaining devices such as disks and tapes.

Where Do the Base Programs Reside?

After initial installation, the base programs reside in the system libraries listed below:

- Sublibrary **IJSYSRS.SYSLIB** contains:
  - *VSE Central Functions* including:

VSE/Advanced Functions
VSE/SP Unique Code
VSE/POWER
VSE/ICCF
VSE/VSAM
VSE/Fast Copy

– ICKDSF (Device Support Facilities)

- Sublibrary **PRD1.BASE** contains:

  VSE Central Functions including:

  OSA/SF
  REXX/VSE
  VSE/OLTEP
  VSE Connectors

  The other base programs included in PRD1.BASE are:

  VTAM
  CICS Transaction Server for VSE/ESA
  TCP/IP for VSE/ESA
  High Level Assembler for VSE
  DITTO/ESA for VSE
  EREP

- Sublibrary **PRD1.MACLIB** contains:

  VSE/Advanced Functions Macros
  VSE/VSAM Macros
  VSE/POWER Macros

- Sublibrary **PRD2.SCEEBASE** contains:

  LE/VSE

- Sublibrary **PRD2.DB2740** contains:

  DB2 Server for VSE

- Sublibrary **PRD2.ASN740** contains:

  DataPropagator Relational Capture

- Sublibrary **PRD2.GEN1**, which is optional, contains the:

  VSE/Advanced Functions Generation Feature.

Base Programs That Require Special Attention

This concerns the following base programs:

1. LE/VSE

   This program is automatically installed into library **PRD2.SCEEBASE**. LE/VSE is always available as part of VSE Central Functions. Refer to "Considerations for LE/VSE and the VSE C Language Run-Time Support" on page 78 for details.

   The *LE/VSE DBCS Locales* component is shipped on the Extended Base Tape and must be installed like a z/VSE optional program.

2. TCP/IP for VSE/ESA

   This program is automatically installed into **PRD1.BASE**. There are different levels of functionality available which is further discussed under "TCP/IP for VSE/ESA" on page 153.

3. DB2 Server for VSE

   The *DB2 Server for VSE* is not installed automatically but on request into default sublibrary **PRD2.DB2740**. The related *DataPropagator Relational Capture* option must be installed separately (default sublibrary **PRD2.ASN740**). The *DB2 Server for VSE Help* component is to be installed during DB2 customization. Except for the *DB2 Server for VSE Help* component, which is shipped on a separate tape, the other two programs are shipped on the Extended Base Tape. Refer also to Chapter 10, “z/VSE e-business Connectors and Tools,” on page 139.
4. CICS/VSE

   CICS/VSE is shipped on the Extended Base Tape and must be installed separately if a CICS Coexistence environment is to be established. The default sublibraries are **PRD2.CICSOLDP** and **PRD2.CICSOLDG**.

Further Information

Further details about the contents of system libraries are provided in Chapter 7, “Files and Libraries,” on page 103 and with regard to optional programs in Table 7. To know the contents of the system libraries is important for tasks such as applying service which is discussed in detail in the manual *z/VSE System Upgrade and Service*.

z/VSE Optional Programs

z/VSE optional programs must be installed separately. They are not part of the initial installation process.

Table 7 lists optional programs for z/VSE available when this manual was printed. IBM may have announced other optional programs since then. Consult the *Program Directory*, which is shipped with z/VSE, for the latest information on optional programs or contact your IBM representative.

Table 7 also shows the default sublibrary displayed by z/VSE for selection when installing an optional program. Instead of using the default system sublibrary, you may select your own private sublibrary. The approximate space requirements in cylinders (for an IBM 3390) and FBA blocks (for IBM FBA disk devices) are shown as well. Note that these are approximate values for the English version of a program.

Table 7. Optional Programs for z/VSE 3.1

z/VSE Optional Programs	Program Number	Version Release	Sublibrary	Cylinders (IBM 3390)	FBA Blocks
VSE/Access Control-Logging and Reporting (VSE/ACLR) (Note 1)	5746-XE7	1.2.1	PRD2.PROD	1	214
ACF/NCP	5648-063	7.8.1	PRD2.COMM	36	35124
ACF/SSP	5686-064	4.8.1	PRD2.COMM2	37	35962
AFP Font Collection	5648-B33	2.1.0	PRD2.PROD	(Note 2)	(Note 2)
CCCA/VSE	5686-A07	2.1.0	PRD2.PROD	11	10884
CICSVR/VSE	5686-011	1.2.0	PRD2.PROD	7	6046
COBOL for VSE/ESA	5686-068	1.1.0	PRD2.PROD	15	13232
C for VSE/ESA	5686-A01	1.1.0	PRD2.DBASE	14	13092

Table 7. Optional Programs for z/VSE 3.1 (continued)

z/VSE Optional Programs	Program Number	Version Release	Sublibrary	Cylinders (IBM 3390)	FBA Blocks
(DB2 Server for VSE)	5697-F42		PRD2.DB2740	60	58700
Control Center VSE		7.3.0	PRD2.CCF730	9	8576
Data Restore VSE		7.3.0	PRD2.RCV730	4	3550
QMF for VSE		7.2.0	PRD2.PROD	25	24614
DataPropagator Capture (Note 4)		7.4.0	PRD2.ASN740	8	7646
DFSORT/VSE	5746-SM3	3.4.0	PRD2.PROD	5	4082
Debug Tool VSE	5686-A02	1.1.0	PRD2.PROD	10	9418
DL/I DOS/VS (Note 3)	5746-XX1	1.10.0	PRD2.DLI1A0	44	41746
DL/I VSE (Note 3)	5746-XX1	1.11.0	PRD2.DBASE	40	38796
EP	5735-XXB	1.14.0	PRD2.COMM2	1	768
GDDM					
GDDM/IMD	5668-801	2.1.3	PRD2.PROD	5	4814
GDDM/IVU	5668-723	1.1.3	PRD2.PROD	2	1922
GDDM/PGF	5668-812	2.1.3	PRD2.PROD	5	4754
GDDM/VSE	5686-057	3.2.0	PRD2.PROD	49	48200
High Level Assembler Toolkit	5696-234	1.5.0	PRD2.PROD	14	13121
MQSeries for VSE/ESA	5686-A06	2.1.2	PRD2.MQSERIES	16	15368
NetView FTP	5686-013	1.1.1	PRD2.PROD	11	10320
OGL/370	5688-191	1.1.0	PRD2.AFP	5	4258
PL/I for VSE/ESA	5686-069	1.1.0	PRD2.PROD	11	10230
PPFA/370	5688-190	1.1.0	PRD2.AFP	2	1666
PSF/VSE	5686-040	2.2.1	PRD2.AFP	6	5906
DOS/VS RPG II	5746-RG1	1.3.0	PRD2.PROD	4	3100
SDF II VSE	5746-XXT	(Note 5) 1.6.0	PRD2.PROD	28	24524
VisualGen Host Services	5648-B02	1.2.0	PRD2.DBASE	22	21938
X.25 NPSI	5688-035	3.9.0	PRD2.COMM	5	4572

Notes:

1. Although VSE/Access Control-Logging and Reporting is installed in PRD2.PROD, you must copy the phases DSPLLOG and $SVALOG to the system sublibrary IJSYSRS.SYSLIB before it can be used.
2. The values depend on the fonts required for a particular installation. The complete AFP Font Collection requires about 990 cylinders on an IBM 3390 or 947620 FBA blocks.
3. DL/I DOS/VS can run with CICS/VSE, but not with the CICS Transaction Server.

   DL/I VSE can run with the CICS Transaction Server, but not with CICS/VSE.
4. The *DB2 Server for VSE* itself (as well as the *DataPropagator Relational Capture* option) are shipped as base programs on the Extended Base Tape. However, some of the programs and components that belong to the DB2 support must be ordered separately as z/VSE optional programs such as the *Control Center for*

VSE, *Data Restore for VSE* , and *QMF for VSE*. Refer also to the "z/VSE 3.1 Base Program Changes" on page 39. For the latest information, consult the z/VSE 3.1 *Program Directory*.

5. SDF II VSE can run with both the CICS Transaction Server and CICS/VSE.

Chapter 4. System Organization and Concepts

After initial installation, z/VSE will reside on two disk volumes named **DOSRES** and **SYSWK1**. These two volumes contain the system files and libraries required by z/VSE. The layout of DOSRES and SYSWK1 is predefined and depends on the disk device type used. The detailed layouts are shown in Appendix B, "z/VSE Disk Layouts (DOSRES, SYSWK1)," on page 291.

Chapter 5, "Installing z/VSE," on page 75 provides information on what is to be considered prior to installing z/VSE.

Note: Depending on your configuration and workload it may be required for performance reasons to move system files to additional disk volumes, for example, page data set extents. If doing so, ensure that no file overlap occurs.

Minimum System Configuration

z/VSE 3.1 requires the following minimum system configuration:

- 32 MB of processor (real) storage.

  Since the processor storage available is usually much higher, this value is mainly of interest if z/VSE is running in LPAR mode or under VM.
- Two disk volumes, DOSRES and SYSWK1, providing together at least 916 MB. This includes a page data set of 150 MB (predefined environment A) *providing* a partition data set is being used (for details, see "Page Data Set Considerations" on page 49).
  - For predefined environment B, 114 MB are additionally required for the page data set, resulting in a total of 1030 MB (equal to 916 MB + 114 MB).
  - For predefined environment C, 1898 MB are additionally required for the page data set, resulting in a total of 2814 MB (equal to 916 MB + 1898 MB).

  The volumes must be of the same device type. Also, it is recommended that they are of the same device model. See also "z/VSE System Size" on page 48.
- A tape device. z/VSE 3.1 is shipped on 3480 (compressed) or 3590 cartridges. z/VSE 3.1 is also available on CD-ROM, or can be downloaded via the Internet.
- A system printer. This may be a channel-attached or adapter-attached printer controlled by VSE/POWER or a local terminal printer controlled by CICS. A terminal printer should have a minimum speed of 300 lines per minute.
- A display station. This can be a terminal or programmable workstation of any supported type.
- A non-SNA connected system console. This can be the Integrated Console of a processor or any display station supported as system console.

Note: The IBM 3380 Models A and D **cannot** be used for initial installation of z/VSE 3.1 since the available space is too small. The installation of z/VSE 3.1 requires at least 1770 cylinders on the installation volumes.

z/VSE System Size

For installation, z/VSE requires a minimum of two disk volumes (DOSRES and SYSWK1) of the same device type. The total system size (required space on DOSRES and SYSWK1) of:
- Predefined environment A is about 916 MB (which includes a page data set of 150 MB).
- Predefined environment B is about 1030 MB (which includes a page data set of 264 MB).
- Predefined environment C is about 2814 MB (which includes a page data set of 1898 MB).

For the predefined environments, about 86 MB of free library space are available in the VSE/VSAM master catalog. This applies if no Generation Feature and no optional programs have been installed.

The predefined environment C can only be installed on disks such as the IBM 3390-3 or equivalent size.

Supervisor Mode ESA

z/VSE requires an IBM processor as listed in "Processor Support" on page 4. It provides through the **ESA-mode supervisor** all enhanced z/VSE functions and capabilities such as 31-bit addressing, data spaces, and virtual disks. Please be aware that the z/VSE operating system can execute in 31-bit mode only. It does not implement z/Architecture, and specifically does not implement 64-bit mode capabilities.

An ESA-mode supervisor can run natively and under VM, and supports:
- Static partitions.
- Dynamic partitions

Other major characteristics and capabilities are listed in Table 8 on page 52.

Storage Support and Layout

Real Storage Support

z/VSE can support a maximum of 2 GB of processor (real) storage. Therefore, environment C can run on 2 GB of real storage without the use of a page data set.

Virtual Storage Support

Virtual storage is used up by address spaces and data spaces. An address space can be occupied by one or more static partition(s) or by one dynamic partition. A data space usually holds data. A data space can also be defined to hold a virtual disk. Details about these system resources are provided on the following pages.

The maximum value of virtual storage (VSIZE) that z/VSE Version 3 can support depends on the capacity of the installation's page data set devices. Up to 15 extents can be specified for a page data set, and an entire volume may be used for one extent. For more details, refer to the manual *z/VSE System Control Statements*, SC33-8225.

The **actual usable** VSIZE on a z/VSE system depends on a number of factors: for example, the type of workload, the system setup, and the available processor capacity. The most limiting factor, however, is the size of the processor (real) storage since the paging rate should not exceed an acceptable level.

As an example, some practical values are given:

Processor Storage	Concurrently Usable VSIZE
256 MB	About 800 MB to about 1.6 GB
512 MB	About 1.6 GB to about 3.2 GB
2 GB	About 6.0 GB to about 20 GB

As a rule of thumb you can calculate for VSIZE 3 to 6 times the processor (real) storage available.

Page Data Set Considerations

Starting with VSE/ESA 2.1, it was possible for the first time to run VSE without a page data set. Refer to “Migration Items of Previous Releases” on page 94 for further details.

For performance reasons, consider a separate volume for the page data set.

VSE **without** a page data set is characterized by the following:

- The minimum processor storage or virtual machine size (under VM) required is 32 MB. However, a suitable size will depend upon either the real storage size of your processor, or the storage size of your z/VSE virtual machine.
- The system is initialized by a new option in the IPL supervisor command: **NOPDS**. See also “IPL Supervisor Parameters Command” on page 62.
- VSIZE need not be specified; it is calculated by the system from the processor storage available for VSE and the specified VIO size:

  ```
  VSIZE = processor storage - VIO
  ```

Before VSE/ESA 2.6, if VSE was installed on a large processor, there was no way to prevent the system from formatting the page data set even if it was never used. Only after initial installation was it possible to tailor VSE as a NOPDS system.

From VSE/ESA 2.6 onwards, VSE checks during the initial installation whether the real storage available is sufficient for a NOPDS system. If yes, no page data set is created. This enhancement is related to large processors that provide sufficient storage so that no page data set (NOPDS) is required.

The initial installation process now checks whether the processor storage size is sufficient to accommodate both:

- All virtual storage.
- The real storage required by the system.

The virtual storage size is the sum of the specified values for VSIZE and VIO.

If the processor storage available is sufficient, the DPD commands are ignored and no page data set is created. In addition, message 0I35I is issued:

```
0I35I  PROCESSOR STORAGE LARGER THAN VIRTUAL STORAGE - DPD COMMAND IGNORED
```

Address Spaces, Data Spaces, and Virtual Disks

In addition to **address spaces** (in which static and dynamic partitions reside and the programs executed in them), z/VSE supports also data spaces.

A **data space** is an area in virtual storage like an address space but contains data only. It can have a maximum size of up to 2 GB. A data space can also be used as a **virtual disk** for holding temporary files.

Chapter 12, "Using Data Spaces and Virtual Disks," on page 167 provides further details about data spaces and virtual disks.

Shared and Private Areas

The supervisor distinguishes two types of virtual areas: shared and private.

Shared Areas

Some areas are shared among all address spaces. They are provided for code and data that must be directly accessible to several or all partitions. These areas include the supervisor and the shared virtual areas (SVA 24-Bit and SVA 31-Bit) and may include shared partitions.

Private Areas

The space between the end of the SVA (24-Bit) up to the address space limit or the beginning of the SVA (31-Bit) is available as private area.

Figure 4 on page 56 illustrates this concept further.

Static and Dynamic Partitions

Partitions always reside in an address space. The number of partitions and address spaces depends on the predefined environment chosen during initial installation.

Static Partitions

Once defined, static partitions are always existent and occupy virtual storage. This is in contrast to dynamic partitions as shown later. z/VSE supports a maximum of 12 static partitions. An address space may include more than one static partition but only one may cross the 16 MB line.

Dynamic Partitions

You can define up to between 150 and 200 dynamic partitions in addition to a maximum of 12 static partitions. The actual number of dynamic partitions that can be specified and used depends on the processor, the system configuration, the job profiles, and the workload. Dynamic partitions allow you to exploit the available virtual storage up to the VSIZE maximum available. Each dynamic partition occupies an address space of its own. Contrary to static partitions, z/VSE creates and activates dynamic partitions as they are needed for jobs intended to run in a dynamic partition. Also, it releases the occupied space when a dynamic partition finishes processing. For further details, refer to Chapter 13, "Dynamic Partition Support," on page 175.

GETVIS Areas

Certain functions require virtual storage dynamically during program execution. z/VSE maintains three GETVIS (Get Virtual Storage) areas for this purpose located within an address space:

- System GETVIS area
- Partition GETVIS area
- Dynamic space GETVIS area

Figure 6 on page 61 shows the location of these areas. "GETVIS Areas" in the manual *z/VSE Guide to System Functions* provides additional details about the GETVIS areas and their layout.

Partition IDs and Address Space IDs

z/VSE assigns for each ID (identification) of a **static** partition a **default** address space ID as follows:

Partition ID	Address Space ID
BG	0
F1	1
F2	2
:	:
FA	A
FB	B

The following rules apply:

1. If an ALLOC command does not include an address space ID, z/VSE uses the defaults. You can change the defaults by explicitly defining an address space ID in the ALLOC command, but it is recommended to use the defaults.
2. The BG partition **must** always reside in address space 0.

For **dynamic** partitions the address space ID is identical with the partition ID; C1, for example. It is assigned automatically according to the definitions in the dynamic class table.

Summary of System Characteristics and Capabilities

Table 8. z/VSE 3.1 System Characteristics and Capabilities

Support/Function	Comment
Real storage maximum	2 GB
VSIZE maximum	xx GB (see Note 1)
Number of virtual address spaces	12 (max. static partitions) plus number of dynamic partitions
Maximum size of a virtual address space	2 GB
VTAM in a private address space	YES
VSE/POWER in a private address space	YES
Number of static partitions	12
Number of dynamic partitions	max. 150-200 (NPARTS-12, SYS command)
31-bit addressing	YES
64-bit addressing	NO (see Note 4)
Access registers	YES
Data spaces	YES
Virtual disks	YES
Linkage stack	YES
Callable cell pool services	YES
Space switching	YES
Subsystem storage protection	YES
Location of SVA areas	low (24-bit) high (31-bit)
Hardware data compression	YES
Max. number of devices	1024 (see Note 2)
Dynamic channel subsystem	YES
ESCON/FICON channels	YES
Runs on zSeries processors	YES (see Note 4)
Runs under PR/SM (LPAR mode)	YES
Runs under z/VM	YES (see Note 3)
VM linkage support	All (including IUCV)

Notes:

1. The VSIZE maximum possible (xx GB) is discussed under "Virtual Storage Support" on page 48.
2. During IPL, z/VSE can support more than 1024 devices.
3. For further details about how to run z/VSE under z/VM, refer to these manuals:

   z/VM Version 4, Running Guest Operating Systems, SC24-5997.

   z/VM Version 5, Running Guest Operating Systems, SC24-6115.

4. The z/VSE operating system can execute in 31-bit mode only. It does not implement z/Architecture, and specifically does not implement 64-bit mode capabilities.

Predefined System Environments

To ease your planning and installation efforts, z/VSE provides three predefined environments as shown in Table 9.

Selecting a Predefined Environment

During initial installation, you *must* choose one of the predefined environments (A, B, or C) as the basis for your system. If none of these environments meets your needs, select the most suitable one and tailor it according to your requirements.

Before you install your system, be sure to read the information given here. *By carefully planning your system's initial environment you may considerably reduce any tailoring effort required later*.

Note that the following functions are available with predefined **Environments B and C only**:

- A CICS coexistence environment (running the CICS Transaction Server and CICS/VSE together).
- Installing a second CICS Transaction Server.

Table 9. Predefined Environments

Predefined Environment	Virtual Storage (VSIZE)	Number of Address Spaces	Number of Partitions	Characteristics
A	150 MB	12+dyn	12+dyn	Entry System
B	264 MB	12+dyn	12+dyn	Medium System
C	2 GB	12+dyn	12+dyn	Large System

The maximum VSIZE possible (xx GB) for a z/VSE system depends on various factors explained under "Virtual Storage Support" on page 48.

Environment Characteristics

Predefined environments A, B, and C offer static and dynamic partitions. Both environments include 12 predefined static partitions (12 is the maximum possible).

The abbreviation **dyn** in Table 9 indicates that dynamic partitions are supported. Each dynamic partition occupies an address space of its own. The maximum number of dynamic partitions that can be activated is:

- 28 for environment A
- 48 for environment B
- 108 for environment C

This value are determined by NPARTS (explained below) minus the number of static partitions. However, the number of dynamic partitions that can actually be active at the same time also depends on the virtual storage (VSIZE) available and the number of dynamic partitions specified for each class in the dynamic class table. NPARTS is an IPL SYS command parameter and defines the maximum number of partitions (static and dynamic) that can be specified.

The shared virtual area (SVA) is split into a 24-bit and a 31-bit area.

Figure 3 shows the system layout of environment A, Figure 4 on page 56 of environment B, and Figure 5 on page 58 of environment C.

Storage Layout of Predefined Environment A

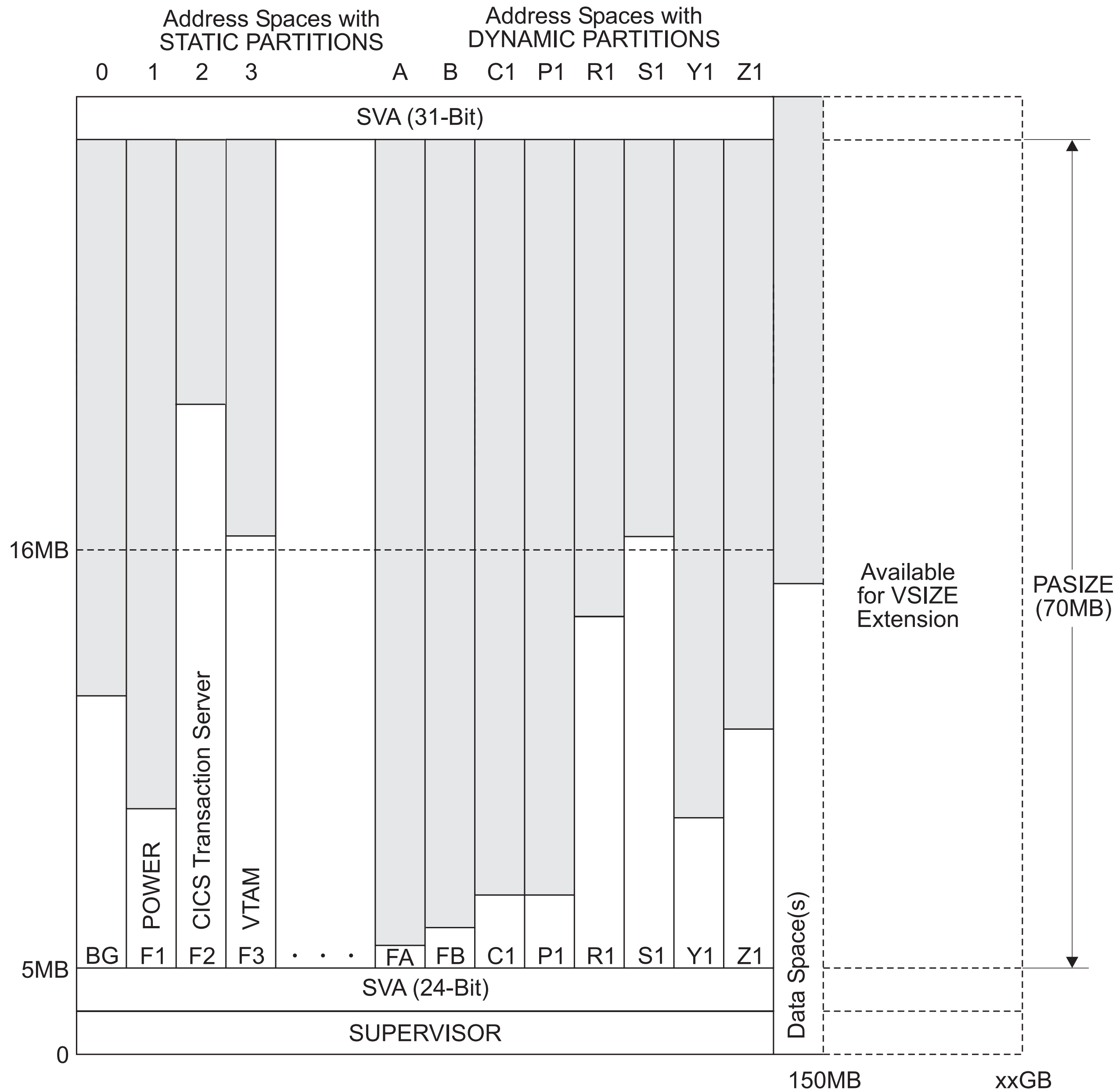


Figure 3. z/VSE Predefined Environment A

Explanations:

1. Environment A offers a default system with a VSIZE of 150 MB and a maximum of 40 partitions, 12 static and 28 dynamic partitions (NPARTS=40). The VSIZE of 150 MB is divided into about:

   81.5 MB for static partitions

   36.5 MB for dynamic partitions

5 MB for supervisor and SVA (24-bit)

12 MB for SVA (31-bit)

15 MB for data spaces (VTAM applications and label area)

The dynamic partitions shown in Figure 3 on page 54 (1 partition of each class active) are an example and have the following storage values (as specified in the predefined dynamic class table DTR$DYNC):

- 1 MB (for 1 C-class partition)
- 1 MB (for 1 P-class partition)
- 8 MB (for 1 R-class partition)
- 15 MB (for 1 S-class partition)
- 2 MB (for 1 Y-class partition)
- 5 MB (for 1 Z-class partition)

This combination occupies 32 MB of the available 39 MB.

A detailed description of DTR$DYNC is given under "The Predefined Dynamic Class Table" on page 180.

2. The default PASIZE is 70 MB resulting in an address space size of about 86 MB for supervisor and SVAs.
3. The areas for the SVAs (shared virtual areas) and the supervisor are shared by all address spaces.
4. Besides the CICS Transaction Server, VSE/ICCF also resides in the F2 partition.
5. VTAM resides in a private address space (F3) but VTAM applications need not reside in the same address space.
6. 15 MB are reserved for data spaces (VTAM applications and label area on virtual disk). This value is defined in the ALLOC procedure (skeleton SKALLOCA). Refer to the manual *z/VSE Administration* under "Skeletons for Static Partition Allocations" for further details about ALLOC and SKALLOCA.
7. The FB partition is reserved for the z/VSE security support. After initial installation, the Security Server of the BSM (Basic Security Manager) runs in the FB partition. Chapter 19, "Security Support," on page 233 provides further details.
8. The F7 partition is reserved for TCP/IP.

Note that with predefined Environment A you cannot have a second CICS TS or a CICS coexistence environment as it is possible with predefined Environment B.

Storage Layout of Predefined Environment B

This environment provides a default system with a VSIZE of 264 MB and a maximum of 60 partitions 12 static and 48 dynamic partitions (NPARTS=60). The VSIZE is divided into about:

169.5 MB for static partitions

57.5 MB for dynamic partitions

5 MB for supervisor and SVA (24-bit)

12 MB for SVA (31-bit)

20 MB for data spaces (VTAM applications and label area)

The default value for PASIZE is 70 MB, resulting in an address space size of about 86 MB for supervisor and SVAs.

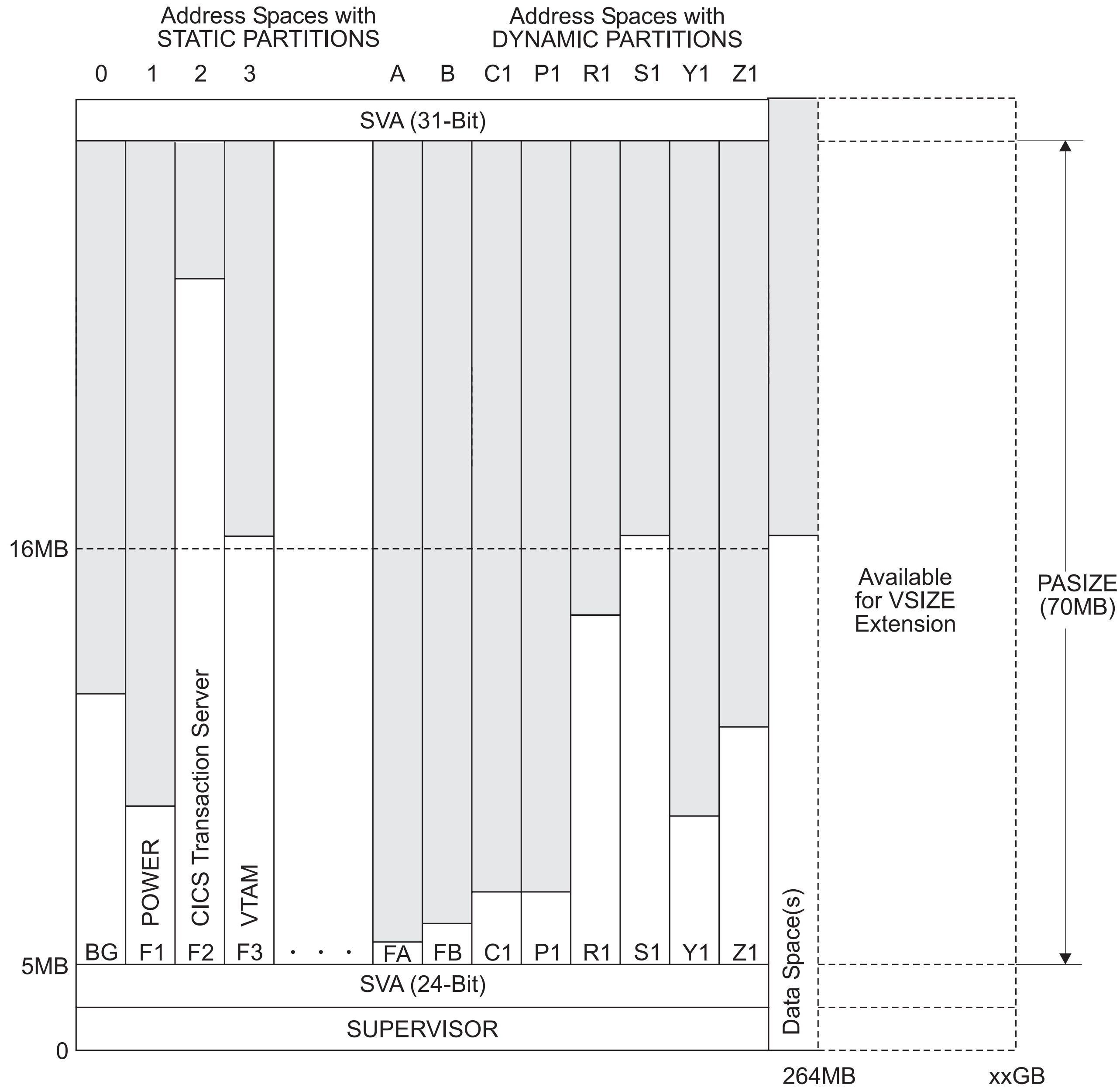


Figure 4. z/VSE Predefined Environment B

Explanations:

1. The dynamic partitions shown in Figure 4 (1 partition of each class active) are an example and have the following storage values (as specified in the predefined dynamic class table, DTR$DYNC):

   1 MB (for 1 C-class partition)

   1 MB (for 1 P-class partition)

   8 MB (for 1 R-class partition)

   15 MB (for 1 S-class partition)

   2 MB (for 1 Y-class partition)

   5 MB (for 1 Z-class partition)

   This combination occupies 32 MB of the available 57.5 MB.

A detailed description of DTR$DYNC is given under “The Predefined Dynamic Class Table” on page 180.
2. The SVA and supervisor areas are shared by all address spaces.
3. Besides the CICS Transaction Server, VSE/ICCF also resides in the F2 partition.
4. VTAM resides in a private address space (F3) but VTAM applications need not reside in the same address space.
5. 20 MB are reserved for data spaces (VTAM applications and label area on virtual disk). This value is defined in the ALLOC procedure (skeleton SKALLOCB). Refer to the manual *z/VSE Administration* under “Skeletons for Static Partition Allocations” for further details about ALLOC and SKALLOCB.
6. The FB partition is reserved for the z/VSE security support. After initial installation, the Security Server of the BSM (Basic Security Manager) runs in the FB partition. Chapter 19, “Security Support,” on page 233 provides further details.
7. The F7 partition is reserved for TCP/IP.
8. After initial installation, the F4 partition is reserved for CICS/VSE, and the F8 partition for a second CICS Transaction Server.

Storage Layout of Predefined Environment C

This environment provides a default system with a VSIZE of 2 GB and a maximum of 120 partitions 12 static and 108 dynamic partitions (NPARTS=120). The VSIZE is divided into about:

1040 MB for static partitions

735 MB for dynamic partitions

5 MB for supervisor and SVA (24-bit)

12 MB for SVA (31-bit)

256 MB for data spaces (VTAM applications and label area)

The default value for PASIZE is 512 MB, resulting in an address space size of about 528 MB for supervisor and SVAs.

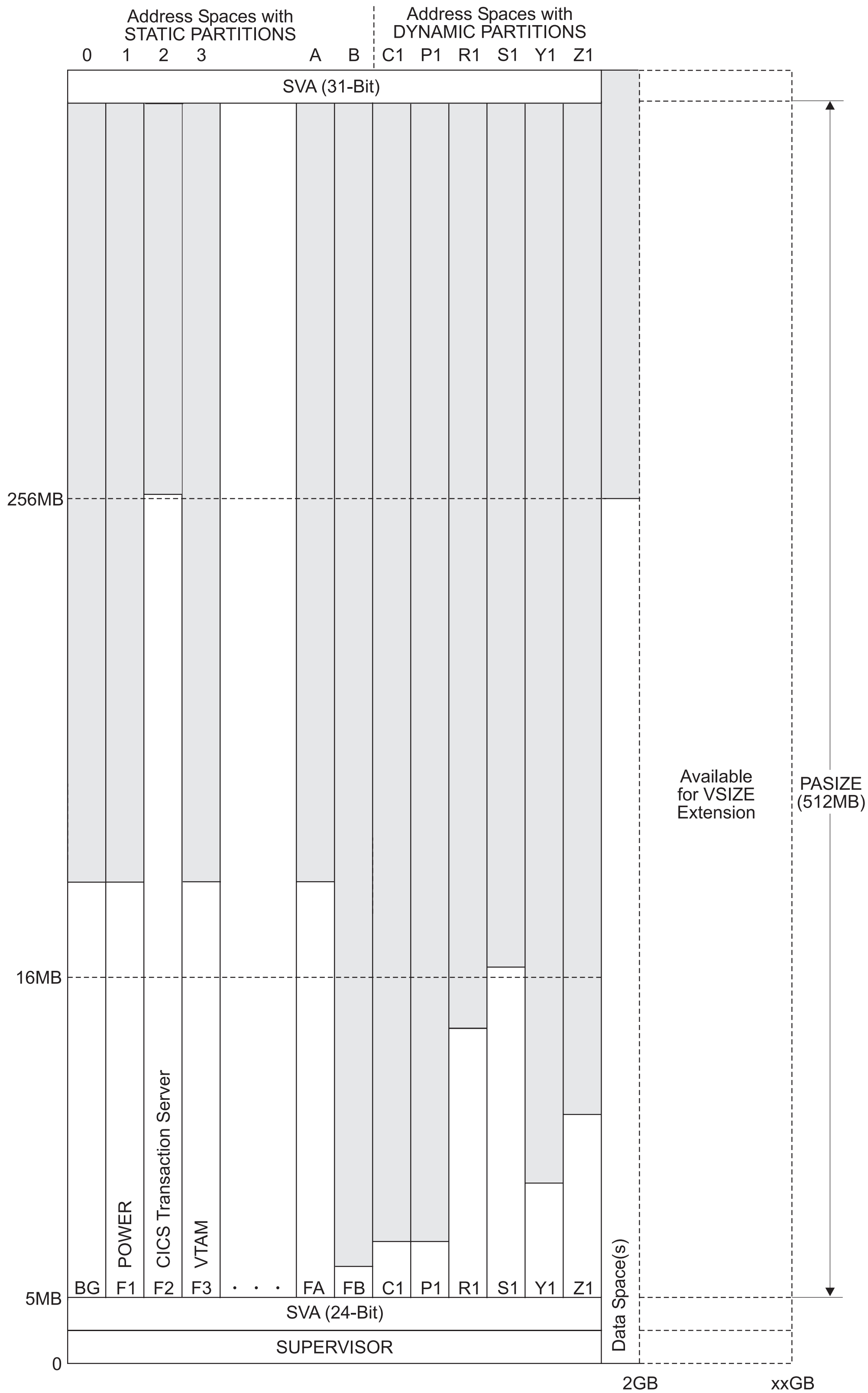


Figure 5. z/VSE Predefined Environment C

Explanations:

1. The dynamic partitions shown in Figure 5 on page 58 (1 partition of each class active) are an example and have the following storage values (as specified in the predefined dynamic class table, DTR$DYNC):

   1 MB (for 1 C-class partition)

   1 MB (for 1 P-class partition)

   8 MB (for 1 R-class partition)

   15 MB (for 1 S-class partition)

   2 MB (for 1 Y-class partition)

   5 MB (for 1 Z-class partition)

   This combination occupies 32 MB of the available 735 MB.

   A detailed description of DTR$DYNC is given under "The Predefined Dynamic Class Table" on page 180.
2. The SVA and supervisor areas are shared by all address spaces.
3. Besides the CICS Transaction Server, VSE/ICCF also resides in the F2 partition.
4. VTAM resides in a private address space (F3) but VTAM applications need not reside in the same address space.
5. 256 MB are reserved for data spaces (VTAM applications and label area on virtual disk). This value is defined in the ALLOC procedure (skeleton SKALLOCC). Refer to the manual *z/VSE Administration* under "Skeletons for Static Partition Allocations" for further details about ALLOC and SKALLOCC.
6. The FB partition is reserved for the z/VSE security support. After initial installation, the Security Server of the BSM (Basic Security Manager) runs in the FB partition. Chapter 19, "Security Support," on page 233 provides further details.
7. The F7 partition is reserved for TCP/IP.
8. After initial installation, the F4 partition is reserved for CICS/VSE, and the F8 partition for a second CICS Transaction Server.

Page Data Set Definitions

The two predefined page data set extents reside on the system disk DOSRES. For all predefined environments, the system keeps a minimum of 150 MB. This is also the total size of the page data set for predefined environment A. The additional 1898 MB for the page data set of predefined environment C are included in the definition for the second extent. Details on the DOSRES layout, including the page data set, are provided under Appendix B, "z/VSE Disk Layouts (DOSRES, SYSWK1)," on page 291.

Notes:

1. In case of high paging rates, it is recommended to move both extents to a separate volume. In this way, you can improve performance because of a better I/O distribution. You specify or redefine page data set extents with the DPD parameter of the *Tailor IPL Procedure* dialog.
2. The 150 MB minimum reserved for the page data set on DOSRES are not accessible via dialog. To reuse them, especially if you either moved the page data set to another volume or if you run with more real storage, the *Tailor IPL Procedure* dialog (Page Data Set Definition) offers PF6=FREE. After pressing the key, the space can be reused by other dialogs.

PASIZE Considerations

With PASIZE you define the maximum size available as private area in an address space. This area is available for partition allocation. The size of an address space is the sum of:

```
PASIZE + size of shared areas 24-bit and 31-bit
```

The resulting value must be equal or smaller 2 GB.

You should limit the PASIZE to the maximum value required for the largest partition in a single address space or to the sum of all partitions in one address space (whatever is larger) since this also reduces the amount of storage needed by the system for storage management.

Program SIZE Restrictions

The size (defined via the SIZE parameter of the EXEC statement or the SIZE command) of a program running in a partition that crosses the 16 MB line is restricted as follows:

```
16 MB - (xx + shared areas 24-bit)
```

where xx is the partition GETVIS size required by z/VSE which must be ≥48 KB (the minimum of 48 KB must be available below 16 MB). The resulting value is further reduced if additional static partitions reside in an address space or by the dynamic space GETVIS area in case of a dynamic partition.

A program, however, can load additional phases above 16 MB into the partition GETVIS area by using the CDLOAD macro (if 31-bit addressing is used).

Considerations for the BG Partition

The minimum storage allocation for the BG partition is 6 MB. This value should not be reduced.

Storage Allocation During System Startup

The z/VSE system startup sequence consists of two parts:

1. The processing of the IPL (initial program load) procedure.
2. The processing of JCL startup procedures and jobs.

It is during the processing of these procedures that z/VSE allocates storage and defines the storage layout as requested. For conceptual details about system startup refer to Chapter 14, “Tailoring System Startup,” on page 183.

The following figure shows the IPL commands and parameters as related to the subareas of an address space. These commands and parameters are part of the IPL procedure. z/VSE provides predefined IPL procedures. A sample is shown under Appendix A, “Startup Procedures,” on page 289.

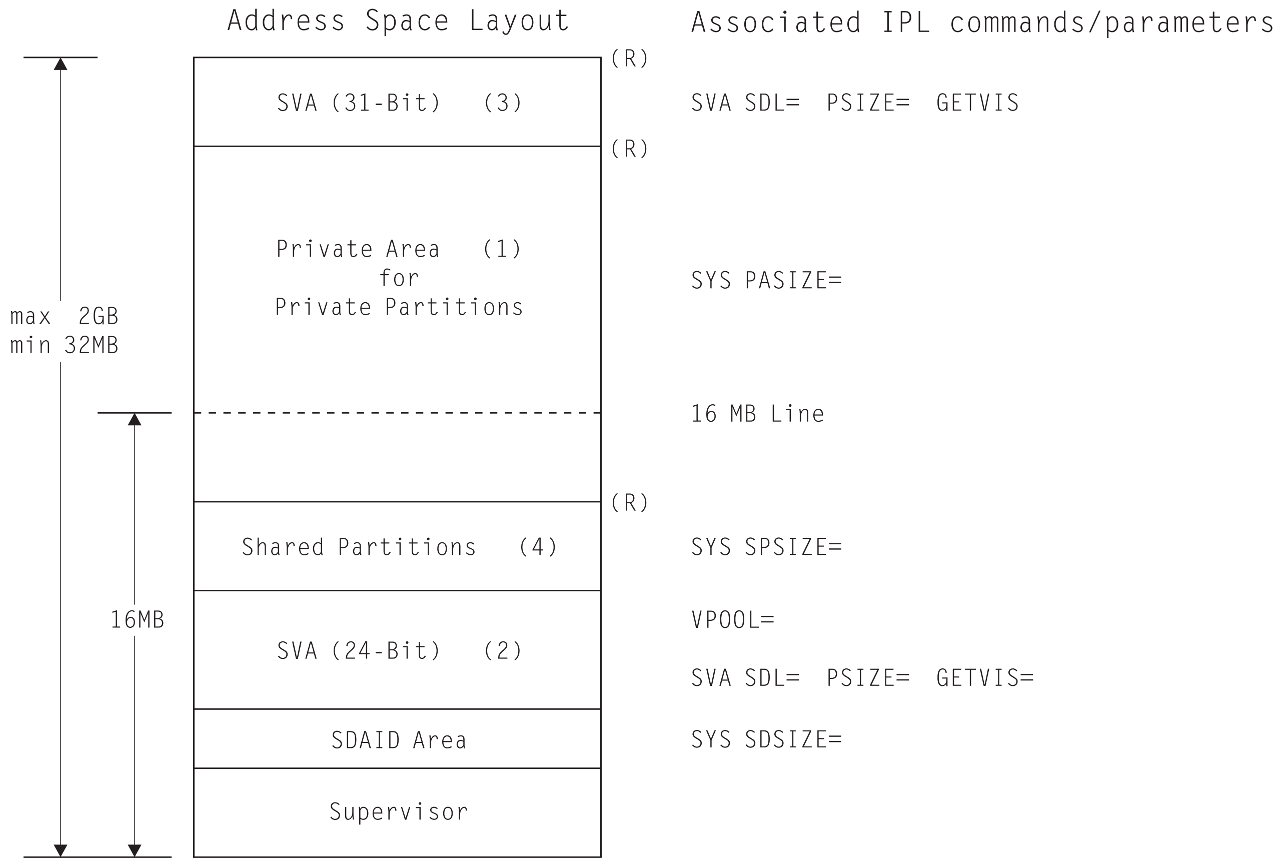


Figure 6. Address Space Layout and Related IPL Parameters

(1) The private area includes:
- Partition GETVIS area (static or dynamic partition)
- Program area
- Dynamic space GETVIS area (dynamic partition)

(2) The SVA (24-Bit) includes:
- VPOOL (virtual pool)
- SLA (system label area)
- System GETVIS area
- VLA (virtual library area)
- SDL (system directory list)

(3) The SVA (31-Bit) includes:
- System GETVIS area
- VLA (virtual library area)

(4) Not existing in predefined environments.

Note on **SLA** (SVA 24-Bit):

This area includes control information for label processing (12 KB for the whole system and an additional 8 KB for each static partition).

Note on the **SVA** command:

Although the SVA command is shown twice in Figure 6, in the IPL procedure only **one** SVA command is present. This single SVA command defines space for both the SVA (31-Bit) and the SVA (24-Bit) as described in the manual *z/VSE System Control Statements* under “SVA”.

(R) identifies storage boundaries. At these boundaries rounding may occur up to the next 1 MB value. The chapter “Storage Management” in the manual *z/VSE Guide to System Functions* provides further details about address space and partition layout.

Loading Phases into the SVA

z/VSE loads system phases that must reside in the shared virtual area (SVA) automatically into this area of virtual storage during system startup. Phases that are to be loaded can be defined in load lists.

Under “Loading Phases into the SVA”, the manual *z/VSE Guide to System Functions* provides background information and describes what is to be considered when loading user phases into the SVA and which tasks are involved.

IPL Storage Allocation Parameters

This section describes the IPL parameters related to **storage allocation**. The values for these parameters can be changed with the *Tailor IPL Procedure* dialog described in the manual *z/VSE Administration* under “Tailoring the IPL Procedure”.

Refer to Table 45 on page 295 for IPL storage values.

For a detailed description of each IPL command and its parameters refer to the manual *z/VSE System Control Statements*.

IPL Supervisor Parameters Command

NOPDS
Defines that the system is to operate without a page data set. You cannot specify both, NOPDS and VSIZE. Only one of these parameters is allowed. Refer to “Migration Items of Previous Releases” on page 94 for further details about NOPDS.

VSIZE
Defines the total virtual storage size of a z/VSE system.

Note that the size of the page data set must correspond to VSIZE+VIO. The value for VSIZE must be sufficient to accommodate all of the following of your z/VSE system:

- ADDRESS SPACES (including shared areas, static and dynamic partitions). Note that the shared areas have to be added only once for all address spaces.
- DATA SPACES (including virtual disks if used).
- You must also consider that z/VSE itself requires virtual storage for system management tasks. As a rule of thumb, z/VSE requires about 4 KB for managing 1 MB of virtual storage (VSIZE).

Note that VSIZE cannot be specified together with NOPDS. In a system with no page data set (NOPDS), z/VSE calculates the VSIZE as follows: processor storage available minus the value specified for VIO.

VIO
Defines the virtual I/O area which can be considered as an extension of the page data set and is used as system work area.

VPOOL
Defines storage needed to exchange data with the VIO area.

IPL SYS Command

PASIZE
: Defines the maximum size of the private area (available for the allocation of private partitions) within an address space. The system will decrease the PASIZE to the largest possible value if necessary.

RSIZE
: Defines the storage that can be allocated for the real execution of programs (the maximum available for ALLOC R definitions).

SDSIZE
: Defines the size of a shared area for system monitor functions such as provided by the SDAID program.

SPSIZE
: Defines the storage to be reserved for shared partitions. The default is 1 MB.

IPL SVA Command

PSIZE
: Defines additional PSIZE space in the SVA (24-Bit) and SVA (31-Bit) areas. PSIZE reserves space for SVA eligible phases loaded via the SET SDL command. It also increases the size of the VLA (virtual library area).

GETVIS
: Defines additional system GETVIS space in the SVA (24-Bit) and SVA (31-Bit) areas. z/VSE automatically reserves system GETVIS space for its own requirements.

SDL
: Defines the number of entries in the system directory list to be reserved for user phases and SVA-eligible phases of z/VSE which are not automatically loaded during IPL.

IPL Commands and Storage Values

The following table shows the possible range of IPL storage values and the defaults chosen by the system if necessary. The leftmost column identifies the related **IPL command**. For the predefined environments, z/VSE allocates storage according to the values shown in Table 11 on page 64.

Table 10. IPL Commands and Storage Values

SUPERVISOR	VSIZE (see Note 1)	maximum minimum default	xxGB 16MB 32MB
	VIO	maximum minimum default	128MB VPOOL value 512KB
	VPOOL	maximum minimum default	16MB 64KB 256KB

Table 10. IPL Commands and Storage Values (continued)

SYS	PASIZE (see Note 2)	maximum minimum default	2048MB 64KB 256KB	
	RSIZE	maximum minimum default	16MB 0KB 64KB	
	SDSIZE	maximum minimum default	256KB 0KB 64KB	
	SPSIZE	maximum minimum default	16MB 0KB 0KB	
SVA	PSIZE (see Note 3)		**24Bit**	**31Bit**
		maximum minimum default	16MB 0KB 1MB	2048MB 0KB 0KB
	GETVIS (see Note 3)		**24Bit**	**31Bit**
		maximum minimum default	16MB 0KB 0MB	2048MB 0KB 0KB

Notes:

1. For an explanation of the maximum VSIZE value (xx GB) refer to "Virtual Storage Support" on page 48.
2. 2048 MB (2 GB) is the theoretical maximum size of an address space. The storage available for PASIZE, however, is 2048 MB minus the size of the shared areas.

   The minimum PASIZE is 6 MB if the VSIZE is equal or larger than 256 MB, and 1 MB if the VSIZE is smaller than 256 MB.

   In a system without a page data set (NOPDS), no VSIZE can be specified. The system uses the processor storage size to calculate the minimum values for the PASIZE.
3. Although the default value is 0 KB, the system acquires and occupies storage needed for system tasks.

IPL Storage Values for Predefined Environments

The following table shows the IPL storage values as set for predefined environments A and B. The leftmost column identifies the related **IPL command**.

Table 11. IPL Storage Values as Set for Predefined Environments

		ENVIRONMENTS		
		A	**B**	**C**
SUPERVISOR	VSIZE	150MB	264MB	2GB
	VIO	512KB	512KB	512KB
	VPOOL	64KB	64KB	64KB

Table 11. IPL Storage Values as Set for Predefined Environments (continued)

		ENVIRONMENTS		
		A	B	C
SYS	PASIZE	70MB	70MB	512MB
SYS	RSIZE	64KB	64KB	64KB
SYS	SDSIZE	64KB	64KB	64KB
SYS	SPSIZE	0KB	0KB	0KB
SVA	PSIZE (24) PSIZE (31) (see Notes)	652KB 6MB	652KB 6MB	652KB 6MB
SVA	GETVIS (24) GETVIS (31) (see Notes)	768KB 6MB	768KB 6MB	768KB 6MB

Notes:

1. The values for PSIZE and GETVIS are the values predefined for the SVA (24-Bit) and SVA (31-Bit) areas. These values are added by the system to its own requirements.
2. If your installation uses 31-bit addressing, you must define storage values for the SVA (31-bit) area according to your requirements. Refer to the IPL SVA command in the manual *z/VSE System Control Statements* under “SVA” for details.
3. To get the current SVA storage values displayed, you can use the *Display Storage Layout* dialog. The dialog is described in detail in the manual *z/VSE Administration* under “Using the Display Layout Dialog”.

Real Storage Allocation

IPL SYS Command RSIZE

The IPL SYS command RSIZE specifies the amount of real storage that may be allocated for the execution of programs in real mode in static partitions. It is the maximum size available for ALLOC R definitions. The RSIZE default of 64 KB is sufficient for static partitions. If dynamic partitions are to be used for programs running in real mode, you must increase the RSIZE.

Note: It is recommended to use ALLOC R definitions **only** if programs are to be executed in real mode (EXEC program,REAL). Whenever a program needs real storage to PFIX pages, the SETPFIX statement should be used.

JCL SETPFIX Statement

The JCL SETPFIX statement sets storage limits per partition for fixing pages (PFIX).

The shipped startup jobs and JCL startup procedures for the predefined environments include SETPFIX statements where page fixing is needed. The values specified for SETPFIX are assigned permanently and are shown in Table 14 on page 68 and Table 15 on page 69.

If you have job streams with programs that PFIX pages, you should use the JCL SETPFIX statement in the job directly to have storage temporarily assigned for the duration of the job.

Note: For a single partition, an ALLOC R specification and a SETPFIX specification (BELOW) are mutually exclusive. Refer to the manual *z/VSE System Control Statements* under "ALLOC" and under "SETPFIX" for details.

Additional Considerations

The size of the shared areas (supervisor, SVA 24-Bit, shared partitions) is taken from the real storage available. If your real storage size is relatively small, it might be that in exceptional cases the remaining size for ALLOC R is not sufficient to execute programs in real mode.

Startup Processing after IPL

Once IPL processing is complete, various JCL procedures are processed mainly for allocating space to static partitions and starting them.

Storage Allocation for Static Partitions

For the predefined environments, the following skeletons (stored in VSE/ICCF library 59) and procedures are provided for allocating storage to static partitions.

Table 12. Skeletons and Procedures for Storage Allocation (Static Partitions)

Name	Environment	Description
SKALLOCA	A	Skeleton
SKALLOCB	B	Skeleton
SKALLOCC	C	Skeleton
ALLOC	A, B, C	Procedure for running system
ALLOCBSX	A, B, C	Procedure for basic startup

Startup Overview of Predefined Environments

The following table provides an overview for the predefined environments about page data set extent books, partition allocation and startup, and the JCL startup procedures, jobs, and skeletons associated with it.

Table 13. Startup Overview of z/VSE Predefined Environments

	ENVIRONMENT A	ENVIRONMENT B	ENVIRONMENT C
Partitions (see Note 1)	12 Static 28 Dynamic	12 Static 48 Dynamic	12 Static 108 Dynamic
Address Spaces	12/28	12/48	12/108
VSIZE (in MB) and related Page Data Set Extent Books (see Note 2)	150 PDSxxxx0/2	264 PDSxxxx0/2	2048 PDSxxxx0/2
Partition Allocation Procedure: Skeletons: (see Note 3)	ALLOC SKALLOCA	ALLOC SKALLOCB	ALLOC SKALLOCC
BG Startup	Procedures: $0JCL, USERBG Skeletons: SKJCL0, SKUSERBG		
F1 Startup (VSE/POWER)	Procedures: $1JCL Skeletons: SKJCL1		

Table 13. Startup Overview of z/VSE Predefined Environments (continued)

	ENVIRONMENT A	ENVIRONMENT B	ENVIRONMENT C
VSE/POWER Autostart Procedures: Skeleton:	`POWSTRTA` `SKPWSTRT`	`POWSTRTB` `SKPWSTRT`	`POWSTRTC` `SKPWSTRT`

F2 Startup
(CICS TS, ICCF)
	`Procedure: $2JCL`
	`Startup Job: CICSICCF`
	`Skeletons: SKJCL2, SKCICS`		

F3 Startup (VTAM)	`Procedure: $3JCL`
	`Startup Job: VTAMSTRT`
	`Skeletons: SKJCL3, SKVTAM`		

F4 Startup	`Procedure: $4JCL` `Skeleton: SKJCL4`		
F5 Startup	`Procedure: $5JCL` `Skeleton: SKJCL5`		
F6 - FA Startup	`Procedure: $6JCL - $AJCL` `Skeletons: SKJCL6 - SKJCLA`		
FB Startup (Security Server)	`Procedure: $BJCL` `Skeleton: SKJCLB`		
Dynamic Partition Startup	`Profile: STDPROF/PWSPROF` `Skeleton: SKJCLDYN`		
Library Definitions	`Procedure: LIBDEF` `Skeleton: SKLIBCHN`		

Notes:

1. The maximum number of partitions (static plus dynamic) that can be active is defined with the NPARTS parameter of the IPL SYS command.
2. Listed are the names of the Z-books stored in IJSYSRS.SYSLIB. Such a Z-book contains the DPD commands specifying the page data set extents. xxxx represents the disk device type (such as 3390) on which the page data set extents reside.
3. During initial installation, z/VSE changes the procedure name for partition allocation automatically to ALLOC. This procedure is active once initial installation has been completed.

Static Partition Allocations for Predefined Environments

The following tables reflect the storage values allocated to the static partitions of the predefined environments. z/VSE provides skeletons (SKALLOCx) for changing these values. Refer to "Skeletons for Static Partition Allocations" in the manual *z/VSE Administration* for details about the skeletons provided. For details about the ALLOC command used to allocate virtual and real storage, see the manual *z/VSE System Control Statements* under "ALLOC".

For Table 14 on page 68 and Table 15 on page 69 the following applies:

- The SETPFIX values are defined in the corresponding JCL procedure for partition startup and in the startup jobs.
- All other storage values are defined through the ALLOC procedure active after initial installation has been completed.

Predefined Environment A

Environment A provides 12 address spaces with 12 static partitions. Additionally, environment A provides support for dynamic partitions as described under "Predefined Dynamic Partition Support" on page 70.

Table 14. Storage Values for Static Partitions (Environment A)

Address Space	Partition	Initial ALLOC	Virtual SIZE	Partition GETVIS	Program	SETPFIX below/above
0	BG	6M	1280K	4816K	---	---
1	F1	4M	1280M	2816M	POWER	200K
2	F2	30M	2M	28M	CICS TS ICCF	144K
3	F3	15M	600K	14400K	VTAM	424K/300K
4	F4	1024K	768K	256K	---	---
5	F5	1024K	768K	256K	---	---
6	F6	512K	256K	256K	---	---
7	F7	16M	1M	15M	TCP/IP	900K/2100K
8	F8	6M	3584K	2560K	---	---
9	F9	512K	256K	256K	---	---
A	FA	512K	256K	256K	---	---
B	FB	1M	872K	512K	Security Server	---

SETPFIX Details

Notes:

1. Each OSA Express link causes TCP/IP to acquire 1 MB SETPFIX storage above if more than 16 MB real storage is available. 2100 KB is sufficient for 2 links.
2. To activate the SETPFIX limit above, you must modify the appropriate startup jobs for TCP/IP and VTAM.

Predefined Environment B

Environment B provides 12 address spaces with 12 static partitions. Partitions F2, F3, F4, F7, and F8 cross the 16 MB line making use of the 31-bit addressing support. Besides static partitions, environment B supports also dynamic partitions as described under “Predefined Dynamic Partition Support” on page 70.

Table 15. Storage Values for Static Partitions (Environment B)

Address Space	Partition	Initial ALLOC	Virtual SIZE	Partition GETVIS	Program	SETPFIX below/above
0	BG	6M	1280K	4816K	---	---
1	F1	5M	1280M	3840M	POWER	200K
2	F2	50M	2M	48M	CICS TS ICCF	144K
3	F3	15M	600K	14400K	VTAM	424K/300K
4	F4	20M	2M	18M	---	---
5	F5	1024K	768K	256K	---	---
6	F6	512K	256K	256K	---	---
7	F7	20M	1M	19M	TCP/IP	900K/2100K
8	F8	50M	2M	48M	---	---
9	F9	512K	256K	256K	---	---
A	FA	512K	256K	256K	---	---
B	FB	1M	512K	512K	Security Server	---

Refer also to “SETPFIX Details” on page 68.

After initial installation, partition F8 is reserved for a second CICS Transaction Server and partition F4 for CICS/VSE (CICS Coexistence Environment).

Predefined Environment C

Environment C provides 12 address spaces with 12 static partitions. Partitions F2, F4, and F8 cross the 16 MB line making use of the 31-bit addressing support. Besides static partitions, environment C also supports dynamic partitions, as described under "Predefined Dynamic Partition Support."

Table 16. Storage Values for Static Partitions (Environment C)

Address Space	Partition	Initial ALLOC	Virtual SIZE	Partition GETVIS	Program	SETPFIX below/above
0	BG	32M	1280K	31488K	---	---
1	F1	32M	1280K	31488K	POWER	200K
2	F2	256M	2M	254M	CICS TS ICCF	144K
3	F3	15M	600K	14760K	VTAM	424K/300K
4	F4	32M	2M	30M	---	---
5	F5	32M	1M	31M	---	---
6	F6	32M	1M	31M	---	---
7	F7	32M	1M	31M	TCP/IP	900K/400K
8	F8	512M	2M	510M	---	---
9	F9	32M	1M	31M	---	---
A	FA	32M	1M	31M	---	---
B	FB	2M	512K	1536K	Security Server	---

Refer also to "SETPFIX Details" on page 68.

After initial installation, partition F8 is reserved for a second CICS Transaction Server and partition F4 for CICS/VSE (CICS Coexistence Environment).

Predefined Dynamic Partition Support

z/VSE provides a dynamic class table and default startup profiles (procedures) for dynamic partition support.

Dynamic Class Table

The dynamic class table (DTR$DYNC) is activated for predefined environments A and B. You can modify the table or create new tables with the *Maintain Dynamic Partitions* dialog. z/VSE supports multiple dynamic class tables.

Startup Profile

z/VSE provides default startup profiles (STDPROF and PWSPROF) for the use of predefined dynamic partitions. To maintain and catalog such a profile, use skeleton SKJCLDYN described in the manual *z/VSE Administration* under "Cataloging JCL Startup Procedures".

For further details, refer to "Predefined Dynamic Partition Support" on page 180.

CICS Transaction Server and VSE/ICCF Interrelationship

Concepts and Partition Layout

The following list provides an overview of the CICS Transaction Server and VSE/ICCF interaction, operation, and partition layout:

- VSE/ICCF runs as a subtask system under the primary CICS Transaction Server.
- VSE/ICCF resides below 16 MB and is loaded and initialized during CICS/VSE startup in the partition GETVIS area.
- VSE/ICCF, including the VSE/ICCF interactive partitions, requires about 3 MB of virtual storage below 16 MB. As shipped, this value is added to the GETVIS space below 16 MB reserved for the CICS/VSE partition (size value).
- During CICS/VSE startup VSE/ICCF is initialized via the initialization program DTSPOSTI. To restart VSE/ICCF during production, transaction I$ST is available. This may be necessary to implement VSE/ICCF configuration changes which is possible via the VSE/ICCF generation table. The name of the default generation table (DTSIGEN) has then to be specified as parameter for transaction I$ST.

  The table DTSIGEN must be assembled and cataloged in a VSE/ICCF regeneration run as described under “Regenerating VSE/ICCF” on page 282.

  z/VSE provides table DTSIGENM for generating larger interactive partitions if needed.
- Together with the CICS Transaction Server, VSE/ICCF can run in a dynamic partition.
- For VSE/ICCF shutdown, the command /ICCFEND and transaction I$SH are available.
- VSE/ICCF transaction programs have been converted from CICS macro to CICS command level.
- The DSA storage limit (EDSALIM) for the CICS Transaction Server above the 16 MB limit is set to:
  - 14 MB for predefined environment A.
  - 25 MB for predefined environment B when the CICS TS runs in either partition.
  - 200 MB for predefined environment C when the CICS TS runs in partition F2.
  - 450 MB for predefined environment C when the CICS TS runs in partition F8.

The value for environment B can be increased to about 35 MB with a partition size of 50 MB. For environment C, the value can be increased to about 240 MB for F2 and 490 MB for F8.

In a z/VSE system with several CICS Transaction Servers only one can include VSE/ICCF. Figure 7 on page 72 shows the layout of the CICS - ICCF partition (F2).

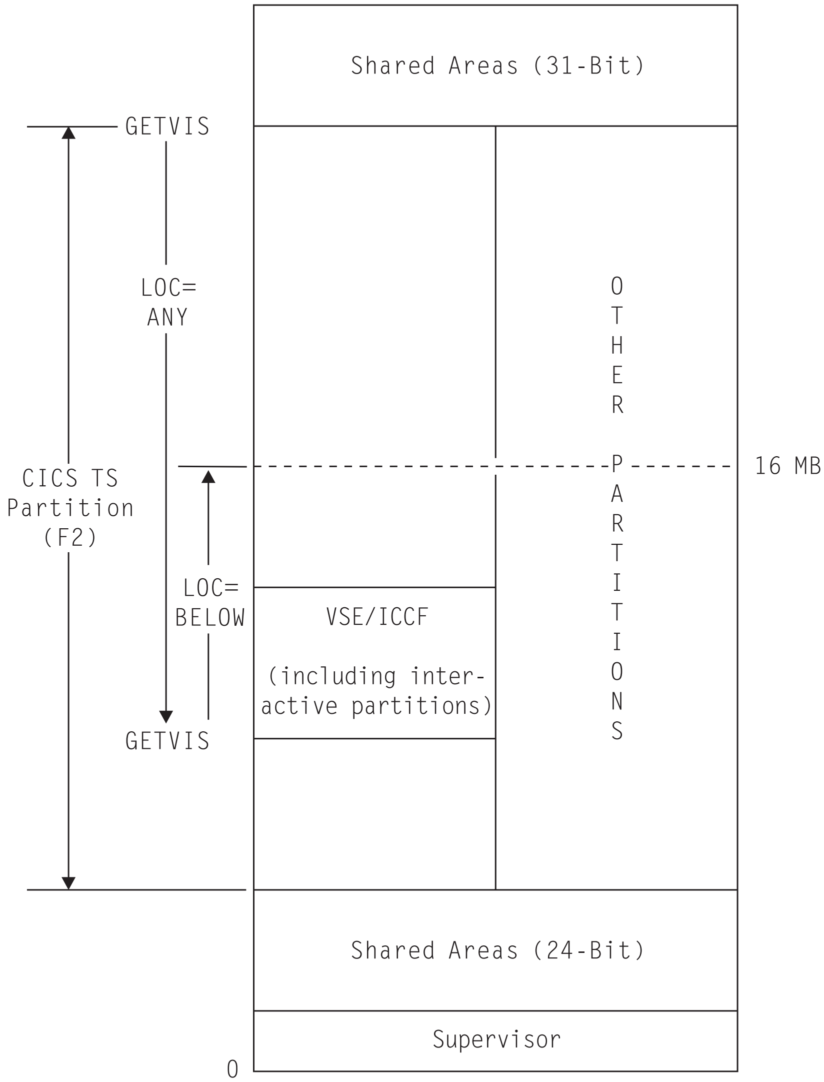


Figure 7. CICS Transaction Server - VSE/ICCF Partition Layout

Note: Compared to the former CICS/VSE, the CICS Transaction Server provides more space below 16 MB for user applications.

VSE/ICCF Interactive Partition Layout and Characteristics

The Interactive Interface uses **interactive partitions** that are allocated within VSE/ICCF in the CICS TS partition (F2).

Table 17 shows the characteristics of the predefined VSE/ICCF interactive partitions. You can change their size through the VSE/ICCF generation table DTSIGEN or DTSIGENM. Refer also to “Regenerating VSE/ICCF” on page 282.

If you increase the size of the existing interactive partitions or add partitions, you should also make a corresponding increase in the size of the F2 partition.

Note that you should **not**

- Add a second class I partition. Some functions of the Interactive Interface use class I. These functions should *not* be performed concurrently.
- Decrease the size of any partition, though you may be tempted to save 24-Bit storage.
- Remove all class A and B partitions. The Interactive Interface requires at least one partition of each class.

To improve Interactive Interface performance, you should also consider regenerating VSE/ICCF, for example, to increase the number of DTSFILE buffers.

Table 17. Characteristics of the Predefined VSE/ICCF Interactive Partitions

Interactive Partition	Class	Minimum Size	Used By
1	I	1024K	Interactive Interface
2	A	384K	Interactive Interface
3	A	384K	Interactive Interface
4	A,B	512K	Interactive Interface
5	A,B	512K	Interactive Interface

Displaying Storage Layout Information

Before you start modifying the storage and partition layout of your system, you should use the *Display Storage Layout* dialog to get the real and virtual storage values of your current system. The dialog is described in detail in the manual *z/VSE Administration*.

Chapter 5. Installing z/VSE

Introducing the Installation of z/VSE 3.1

The following IBM disk devices are supported for the initial installation of z/VSE 3.1:

3380 (with a minimum of 1770 cylinders)
3390
FBA

Notes:

1. The device types 3380 and 3390 can also be devices of the IBM RAMAC Array family, the IBM TotalStorage Enterprise Storage Server (ESS), or can be the Internal Disk of the IBM Multiprise 3000 processors.
2. The FBA disk device can be either:
   - A "hardware" FBA-SCSI disk in z/VSE.
   - An FBA disk under z/VM which may be a SCSI disk under z/VM, or an FBA minidisk.

How to Install z/VSE 3.1

- If you install z/VSE for the first time, you must perform an **initial installation**.
- If you have already installed a VSE/ESA 2.6.x or 2.7.x system, you can perform a *Release Upgrade* through an **FSU** (Fast Service Upgrade).
- You cannot perform an FSU from FBA devices.
- Older releases of VSE/ESA cannot be upgraded via FSU.

Initial Installation

This is the standard method if you install z/VSE for the first time or if your current system does not fulfill the requirements for an FSU. Refer to "Initial Installation of z/VSE 3.1" on page 80 for an overview of the initial installation process.

The manual *z/VSE Installation* provides a step-by-step description for performing an initial installation.

FSU from VSE/ESA 2.6.x or 2.7.x to z/VSE 3.1

The great advantage of an FSU is the fact that your migration effort is reduced to a minimum. The FSU uses your current system as installed on DOSRES and SYSWK1 and upgrades it to the z/VSE 3.1 level by replacing IBM code as follows:

- z/VSE base programs.
- VSE/ICCF members provided by IBM.
- The online messages file.

User libraries, hardware configuration tables, and other installation-unique data are not changed by the FSU.

An FSU does not install or update z/VSE optional programs or programs shipped on the Extended Base Tape. If you want to add or update such programs you can do so after FSU completion.

You perform an FSU by using the *Fast Service Upgrade* dialog. This chapter provides planning information and discusses important aspects of an FSU (refer also to "Installing z/VSE 3.1 via FSU" on page 82). The manual *z/VSE System Upgrade and Service* describes the FSU dialog in detail.

Comparing Initial Installation and FSU

Table 18 provides an overview of the advantages and disadvantages comparing the two installation methods.

Table 18. Comparing Initial Installation against FSU

	Initial Installation	FSU
Eligible for:	All users	VSE/ESA 2.6.x or 2.7.x users
Requirements:	No special requirements	(see "Required Status of Current System" on page 82)
Hardware configuration:	Can be migrated (which must be requested)	Kept as is (only system data on DOSRES and SYSWK1 replaced)
User profiles, selection panels, application profiles:	Can be migrated (which must be requested)	Kept as is (only system data on DOSRES and SYSWK1 replaced)
User-specific data:	Access to VSE/VSAM data must be re-established VSE/ICCF user libraries must be restored	Kept as is For example: System layout, VSE/VSAM data, VSE/ICCF file, etc. System startup is to be adapted
z/VSE optional programs:	Must be installed	Should be refreshed
Additional IBM or non-IBM programs:	Must be installed	Should be refreshed
User application programs:	Must be re-established (update of VSE/VSAM catalogs, CICS tables, etc.)	Kept as is (recompile may be necessary)
CICS Coexistence Environment:	Must be installed	Kept as is
System Modifications:	Possible	(see "Restrictions for System Modifications" on page 82)

Shipment of z/VSE 3.1

z/VSE 3.1 is delivered in the following three ways:

- **As distribution tapes**. Either:
  - Three IBM 3480 compressed distribution tapes (compacted using IDRC).
  - Three IBM 3590 distribution tapes (128 tracks).
- **As a CD–ROM**.
- **Via the Internet**.

These three possible ways to ship z/VSE are now described.

Shipment of z/VSE 3.1 on Distribution Tapes

z/VSE 3.1 is shipped on **three** distribution tapes as follows:

- One **base tape**.
- One **extended base tape**.
- One **extra tape** for the *DB2 Server for VSE Help* component.

The distribution tapes you receive are IBM 3480 or IBM 3590 cartridges. The IBM 3480 is in compressed format (compacted with IDRC).

The base tape is labeled **Z/VSE3.1.X-YY**, the extended base tape has the label **Z/VSE3.1.X-XBASE**. For the tape containing the *DB2 Server for VSE Help* component, refer to the z/VSE 3.1 Program Directory.

The ″X″ identifies the latest modification level of the current release. IBM provides National Language Support (NLS) for z/VSE. ″YY″ is the language indicator:

EN for English
GE for German
KA for Kanji (Japanese)
SP for Spanish

Note: You receive additional tapes if you have ordered z/VSE optional programs.

Extended Base Tape

The extended base tape for z/VSE 3.1 includes the following programs and components:
- DB2 Server for VSE
- DataPropagator Relational Capture
- LE/VSE DBCS Locales
- CICS/VSE

Programs shipped on the Extended Base Tape are not installed automatically. An exception is the **DB2 Server for VSE** for which you are asked during initial installation (but not during an FSU). If you answer with YES, you must mount the tape and the program is then installed automatically.

Installing Programs from the Extended Base Tape: From an installation point of view, programs and components on the Extended Base Tape are to be treated as z/VSE optional programs. They have to be installed using the *Install Program(s) from Tape* dialog as described in the *z/VSE Installation* manual. You reach this dialog by first selecting the *Install Programs -V2 Format* dialog.

Extra Tape

This tape contains only one component: the *DB2 Server for VSE Help*. For installation details, refer to “Use of Installation Member ARISIVAR.Z” on page 145. Note that you cannot install this component like a z/VSE optional program.

For details of how to obtain and use the above z/VSE 3.1 distribution tapes, refer to the manual *z/VSE Installation*, SC33-8222.

Shipment of z/VSE 3.1 on CD-ROM

There are two CD-ROMs that can be used directly at your workstation. The first CD-ROM is labelled VSE_BASE_CDR_YYY and contains these files in virtual tape format:
- VSE31xYY.AWS
- VSE31xXB.AWS
- DB2HELP.AWS

The second CD-ROM contains the file VSE31xOP.AWS in virtual tape format, which contains the optional products.

The above files contain the contents of the three tapes described in "Shipment of z/VSE 3.1 on Distribution Tapes" on page 76.

The "YYY" part of the label VSE_BASE_CDR_YYY refers to the language used, where:
- ENU is for English
- DEU is for German
- ESP is for Spanish

For details of how to obtain and use the above z/VSE 3.1 CD-ROM, refer to the manual *z/VSE Installation*, SC33-8222.

Downloading of z/VSE 3.1 Via the Internet

If you download from the Internet, the tape image files are in zipped format. These zipped files contain the contents of the three tapes described in "Shipment of z/VSE 3.1 on Distribution Tapes" on page 76.

Note: It is currently not possible to download Japanese tape image files via the internet.

For details of how to obtain and use the above z/VSE 3.1 zipped files, refer to the manual *z/VSE Installation*, SC33-8222.

Considerations for LE/VSE and the VSE C Language Run-Time Support

From z/VSE 3.1 onwards, LE/VSE is a z/VSE base program. This means, you receive LE/VSE as part of VSE Central Functions. LE/VSE is installed automatically (during initial installation) into sublibrary PRD2.SCEEBASE.

The VSE C Language Run-Time Support is also part of VSE Central Functions. It enables you to run applications compiled with the *C for VSE/ESA* compiler.

TCP/IP Considerations

TCP/IP for VSE/ESA is installed automatically into sublibrary PRD1.BASE. Refer to "Installing TCP/IP for VSE/ESA" on page 153 and "Customizing TCP/IP for VSE/ESA" on page 154 for further details.

Note that some components require that TCP/IP is up and running before they can be started. For example, the DB2 connector support.

DB2 Considerations

Refer to Chapter 10, "z/VSE e-business Connectors and Tools," on page 139 for installation details of the DB2 Server for VSE.

VTAM Considerations

VTAM Passwords

You receive VTAM *Version 4 Release 2* with z/VSE 3.1. VTAM consists of three different levels. When you ordered VTAM, you had to specify one of these levels. With the z/VSE distribution tapes, you also receive a unique VTAM password customized for your site.

For initial installation, z/VSE uses a predefined password and no action on your side is necessary. However, when initial installation is complete and you tailor your system, you must activate your own VTAM password by modifying skeleton SKVTAM stored in VSE/ICCF library 59. Proceed as follows:

- Copy the skeleton into your primary VSE/ICCF library.
- Use the copy of the skeleton to modify the VTAM password.
- Submit the skeleton for processing.

For further introductory information refer to the manual *VTAM Overview*, for further technical details refer to the *VTAM Network Implementation Guide*.

VTAM storage requirements are discussed under "VTAM 4.2 Storage Requirements" on page 87.

Initial Installation of z/VSE 3.1

Latest Information

For the latest information on installing z/VSE, consult the *Program Directory* shipped with z/VSE 3.1.

You must plan for initial installation as well as for the subsequent customization and tailoring steps. After initial installation, you can use the Interactive Interface to customize and tailor z/VSE for your particular needs.

Any preparatory steps, like defining and planning your system requirements or loading the Input/Output Configuration Data Set (IOCDS), should be completed before you install z/VSE.

z/VSE Input/Output Configuration Program (IOCP)

Support for the stand-alone Input/Output Configuration Program (IOCP) is supplied with the **hardware system's service processor** or **processor controller**. IOCP describes a system's I/O configuration to the Central Processing Unit (CPU).

Note: Before you install z/VSE natively (not under z/VM or LPAR), make sure that you have fully configured your system via IOCP.

The IOCP of z/VSE is automatically installed during initial installation of z/VSE. You can use this z/VSE batch program to create a new IOCDS when you change the hardware configuration. You can use it to define and validate the IOCP macro instructions if you prepare for the installation of a new processor. Use skeleton SKIOCPCN (available in VSE/ICCF library 59) as a base for configuration changes.

As mentioned above, an Input/Output Configuration Data Set (IOCDS) must be generated for the hardware the first time through the stand-alone IOCP delivered with the processor. For this program you can use prepared IOCP macro instructions which have been generated on another processor. The specified device numbers for z/VSE must be within the range of 0000 through 0FFF. The devices themselves may be attached to any available channel.

For detailed information about IOCP, consult the IOCP manual of your processor.

The Initial Installation Process

The initial installation process is divided into three installation parts and should take about two hours to complete. The three installation parts consist of:

1. Initializing the system disks by using the program Device Support Facilities. The system disks, named DOSRES and SYSWK1, are reserved for use by z/VSE.
2. Transferring z/VSE from its distribution tapes to disk and starting up the system for the first time.
3. Using dialogs of the Interactive Interface to provide z/VSE with specific information vital to the operation of your system.

The first two parts can be done using either an "automatic" installation or a "manual" installation. The automatic installation initializes your disk devices, places the Volume Table of Contents (VTOC), and restores the system library IJSYSRS.SYSLIB. A predefined customization table shipped with the z/VSE installation tape provides the required values.

Planning Steps for Initial Installation

The following planning steps are required:

1. Define your hardware configuration and collect related information like device types and addresses.

   At initial installation, z/VSE supports up to 1024 input/output devices. If more than 1024 input/output devices exist, these additional devices will be ignored by the device sensing process. After initial installation, z/VSE can support more than 1024 I/O devices during IPL providing you add these devices using the *Hardware Configuration* dialog.

   The following devices must be ADDed:
   - Channel-attached devices like disk and tape devices, printers, and communication controllers.
   - Local non-SNA terminals. Each terminal attached requires one IPL ADD command. This may be a constraint. However it can be removed by using local SNA terminals and VTAM.
2. Determine the predefined environment (A, B, or C) that fits your needs best.

   Note that if you want to run a second CICS Transaction Server or the primary CICS Transaction Server together with CICS/VSE (CICS coexistence environment), you must choose predefined environment B or C. Predefined environments B or C are also required for installing the *DB2 Server for VSE.*
3. Decide whether you want to do migration during initial installation or later. Refer to Chapter 6, "Migrating From Earlier Releases," on page 91 for further details.
4. Decide whether you want to install the *DB2 Server for VSE* during initial installation or later.
5. Determine whether you want to run your system with security active. With security active, you can control and restrict the access to resources such as files and libraries. Refer to Chapter 19, "Security Support," on page 233 for further details.
6. Determine the procedure name for the FCB (Forms Control Buffer) or UCB (Universal Character Set Buffer) to be used for your system printer.

To simplify initial installation, z/VSE includes a master procedure ($ASIPROC) and a predefined **customization table**.

Master Procedure $ASIPROC

z/VSE includes an $ASIPROC (TYPE=INSTALL) for initial installation. It selects the IPL and JCL procedures for initial installation according to the following list of IBM disk devices that can be used as DOSRES and SYSWK1 devices:

3380 (with a minimum of 1770 cylinders)

3390

FBA

An FBA device reflects the generalized fixed blocked architecture (GFBA) disk layout, which is used by:
- Minidisks or virtual disks under z/VM
- SCSI FCP-attached disks, as described in Chapter 9, "Using SCSI Disks With Your z/VSE System," on page 133.

Related information is provided in Appendix A, "Startup Procedures," on page 289 and in Appendix B, "z/VSE Disk Layouts (DOSRES, SYSWK1)," on page 291.

Customization Table

To automate installation, z/VSE uses a predefined customization table. The table relieves the operator from entering commands and responses during the installation process. This minimizes the possibility of entering wrong or incorrect information and reduces the overall time required for the initial installation of z/VSE.

A message asks the user whether ″automated installation″ is wanted. A ″NO″ response enables the operator to enter all required information manually at the console. This may be necessary if a review of the customization table shows that the predefined values and parameters defined there do not fit your system's requirements.

Installing z/VSE 3.1 via FSU

This section provides planning details for the installation of z/VSE 3.1 through a Release Upgrade via FSU (Fast Service Upgrade). An FSU is possible if you currently have a VSE/ESA 2.6.x or 2.7.x system.

Required Status of Current System

Your current system **must** have the standard (shipped) system layout of a VSE/ESA 2.6.x or 2.7.x system including the z/VSE library structure and VSE/VSAM catalogs. This is because the following libraries must be present for FSU: IJSYSRS, PRD1, and PRD2. Note that if you have combined or renamed these libraries and sublibraries, the FSU job stream will not work without modification.

Restrictions for System Modifications

You **cannot** use the FSU to modify your system. For example, you cannot:

- Perform an FSU if you are using an FBA disk as your system residence device.
- Use different disk device types from those you are using presently (for example, if you have installed your system on IBM 3380 and now want to use IBM 3390 or SCSI disk devices).
- Change your system from one language to another (English to Spanish, for example).
- Switch from the 2-digit subarea naming convention to the 4-digit subarea naming convention.

In each of these cases, you have to do an **initial installation**.

You **cannot** use the FSU to change your environment, for example from environment B to C. However, using the provided skeletons, you can perform a change of environment *after* the FSU is complete.

DB2 Server for VSE 7.4

z/VSE 3.1 includes the DB2 Server for VSE 7.4 on the Extended Base Tape.

In case of VSE/ESA 2.6.x or 2.7.x, the FSU allows the installation of the DB2 Server for VSE 7.4

This means that after an FSU from 2.6.x, the optional program *DB2 Server for VSE & VM* still exists (if it was installed before), and that (after an FSU from 2.6.x or 2.7.x) the base program *DB2 Server for VSE 7.x* is still part of your z/VSE system. The *DB2 Server for VSE 7.x* can be deleted by running the delete job provided with skeletons:

- DELDB2
- DELDB273
- DELDB274

(they also delete the *Data Propagator Relational Capture 7.1* program).

VSE/VSAM Space Considerations for an FSU

This concerns VSE/VSAM space that must be available on **DOSRES** and **SYSWK1** for the system libraries **PRD1** and **PRD2**. The DOSRES and SYSWK1 layouts are shown in Appendix B, “z/VSE Disk Layouts (DOSRES, SYSWK1),” on page 291.

PRD1 and PRD2 Libraries

Library Blocks Required for the PRD1 LIBRARY:

- To perform an FSU from VSE/ESA 2.6 to z/VSE 3.1, you require about 3000 less library blocks.
- To perform an FSU from VSE/ESA 2.7 to z/VSE 3.1, you require about 2000 less library blocks.

Library Blocks Required for the PRD2 LIBRARY:

- To perform an FSU from VSE/ESA 2.6 to z/VSE 3.1, you require about 10000 additional library blocks.
- To perform an FSU from VSE/ESA 2.7 to z/VSE 3.1, you require about 4000 additional library blocks.

You are strongly recommended to check that the space required for the additional library blocks is available in the VSE/VSAM master catalog. If there is *not* enough space available, the FSU might fail during stage 2 and it *will be difficult to recover from such an error situation*.

To find out about the VSE/VSAM space available on DOSRES and SYSWK1 of your old system, you can use one of the following methods:

- The SHOW SPACE function of the *Display or Process a Catalog, Space* dialog.
- The VSE/VSAM LISTCAT command.

Check whether the values in your VSE/VSAM user catalog are sufficient. Refer to the corresponding disk layouts in Appendix B, “z/VSE Disk Layouts (DOSRES, SYSWK1),” on page 291 for details.

Performing the FSU

After completing the above planning steps, you can select the *Fast Service Upgrade* dialog to start the FSU. Do not use the downlevel check function when upgrading to z/VSE 3.1. The FSU process includes the following steps:

1. In the prepare step, the FSU programs are loaded automatically from the z/VSE 3.1 base tape.
2. During the installation step, IBM code is replaced in two stages:
   a. During **stage 1**, the new system (library IJSYSRS) is copied from the z/VSE 3.1 base tape into library IJSYSR1 on SYSWK1. This includes the upgrading of the system history file. In addition, PRD1.MACLIB is restored. If requested, it also includes the loading of sublibrary PRD2.GEN1 if the generation feature is installed on your current z/VSE system and requested in the dialog.

      Note: For an upgrade from VSE/ESA 2.7 to z/VSE 3.1, a cold start of VSE/POWER is not required. For an upgrade from VSE/ESA 2.6 to z/VSE 3.1, a cold start of VSE/POWER **is** required. Therefore, for an upgrade from VSE/ESA 2.6 to z/VSE 3.1 a POFFLOAD will be requested at the end of stage 1.

b. An IPL from SYSWK1 starts **stage 2**.

   Sublibraries PRD1.BASE and PRD2.SCEEBASE are restored from the z/VSE 3.1 base tape. The CICS CSD file, the VSE/ICCF DTSFILE, the text repository file, and the online messages file are upgraded to the z/VSE 3.1 level. In a final step, IJSYSR1.SYSLIB on SYSWK1 is copied to DOSRES under the name IJSYSRS.SYSLIB.

   After stage 2 has finished, VTAM and CICS are started first in BASIC startup mode.

   FSU uses the existing IPL procedure (of your current z/VSE system) and its own JCL procedures for startup.

3. Because of new system characteristics and also in case of private z/VSE system modifications, tailoring is required after FSU completion. This is described under "Tailoring Tasks after the FSU."

Error Recovery

If you experience space problems during stage 2 of the FSU, you may use EXEC IDCONS to recover from the problem. Proceed as follows:

1. Change class of job PAUSEFSU to CLASS=2, for example.
2. Release job PAUSEFSU.
3. After checking for free space using LVTOC, define new VSE/VSAM space as shown in the following example:

```
F2-0002 // PAUSE
0 EXEC IDCONS
F2 0002 IDCONS400I IDCONS 1.0  CONSOLE INTERFACE TO ACCESS METHOD SERVICES

F2 0002 IDCONS402I SELECT OUTPUT DEVICE:
F2 0002 IDCONS403I 1= SYSLOG (COMPRESSED); 2= SYSLOG (NORMAL); 3=SYSLST
F2 0002 IDCONS405D REPLY 1, 2 OR 3

F2-0002
2 1
F2 0002 IDCONS406D ENTER IDCAMS COMMAND OR QUIT TO EXIT
2 DEFINE SPACE ( TRACKS (xxxx) ORIGIN (xxxxx) VOLUME(SYSWK1)) -
F2-0002 IDCONS409D CONTINUE IDCAMS COMMAND OR QUIT TO EXIT
2      CATALOG(VSESP.USER.CATALOG)
F2 0002 IDCAMS SYSTEM SERVICES TIME: 09:07:52 09/28/2000 PAGE 12
F2 0002
F2 0002  DEFINE SPACE ( TRACKS (xxxx) ORIGIN (xxxxx) VOLUME(DOSRES)) -
F2 0002  CATALOG(VSAM.MASTER.CATALOG)
F2 0002 IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
F2-0002 IDCONS406D ENTER IDCAMS COMMAND OR QUIT TO EXIT
```

Tailoring Tasks after the FSU

The IPL and JCL startup procedures as well as the LIBDEF procedures of z/VSE 3.1 include changes compared to your old system. These changes are **not** "transferred" by the FSU which uses the resources of your old system and performs updates only as far as required for a successful BASIC startup.

Therefore ...

- Use the *Tailor IPL Procedure* dialog and skeletons for the JCL startup procedures (such as SKALLOCx, SKJCL0, SKJCL1, SKUSERBG and so on) to get the z/VSE 3.1 system changes established for your installation.
- At the same time add your own private modifications you had on your old system. The manual *z/VSE Administration* describes the dialog and the skeletons in detail.
- Do not change the selected environment.
- PASIZE has changed, therefore use the dialog *Tailor IPL* to adjust the PASIZE.

You can do this after the FSU has finished and you have performed an IPL from DOSRES.

Further tasks to be performed after the FSU:

- Delete all CICS tables and re-compile them with CICS TS 1.1.1.

  If your system includes an old-level DFHSIT, CICS TS will not come up and you must rename or delete the old DFHSIT.
- Update the startup job for the CICS TS using skeletons SKCICS and SKCICS2.
- Update system startup using skeleton SKJCL0 to add the LE/VSE load list.
- Update VSE/POWER partition definitions by using skeleton SKPWSTRT.
- Use the *Maintain Dynamic Partitions* dialog to add the new dynamic classes R and S to the system.
- Make sure the following entries are included in procedure SETSDL.PROC:

```
EZASOH03,SVA     CICS LISTENER
EZASOHTR,SVA     CICS LISTENER
IKQVEX01,SVA     VSAM EXIT
```

- Make sure the following entry is deleted from procedure SETSDL.PROC:

```
$IJBIXFP,SVA     SNAPSHOT
```

- Activate the SVA load lists $SVACEE and $SVAEDCM by submitting skeleton SKJCL0.
- Add libraries PRD2.SCEEBASE and PRD2.CONFIG to the // LIBDEF statement in procedure **LIBSDL.PROC** and library PRD2.SCEEBASD (and PRD2.SCEEBASE) to the // LIBDEF statement in procedure **LIBSDLS.PROC**. Use LIBRP/LIBRC for updating the procedures.

  Perform these steps at the end of stage 2.
- Update the LE/VSE definitions in the CICS/VSE CSD file (if CICS/VSE has a separate CSD file).
- Because of a release change, the VSE/POWER phase of your old system does not run on a z/VSE 3.1 system. Therefore, a user generated VSE/POWER phase must be regenerated on z/VSE 3.1. The name of the phase is IPWPOWER if you did not choose your own name.

  FSU will use for startup the IBM-provided IPWPOWER phase.

Note that the FSU uses sublibrary PRD2.SAVE as save library. Therefore, you should not store any phases in it except for FCBs and UCBs.

Installing the DB2 Server for VSE

Refer to Chapter 10, “z/VSE e-business Connectors and Tools,” on page 139 for details on installing the *DB2 Server for VSE*, either during initial installation or after an FSU.

Installing z/VSE Optional Programs

You install optional programs after initial installation or FSU by using dialogs. It is recommended to use the dialogs since they support selective installation from tape. z/VSE provides dialogs for installing optional programs that are shipped in either the Version 2 (V2) or the Version 1 (V1) format. Most IBM programs you can order for z/VSE are V2 programs. Such programs are shipped in the library format introduced with VSE/Advanced Functions Version 2. **All z/VSE optional programs are distributed in the V2 format.**

For installation, select first the *Install Programs - V2 Format* dialog and then the *Install Program(s) from Tape* dialog. Refer to the *z/VSE Installation* manual for further details.

Further Installation and Customization Tasks

Installation Tasks

Installing z/VSE under VM

Basically, an initial installation of z/VSE on a VM system works just like a native installation. Before installing z/VSE, you must first prepare the VM host as described in Chapter 8, “Running z/VSE Under VM,” on page 119.

Installing the Generation Feature

The z/VSE distribution tape or cartridge contains source code which provides generation capability for the supervisor. The installation of this code is optional. To install the source code, z/VSE provides the *Install Generation Feature* dialog. With the installed source code you can regenerate the supervisor with changed values for the supervisor options.

Customization Tasks

Once you have installed z/VSE, a number of customization tasks may be required to tailor z/VSE to your specific needs. For example:

Adding a Second CICS Transaction Server

z/VSE offers a second predefined CICS Transaction Server. It is designed for use as a “production” system, providing an adequate base for running your applications. The primary CICS Transaction Server offers Interactive Interface support through VSE/ICCF, which the second CICS Transaction Server does not have.

Refer to “Installing a Second CICS Transaction Server” on page 200 for details.

Creating a CICS Coexistence Environment

Refer to Chapter 16, “Setting Up Your CICS Environment,” on page 195 for details.

VTAM 4.2 Storage Requirements

VTAM 4.2, which is part of z/VSE Version 3, has specific storage requirements as discussed below.

SVA Size

The size of the SVA as shipped is:

```
SVA PSIZE=(652K,6M),SDL=700,GETVIS=(768K,6M)
```

This value must be increased if user programs or vendor programs are additionally loaded into the SVA. The above values cover all base programs including the High Level Assembler, REXX, CICS, VTAM, TCP/IP, and LE/VSE where about 1 MB is left in the SVA (31-bit) area for user purposes.

Data Space Sizes

1. General Requirements

   VTAM requires 1 MB for initialization and in addition 1 MB for each partition running VTAM applications. This would mean, for example, if VSE/POWER with PNET and a VTAM line is used:

   ```
   1 MB for VTAM, 1 MB for VSE/POWER, and 1 MB for CICS
   ```

   That is, in an environment with VSE/POWER and PNET, 3 MB are needed as minimum. Without VSE/POWER and PNET, 2 MB is the minimum since CICS always needs VTAM. The first data space (1 MB) is used by VTAM itself, the second data space is created when the ACB for CICS is opened. If there are VTAM user applications active, for each partition running such applications, 1 MB has to be added.
2. Maximum Value for VTAM Data Spaces

   There is a maximum size for VTAM data spaces which may be specified either through the DFSIZE operand of the SYSDEF statement or by the DSPACE parameter of the EXEC job control statement. VTAM expands data spaces in 1 MB increments up to this specified limit.

   The DFSIZE is a global value for all VTAM applications (but only VTAM applications), private usage of data spaces is not affected. The value specified in the EXEC JCL statement is only for the related application and it overrides the DFSIZE specification. If DFSIZE is smaller than 1 MB, VTAM will take 1 MB.

   The IBM provided settings are: VTAM, CICS and VSE/POWER have a DSPACE parameter of 2 MB. DFSIZE is not used. The maximum of 2 MB will be sufficient in most cases; if many different request unit sizes (RU sizes) are used, we recommend to change the maximum size to 3 MB.

   The DSPACE parameter in the VTAM startup job limits the size for the VTAM data space that supports the VTAM Control Point application. It does not set a maximum for data spaces that VTAM gets for the support of other applications. It is the maximum value of data space VTAM can obtain for itself from the z/VSE system data space pool. The DSPACE parameter in an application startup job limits the value of data space which can be used by VTAM to support this specific application.
3. How VTAM uses Data Spaces

   The default size allocated is 1 MB for each partition. From the first 1 MB data space, 0.25 MB is used for data space management, the remainder is also divided into 0.25 MB segments and is reclaimed as needed. There are two major considerations when trying to estimate the size needed for VTAM data spaces:

a. The range of sizes of the request units (RU) being received by applications in the partition.
b. The number of RUs that have to be queued because the application does not have RECEIVE RPLs issued to read incoming RUs.

There are 10 possible different RU size groups. Each group will obtain a 0.25 MB segment when the first RU of this size is queued. The groups have the following size limits (in hexadecimal):

GROUP	1	2	3	4	5	6	7	8	9	10
SIZE (upper limit)	78	F8	1F8	3F8	7F8	FF8	1FF8	7DF8	FBF8	10108

3 MB of data space would allow to have RUs of all 10 groups (0.25 MB would be left). A data space of 2 MB would allow 7 groups, a 1 MB data space could only have 3 different groups. If one 0.25 MB segment is filled up with RUs not being processed (which means the RUs are not received), the related segment will be extended by another 0.25 MB segment if further RUs need to be stored for this group. The first 0.25 MB segment reclaimed is not freed till VTAM is shut down; additionally used segments for a certain group are freed if no longer used. This means that the maximum size of the data space needed for VTAM and its applications varies and VTAM extends dynamically.

4. Detailed Calculation

```
Minimum size used = 1 + N1 + N2 + N3       (MB)

       where N1  is the number of VTAM applications using
                 RUs with sizes of 3 different groups

             N2  is the number of VTAM applications using
                 RUs with sizes of 4 to 7 different groups

             N3  is the number of VTAM applications using
                 RUs with sizes of 7 to 10 different groups
```

The above minimum size is the minimum value that should be specified in the DSIZE operand of the SYSDEF command.

5. How to Verify the Settings

The total amount of storage available for data spaces is limited through the DSIZE parameter of the SYSDEF command. The predefined environments A (and C) have a DSIZE value of 12 MB, predefined environment B has one of 20 MB.

In case data compression is used, 2 MB of data space are needed additionally. This data space may extend by another 2 MB if needed. The DSIZE parameter should be increased accordingly if data compression is used.

The value of DSIZE should be the sum of all VTAM related data spaces, virtual disks, and private data spaces used in the system. We recommend to start with a higher DSIZE value than calculated and adjust the value according to the space actually used.

a. The QUERY DSPACE command will show the usage of data spaces:

```
                DSIZE    MAX  PARTMAX    COMMAX  VDISK  DFSIZE

DEFINED:        8192K    256       16         5      1    960K
```

```
ACTUAL:         3072K      3        3         0      0

AREA DSPS  AREA DSPS  AREA DSPS  AREA DSPS AREA DSPS AREA DSPS
F3    3
```

ACTUAL shows the amount of space used (3072K), the last line shows how many data spaces are used by VTAM (F3).

b. The QUERY DSPACE,F3 command will show more details about the data spaces of VTAM:

```
AREA DSPNAME      SIZE  MAXSIZE  SCOPE  OWNER DU-AL PASN-AL

 F2  ISTA95B8    1024K    2048K  ALL      F3           X

 F3  IST9A785    1024K    2048K  ALL      F3           X
 F3  ISTFB726    1024K    2048K  ALL      F3           X
 F3  ISTA95B8    1024K    2048K  ALL      F3           X

 F4  ISTFB726    1024K    2048K  ALL      F3           X
```

6. Changing the Data Space Size

   If the maximum value of the data space size used by VTAM needs to be changed, the related EXEC statement has to be modified:

   ```
   // EXEC ISTINCVT,SIZE=ISTINCVT,PARM='CUSTNO=...',DSPACE=2M
   ```

   This change is in VTAMSTRT - use skeleton SKVTAM. For the other VTAM applications proceed accordingly.

   If the global value is to be used, change the SYSDEF command in the ALLOC procedure using the related skeleton SKALLOCx where x is the environment character A, B, or C.

   Note: Do not add the SYSDEF command to your IPL procedure, the subsequently called ALLOC procedure will reset such a specification. If you need it in the IPL procedure, for example if you use a virtual disk for the label area, be sure that you have the same values specified in the ALLOC procedure or that you have deleted the SYSDEF command there.

Verifying Storage Allocations and Buffer Definitions

It is recommended to check storage allocations and buffer definitions and adjust them if necessary. The following commands are useful for finding out the current system values.

1. System Information

   ```
   GETVIS F3
   ```

   Shows the usage of the 24-Bit and 31-Bit GETVIS area in the VTAM partition F3.

   ```
   MAP F3
   ```

   Displays allocation information about VTAM partition F3.

2. VTAM Information

   ```
   D NET,BFRUSE
   ```

   Displays information about all the buffers and the usage of the 24-Bit and 31-Bit system GETVIS area.

   ```
   D NET,VTAMOPTS
   ```

   Displays information about all startup options.

```
D NET,STATS,TYPE=VTAM
```

Displays information about the VTAM network.

```
D NET,STORUSE,DSPNAME=*
```

Displays information about all data spaces accessed by VTAM.

```
D NET,STORUSE,APPL=appl-name
```

Displays information about the data space accessed by VTAM for the application identified by appl-name.

```
D NET,STORUSE,POOL=*
```

Displays usage information about all VTAM storage pools.

Chapter 6. Migrating From Earlier Releases

Basically, migration means for z/VSE users transferring **hardware configuration** and **user profile** data stored in the VSE/ICCF DTSFILE and the z/VSE control file to the new system.

In addition, there are various migration tasks to be done depending on your system setup and the z/VSE version and release you are currently using. The following sections provide an overview.

Ways of Performing Migration

Existing VSE system information can be migrated to z/VSE 3.1 as follows:

1. Migrating during initial installation

   In this way, you can migrate hardware information as well as VSE.CONTROL.FILE and VSE/ICCF DTSFILE information. This means that you need as input for initial installation a copy of the DTSFILE and the control file of your **old system** on separate tape volumes. Refer to "Migrating from VSE/ESA 2.x to z/VSE 3.1" on page 92 for an outline of the tasks involved.

   From z/VSE 3.1 onwards, some hardware devices are no longer supported. During the hardware migration, the related addresses will be displayed on the "unidentified service list" for further processing or deletion. For a list of supported hardware devices, refer to Appendix G, "Devices Supported," on page 319.
2. Migrating after initial installation

   This method is not recommended since it allows you to migrate VSE.CONTROL.FILE and DTSFILE information but **not** your hardware information.

z/VSE 3.1 Migration Items

This section describes the migration items resulting from z/VSE 3.1 system changes.

Important:

Consult also the *z/VSE Release Guide* for a summary of new and changed functions to identify possible migration items for your installation and refer also to "Migration Items of Previous Releases" on page 94.

Increased PASIZE

The PASIZE is the maximum value for the partition size. The High Level Assembler (HLASM) requires a larger partition. For example, when the z/VSE supervisor is generated a partition size of at least 60 MB is required. For further details, see "Supervisor Generation Example" on page 278.

The increased PASIZE is not active after an FSU, you must increase the PASIZE *after* you have performed an FSU.

VSE/POWER 7.1

VSE/POWER 7.1 (as part of z/VSE 3.1) cannot coexist with previous levels of VSE/POWER in a shared spooling environment. All systems must be of the same level.

High Level Assembler (HLASM) Release 1.5

The HLASM (High Level Assembler) Release 1.5 no longer uses workfiles during the assembly process. Instead, partition GETVIS storage is used. As a result, the partition allocation must be larger than it was for previous releases. The minimum allocation size of the partition depends upon either the:
- Size of the assembler source.
- Number of source statements generated (where macros are used).

During initial installation and when the transaction security dialog is used, HLASM runs in BG. In addition, most of the CICS-related skeletons use HLASM (all are setup to run in BG).

The partition BG has at least 6 MB allocated in the shipped environments A, B, and C. This value allows you to use HLASM to run all installation assemblies and skeletons, except for the Supervisor-generation as provided in the skeleton SKSUPASM.

Another routine you should consider is the ASSEMBLE routine in ICCF. This routine should run in Class I, which is the largest ICCF pseudo partition. However, this 1 MB class is not sufficient to assemble programs of about 2000 or more source statements. To assemble such programs, you must use a larger ICCF pseudo partition.

Migrating from VSE/ESA 2.x to z/VSE 3.1

For migration **during initial installation** you need:
- A backup copy on tape of your current VSE/ICCF DTSFILE created with the VSE/ICCF utility program DTSUTIL. It is recommended to use the *Backup/Restore Library Objects* dialog. The volume serial number of the tape must be ICCF01.
- A VSE/VSAM backup copy on tape (must be of the same device type as used for initial installation) of your current VSE.CONTROL.FILE created with the REPRO command (which **must** be used). The following should be specified:
  - RECORD FORMAT=variable-length blocked records
  - BLOCK SIZE=4000

  Following is a job stream with the *VSE/VSAM REPRO* command. You can create a similar job stream by using the *Display and Process a File* dialog and select option 4 (COPY).

```
* $$ JOB JNM=COPYCNTL,CLASS=A,DISP=D,NTFY=YES
// JOB  COPYCNTL COPY 'VSE.CONTROL.FILE'
// DLBL COPYIN,'VSE.CONTROL.FILE',,VSAM,                        X
               CAT=VSESPUC
// TLBL COPYOUT,'REPRO.CNTRL.FILE',,CF0001
// ASSGN SYS005,cuu
// EXEC IDCAMS,SIZE=AUTO
 REPRO INFILE (COPYIN) -
       OUTFILE (COPYOUT -
       ENVIRONMENT (BLOCKSIZE (4000 ) -
                   RECORDFORMAT (VARBLK) -
                   STDLABEL -
```

```
                    PRIMEDATADEVICE (2400) REW)) -
        NOREUSE
/*
/&
* $$ EOJ
```

In the job stream example, the volume serial number is CF0001 which must be used.

The tapes you use for backup must be of the same type as the distribution tape(s) you received. Refer to "Backing Up and Restoring Data" in the manual *z/VSE Operation* for information on how to create backup copies.

You can migrate your

hardware configuration and

your user profiles (up to 199 VSE/ICCF users and their library names)

by answering 'YES' to the migration question displayed on the system console during initial installation. Keep in mind that the migration performed during initial installation is a *predefined migration*. If you want to migrate **after initial installation**, you can use the migration utility IESBLDUP introduced under Migration Utilities to migrate the control file information.

After migration you must do the following:

- Assemble and catalog the migrated hardware tables. You do this by invoking the *Hardware Configuration* dialog in your newly installed system. Devices which are no longer supported by the z/VSE dialogs or which are not identified uniquely are displayed on the *Unidentified Device List* panel. Verify the migrated tables and execute the function by pressing PF5 (PROCESS).
- Validate and process the migrated VTAM startup options and VTAM APPLIDs. You do this by invoking the dialogs *Maintain Startup Options* and *Maintain VTAM Application Names*.

Migration Utilities

z/VSE provides a number of migration utilities to help you perform migration:

- The utility program *IESBLDUP* to migrate user profile information to the new system. The utility can be used, for example, if you did not migrate this information during initial installation.

  "Overview of IESBLDUP" below summarizes the functions of the utility program.
- Dialogs of the Interactive Interface to back up and restore libraries and VSE/VSAM files. Refer to the manual *z/VSE Operation* for details.
- VSE/POWER's POFFLOAD facility to save and later reload jobs and output in the VSE/POWER queues. Refer to "POFFLOAD: Saving Queue Entries on Tape and Restoring Them" in the manual *VSE/POWER Administration and Operation* for details.

Overview of IESBLDUP

The utility program IESBLDUP helps z/VSE users migrate old VSE control file data such as user profiles, application profiles, and selection panels (but not VSE/ICCF user libraries) to the new system.

To do this, you must first create copies of the following resources on your old system:

- A copy of the VSE control file.

- A copy of the VSE/ICCF DTSFILE.

IESBLDUP uses these files as input and updates the user profile information of the new VSE control file. In addition, it creates the job *DTRMIGR*, and places DTRMIGR into the VSE/POWER reader queue. Job DTRMIGR is automatically processed and invokes the VSE/ICCF utility DTSUTIL to:

- Add new user profiles to the new VSE/ICCF DTSFILE.
- Alter VSE/ICCF DTSFILE options that are incompatible with VSE/ESA Version 1.

IESBLDUP must run in a batch partition controlled by VSE/POWER.

Note: You can also use IESBLDUP to create a *status report* of your system's users. Such a report lists all user IDs and related data defined in the VSE control file, VSE/ICCF DTSFILE, and (optionally) the CICS/VSE Sign-on table DFHSNT. It thus provides valuable information for maintaining user profiles and data.

The manual *z/VSE Installation* provides a detailed description of the IESBLDUP program.

Migration Items of Previous Releases

You need to evaluate the items listed in this section and find out how far they may have an impact on your system when migrating from an earlier release of VSE/ESA to z/VSE 3.1.

Migrating from Releases Older than VSE/ESA 2.3!

For information about how to migrate from a VSE/ESA release that is older than VSE/ESA 2.3, refer to the edition of the *VSE/ESA Planning* manual that was supplied with VSE/ESA 2.7.

VSE/ESA 2.7 Migration Items

These are the items you should consider when migrating from VSE/ESA 2.7 to z/VSE 3.1:

- HLASM (High Level Assembler) Release 1.5 is introduced as part of z/VSE 3.1. For details, see "High Level Assembler (HLASM) Release 1.5" on page 92.
- VSE/POWER 7.1 is introduced as part of z/VSE 3.1. For details, see "VSE/POWER 7.1" on page 92.
- With z/VSE 3.1, the 9345 disk device is no longer supported. Therefore, a Fast Service Upgrade (FSU) to z/VSE 3.1 is not possible from an IBM 9345 disk.
- With z/VSE 3.1, the *IXFP/SnapShot for VSE/ESA* feature is no longer supported for the IBM 9393 RAMAC Virtual Array (RVA). Therefore, you might wish to remove the entry for *IXFP/SnapShot for VSE/ESA* from the procedure SETSDL.PROC.

VSE/ESA 2.6 Migration Items

There are no specific migration items resulting from VSE/ESA 2.6 system changes except for VSE/POWER. VSE/POWER 6.6 (as part of VSE/ESA 2.6) cannot coexist with previous levels of VSE/POWER in a shared spooling environment. All systems must be of the same level.

VSE/ESA 2.5 Migration Items

There are no specific migration items resulting from VSE/ESA 2.5 system changes except for VSE/POWER. VSE/POWER 6.5 (as part of VSE/ESA 2.5) cannot coexist with previous levels of VSE/POWER in a shared spooling environment. All systems must be of the same level.

VSE/ESA 2.4 Migration Items

The main migration item for VSE/ESA 2.4 is the switch from CICS/VSE to the CICS Transaction Server and the new security concept. The following chapters provide planning information and help you evaluate the migration effort for CICS and security:

- Chapter 19, "Security Support," on page 233.

  "Migrating TRANSEC Values from CICS/VSE" on page 238 is a typical migration task related to security.
- Chapter 16, "Setting Up Your CICS Environment," on page 195.

  "Security Migration Aid" on page 210 introduces a CICS/VSE tool for migrating CICS/VSE security data to the CICS Transaction Server.

You should also consult the migration manuals of the CICS Transaction Server:

CICS TS Migration Guide, GC33-1646

CICS TS Application Migration Aid Guide, SC33-1943

VSE/POWER 6.4 (as part of VSE/ESA 2.4) cannot coexist with previous levels of VSE/POWER in a shared spooling environment. All systems must be of the same level.

VSE/ESA 2.3 Migration Items

The following items were new with VSE/ESA 2.3:

- TCP/IP

  Refer to the *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information* for TCP/IP migration information.
- LE/VSE and VSE C Language Run-Time Support.

  For items related to migration when using the VSE C Language Run-Time Support and LE/VSE refer to "Considerations for LE/VSE and the VSE C Language Run-Time Support" on page 78.

Changing from DOS/VSE Assembler to High Level Assembler

Starting with VSE/ESA 2.1, the DOS/VSE Assembler has been replaced by the **High Level Assembler for VSE**. The implications and the characteristics of the High Level Assembler are as follows:

- It is automatically installed into system sublibrary PRD1.BASE during initial installation.
- During startup, the High Level Assembler is automatically loaded into the SVA (31-Bit) area through procedure $0JCL via load list $SVAASMA.
- The High Level Assembler is to be called with // EXEC ASMA90.... (not with // EXEC ASSEMBLY).
- The High Level Assembler cannot create E-Decks (as the DOS/VSE Assembler could) but can process existing E-Decks via a special library exit shipped as EDECKXIT. This is to support existing IBM or user programs that depend on E-Decks.

Increasingly, most macros are shipped as A-Books (not as E-Decks).

- The High Level Assembler requires more virtual storage: a minimum partition size of 6 MB is recommended. Depending on the size of your programs much more virtual storage may be needed.

  For customers running assemblies in VSE/ICCF partitions, it is recommended to consider the use of dynamic partitions instead.

Environments with no Immediate Need for the High Level Assembler

Customers who have no immediate need for advanced High Level Assembler functions (such as 31-bit addressing or data space support) do not need to change their job streams or compile procedures. z/VSE automatically converts an `// EXEC ASSEMBLY` statement into the following call for the High Level Assembler:

```
// EXEC ASMA90,SIZE=(ASMA90,64K),                                 C
        PARM='CPAT(SYSL),EX(LBX(EDECKXIT)),FOLD,SZ(MAX-200K,ABOVE),    C
        LINECOUNT(056)'
```

The same statement is created when calling the High Level Assembler via VSE/ICCF.

The statement created invokes the High Level Assembler in a way as much compatible to the former // EXEC ASSEMBLY call as possible. This is achieved mainly by the following parameters:

SIZE=(ASMA90,64K) and SZ(MAX-200K,ABOVE)
: Sets up the partition for the compile step in such a way that a maximum amount of storage will be available as GETVIS storage. This is achieved by reducing the program area of the partition to the possible minimum. The 64K are provided for the ESERV program which is called by EDECKXIT.

EX(LBX(EDECKXIT))
: Activates a library exit which temporarily translates macros from E-Deck format back into source format (A-Books) via the ESERV utility program. This enables the High Level Assembler to process such macros.

Refer to the documentation of the High Level Assembler for a detailed and complete parameter description.

Notes:

1. Programs compiled in this way cannot exploit the full range of the High Level Assembler capabilities.
2. When using CICS or RACROUTE macros in your assembler program, **do not** use for your compile run the DOS assembler option OPT(DOS).

Environments Making Full Use of the High Level Assembler

In order to make full use of High Level Assembler functions, job streams must call the High Level Assembler directly. In its simplest form the High Level Assembler can be called with:

```
// EXEC ASMA90,SIZE=ASMA90
```

If E-Decks are to be processed, the following statement is required:

```
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT))'
```

It calls the High Level Assembler and enables the processing of E-Decks (created by the DOS/VSE Assembler). As long as macros in E-Deck format are to be processed, the High Level Assembler must be activated with the library exit (EDECKXIT) which is capable of translating E-Decks back to source format (A-Books).

Note: Note that supervisor generation is dependent on functions of the High Level Assembler. Refer to "Supervisor Generation Example" on page 278 for further details.

Further Information

For further details about E-Deck processing and the library exit, refer to the manual *z/VSE Guide to System Functions* under "Using the High Level Assembler Library Exit for Processing E-Decks". For a detailed description of High Level Assembler functions and statements and how to use them, refer to the following manuals of the High Level Assembler:

Programmer's Guide
Language Reference

Compatibility of Link Information

The following information applies to customers who use in their data processing environment different VSE releases. If they create applications, they may wonder whether they need different program versions: one for VSE/ESA 1.3 or higher with AMODE/RMODE information to exploit 31-bit addressing, and one for the other VSE systems up to VSE/ESA 1.2. The answer is that **one program version is sufficient**, if the link information is built in one of the ways described below.

It is important to understand how the linkage editor in VSE/ESA 1.3 acquires the AMODE/RMODE information. As shown in the sequence below, the linkage editor retrieves it from:

1. ESD entries in the object modules, created by a compiler or assembler.
2. The PARM field of the linkage editor EXEC statement used in the link job.
3. The linkage editor MODE control statement, introduced with VSE/ESA 1.3.0 and used in the link job or link books.

Note that the sequence of this list also reflects which linkage information overrides the previous information (2. overrides 1. and 3. overrides 2.). For more information on AMODE/RMODE, linkage editor, and downward compatibility refer to the manual *VSE/ESA Extended Addressability* under "Linkage Editor and Librarian Support".

The information in the ESD entries is ignored by the linkage editor of VSE systems prior to VSE/ESA 1.3. It becomes effective only when a program is linked on VSE/ESA 1.3 or higher.

The new linkage editor MODE control statement serves to specify the desired AMODE/RMODE in the link job or link books shipped with the program, if it is not already contained in the ESD entries or if the ESD information is to be overwritten.

The MODE statement will only be effective with VSE/ESA 1.3 or higher. It will be ignored on older releases, **provided the appropriate PTFs have been applied**. Consequently, there are two possibilities to create a program installable on all VSE releases:

1. The program is distributed in object format and link-edited as it is being installed and was

- compiled, for example, with VS COBOL II or the High Level Assembler using the respective method to set AMODE/RMODE so that the information is available in the ESD entries, or it
- specifies the AMODE/RMODE information via MODE control statements in the link books.

2. The program is distributed in phase format after being link-edited on a VSE/ESA 1.3. system with the appropriate linkage information.

Note: To work correctly, a program must be either installed and linked on a new VSE system (VSE/ESA 1.3 or higher) or at least relinked there, depending on how it is distributed. If the program is copied from an older VSE system, it can only run with the default: AMODE=24, RMODE=24 since an old linkage editor removes all mode information.

Migrating User-Defined System Information

The system sublibrary (IJSYSRS.SYSLIB) of your z/VSE system may include user-defined system information. Examples of such information are:

- FCB- and UCB-image phases of your own stored by the IBM standard names.
- Label information.
- Private SVA load lists.

Any such information is lost if you simply perform **initial installation**. To avoid this, proceed as outlined below.

Note: Numbers within parentheses in front of individual statements refer to additional information under "Explanations to Sample Jobs" on page 100.

1. **Produce a backup copy of sublibrary PRD2.SAVE**

   This step assumes that you had defined and processed your FCB- and UCB-image phases, using the Interactive Interface of z/VSE as described in the *z/VSE Administration* manual. In that case, these phases were saved in the sublibrary PRD2.SAVE of your system. Perform this step on your current VSE system.

   To produce the backup, either use the *Backup/Restore Library Objects* dialog with *Backup VSE Library on Tape* or run a job similar to the sample below.

   ```
       // JOB BACKUP PRD2.SAVE SUBLIBRARY
       // MTC  REW,cuu
       // EXEC LIBR
   (1) BACKUP S=PRD2.SAVE TAPE=cuu
       /*
       /&
   ```

   This step saves all control information stored in your PRD2.SAVE library.

   If, because of a locally defined procedure, this control information is not stored in the sublibrary PRD2.SAVE, produce a backup of the system sublibrary. The corresponding BACKUP statement would be:

   ```
   BACKUP S=IJSYSRS.SYSLIB TAPE=cuu
   ```

   As concerns label information stored on the system residence volume, ensure that your original load procedures are stored in a sublibrary on a private volume.

2. **Edit your JCL ASI and label procedures**

   Do this after having performed the first system startup with the newly installed refresh release. This is a multi-step process as follows:

a. Restore the sublibrary PRD2.SAVE into a separate sublibrary. Either use the *Backup/Restore Library Objects* dialog with *Restore VSE Library from Tape* or run a job similar to the sample below.

```
    // JOB  RESTORE PRD2.SAVE
    // MTC  REW,cuu
    // EXEC LIBR,PARM='MSHP'
(2) RESTORE S=PRD2.SAVE:PRD2.SAVEOLD TAPE=cuu DATE=OLD
    /*
    /&
```

If there was a need to backup your system sublibrary, replace the librarian RESTORE command by a number of selective restore requests. Each of your restore requests would have to specify the generic name of the procedures that are to be restored. The librarian RESTORE command is described in your *z/VSE System Control Statements* manual under “RESTORE”.

b. Select VSE/ICCF command mode.

c. Punch, into your primary VSE/ICCF library, the startup procedures that you want to migrate. To do this, use the LIBRP command as shown:

```
LIBRP PRD2.SAVEOLD vsemembername.PROC iccfmembername
```

If you have used the default names for your z/VSE startup procedures, you may have to issue such a request for each of the following procedures:

ALLOC
If you changed the partition allocations.

Note that with VSE/ESA 1.3 and 2.1 changes of the procedure names were introduced. Refer to Table 12 on page 66 for the current default procedure names used.

$0JCL
If you changed the startup procedure for BG.

$1JCL
If you changed the startup procedure for the VSE/POWER partition.

POWSTRT
If you changed the VSE/POWER autostart procedure.

Any of $2JCL through $BJCL
If you changed the startup procedures for partition F2 through FB.

STDLABEL and STDLABUP
Get the new label procedures and edit them to get your own label definitions into the procedures.

Use the corresponding names of your own choice if you have not used the IBM standard names.

d. Ensure that IBM supplied specifications relating to new support (if any) will be included in the procedures that are to be migrated.

e. Verify your changes – on a test system, if possible.

Perform step 3 on a test system rather than on your system for normal operation and subsequently start up the test system.

f. Return the edited procedures to the sublibrary from which they were punched into your VSE/ICCF library.

Use LIBRC macro requests to accomplish this. The macro works in a way similar to the LIBRP macro but in reverse direction.

3. **Restore the procedures to the system sublibrary**

Copy the edited procedures to the system sublibrary. Use a job similar to the one shown below:

```
    // JOB COPY PROCEDURES
    // EXEC LIBR,PARM='MSHP'
    CONNECT S=PRD2.SAVEOLD:IJSYSRS.SYSLIB
(3) COPY vsemembername.PROC REPLACE=YES DATE=OLD
    /*
    /&
```

4. **Restore the FCB and UCB-image phases**

   If you use the interactive interface, do an administrator fast path selection of **3722** and provide the required control information.

   If you do not use the interactive interface, run a job similar to the one below:

```
    // JOB RESTORE FCB PHASES
    // MTC  REW,cuu
    // EXEC LIBR,PARM='MSHP'
(4) RESTORE PRD2.SAVE.$$BFCB*.PHASE : IJSYSRS.SYSLIB -
            TAPE=cuu REPLACE=YES DATE=OLD
(5) RESTORE PRD2.SAVE.$$BUCB*.PHASE : IJSYSRS.SYSLIB -
            TAPE=cuu REPLACE=YES DATE=OLD
    /*
    /&
```

5. **Make the changes effective**

   IPL the modified system to make the changes effective.

Explanations to Sample Jobs

(1) Replace the tape-unit address (TAPE=cuu) by the address of the tape drive on which you mounted the backup tape. Use an unlabeled scratch tape.

(2) You may replace PRD2.SAVEOLD by the qualified name of an existing sublibrary of your own choice to temporarily store the information you want to migrate.

Replace the tape-unit address by the address of the tape drive on which you mounted the backup tape.

(3) You must submit one such statement for each distinct (generic) member name.

(4) This librarian step restores the previously saved FCB-image phases. The IBM supplied FCB-image phases are overwritten. In the command, replace the tape unit address by the address of the tape drive on which you mounted the backup tape.

(5) This step is not required if your printer automatically loads the correct image into its UCB from internal storage, such as does the IBM 4248, for example.

z/VSE-Supplied CICS CSD TYPETERM Definitions

The Interactive Interface dialogs of VSE/ESA 1.1 or higher store the terminal definitions for VTAM users in the **CICS System Definition (CSD)** file.

There is no TCT support available with the CICS Transaction Server. A CICS terminal definition in the CSD consists of two parts:

The type terminal definition (DEFINE TYPETERM).

The terminal definition (DEFINE TERMINAL).

One CICS parameter table has **two** corresponding CSD TYPETERM definitions. One is for SNA, the other one for non-SNA definitions. Table 19 on page 101 lists the VSE/SP Version 4 CICS table names and the migrated z/VSE CSD TYPETERM definitions provided as models.

Table 19. Mapping of CICS Parameter Table Names to CSD TYPETERM Definitions

Table name	CSD non-SNA	CSD SNA	Remarks
D3178	VSE32782	VSELU2A	DSCRS = 24,80 PAGE (AUDALARM)
D3179	VSE32792	VSELU2E	(COLOR EXTDS HILIGHT)
D3277	VSE3277	VSELU2A	DPGES = 24,80 PAGE (AUDALARM)
D32782	VSE32782	VSELU2A	DSCRS = 24,80 PAGE (AUDALARM)
D32783	VSE32783	VSELU2B	ASCRS = 32,80
D32784	VSE32784	VSELU2C	ASCRS = 43,80
D32785	VSE32785	VSELU2D	ASCRS = 27,132
D3279S2A	VSE32782	VSELU2A	-
D3279S2B	VSE32792	VSELU2E	(COLOR EXTDS HILIGHT)
D3279S3A	VSE32783	VSELU2B	ASCRS = 32,80
D3279S3B	VSE32793	VSELU2F	ASCRS = 32,80 (COLOR EXTDS HILIGHT)
D3279S3G	VSE3279G	VSELU2G	ASCRS = 32,80 (COLOR EXTDS HILIGHT PS)
D32792X	VSE32782	VSELU2A	-
D32793X	VSE32783	VSELU2B	ASCRS = 32,80
D3290	VSE3290	VSELU2I	ASCRS = 62,160
D5550	VSE5555	VSELU2H	(COLOR EXTDS HILIGHT PS KATAKANA SOSI OUTLINE)
P3262	VSEDSCP	VSELU3	DPGES = 24,80 AUTOPAGE
P3268	VSEDSCP	VSESCS	—
P3287	VSEDSCP	VSELU3	—
P3287C	VSEDSCP	VSESCS	—
P3289	VSEDSCP	VSELU3	—
P5210	VSEDSCP	VSESCS	—
P555X	VSE3278	VSELU2	(PS OUTLINE SOSI)

User-Defined and Modified CICS Parameter Tables

For user-defined and modified CICS parameter tables the z/VSE migration program defines new TYPETERM definitions (during initial installation). These definitions are stored into a CSD group with the group name VSETYPE1. The group VSETYPE1 will be exclusively used by z/VSE.

TYPETERM definitions in the CICS Transaction Server and in the coexistence environment are identical.

The naming convention for user defined and modified CICS TYPETERM definitions, will be as follows:

The new TYPETERM name is **VSEDIxxx** for displays and **VSEPRxxx** for printers (with **xxx=001-999)**. The number is incremented by 1 for each new generated name.

If you have used, by any chance, the same TYPETERM names for your own TYPETERMs, you have to *manually* rename your TYPETERM names.

Migrating Device Definitions

When you log on with the user ID POST and have selected *migration* during initial installation, your previously made hardware configurations are checked by the migration program.

Depending on how many devices you had defined in your previous system, this step may take some time. Do **not** interrupt this process after the message has been displayed:

```
INPUT ACCEPTED.  PLEASE WAIT.
```

For devices that are no longer supported under z/VSE you get an *Unidentified Device List* panel. In this panel, the unsupported device addresses are displayed with question marks. If you get this panel, redefine or delete the devices which show question marks. The following message also appears on the *Unidentified Device List* panel:

```
JOB HAS BEEN SUBMITTED AND FILED AS CSDMIGR
```

This job is stored in VSE/ICCF library 10.

For information about how to continue, refer to “Completing Initial Installation” in the manual *z/VSE Installation*.

To migrate your previously configured hardware, you must at least once enter the *Hardware Configuration* dialog. Select fast path 241 to get to the *Unit Address List* panel. The system may display another unidentified device list for devices which are not defined uniquely. You can define or delete them using the dialog. Once you press **PF5** (PROCESS), your hardware specifications are stored in the CICS CSD File.

Chapter 7. Files and Libraries

This chapter contains these main sections:

Standard Label Procedures

Labels and the information they contain serve to identify and describe files stored on disk devices, or, less frequently, stored on tape. In this context, a library is considered to be a file. In a job stream, label information is provided through DLBL and EXTENT statements for disk devices, and files on tape are identified through a TLBL statement. The following manuals provide background information on file and label processing:

z/VSE System Macros User's Guide

z/VSE Guide to System Functions

z/VSE System Control Statements

As outlined below, z/VSE uses three standard procedures for maintaining label information in the system's label information area. The area itself is divided into the subareas STDLABEL and PARSTD. STDLABEL identifies the **system** standard subarea and PARSTD the **partition** standard subarea. Both, STDLABEL and PARSTD are parameters of the JCL OPTION statement.

z/VSE provides and uses the following standard label procedures:

- **STDLABEL.PROC**

  This procedure is automatically created during initial installation. It is based on the disk device type (of DOSRES and SYSWK1) you use for installation, and it contains labels for all **non-VSE/VSAM** system files.

  The startup procedure for the BG partition ($0JCL) executes STDLABEL.PROC. You should modify STDLABEL.PROC only if you extend a system file, for example.

- **STDLABUP.PROC**

  This procedure is used for labels of **VSE/VSAM** system files only. These are labels for files that:

  – Are created automatically during installation.

  – You define using the Interactive Interface.

  When you use the dialogs for defining or deleting VSE/VSAM files, the jobs created by the dialog automatically update the STDLABUP procedure.

  Note: If you define VSE/VSAM files without using the Interactive Interface (but through editing job streams), you should not use EXTENT statements. The reason is that all these files are deleted from STDLABUP.PROC when you use the Interactive Interface for file creation next time.

  The STDLABEL procedure executes STDLABUP.PROC.

- **STDLABUS.PROC**

  z/VSE provides the STDLABUS skeleton in VSE/ICCF library 59. You can use this skeleton to include standard labels for non-VSE/VSAM user files.

  The STDLABEL procedure executes the STDLABUS.PROC.

  z/VSE initially provides a dummy STDLABUS procedure in IJSYSRS. “Creating Standard Labels for Non-VSE/VSAM Files” in the manual *z/VSE Administration* describes how to use the supplied skeleton (STDLABUS) to create labels for non-VSE/VSAM files.

System Files

System files reside on the system disks DOSRES and SYSWK1 and are discussed on the following pages.

Appendix B, “z/VSE Disk Layouts (DOSRES, SYSWK1),” on page 291 shows the layouts of DOSRES and SYSWK1 for all the disk device types supported for initial installation.

System File Assignments

z/VSE supplies procedures which contain the assignments for key system files. These are automatically invoked by the startup job stream and by other z/VSE job streams. You should change them only if you move the corresponding files.

Table 20. Procedures for System File Assignments

PROCEDURE:	ASSIGNMENT:	FOR:
DTRICCF.PROC	SYS010	VSE/ICCF DTSFILE
DTRPOWR.PROC	SYS000 to SYS002	VSE/POWER
DTRINFOA.PROC	SYS016 to SYS017	Info/Analysis work files
DTRCICST.PROC	SYS018	CICS SD files
	SYS001 to SYS002	Work files for DTSANALS to recover DTSFILE

Requirements for VSE/POWER Files

VSE/POWER Queue File

The disk space requirements for the VSE/POWER queue file on DOSRES (VSE.POWER.QUEUE.FILE) are directly related to the number of queue file entries you plan to hold at one time. For an extension of the queue file, free space is provided right after the queue file location on DOSRES.

Refer to “Estimating Disk Space for the VSE/POWER Spool Files” in the *VSE/POWER Administration and Operation* manual for more information.

In storage, the VSE/POWER queue file resides in the partition GETVIS area. This must be reflected in the value specified for the partition size.

VSE/POWER Queue File Considerations

Starting with VSE/ESA 2.4, the storage copy of the Queue File is placed into the VSE/POWER partition GETVIS-31 storage, as soon as the partition ALLOC provides GETVIS space beyond the 16MB line. This provides storage relief for the GETVIS-24 area, where VSE/POWER control blocks and data blocks (DBLK) are acquired per task. The conditional PDISPLAY STATUS line, reading

```
QUEUE FILE STOR. COPY, PART IN PART. GETVIS-31     nnn K-BYTES
```

informs which part of the queue file copy stretches beyond the 16MB line. Vendor products, for example, that do NOT address queue records by standard VSE/POWER interfaces (but by their own direct access to the GETVIS storage copy of the queue file), must be prepared for AMODE-31 addressing. For debugging purposes the following autostart statement

```
SET GETVQFL=24
```

can be used. The statement causes VSE/POWER to request Queue File storage below the 16MB line, although GETVIS-31 space has been allocated.

VSE/POWER Data File

Table 21 shows possible allocations for the VSE/POWER data file on SYSWK1 (VSE.POWER.DATA.FILE). These values are shown for the disk devices supported for initial installation. The DBLK (data block) size corresponds to the physical record size of the data file on disk. The DBLK influences the VSE/POWER spooling performance and is a parameter of the VSE/POWER generation macro (refer also to "Regenerating VSE/POWER" on page 279).

Starting with VSE/ESA 2.6, it is possible to extend the data file during a WARM start of VSE/POWER. Refer to the *z/VSE Release Guide* for further details.

Table 21. Examples of DBLK Sizes and Default DBLK

Device Type	DBLK (Sizes in Bytes)	DBLK/ Track	Bytes/ Track Used	Minimum 80-Cards /DBLK	Minimum 132-Lines /DBLK	Default DBLK used by VSE/POWER (in Bytes)
3380	3860	11	42460	44	28	7476
	4276	10	42760	49	31	
	6356	7	44492	73	45	
	7476	6	44856	85	54	
	11476	4	45904	131	82	
	15476	3	46428	176	111	
	23476	2	46952	267	168	
3390	4566	11	50226	52	32	7548
	5726	9	51534	65	41	
	7548	7	52836	86	54	
	10796	5	53980	123	78	
	13682	4	54728	156	98	
	18452	3	55356	210	132	
	27998	2	55886	319	200	
SCSI (FBA)	4096	---	---	46	29	7680
	7680	---	---	87	54	
	15360	---	---	174	109	
	23040	---	---	261	164	
	30720	---	---	349	219	

The default value of DBLK as used in the pregenerated IPWPOWER phase, is about 7500 for the common disk device types. You can use the DBLK operand of the VSE/POWER generation macro, or the SET DBLK statement in the VSE/POWER startup procedure to modify the DBLK value which is to be activated with a COLD startup.

The maximum DBLK value for CKD and FBA disk devices is 65024 bytes.

Notes:

1. If you use shared spooling, consider the following:
   - Place the queue and data files on separate shared disks and not on the same actuator as the lock file.
   - Define multiple extent data files on multiple actuators.
2. The manual *z/VSE Administration* under "Extending VSE/POWER Files" explains how to extend the space reserved for the VSE/POWER files. Only do this, however, when absolutely necessary. Incorrect specifications will most likely cause startup problems.
3. The optimal performance is reached if you use the highest allocation values for the DBLK size, as shown in Table 21 on page 105.

VSE/ICCF DTSFILE

The VSE/ICCF DTSFILE includes the VSE/ICCF libraries (file ICCF.LIBRARY on SYSWK1). Refer to "VSE/ICCF Libraries" on page 116 for further details about these libraries.

System Work Files

Note that it is possible to place workfiles on a virtual disk. Refer to "Defining Work Files on a Virtual Disk" on page 109 for details.

The following is important for system work files:

- All work files are allocated in VSAM-managed space including, for example:
  - The VSE/ICCF work files IKSYS11 to IKSYS54.
  - A standard SORT work file (SORTWK1) for which you should ensure that the allocation is sufficient for your sort requirements.
  - The work files IJSYS01 to IJSYS07 for use by compilers.
- All work file labels are part of the system standard labels (procedure STDLABUP.PROC). Refer also to "Standard Label Procedures" on page 103.

Notes:

1. Files defined implicitly by the system, such as SORTWK1, cannot be redefined (EXTENTs) through the Interactive Interface. They can be changed only by editing the job stream.
2. If system work files are defined on virtual disk, you must also define a user catalog for them on the same virtual disk.
3. High Level Assembler (HLASM) no longer uses workfiles, but instead uses the partition storage as working storage in which to assemble your programs. Therefore, you must ensure that the partition in which you wish to run the ASMA90 program is large enough to assemble your programs.

Control File

The control file (VSE.CONTROL.FILE) is the central repository file for system access information for the Interactive Interface and CICS TS. It contains records such as:

- User profile records
- Selection panel records
- Application profile records
- Synonym records
- News records (messages displayed to users after they sign on)

The user profile information provides coordination between CICS/VSE, VSE/ICCF, and the Interactive Interface.

Shareoption 4, which allows multiple write access, is required.

The control file is also accessed by the Basic Security Manager (BSM). Only one control file can be part of the system.

Text Repository File

The text repository file (VSE.TEXT.REPSTORY.FILE) contains Interactive Interface information such as HELP text and messages which the Interactive Interface dialogs display. You can redefine this system file using skeleton SKRSTRFL.

Online Messages File

The online messages file (VSE.MESSAGES.ONLINE) must be in the VSE/VSAM master catalog. It contains console messages which may be issued by z/VSE component programs. Via the *System Console* dialog, you can request an explanation of a console message online at your display station. You can redefine this system file using skeleton SKOMERST.

Online Problem Determination (OLPD) File

The OLPD file (VSE.ONLINE.PROB.DET.FILE) contains information about CICS/VSE transaction abends. Abend information is collected and stored in this file. You can use the *Online Problem Determination* dialog to view the abend information. Each CICS/VSE subsystem has its own OLPD file. An OLPD file has shareoption 2 (not shareable and single write access).

You can redefine the OLPD file by using as an example the related job step of source member VSAMDEFS.Z.

Message Routing File

The messages routing file (VSE.MESSAGE.ROUTING.FILE) supports the exchange of messages between the system and its Interactive Interface users. Messages sent by the system to a user are stored in this file. They can be retrieved by the user after notification.

When a user retrieves a message, it is deleted from the file. If a user is not signed on, the system keeps any messages for that user in the file until a sign on is performed. To avoid a possible overflow, the administrator may need to sign on for a user who is not working with the system.

You can redefine this system file by using as an example the related job steps in source members VSAMDEFS.Z and VSAMINIT.Z.

Other System Files

The remaining predefined system files on DOSRES and SYSWK1 are listed below.

Non-VSE/VSAM System Files

```
DOS.LABEL.FILE.CPUID          (on DOSRES)
INFO.ANALYSIS.DUMP.MGNT.FILE  (on SYSWK1)
INFO.ANALYSIS.EXT.RTNS.FILE   (on SYSWK1)
VTAM.TRACE.FILE               (on SYSWK1)
VSESP.JOB.MANAGER.FILE        (on SYSWK1)
VSE.HARDCOPY.FILE             (on SYSWK1)
VSE.RECORDER.FILE             (on SYSWK1)
VSE.SYSTEM.HISTORY.FILE       (on DOSRES)
VSE.POWER.ACCOUNT.FILE        (on SYSWK1)
VSE.POWER.DATA.FILE           (on SYSWK1)
VSE.POWER.QUEUE.FILE          (on DOSRES)
WORK.HIST.FILE                (on SYSWK1)
ICCF.LIBRARY                  (on SYSWK1)
CU37XX.LOAD.FILE              (on SYSWK1)
PAGING.DATA.SET.ONE           (on DOSRES)
PAGING.DATA.SET.TWO           (on DOSRES)
```

Note on CICS/VSE Journaling

z/VSE provides predefined (non-VSE/VSAM) system files on DOSRES for journaling, which is optional, as follows:

For primary CICS Transaction Server:

```
CICS.SYSTEM.LOG.A
CICS.SYSTEM.LOG.B
CICS.USER.JOURNAL.A
CICS.USER.JOURNAL.B
```

For secondary CICS Transaction Server:

```
CICS2.SYSTEM.LOG.A
CICS2.SYSTEM.LOG.B
CICS2.USER.JOURNAL.A
CICS2.USER.JOURNAL.B
```

For CICS/VSE:

```
CICSO.SYSTEM.LOG.A
CICSO.SYSTEM.LOG.B
CICSO.USER.JOURNAL.A
CICSO.USER.JOURNAL.B
```

Refer also to “Overview on CICS Skeletons and Tables” on page 195.

VSE/VSAM System Files

VSE/VSAM system files reside in the VSE/VSAM user catalog, **except** for the VSE.MESSAGES.ONLINE file which resides in the master catalog. The workfiles defined in the catalogs are not listed.

```
VSAM.MASTER.CATALOG
VSAM.COMPRESS.CONTROL
VSESP.USER.CATALOG
DFHTEMP
CICS.CSD
CICS.RSD
CICS.TD.INTRA
CICS.DUMPA
CICS.DUMPB
CICS.AUXTRACE
CICS.DBDCCICS.DFHDMFA
CICS.DBDCCICS.DFHDMFB
CICS.GCD
CICS.LCD
PTF.FILE
VSE.CONTROL.FILE
VSE.BSTCNTL.FILE
VSE.TEXT.REPSTORY.FILE
VSE.MESSAGES.ROUTING.FILE
VSE.ONLINE.PROB.DET.FILE
VSE.VSAM.RECORD.MAPPING.DEFS
VSE.CICREX.FP01.DIR
VSE.CICREX.FP02.DIR
VSE.CICREX.FP01.FILE01
VSE.CICREX.FP02.FILE01
VSE.EZACICS.CONFIG
VSE.EZACICS.CACHE
VSE.MESSAGES.ONLINE
```

Defining Work Files on a Virtual Disk

z/VSE offers skeleton **SKWRKFIL** in VSE/ICCF library 59 to define system work files on a virtual disk. You can create a virtual disk using the *Hardware Configuration* dialog which supports the device type FBAV for virtual disks.

The system startup must reflect the size of the virtual disk in the DSPACE settings.

Skeleton SKWRKFIL modifies and catalogs procedure IESWORK which is executed during system startup. Procedure IESWORK defines a user catalog (VSEWKUC) on the virtual disk, defines the model cluster for SAM ESDS files, and redefines the labels for the following work files:

```
 IJSYS01 - IJSYS07
 IKSYS21 - IKSYS24
 IKSYS31 - IKSYS34
 IKSYS41 - IKSYS44
 IKSYS51 - IKSYS54
 SORTWK1
```

The whole virtual disk is dedicated to the user catalog VSEWKUC.

Implementation Details

For size flexibility, the work files are allocated in VSAM space. The size required for the virtual disk depends on the amount of partitions active at a time using work files, for example the number of partitions with compile jobs, and the average size of the work file needed. A supervisor assembly for example will need a lot of space, more than 40 MB which is 81920 blocks.

The virtual disk must be defined in the IPL procedure and the VDISK command must be active in the $0JCL procedure. The IESWORK procedure must also be included in the $0JCL procedure. Skeleton SKJCL0 has comments where to add the procedure call and the VDISK command. To make sure that the DSPACE settings are correct, the SKALLOCx skeleton has to be considered for changing the size.

After defining the virtual disk, use the *Tailor IPL Procedure* dialog for increasing the VSIZE in the IPL procedure. Run skeletons SKALLOCx and SKJCL0 to increase the DSPACE value and to modify the system startup by invoking the IESWORK procedure.

The size of the virtual disk must be defined individually for each z/VSE system. Suggested is a minimum of 81920 blocks which means a minimum of 40 MB.

The FSU keeps the work files in the VSESPUC catalog on real disk. Also, during initial installation the work files are on real disk.

Predefined z/VSE Libraries

This section discusses the predefined libraries shipped with z/VSE. z/VSE distinguishes two types of libraries:

VSE Libraries

VSE/ICCF Libraries

Predefined VSE libraries reside on the system disks DOSRES and SYSWK1 in the files:

```
VSE.SYSRES.LIBRARY      (on DOSRES)
VSE.PRD1.LIBRARY        (on DOSRES)
VSE.PRD2.LIBRARY        (on DOSRES)
VSE.PRIMARY.LIBRARY     (on DOSRES)
VSE.CRYPTO.LIBRARY      (on SYSWK1)
VSE.DUMP.LIBRARY        (on SYSWK1)
```

You may refer to Appendix B, “z/VSE Disk Layouts (DOSRES, SYSWK1),” on page 291 for the DOSRES and SYSWK1 layout as shipped with z/VSE.

VSE/ICCF libraries reside in the VSE/ICCF DTSFILE:

```
ICCF.LIBRARY            (on SYSWK1)
```

VSE Libraries

Table 22 lists the VSE libraries and sublibraries predefined for z/VSE and available after initial installation. z/VSE includes the following **system** libraries:

IJSYSRS, PRD1, PRD2, SYSDUMP, and **CRYPTO**.

The **PRIMARY** library is a **private** library, but it is included here because it is created during initial installation of z/VSE.

Table 22. Overview of VSE Libraries and Sublibraries

Library	Sublibraries	Type of Space	Creation Time
IJSYSRS	SYSLIB	Non-VSAM	standalone restore (initial installation)
PRD1	BASE BASED	VSAM	initial installation
PRD1	MACLIB MACLIBD	VSAM	initial installation
PRD2	CONFIG	VSAM	initial installation
PRD2	SAVE	VSAM	initial installation
PRD2	COMM	VSAM	initial installation
PRD2	COMM2	VSAM	initial installation
PRD2	DBASE	VSAM	initial installation
PRD2	AFP	VSAM	initial installation
PRD2	PROD	VSAM	initial installation
PRD2	AFP	VSAM	initial installation
PRD2	DFHDOC	VSAM	initial installation
SYSDUMP	BG F1 through FB DYN	Non-VSAM	initial installation
PRIMARY	$$C	VSAM	initial installation
PRIMARY	SUF	VSAM	initial installation
PRD2	GEN1 GEN1D	VSAM	generation feature installation
PRD2	DB2740 DB2STP SCEEBASE SCEEBASD	VSAM	initial installation
PRD2	CICSOLDG CICSOLDP DLI1A0 ASN740 CCF730 RCV730	VSAM	installation of program(s)
CRYPTO	KEYRING	VSAM	initial installation

The z/VSE base programs and optional programs installed in these sublibraries are listed in Table 6 on page 40 and Table 7 on page 43. Sublibraries with a D at the end are required for PTF application.

IJSYSRS Library

z/VSE allocates space for IJSYSRS on the DOSRES volume. As shown above, IJSYSRS has only one sublibrary, named **SYSLIB**. IJSYSRS and IJSYSRS.SYSLIB are also referred to as **system library**. **IJSYSRS.SYSLIB** contains the following base programs:

- *VSE Central Functions* including:
  - VSE/SP Unique Code
  - VSE/Advanced Functions
  - VSE/POWER
  - VSE/VSAM
  - VSE/ICCF
  - VSE/Fast Copy
- ICKDSF (Device Support Facilities)

IJSYSRS is intended for those base programs which provide hardware and functional support for starting up the system. You should:

1. **Not** copy members into it, *except* for the following:
   - IPL or JCL procedures (must be in IJSYSRS for startup)
   - FCBs and UCBs
   - User exits (such as IPL and JCL exit programs)
2. **Not** move this library or change its size.
3. **Not** create another sublibrary within the library.

Service Aspects of IJSYSRS

Any of your own members that you catalog into IJSYSRS may be affected by a *system refresh* for z/VSE. Because of this, you should also catalog a copy of them into PRD2.SAVE. PRD2.SAVE is created during initial installation and later used for FSU only (Fast Service Upgrade). For a detailed description of the service aspects, refer also to the manual *z/VSE System Upgrade and Service* under “General Service Concept for z/VSE”.

Notes:

1. A system refresh is also referred to as a *Fast Service Upgrade (FSU)*.
2. IBM supplied members which you modify by using specific IBM supplied skeletons or dialogs may automatically be saved in PRD2.SAVE (for example, skeletons for tailoring startup procedures).
3. If you **modify other IBM supplied** members of IJSYSRS, **save neither** a copy of the original member **nor** your changed version in **PRD2.SAVE**. If you want to save copies of the original member and your changed version, you should create another VSE sublibrary and catalog them there.

   It is **your responsibility** to check whether service has been applied to the IBM-supplied members that you have modified. Also, you must ensure that the members work correctly when you have modified them.

Backup/Restore Aspects of IJSYSRS

Refer to the description of the BACKUP and RESTORE command in the manual *z/VSE Guide to System Functions* under “Backup a SYSRES File, Library, Sublibrary, or Member” and under “Restore a SYSRES File, Library, Sublibrary, or a Member”.

PRD1 and PRD2 Libraries

The PRD1 and PRD2 libraries are allocated on DOSRES and SYSWK1. During initial installation, z/VSE allocates them in VSE/VSAM space. This VSE/VSAM space is owned by the VSE/VSAM master catalog. On the condition that the master catalog has sufficient space available, PRD1 and PRD2 will be extended automatically as soon as they are full.

The extensions will be done according to the values given in Table 23 on page 114. Secondary allocations are done on the same volume and must start on a cylinder boundary. If you run out of VSE/VSAM space, use the *Display or Process a Catalog, Space* dialog to define new space to the master catalog. The defined space should be a multiple of the size of the secondary allocations provided in Table 23. A primary allocation is required for a new volume if the old volume is full. As shown in Table 23, the initial allocation depends on the disk device type you are using.

Note: If you use 3380 disk devices, extend the space of the VSE/VSAM master catalog for PRD1 and PRD2 before installing any additional programs.

PRD1 Library

PRD1 contains the macros and base programs of z/VSE that are not required to be in IJSYSRS. It includes the sublibraries **BASE** (BASED) and **MACLIB** (MACLIBD). Note that you should **not** move library PRD1, or change its size, or add another sublibrary to it.

PRD1.BASE contains:

VSE Central Functions including:
- OSA/SF for VSE/ESA
- REXX/VSE
- VSE Connectors
- VSE/OLTEP

The other programs included in PRD1.BASE are:
- VTAM
- CICS Transaction Server for VSE/ESA
- TCP/IP for VSE/ESA
- High Level Assembler for VSE
- DITTO/ESA for VSE
- EREP

PRD1.MACLIB contains:
- VSE/Advanced Functions macros
- VSE/POWER macros
- VSE/VSAM macros

PRD2 Library

PRD2 includes the following sublibraries:

- **CONFIG**

  Contains installation-unique members that are not required in IJSYSRS. These include:

– Members created during initial installation.
– Members created when you use the Interactive Interface (for example, CICS/VSE tables and VTAM startup books).

This sublibrary is not used when you apply service. You should **not** change the name of this sublibrary.

- **GEN1** (GEN1D)

  These sublibraries are optional and contain the VSE/Advanced Functions Generation Feature when installed. You should **not** change the name of this sublibrary.
- **SAVE**

  Contains mainly system procedures (PROCs) and is for FSU only. For adding members to IJSYSRS and PRD2.SAVE, refer to "Service Aspects of IJSYSRS" on page 112.
- **AFP, ASN740, CCF730, COMM, COMM2, DBASE, DB2740, DB2STP, DFHDOC, DLI1A0, PROD, SCEEBASE,** (SCEEBASD), **RCV730, CICSOLDG,** and CICSOLDP.

  These are default sublibraries for z/VSE optional and base programs (except for DB2STP which is a working library). Some of these libraries do not exist after initial installation. They are created later when the programs are actually installed.

Note: The following **base** programs are installed in **PRD2** sublibraries:
- DB2 Server for VSE is installed in PRD2.DB2740.
- LE/VSE is installed in PRD2.SCEEBASE.

Table 7 on page 43 lists the z/VSE optional programs and the corresponding default sublibraries. Besides the predefined sublibraries, you can define additional sublibraries in PRD2 for z/VSE optional programs or other VSE licensed programs. It is recommended, however, that you create separate VSE user libraries for your own application programs.

Space Allocations for PRD1 and PRD2

Table 23 shows the total initial allocations for PRD1, PRD2, and the OME file for the disk devices supported for initial installation. Note that a value without a qualifier means library blocks. In case of FBA disk devices, one library block has a size of 1024 bytes (1KB) including 2 FBA blocks of 512 bytes each. In case of (E)CKD disk devices (3380 and 3390), the number of tracks is shown in the table in addition to the values for library blocks.

Table 23. Allocations for PRD1 and PRD2 Library

Disk Device	System-Provided VSE/VSAM Space		PRD1 Allocations in Library Blocks/Tracks		PRD2 Allocations in Library Blocks/Tracks	
	in Blocks/Tracks	in Library Blocks	Primary Allocation	Secondary Allocation	Primary Allocation	Secondary Allocation
3380	10380 692 Cyl.	319100	21212 690 Trks.	10606 345 Trks.	10606 345 Trks.	21673 705 Trks.
3390	11010 734 Cyl.	360100	19624 600 Trks.	9812 300 Trks.	10793 330 Trks.	21586 660 Trks.
FBA	809984 Blks.	404992	20480	10240	10240	20480

In some cases, the amount of space predefined for PRD2 may not be sufficient. To extend PRD2, define additional VSE/VSAM data space in the master catalog, for example, if you:

- Install a large number of z/VSE optional programs.
- Install additional VSE licensed programs in PRD2.

If you install z/VSE optional programs using the Interactive Interface dialog, the dialog scans the tape and prints information about the required space for each optional program.

Notes:

1. The Librarian command **LD** (List Directory) provides useful space information about libraries.
2. The predefined allocations for PRD2 may be checked using the *File and Catalog Management* dialogs.

SYSDUMP Library

The size of the VSE.DUMP.LIBRARY is as follows:

```
Disk Device:   3380   3390   FBA
----------------------------------
Cyls/Blks:     78     95     409600
```

SYSDUMP is the VSE system library that is used for storing partition dumps and data space dumps. It includes one sublibrary for each static partition and a single sublibrary for all dynamic partitions.

You may have to reallocate the file *VSE.DUMP.LIBRARY* with a larger size in any of the following situations:

- You store many dumps in the SYSDUMP library.
- Your partitions are larger than 50 MB in size.
- Your z/VSE system uses the large partitions allocated for environment C.

If you use the Info/Analysis program to view dumps from other systems, you should reallocate this library to meet this requirement. “The Dump Library” in the manual *z/VSE Guide for Solving Problems* has information about working with the SYSDUMP library.

For environment C, you must extend the dump library so that it can hold a dump of the largest partition (512 MB).

PRIMARY Library

The PRIMARY library is a **private** library and includes sublibraries intended for environments which do not use VSE/ICCF and its libraries. z/VSE creates a PRIMARY library during initial installation including:

- A common sublibrary for the exchange of data among different users named PRIMARY.$$C.
- A sublibrary named PRIMARY.SUF used as intermediate storage for PTFs requested from the IBM service server.

The PRIMARY library is located in VSE/VSAM space and defined in the VSE/VSAM user catalog (VSESPUC). It will include PRIMARY sublibraries defined for individual users and which are named **PRIMARY.userid**. Primary sublibraries can be created for individual users (if they are authorized in their user profile) through the:

- *Maintain Primary Sublibraries* dialog which submits a job for creating/deleting PRIMARY sublibraries.
- *IESUPDCF* batch utility described in Appendix C of the *z/VSE Administration* manual.

If security is active, PRIMARY sublibraries are specially protected as described in the manual *z/VSE Administration* under “Access Control for Libraries”.

Size of the PRIMARY Library

The initial allocation values are as follows:

```
Disk        Allocation in Blocks/Tracks
Type          Primary     Secondary

FBA             5120        10240
3380              75          225
3390              75          225
```

CRYPTO Library

This library is required by the Secure Sockets Layer (SSL) support for storing keys and certificates. The library is protected by an entry in the security table DTSECTAB.

The CRYPTO library has the same initial allocation values as shown above for the PRIMARY library.

VSE/ICCF Libraries

The VSE/ICCF DTSFILE is allocated with approximately 40MB and defines 199 VSE/ICCF libraries and users during initial installation. These libraries are also referred to as *program development* libraries.

Table 24 shows the VSE/ICCF libraries and their use. Note that some libraries are reserved for z/VSE. The members that z/VSE ships in these libraries take up approximately 20% of the space reserved for the DTSFILE. After allocating a library for each user, determine the total requirement for the DTSFILE. Compare this to the default allocation. If the space is insufficient, you should extend the file.

For information on how to extend the DTSFILE, see the manual *z/VSE Administration* under “Using Skeleton SKDTSEXT”. It describes how to use the skeleton *SKDTSEXT* for this task.

Table 24. VSE/ICCF Libraries

Library	Type	Contents	Usage
1	Private	VSE/ICCF administrative library. Contents shipped with VSE/ICCF.	system
2	Common	Common library. Macros and procedures. VSE/ICCF and z/VSE code members.	system
3 - 6	Public	Empty.	user
7	Private	Empty.	user
8	Private	Default primary library for operator profile.	user
9	Private	Default primary library for programmer profile.	user
10	Private	Default primary library for administrator profile.	user

Table 24. VSE/ICCF Libraries (continued)

Library	Type	Contents	Usage
11 - 49	Private	Empty.	user
50 - 58	Public	Reserved for z/VSE.	system
59	Public	z/VSE job streams, skeletons, CICS/VSE tables, and sample programs for the workstation file transfer support.	system
60 - 67	Public	Reserved for z/VSE.	system
68	Public	z/VSE members for Personal Computer tasks.	system
69	Public	Reserved for z/VSE.	system
70 - 199	Private	Empty.	user

For information about changing the characteristics of the DTSFILE, refer to "VSE/ICCF DTSFILE Generation Parameters" on page 284.

Note: The *Program Development Library* dialog helps you access and use VSE/ICCF libraries. For detailed information about this dialog, refer to the manual *VSE/ESA Programming and Workstation Guide* under "Handling VSE/ICCF Library Members".

Planning for Private Files

A private file is a user-owned file which is separate and distinct from system files.

VSE/VSAM Files

With the *File and Catalog Management* dialog, you can define files (or libraries) in VSAM-managed space. These files have the following characteristics:

- The space is defined as a VSE/VSAM cluster.
- A secondary allocation allows for dynamic extension.
- Secondary extents are *not* dynamically reclaimed when they are empty.

"Overview of File and Catalog Management Dialogs" in the manual *z/VSE Administration* explains how to use the *File and Catalog Management* dialog.

Non-VSE/VSAM Files

Although it is recommended to work with VSE/VSAM files, you can define files for other access methods such as SAM and DAM. If you use such access methods, consult the following manuals:

z/VSE System Macros Reference

z/VSE System Macros User's Guide

Creating Standard Labels for Non-VSE/VSAM User Files

When using dialogs for VSE/VSAM file definition, z/VSE automatically adds standard labels to the procedure STDLABUP.

For non-VSE/VSAM files, you can use the skeleton *STDLABUS* to create standard labels as described in the manual *z/VSE Administration* under "Creating Standard Labels for Non-VSAM User Files".

Planning for Private VSE Libraries

A private library is a user-owned library which is separate and distinct from the system library.

PRIMARY sublibraries belong to private libraries. However, they are discussed under VSE system libraries (“PRIMARY Library” on page 115) since the PRIMARY library is created by z/VSE during initial installation.

General Recommendations

It is recommended that you create separate user libraries and sublibraries for your applications instead of using system library PRD2.

This keeps those libraries independent of any future changes that IBM service may make to PRD2. In addition, separate libraries offer you more flexibility with regard to size and organization.

You can define VSE user libraries either in VSAM-managed space or in space that is not managed by VSAM. In either case, you must define at least one *sublibrary* before a new library can be used. This is a requirement of the Librarian, described in the manual *z/VSE Guide to System Functions* under “The Librarian Program”, which controls all VSE libraries in your system.

Libraries in VSE/VSAM Space

The same applies as stated for “VSE/VSAM Files” on page 117.

After defining a library, you must add its name to the library definition chain using a *LIBDEF* command for the partition needed.

Libraries in Non-VSE/VSAM Space

z/VSE provides **skeletons** to define, extend, and delete libraries in space **not** managed by VSE/VSAM. These skeletons are described in the manual *z/VSE Administration*.

You can allocate both multi-extent and multi-volume user libraries. When you define a multi-volume library, you must use the same disk device type. After defining a library, you must add its name to the library definition chain using a *LIBDEF* command for the partition needed.

Note: The manual *z/VSE System Control Statements* describes the LIBDEF command in detail. A detailed description of the Librarian program is given in the manual *z/VSE Guide to System Functions*.

Chapter 8. Running z/VSE Under VM

The term VM in this chapter stands for **z/VM**.

Use the information in this chapter together with the following manuals: *z/VM Version 4, Running Guest Operating Systems*, SC24-5997. *z/VM Version 5, Running Guest Operating Systems*, SC24-6115. Table 26 on page 120 provides a summary of the major aspects you should consider when installing z/VSE under VM; you should use this figure together with Table 8 on page 52. The manual *z/VSE Installation* describes in detail how to install a z/VSE system under VM.

Refer also to “Advantages when Running z/VSE Natively in LPAR mode” on page 1 for details on running z/VSE as multiple preferred guest under VM.

Introduction

z/VSE 3.1 can run as a guest system under **z/VM** Version 4 Release 1.0 (or higher). However, Table 25 shows the releases that are required in order to use specific functions.

Table 25. z/VM Levels Required for Various Functions

Function	z/VM Level
HiperSockets	4.2 or later
Hardware Crypto	4.2 or later
z/VSE SCSI Support	4.4 or later
z/VSE IPL from SCSI	4.4 with these service levels: • UM31181 for the English version. • UM31180 for the German version. • UM31179 for the Kanji version. or later
VM-Emulated FBA Disks on SCSI	5.1 with service level UM31214, or later

Note that z/VSE 3.1 runs with supervisor mode ESA only. If your current system runs with a VM or VMESA supervisor, refer to “Migration Items of Previous Releases” on page 94 for migration details.

In an environment where z/VSE runs under VM, you can:

- Log on to VM using the ID and password defined for the z/VSE virtual machine. Your terminal can then serve as the z/VSE console.
- Dial into the z/VSE system and use the functions provided by the Interactive Interface. If your VM system has the *VM/PASSTHRU* licensed program installed, you can quickly switch between VM CMS and z/VSE environments.
- Log on to CMS and interact with z/VSE. z/VSE provides the VM/VSE Interface, which allows CMS users of VM to work with z/VSE systems concurrently. This is discussed under “VM/VSE Linkage Functions” on page 222.

In this chapter, **virtual machine console** refers to a console that is defined for each VM guest system. **z/VSE console** refers to a console that is used to control the operation of a z/VSE system.

Summary of z/VSE and VM Operating Environments

Table 26 summarizes some of the major items you should consider when planning to install z/VSE under VM.

Note that the VM modes have changed with the introduction of z/VM 5.1. If you are planning to run z/VSE under z/VM 5.1, refer to the *z/VM Version 5, Running Guest Operating Systems*, SC24-6115.

Performance Recommendation

Irrespective of which z/VM version you are using, you are strongly recommended to define sufficient storage in the virtual guest machine so that your z/VSE will run *without paging* (that is, as a NOPDS system). As a result, only z/VM will perform paging.

These are the VM modes that are available with releases of z/VM earlier than z/VM 5.1:

V=R (Virtual=Real)

V=F (Virtual=Fixed)

V=V (Virtual=Virtual)

V=R or V=F

In V=R or V=F mode VM does **not** handle paging. You should review the performance considerations for a native z/VSE system, especially the splitting of the Page Data Set. You can run up to 6 V=F guests, or up to 5 if one is V=R.

A z/VSE guest running V=F requires an ESA/390 processor with the PR/SM feature. V=F is not supported in LPAR mode.

V=V

If z/VSE runs as a V=V guest, double paging (VM and z/VSE) will occur if z/VSE has a page data set (PDS). One example of using V=V is if you have multiple z/VSE guest machines and all 6 V=F (and V=R) are used.

Table 26. Summary of Modes of Operation and System Functions Supported

VSE MODE:	**MODE=ESA**	**MODE=ESA**
VM Mode:	V=R/V=F	V=V
VM Paging	NO	YES
PDS (Page Data Set) VSE Paging NOPDS (No Page Data Set) VSE Paging	 YES NO (see Note)	 YES NO

Table 26. Summary of Modes of Operation and System Functions Supported (continued)

VSE MODE:	MODE=ESA	MODE=ESA

VM CCW Translation
 	
VSE CCW Translation
	Not needed/done for V=R if
	`SET NOTRANS ON`
	`or`
	`SET CCWTRAN OFF`
	YES
		YES
		
		YES

VSE Real Size VSE Virtual Size	≥32MB Real from DEF STOR xxGB Virtual from VSIZE definition	≥32MB Real from DEF STOR xxGB Virtual from VSIZE definition

Note: No page I/O but DAT=YES.

Comments to Table 26:

- If you use processors which support PR/SM, it is recommended to use V=R and V=F.
- z/VSE does not page in case of a NOPDS system.
- Refer to "Virtual Storage Support" on page 48 for a description of the maximum VSIZE (xxGB) possible. Note that VSIZE cannot be specified for a NOPDS system.

VM Linkage Support

If you want to submit jobs from VM to a z/VSE guest system, you should have the VM/VSE Interface Routines installed. Specific commands are then available for cross-system communication. Refer to "VM/VSE Linkage Functions" on page 222 for details.

The VM/VSE interface routines must be installed through skeleton SKVMVSE. Under "Installing VM/VSE Interface Routines", the manual *z/VSE Installation* describes in detail the installation of the interface routines.

Initial Program Load (IPL)

Related Section:
- "Defining a CMS Profile to IPL a SCSI Device" on page 128

Initial installation of z/VSE as a VM guest machine involves two IPLs.

1. **The first IPL** (IPL cuu) is from the distribution tape and loads the standalone environment including the required utility programs for installation.
2. **The second IPL** depends upon whether the IPL is performed from a SCSI disk or a non-SCSI disk, as described below.

Performing an IPL From an FCP-Attached SCSI Disk

FCP-attached SCSI disks cannot be sensed. Therefore, if SCSI disks are to be included in the I/O configuration, you must add them using the Interactive Interface *Hardware Configuration* dialog.

An exception to the above is if you use FCP-attached SCSI disks as your system residence disks. The connection path information from the first installation step is used to update the IPL procedure.

The second IPL is then performed automatically using the information provided during the initial installation (the first IPL).

Figure 8 shows an example of an IPL process from a SCSI disk when running z/VSE under VM.

```
0I04I IPLDEV=X'608',VOLSER=DOSRES,CPUID=FF0000032064
      FCP=X'C00',WWPN=5005076300C69A76,LUN=5745000000000
0J01I IPL=$IPLESA ,JCL=$$JCL
$$A$SUPI,VSIZE=2048M,VIO=512K,VPOOL=64K,LOG
        * During initial installation, ADD statements
        * are added for the devices attached.  They
        * also reflect definitions made in the VM
        * directory shown later.  For example:
                  ADD 080,3270  (Terminal for DIAL function)
                  ADD 01F,3270  (z/VSE system console)
                  ADD 00C,3505
                  ADD 00D,3525
                  ADD 00E,3262
:
                  ADD 240:245,ECKD
                  ADD 608:609,FBA
                  ADD C00,FCP
        * The following ADD statements are predefined
        * and are required by z/VSE:
ADD FDF,FBAV            VIRTUAL DISK LABEL AREA, DO NOT DELETE
ADD FEC,3505            POWER DUMMY READER, DO NOT DELETE
ADD FED,2520B2          POWER DUMMY PUNCH, DO NOT DELETE
ADD FEE,PRT1            POWER DUMMY PRINTER, DO NOT DELETE
ADD FEF,PRT1            POWER DUMMY PRINTER, DO NOT DELETE
ADD FFA,3505            ICCF INTERNAL READER, DO NOT DELETE
ADD FFC,3505            ICCF DUMMY READER, DO NOT DELETE
ADD FFD,2520B2          ICCF DUMMY PUNCH,  DO NOT DELETE
ADD FFE,PRT1            ICCF DUMMY PRINTER, DO NOT DELETE
ADD FFF,CONS            DUMMY CONSOLE, DO NOT DELETE
:
DEF SCSI,FBA=608,FCP=C00,WWPN=5005076300C69A76,LUN=5745000000000000
DEF SCSI,FBA=609,FCP=C00,WWPN=5005076300C69A76,LUN=5746000000000000
0S45I SCSI DEVICE 609 CONSISTS OF 09765632 BLOCKS, 09756789 BLOCKS ARE
AVAILABLE, 651 BLOCKS ARE UNUSED
:
```

Figure 8. Example of a z/VSE IPL Process (Using $IPLESA) From a SCSI Disk

Because of device sensing, you should attach the devices to be used by z/VSE *before* the sense IPL. If you do not, you must later modify the IPL procedure created.

Detach devices you do *not* want in the z/VSE IPL procedure (for example, CMS minidisks or MAINT's 190 minidisk) or delete them later via the *Tailor IPL Procedure* dialog.

Performing an IPL From a Non-SCSI Disk

The second IPL from disk (DOSRES) is done by the system automatically. This IPL is known as the *sense* IPL. z/VSE uses device sensing to automatically define:

- Each device attached to z/VSE.
- Each device defined in the VM directory entry for z/VSE.

The sensed devices are used to update the IPL procedure. Figure 9 shows an example of an IPL procedure for running z/VSE under VM.

```
01F $$A$SUPI,VSIZE=264M,VIO=512K,VPOOL=64K,LOG
        * During initial installation, ADD statements
        * are added for the devices attached.  They
        * also reflect definitions made in the VM
        * directory shown later.  For example:
                    ADD 080,3270  (Terminal for DIAL function)
                    ADD 01F,3270  (z/VSE system console)
                    ADD 00C,3505
                    ADD 00D,3525
                    ADD 00E,3262
:
                    ADD 240:245,ECKD
        * The following ADD statements are predefined
        * and are required by z/VSE:
ADD FDF,FBAV            VIRTUAL DISK LABEL AREA, DO NOT DELETE
ADD FEC,3505            POWER DUMMY READER, DO NOT DELETE
ADD FED,2520B2          POWER DUMMY PUNCH, DO NOT DELETE
ADD FEE,PRT1            POWER DUMMY PRINTER, DO NOT DELETE
ADD FEF,PRT1            POWER DUMMY PRINTER, DO NOT DELETE
ADD FFA,3505            ICCF INTERNAL READER, DO NOT DELETE
ADD FFC,3505            ICCF DUMMY READER, DO NOT DELETE
ADD FFD,2520B2          ICCF DUMMY PUNCH,  DO NOT DELETE
ADD FFE,PRT1            ICCF DUMMY PRINTER, DO NOT DELETE
ADD FFF,CONS            DUMMY CONSOLE, DO NOT DELETE
        * Other IPL Commands
SET ZONE=WEST/00/00
DEF SYSCAT=DOSRES
DEF SYSREC=SYSWK1
SYS DASDFP=YES
SYS JA=YES
SYS BUFSIZE=1500
SYS NPARTS=40
SYS SEC=NO
SYS SERVPART=FB
SYS BUFLD=YES
SYS PASIZE=30M
SYS SPSIZE=0K
DLA NAME=AREA1,VOLID=DOSRES,CYL=60,NCYL=3,DSF=N
SVA PSIZE=(652K,6M),SDL=700,GETVIS=(768K,6M)
```

Figure 9. Example of a z/VSE IPL Procedure ($IPLESA) From a Non-SCSI Disk

Because of device sensing, you should attach the devices to be used by z/VSE *before* the sense IPL. If you do not, you must later modify the IPL procedure created.

Detach devices you do *not* want in the z/VSE IPL procedure (for example, CMS minidisks or MAINT's 190 minidisk) or delete them later via the *Tailor IPL Procedure* dialog.

Using z/VSE System Activity Dialogs

The Interactive Interface provides two dialogs that display general information about current system activity. These are the *Display System Activity* and the *Display Channel and Device Activity* dialogs. With them, you can monitor information such as CPU use, paging, and I/O activity.

When using these dialogs for a z/VSE guest machine, note that:

- All data is valid, except for data displayed as the number of events per second (for example, SIO/Sec).
- The information describes z/VSE activity only.

  The percentage of CPU use shows how much time (VTIME) the guest virtual machine has consumed. It does *not* include the time used by CP to service z/VSE (TTIME-VTIME).

Refer to the manual *z/VSE Operation* under "Dialogs for Displaying System Status" for how to use the dialog.

Defining a z/VSE Virtual Machine

Defining a VM Directory Entry

Each guest system running under VM requires a *directory entry* that defines the virtual machine's configuration and operating characteristics. For the z/VSE system volumes DOSRES and SYSWK1, VM requires unique volume IDs if they are shared by two or more z/VSE guest systems running under VM.

Figure 10 shows a sample VM directory entry for a z/VSE system named *VSEESA1* for a V=R (VIRT=REAL) and MODE=ESA system. The letters to the left of the figure are used here as reference points for various statements. They are not part of the actual directory entry.

Note: When you install z/VSE under VM, z/VSE will automatically sense the devices defined in the VM directory for its configuration. If you have any devices not supported under VM, you can bypass device sensing by adding the EML operand in the IPL ADD statements of z/VSE.

```
(A)  USER VSEESA1 password 1024M 2048M G
(B)  OPTION VIRT=REAL MAINTCCW QUICKDSP CPUID 111111
(C)  MACHINE ESA 2
(D)  CPU 0 CPUID 0xxxxx NODEDICATE
     CPU 1 CPUID 1xxxxx
(E)  IPL 240 PARM AUTOLOG
(F)  ACCOUNT ### SYSPROG
(G)    DEDICATE 244 704
       DEDICATE 245 705
(H)    CONSOLE 009 3270 T OPERATOR
(I)    SPECIAL 080 3270
               .
               .
       SPECIAL 01F 3270
(J)    SPOOL   00C 3505 A
       SPOOL   00D 3525 A
       SPOOL   00E 3262 A
       SPOOL   05E 4248 A
               .
               .

            * Link to optional disks as needed....
            * Link to Executable CMS Code

(K)    LINK MAINT 190 190 RR

            * Link to Program Products (Y Disk)

       LINK MAINT 19E 19E RR
(L)    MDISK 240 3390 1 1112 VSADOS MWV
       MDISK 241 3390 1 1112 VSASY1 MWV
       MDISK 242 3390 1 1112 USER01 MWV
       MDISK 243 3390 1 1112 USER02 MWV
```

Figure 10. Example of a VM Directory (V=R)

The statements in the directory define the following:

A **USER** defines the:
- Name and password of the z/VSE guest.
- Virtual storage size (1024M or 1G) for the z/VSE guest when logging on to VM (this is the real storage for z/VSE itself).

- Maximum virtual storage (2048M) size that can be defined for the z/VSE guest after logging on. This is therefore the maximum size for VM.
- User class G (general) is available.

B **OPTION** defines:
- VIRT=REAL specifies the z/VSE guest as a V=R machine.
- MAINTCCW authorizes the z/VSE guest to initialize disk devices (use diagnostic CCWs).
- QUICKDSP causes a virtual machine to be added to the dispatch list immediately when it has work to do, without waiting in the eligible list.
- CPUID specifies a unique processor identification. This option is required if disk devices are shared (DOSRES and SYSWK1, for example).

C **MACHINE** defines the virtual machine mode (which must be ESA). **2** defines the maximum number of processors that can be active.

D **CPU** defines the processors that can be active.

E **IPL** defines automatic IPL to be performed with specified device address.

F **ACCOUNT** defines an account number and a distribution identification.

G **DEDICATE** specifies that a real disk device is to be dedicated to this z/VSE guest (VSEESA1).

H **CONSOLE** defines the virtual machine console. In the statement:
- 009 is the virtual address of the console.
- 3270 defines the terminal type of the virtual machine console.
- T defines the spool class.
- OPERATOR defines the secondary VM user. If the primary user (VSEESA1) is disconnected, the VM user OPERATOR will receive all CP messages for the z/VSE virtual machine. The secondary user also can send CP commands to the disconnected z/VSE virtual machine.

I **SPECIAL** defines a virtual device with device type and address to the z/VSE system. Terminal addresses do *not* have to be real devices on the system.

When you define a terminal as SPECIAL, you can use the address to dial into the z/VSE system. You should ensure that you have a sufficient number of addresses defined as SPECIAL for users who require the DIAL function.

Devices defined in this manner are added to the z/VSE IPL procedure through device sensing.

J **SPOOL** defines virtual unit record devices. One entry is required for each unit record device.

K **LINK** defines a link to a device that belongs to another user.

L **MDISK** defines an extent on a disk device to be owned by VSEESA1. The extent assigned with the statement becomes the user's minidisk.

In case of FBA minidisks, the total number of blocks must be a multiple of the number of blocks per access position for the device type. "Multiple of access position" means cylinder boundary and its values should be obtained from the related device documentation.

For further LINK and MDISK information, refer to "Defining Shared Minidisks" on page 130.

Defining a CMS Profile EXEC

If you want additional options to be executed during startup, you can define a CMS PROFILE EXEC for the z/VSE user ID. The command

```
IPL cuu   (for V=R and V=F) or
IPL CMS   (for V=V)
```

executes the profile before the IPL (for z/VSE). You can add the command to the VM directory entry for z/VSE, or you can enter it under CMS.

If you want to define a CMS profile EXEC for a z/VSE guest system, you must first define a CMS 191 minidisk for the z/VSE guest system for holding the profile.

Note that the sample PROFILE EXEC below is set for two user IDs. *VSEA* is a z/VSE system with 1024 MB and *VSEB* a z/VSE system with 2048 MB.

```
/*
--Next line sets variable nl to hex '15' to separate commands
*/
nl='15'x                            /*linend character*/
parse value diag(8,'QUERY USERID')  with userid . node (nl)
'CP SET RUN ON'
'CP SP RDR CONT'
if userid='VSEA' then cpcom='CP DEF STOR 1024M'
else cpcom='CP DEF STOR 2048M'
cpcom = cpcom ,
        || nl || 'CP TERM CONM 3270 BRE GUEST' ,

        || nl || 'CP IPL 240 CLEAR'
address command cpcom
say 'IPL failed, RC=' RC
exit rc
```

Figure 11. Sample PROFILE EXEC for Two z/VSE Guest Machines (in REXX)

Be sure that the last concatenated command is the IPL command for z/VSE.

You can have the following CP commands in the PROFILE EXEC. Note that if you do not want to use a PROFILE EXEC, you can enter these commands from CP mode.

- **SET RUN ON –** Allows you to activate the Attention key (causing a read of a CP command) without stopping the virtual machine.
- **SPOOL READER CONT –** Ignores intermediate end-of-file indications or CLOSE requests for virtual readers. Reading is continuous, with all end-of-file indicators ignored until all files spooled to the virtual machine are read. If this option is not in effect, a unit exception is reflected to the virtual machine at the end of each spooled file.
- **DEF STOR n..nM –** The DEF STOR command can be used to acquire the maximum amount of virtual storage specified in the directory entry for the z/VSE virtual machine.
- **TERM CONMODE 3270 –** Specifies full screen operation mode of the virtual machine console. This command is required if you use a shared z/VSE console in 3270 mode (see "Defining a Shared Console" on page 129).
- **TERM BREAKIN GUEST –** Prevents CP messages from being displayed on the z/VSE console.

- **IPL cuu LOADPARM** – This is the IPL statement for z/VSE. You should replace **cuu** with the address of DOSRES. *This must be the last CP command in the PROFILE EXEC*.

If you use the TERM CONMODE 3270 command in the PROFILE EXEC, you should combine it with other CP commands into a single command line. This is because TERM CONMODE 3270 causes CMS to terminate. If it is on a separate command line, other commands in the PROFILE EXEC will not be executed.

Combining the commands into a single command line ensures that they are read from the CP console stack when CMS terminates. To combine (concatenate) commands, you can use the coding technique shown in Figure 11 on page 127.

For further details about using a CMS PROFILE EXEC, refer to the VM documentation.

Defining a CMS Profile to IPL a SCSI Device

If your system resource disk is an FBA-SCSI device, you must provide the following definitions in your CMS profile:

```
'CP DEF STOR 32M'
'CP TERM CONMODE 3270'
'CP SET LOADDEV PORT 50050763 00C69A76 LUN 560A0000 00000000'    (see '1')
'CP I C02 '                                                      (see '2')
```

1. The SET LOADDEV command is required in order to specify the:
   - Port (WWPN) that is used to access the DOSRES SCSI disk.
   - Logical unit number of the DOSRES SCSI disk.
2. For the IPL or load address, you specify the FCP device that is used to attach the DOSRES SCSI disk. For further information, refer to the Chapter “Configuring Your System to Use SCSI Disks” in the manual *z/VSE Administration*, SC33-8224.

Defining a z/VSE Console

The CONSOLE control statement in a directory entry specifies the *virtual machine* console for that user. In the z/VSE – VM environment, the way you define the *z/VSE* console (the *system* console) depends in part on the following considerations:

1. Will z/VSE and VM have separate consoles?
2. Will the z/VSE console support VM operations?
3. Will z/VSE be logged on manually or autologged?
4. Will the z/VSE console be disconnected?
5. Will the virtual machine run disconnected?

The rest of this section explains three possible ways to define the z/VSE console.

Note: VM’s *autolog* facility can be used to automatically IPL one or more z/VSE virtual machines without operator intervention. The virtual machine is logged on in disconnected mode. The same restrictions that apply to any disconnected virtual machine also apply to virtual machines logged on with the autolog facility.

When VM autologs a z/VSE virtual machine, the correct IPL and JCL procedures for z/VSE are selected through the automated startup facility.

Defining a Dedicated Console

A VM installation can dedicate (reserve) one of its terminals for use as the z/VSE console. Doing this means that the z/VSE operator can work from this device as if it were a console for a stand-alone z/VSE system.

In Figure 10 on page 125 you can use a DEDICATE statement to specify that the real device at 01F (or a terminal at another address) is to be used as the z/VSE console.

To use a dedicated z/VSE console, the following must be considered:

1. The device specified in the DEDICATE statement must not be enabled before the z/VSE virtual machine is logged on.

   If it has been enabled, you can disable it using the command:

   ```
   CP DISABLE cuu
   ```

   where **cuu** is the address of the real device.
2. If the z/VSE system is not automatically IPLed, you must log on the z/VSE virtual machine and perform an IPL from a VM terminal. This must be a *different* terminal from the one specified in the DISABLE command.

   You can, if you wish, disconnect the z/VSE virtual machine after IPL completes.

Defining a Shared Console

Instead of defining a dedicated console for z/VSE, you can operate the system from a *shared* console. In this environment, the *same* VM terminal used to log on and IPL the z/VSE virtual machine is also used as the z/VSE console.

To use a shared console:

1. From a VM terminal, log on using the ID and password defined for the z/VSE virtual machine.
2. Enter the following CP commands *after* you IPL CMS but *before* you IPL z/VSE:

   ```
   CP SP RDR CONT
   CP SET RUN ON
   CP TERM CONMODE 3270
   CP TERM BREAKIN GUEST
   ```

   Besides manually entering these commands in CP mode, you can also make them a part of a PROFILE EXEC that you define for the z/VSE virtual machine. "Defining a CMS Profile EXEC" on page 127 shows how to do this.
3. IPL z/VSE. The z/VSE console screen will be displayed.
4. If you want to return to VM from the z/VSE console, press the **PA1** key. This puts the screen in CP mode.
5. To return to the z/VSE console from CP mode, enter the command **B** (for **CP BEGIN**).
6. When you are done using the z/VSE console, enter the following command in the command line at the bottom of the console screen:

   ```
   * CP DISC
   ```

Defining a Disconnected Console

You can also define the z/VSE console as a *disconnected* console. The z/VSE console may or may not be active (displayed at a terminal) at any given time.

To define the z/VSE console to operate in disconnect mode:

1. Define the virtual machine console in the z/VSE directory entry. For example, the directory entry could be:

```
CONS 009 3215
```

2. In the directory entry, define the device to be used for the z/VSE console with a **SPECIAL** statement. For example:

   ```
   SPECIAL 2FF 3215
   ```

   The address specified here should be the *highest* one used in the directory entry's SPECIAL statements. This minimizes the chance that a user later will dial into the z/VSE guest machine and get the z/VSE console by mistake.
3. Have an **ADD** statement in the IPL procedure for z/VSE that uses the same device address as in the SPECIAL statement. For example:

   ```
   ADD 2FF,3277
   ```
4. Have an **ASSGN** statement similar to the following in the JCL ASI procedure for the BG partition:

   ```
   ASSGN SYSLOG,2FF
   ```

   After z/VSE is IPLed, this statement re-assigns the terminal at 2FF as the z/VSE console.

To use the disconnected console:
1. From a VM terminal, log on using the ID and password defined for the z/VSE virtual machine.
2. IPL z/VSE.
3. Once IPL is complete, you can disconnect the z/VSE virtual machine and use the terminal for other purposes.
4. Whenever you want to use the z/VSE console, enter the following command from a VM terminal:

   ```
   DIAL xxxxxxxx 2FF
   ```

   where **xxxxxxxx** is the user ID of the z/VSE virtual machine. The z/VSE console will be displayed at the terminal.
5. When you are done using the z/VSE console, enter the following command in the command line at the bottom of the console screen:

   ```
   * CP RESET 2FF
   ```

Defining Shared Minidisks

Under VM, shared disk devices (DASD Sharing) must be defined as minidisks in the VM directory entry for each guest system that is to access them. VM provides multiple access to the same devices through MDISK and LINK definitions. Figure 12 on page 131 shows an example where MDISK and LINK definitions create *one* access path to DASDs.

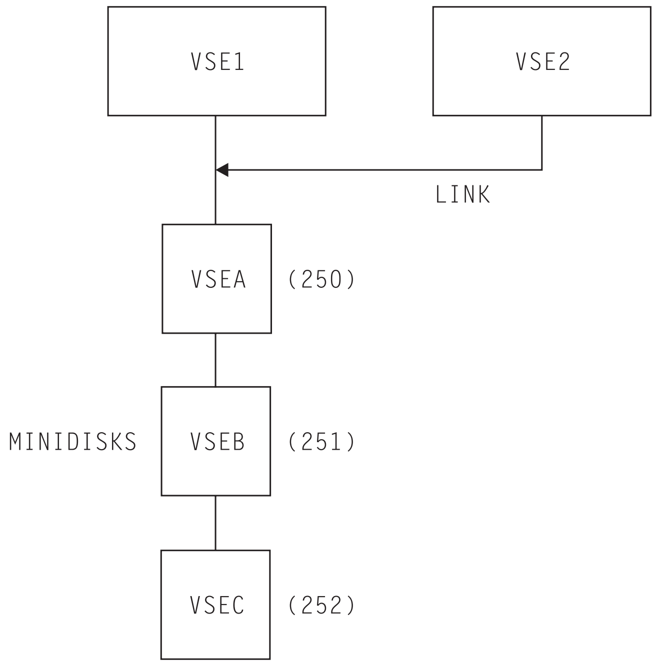


Figure 12. Access Path to Shared VM Minidisks

The MDISK definition of the device on which the lock file resides requires access mode **MWV**. This indicates that a write link is given to the disk and that the CP virtual RESERVE/RELEASE function is used in the I/O operation of that disk.

For performance reasons in a shared system environment, the VSE Lock file should be put on a VM virtual disk or at least on a VM minidisk.

For details about DASD sharing in general, refer to the manual *z/VSE Guide to System Functions* under "DASD Sharing with Multiple VSE Systems".

Using FCP-Attached SCSI Disks Via the VM-Emulated FBA Support

Before you can use FCP-attached SCSI disks via VM's emulated FBA-support, you must have fulfilled the prerequisites described in "Hardware and Software Prerequisites" on page 135.

If you use SCSI disks via VM's emulated FBA-support, these SCSI disks:
- Appear to z/VSE as **9336 Model 20 FBA** disks.
- Are used by z/VSE as disks of type **FBA** (for example, in z/VSE dialogs).
- Must not exceed 2 GB.

Since z/VSE SCSI support is *not* required in order to access VM-emulated FBA disks, these disks can be used by VSE releases prior to z/VSE 3.1.

Processor Resource/Systems Manager Support

The Processor Resource/Systems Manager (PR/SM) allows you to run multiple preferred guest systems under VM as outlined under “Advantages when Running z/VSE Natively in LPAR mode” on page 1.

SQL/DS Guest Sharing Support

Notes:
1. This support relates to DB2 and SQL/DS applications.
2. The guest sharing support requires the *DB2 Server for VSE* program.

The support is provided for a VM environment with z/VSE guest systems. It allows you to run DB2 (or SQL/DS) applications in different VM systems, either under VM or under a z/VSE guest system, and share one or more DB2 (or SQL/DS) data bases. A data base must be installed on a VM system.

The support requires that during IPL z/VSE is initialized as a z/VSE guest system under VM. You must tailor the IPL procedure accordingly (VSE APPC/VM specification). z/VSE provides the *Tailor IPL Procedure* dialog as described in the manual *z/VSE Administration*. Since guest sharing is based on APPC/VM, fixed areas in real storage are required. The amount of PFIX storage needed depends on the number of communication paths. The required PFIX storage can be estimated by the number of links per partition, multiplied by 32KB. For page fixing, use the SETPFIX statement to reserve space for each partition (batch or CICS) that accesses VM databases.

If you are using guest sharing, the z/VSE guest system must have the following entry in the VM directory:

OPTION MAXCONN nn

The value of MAXCONN is based on the estimated number of APPC/VM links required to be established between the z/VSE guest systems and the VM database machines. This number can be estimated as follows:

```
MAXCONN = NUMBER OF CICS LINKS (involved in guest sharing)
          + NUMBER OF BATCH LINKS
          + NUMBER OF DATABASE MINIDISKS
          + SPARE
```

The support is activated if the IPL procedure includes the required SET XPCC commands (created using the *Tailor IPL Procedure* dialog) or by loading the DB2 phase for activation into the SVA.

Chapter 9. Using SCSI Disks With Your z/VSE System

This chapter describes the planning aspects for using Fibre Channel attached SCSI (Small Computer System Interface) disks, referred to simply as SCSI disks, with your z/VSE system.

It contains these main sections:

Related Sections:

- “Support for Fibre-Channel-Attached SCSI Disks” on page 10 provides an introduction to SCSI disk support.
- “Performing an IPL From an FCP-Attached SCSI Disk” on page 121 provides an example of an IPL process from a SCSI disk when running z/VSE under VM.
- “Using FCP-Attached SCSI Disks Via the VM-Emulated FBA Support” on page 131 contains details of using SCSI disks under z/VM.
- “IBM FBA (and SCSI) Disks” on page 296 contains details of the disk layout of a:
  - DOSRES SCSI disk.
  - SYSWK1 SCSI disk.
- The *z/VSE Administration*, SC33-8224 provides:
  - A practical example of a SCSI environment.
  - Details of how to configure FCP adapters, SCSI disks, and connection paths.
  - Details of how to use multipathing to access SCSI disks.
  - Details of how to use shared SCSI disks.
  - A description of the attention routine (AR) OFFLINE / ONLINE commands that you can use.
  - Details of how to perform an IPL of z/VSE from a SCSI disk.
  - A list of errors that might occur during the configuration process.

Please Note That!

- The term *FBA-SCSI disk* is also used to refer to a SCSI disk.

Overview of the z/VSE Support for SCSI Disks

The objective of z/VSE Small Computer System Interface (SCSI) disk support is to offer clients more storage choices as well as help lower Total Cost of Ownership (TCO). Therefore z/VSE 3.1 has been developed to enable selected SCSI disks to be attached, in addition to Extended Count-Key-Data (ECKD) and Fixed Block Architecture (FBA) disks. Only SCSI disks that are contained within the IBM TotalStorage Enterprise Storage Server Model F20, 800, and 800 Turbo are qualified for z/VSE 3.1.

VSE host programs running on a zSeries processor can access Fibre-Channel-attached SCSI disks. Your application programs do not need to be modified in order to access the SCSI disk devices providing these programs are either:
- Device independent.
- Support FBA devices.

Using the appropriate features and service applied to the IBM eServer zSeries server, it is possible to IPL z/VSE from a SCSI disk. Therefore, it is possible to build a “SCSI-only” z/VSE system. However, there are some limitations that apply to a SCSI-only system. For example:
- A standalone dump cannot be created on a SCSI disk. If you wish to create a standalone dump, you must create this dump using a tape or another type of disk.
- Dump information cannot be written to SCSI disks. If you wish to write dump data, you must write this dump to a tape or another type of disk.
- Your applications running under z/VSE cannot use SCSI commands directly. SCSI support is only available through FBA channel commands.

SCSI disk devices use Fixed Block (512-bytes) sectors and are defined in z/VSE as FBA devices. They appear to the user as if they were FBA devices. User-written programs use VSE’s existing Fixed Block Architecture (FBA) support (512-byte blocks) to access SCSI disks.

Special migration considerations also apply:
- It is not possible to use the Fast Service Upgrade (FSU) process to migrate from a VSE/ESA 2.6 or 2.7 system to a z/VSE 3.1 system with SCSI system-residence disks.
- VSE/VSAM Backup/Restore is not possible from ECKD to SCSI. You must use the Import/Export utility to move VSE/VSAM data from ECKD to SCSI disks.
- Similar restrictions (as above) apply when migrating from FBA to SCSI.

z/VSE SCSI-FCP disk support complements SCSI support in z/VM Version 5 and Linux for zSeries. When operating as a guest under z/VM (using SCSI disks not directly attached to z/VSE), z/VM presents SCSI disks as 9336-20 FBA disks. In this situation, z/VSE “sees” them as FBA, not SCSI disks.

z/VSE 3.1 is designed to support SCSI disk volume sizes from 8 MB to 24 GB. Because z/VSE itself uses the first 4 MB for internal purposes, the available user space is equal to the defined size of the disk, minus 4 MB.

Note!

From z/VSE 3.1 onwards, the usable VSE/VSAM space has greatly increased to 16 GB of any SCSI volume. This compares with approximately 4.5 GB for an ECKD volume. The VSE/VSAM space must be within the first 16 GB of a SCSI volume.

The maximum z/VSE 3.1 FBA volume is 2 GB. Of course, multiple 2 GB minidisks can be assigned within the limits of a single physical SCSI disk volume controlled by z/VM. For SCSI disks directly attached to z/VSE under z/VM, the normal z/VSE limits described above apply.

The z/VSE SCSI disk support includes support for:
- *Multipathing*, which is a method of increasing the availability of the device.
- *DASD (disk) sharing* using the z/VSE lock file on a SCSI device (or any other supported device).

Note: z/VSE 3.1 does **not** support FlashCopy for SCSI-FCP disks.

Hardware and Software Prerequisites

If you wish to attach SCSI disk devices to a zSeries processor and access these disk devices from z/VSE, you require:
- An IBM zSeries 800 or 900 processor or their successors (such as a z890 or z990 processor) with at least 32 MB of real storage.
- An IBM zSeries FCP adapter with the following microcode levels:
  - For z800 and z900 processors, J11233.015 or later
  - For z890 and z990 processors, J13471.004 or later
- An FCP switch (such as an IBM 2109).
- An IBM TotalStorage Enterprise Storage Server (ESS) disk controller (such as the IBM 2105) with the microcode level 2.3.1 or later.
- z/VSE Version 3 Release 1 or later.
- z/VM 4.4 or later.
- If you wish to perform an IPL from a SCSI disk:
  - CPU Feature Code 9904 must be installed.
  - For z800 and z900 processors, your microcode level must be EC-Number J12811 or later.
  - For z890 and z990 processors, your microcode level must be EC-Number J12221 or later.
- If you wish to perform an IPL from a SCSI disk under **z/VM 4.4**, you require z/VM Service level:
  - UM31181 for the English version.
  - UM31180 for the German version.
  - UM31179 for the Kanji version.

Prerequisites for Using Emulated FBA Disks: z/VM Version 5.1 onwards supports SCSI FCP disk logical units (SCSI disks) for both system and guest use. The z/VM guest operating system can use SCSI disks either directly, or as emulated 9336 Model 20 fixed-block-architecture (FBA) disks. From VSE/ESA Version 2 Release 6 onwards, VSE supports emulated FBA disks (fullpack or partial minidisks) with these prerequisites:

- APAR VM63560/PTF UM31214 must be installed on z/VM Version 5.
- ICKDSF 1.17 must be installed in order to initialize the emulated FBA disks.
- APAR PQ84848/PTF UQ89212 must be installed for ICKDSF 1.17 on z/VSE or VSE/ESA.

Characteristics of a SCSI Disk

Size of SCSI Disks
z/VSE only supports SCSI disks that have a minimum size of 8 MB and a maximum size of approximately **24 GB**. 4 MB of a SCSI disk is used internally by z/VSE. Therefore, z/VSE restricts your use of each SCSI disk to the actual size minus 4 MB.

Model In z/VSE, SCSI disks are defined as FBA devices and appear to the user as a 9336 Model 20 FBA device.

Block Size
SCSI disks must be configured with a block size of 512 bytes even if the disk controller allows a larger block size than 512 bytes.

ANSI Standards
SCSI disks must support ANSI SCSI Version 3.

FBA CCW Commands Not Supported
The following CCW commands are not supported by z/VSE, and will be terminated with the "command reject" message (X'80' in sense byte 0):
X'02' Read IPL
X'14' Unconditional reserve
X'C4' Diagnostic sense/read

Restrictions When Using Emulated FBA Disks

Before you start, please check the prerequisites listed in "Hardware and Software Prerequisites" on page 135.

From VSE/ESA Version 2 Release 6 onwards (that is VSE/ESA 2.6, VSE/ESA 2.7, and z/VSE 3.1), VSE supports emulated FBA disks (fullpack or partial minidisks) with these restrictions:

- The VTOC of an FBA disk for use by z/VSE or VSE/ESA should **not** be placed at the end of the volume, *except* when the last block is a multiple of 8*777 blocks.
- If the size of the disk is not a multiple of 8*777 blocks, the VTOC should be placed below the highest multiple number of 8*777 blocks that fit into the total number of defined blocks. This is due to the fact that DASD file protection is set to "On" in z/VSE or VSE/ESA.
- The FBA disks must not exceed a size of 2 GB.
- VSAM can only use the first **2 GB** of an emulated FBA disk.

Restrictions When Using VSAM Files On SCSI Disks

- From z/VSE 3.1 onwards, the minimum number of FBA blocks for a VSAM file is 512 blocks compared with 64 blocks for an FBA device (for example, a z/VM FBA minidisk or a virtual FBA device).
- The maximum size of a VSAM **file** is **4 GB**.
- VSAM can use the first **16 GB** of a SCSI disk. If a SCSI disk is larger than 16 GB, the remaining space is not available to VSAM.

Important!
For a list of the restrictions that apply when using VSAM structures on SCSI disks, refer to the manual *VSE/VSAM User's Guide and Application Programming*, SC33-8246.

Space Requirements When SCSI Is Used As a System Disk

From z/VSE 3.1 onwards, these are the main differences to the DOSRES and SYSWK1 disk layouts:

- The layouts of DOSRES and SYSWK1 are designed to take account of the minimum number of FBA blocks (512) for a VSAM file. The layouts of FBA-SCSI and FBA disks (for example, a z/VM FBA minidisk or a virtual FBA device) are identical.
- The space that is available for the master catalog with PRD1 and PRD2 libraries has been increased.
- The dump library has been increased.

For details of the FBA disk layouts of DOSRES und SYSWK1, see "IBM FBA (and SCSI) Disks" on page 296.

Storage Requirements When Using SCSI Disks

To use SCSI with your z/VSE system you require approximately:

- 100 KB 31-bit fixed system Getvis storage per FCP device.
- 10 KB 31-bit fixed system Getvis storage per SCSI disk device.

Installation Considerations for SCSI Disks

You must perform an *initial installation* of z/VSE if you wish to either:

- Install your z/VSE system using SCSI system residence disks.
- Use your SCSI disks as data disks only but use *FBA disks as your system residence disks*.

For details, refer to the manual *z/VSE Installation*, SC33-8222.

You can perform a *Fast System Upgrade (FSU)* of z/VSE if you wish to use your SCSI disks as data disks only and use *ECKD disks as your system residence disks*. For details, refer to the manual *z/VSE System Upgrade and Service*, SC33-8223.

Migration Considerations for SCSI Disks

A VSAM Backup/Restore from FBA to SCSI, or SCSI to FBA, is *not* possible. You can only perform a VSAM Backup/Restore from SCSI to SCSI. To overcome this limitation, you can use the Import/Export facility to import or export:

- FBA to SCSI
- SCSI to FBA
- CKD/ECKD to SCSI
- SCSI to CKD/ECKD

Chapter 10. z/VSE e-business Connectors and Tools

This chapter describes the z/VSE e-business connectors and tools you can use to allow Web browsers to access data stored on the z/VSE host. It consist of these main sections:

Choosing Between 2-Tier and 3-Tier Environments

The support for e-business connectors and tools is based on a 2–tier or 3–tier environment:

- In 2-tier environments, the Web client and z/VSE host communicate directly with each other.
- In 3-tier environments, the Web client or non-Java client, and z/VSE host communicate with each other via an intermediate tier called the *middle-tier*
- The 2-tier environment is **not** the typical environment under which you will develop your Java programs, since it is not:
  - part of the IBM *Application Framework for e-business* (it does not, for example, use the IBM WebSphere Application Server on the middle-tier).
  - secured by the state-of-the-art security services (firewall, and so on) provided by the IBM Application Framework for e-business.

  The 2-tier environment is generally suitable for *intranet* solutions only.
- The Java-based connector (described in “Overview of the Java-Based Connector” on page 141) is normally used in a 3-tier environment.
- The DB2-based connector (described in “Overview of the DB2-Based Connector” on page 142) can **only** be used in a 3-tier environment.
- The 3-tier environment requires a middle-tier server (such as xSeries or pSeries processor) on which the WebSphere Application Server is installed.

For further discussion about the implementation of 2-tier and 3-tier environments, refer to the chapter “Overview of 2- and 3-Tier Environments” in the manual *z/VSE e-business Connectors User's Guide*, SC33-8231.

Installing the VSE Connectors Component

During **Initial Installation** or **FSU** the VSE Connectors component (which is part of VSE Central Functions) is automatically installed into sublibrary PRD1.BASE. The support consists of the:

1. **VSE Connector Server**

   Runs in a dynamic partition of z/VSE and is required for the Java-based connector support. It is described in “Overview of the Java-Based Connector” on page 141.
2. **VSE Connector Client**

   Provides the Java-based connector support on the middle tier and the client side. It is described in “Overview of the Java-Based Connector” on page 141.
3. **VSAM Call Level Interface**

   Supports access to VSE/VSAM data when using the DB2-based connector support.
4. **VSE/VSAM Redirector Connector**

   Enables VSE programs to access data on remote systems, in real-time. It is described in “Overview of the VSAM Redirector Connector” on page 147.

In addition, the **DB2 Server for VSE 7.4** is installed from the Extended Base Tape into sublibrary PRD2.DB2740 if requested during Initial Installation. It is the main component on the host side of the DB2-based connector, and runs in a dynamic z/VSE partition.

In case of an FSU, the DB2 Server for VSE may be installed after the FSU, or during the FSU using the appropriate processing control switch. You should, however, be aware that the DB2 Version might be different from the one installed with the previous release. Refer to “FSU from VSE/ESA 2.6.x/2.7.x to z/VSE 3.1” on page 143 for further details.

Overview of the Java-Based Connector

The Java-based connector provides access to these z/VSE resources and the data they maintain:
- VSE/VSAM
- VSE/POWER
- VSE/ICCF
- DL/I
- Librarian
- Console

This support is based on the functions provided by the *VSE Connector Server* (installed on the z/VSE host) and the *VSE Connector Client* (installed on Web browsers or the middle-tier).

Implementing the VSE Connector Client

To implement the VSE Connector Client, you install these three components on any Java-enabled middle-tier platform:
- A file **VSEConnector.jar** which contains the VSE Java Beans class library. The VSE Java Beans provide a Java programming interface for communicating with VSE/VSAM, VSE/Librarian, VSE/POWER, VSE/ICCF, and Operator Console, on the z/VSE host.
- A set of samples, including Java source code, that show you how to write Java programs that are based upon the use of VSE Java Beans.
- Online documentation (a set of HTML pages) describing the various concepts and samples.

To develop your Web applications, you will probably use all the three components listed above. However, the completed Web applications require only the VSE Java Beans class library at run-time. Therefore, when your Web applications are ready for the production environment (the end-user environment), you install only the **VSEConnector.jar** file (which contains the VSE Java Beans class library) on the:
- Middle-tier (for 3-tier environments)
- Web clients (for 2-tier environments).

This means, you would usually only install **VSEConnector.jar** on each of your Web browsers that are to be used for accessing the z/VSE host.

For a detailed description of how to implement the VSE Connector Client, refer to the chapter “Installing and Operating the Java-Based Connector” in the manual *z/VSE e-business Connectors User's Guide*, SC33-8231.

Implementing the VSE Connector Server

The *VSE Connector Server* is a batch application that implements a TCP/IP socket listener to analyze incoming TCP/IP traffic and initiate subsequent activities as required. It runs per default in a **dynamic partition of class R**. The server communicates with Java-based client applications requesting VSE data and invoking VSE functions.

1. Activating the Server

   During initial installation, FSU, or a COLD startup, the startup job **STARTVCS** for the *VSE Connector Server* is placed into the VSE/POWER reader queue. If STARTVCS is released, the server is activated in a dynamic partition of class R with a partition size of 8 MB. For activation, **TCP/IP must be active**.

2. Configuring the Server

   z/VSE provides skeletons in VSE/ICCF library 59 for configuring the *VSE Connector Server* as follows:

   SKVCSSTJ Place startup job into the VSE/POWER reader queue.

   SKVCSCFG Specify general settings for the server.

   SKVCSCAT Catalog configuration members.

   SKVCSLIB Specify libraries that can be accessed by the server.

   SKVCSPLG Specify plug-ins to be loaded during startup.

   SKVCSUSR Specify users or group of users that can logon to the server.

For a detailed description of how to implement the VSE Connector Server, refer to the chapter “Installing and Operating the Java-Based Connector” in the manual *z/VSE e-business Connectors User’s Guide*, SC33-8231.

Overview of the DB2-Based Connector

In a DB2 relational database environment, the DB2-based connector provides access to this data stored on the z/VSE host:
- VSE/VSAM data
- DB2 data
- DL/I data

The access from a client to VSE/VSAM, DL/I, or DB2 data stored on the z/VSE host is via the middle-tier. The connection from the client to the middle tier is via TCP/IP, from the middle tier to the host via DRDA (Distributed Relational Database Architecture) using TCP/IP (or APPC), where DRDA is the communication protocol and TCP/IP the carrier protocol.

For details and examples of how the DB2-based connector is used to access data on the z/VSE host, refer to the chapter “Using the DB2-Based Connector to Access Data” in the manual *z/VSE e-business Connectors User’s Guide*, SC33-8231

On the z/VSE host, the main component for the DB2–based connector support is the **DB2 Server for VSE 7.4** which runs per default in a **dynamic partition of class S**. By setting up a DB2 environment with the *DB2 Server for VSE* you have the whole range of supported DB2 functions available at your installation. This also includes a DB2 sample database.

For the processing of *Stored Procedures*, which actually perform the access to VSE/VSAM and DL/I data, a **Stored Procedure Server** must be defined. This server runs per default in a **dynamic partition of class R**.

Note:

Mapping VSAM Data to a Relational Structure

Before **VSE/VSAM data** can be accessed, it must first be **mapped into a relational structure**.

Mapping means splitting a VSE/VSAM record into columns (data fields) that have name, length, and other attributes. Optionally, a data view can be created which contains a subset of the fields contained within a map.

All this information is stored in the KSDS file VSE.VSAM.RECORD.MAPPING.DEFS located in the user catalog (VSESP.USER.CATALOG). This file is created automatically during Initial Installation or FSU. VSE/VSAM data can be mapped using the IDCAMS command RECMAP or by using the Java program provided.

For a detailed description of how to perform data mapping, refer to the chapter “Mapping VSE/VSAM Data to a Relational Structure” in the manual *z/VSE e-business Connectors User's Guide*, SC33-8231.

Initial Installation or FSU

The installation process for the *DB2 Server for VSE* is different for an Initial Installation compared to an FSU from VSE/ESA 2.6.x or 2.7.x to z/VSE 3.1.

During the Initial Installation process, the following message appears on the console screen:

```
IESI0063D DB2 IS REQUIRED FOR THE DB2-BASED CONNECTOR. DO YOU WANT
          TO INSTALL DB2 NOW? YES/NO
```

If you respond with YES, you must mount the **Extended Base Tape** (during processing of job DB2REST) and the *DB2 Server for VSE* is installed (restored) into sublibrary PRD2.DB2740.

FSU from VSE/ESA 2.6.x/2.7.x to z/VSE 3.1

The FSU dialog allows the installation of the *DB2 Server 7.4*. Note that an existing DB2 Server is not deleted and that your system may then include more than one DB2 Server. z/VSE 3.1 therefore provides these delete jobs:

- DELDB2 to delete DB2/BASE 7.1.0.
- DELDB272 to delete DB2/BASE 7.2.0.
- DELDB273 to delete DB2/BASE 7.3.0.
- DELDB274 to delete DB2/BASE 7.4.0.

DELDB272 for deleting DB2 Server 7.2, and DELDB273 for deleting DB2 Server 7.3.

Procedure USERBG includes the EXEC IVALPKEY statement to enable the *DB2 Server for VSE*. This statement must be activated after FSU completion via skeleton SKUSERBG available in VSE/ICCF library 59.

Refer to “DB2 Server for VSE 7.4” on page 82 for further FSU details.

If you have already installed a previous version of DB2 on your current system, a migration from your current DB2 to the *DB2 Server for VSE* is required to make use of the DB2-based connector support. For migration details, refer to the *Program Directory* of the *DB2 Server for VSE*.

Enabling the DB2 Server for VSE

To use the *DB2 Server for VSE*, you must first enable it via a **key**. You have two choices: you use the trial key provided (giving you access for 90 days) or use the key you get when buying a DB2 license. The startup procedure **USERBG** (skeleton SKUSERBG) includes the following (inactive) statement with the trial key:

```
EXEC IVALPKEY,PARM='PRODUCT=DB2 KEY=0000-1111-2222-3333-4444CUSTINF*
               O=C111-111-111'
```

where the asterisk (*) in the last position of the first line is the continuation character (column 72) and the second line continues at column 16.

You can use the statement as it is and just activate it via skeleton SKUSERBG available in VSE/ICCF library 59. This allows you to use the *DB2 Server for VSE* **for a trial period of 90 days**. After that time, you must replace the KEY and CUSTINFO information with the data you receive when buying a DB2 license. Otherwise, access to DB2 is denied. In case of a validation error, message 0S38I is displayed.

The trial period allows you to test the support provided via the DB2-based connector and access VSE/VSAM and DL/I data as well as using the DB2 sample data base available with the *DB2 Server for VSE*.

Once the trial period is activated by IVALPKEY, it is valid for 90 days. This is independent of the installation of the *DB2 Server for VSE*. If DB2 is installed 40 days later, for example, it can be used for 50 days only.

In case you decide not to use DB2 after the trial period, you can use the delete job DELDB274 (available in VSE/ICCF library 59) to remove the *DB2 Server for VSE 7.4* from your system. The job also deletes the *DataPropagator Relational Capture* component if installed. The delete job for the *DB2 Server for VSE 7.3* is DELDB273.

Enabling the DB2 Server for VSE with a separate job:

You can run a separate job to activate the *DB2 Server for VSE* without using the EXEC IVALPKEY statement in the startup procedure USERBG (though this is recommended). A sample job is shown below. It uses as example the key for the trial period and must be run each time an IPL is performed **before** the *DB2 Server for VSE* is started.

```
* $$ JOB JNM=IVALTRY,DISP=L,CLASS=0
// JOB IVALTRY
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
EXEC IVALPKEY,PARM='PRODUCT=DB2 KEY=0000-1111-2222-3333-4444CUSTINF*
               O=C111-111-111'
/&
* $$ EOJ
```

The EXEC IVALPKEY statement must be processed before each DB2 startup. Otherwise, DB2 cannot be activated.

Summary of DB2 Planning and Customization Tasks

This section provides is a summary (only) of the tasks required for setting up a running DB2 environment on the host side for accessing VSE/VSAM, DL/I, and DB2 data stored on the host. It is assumed that the *DB2 Server for VSE* has already been restored (installed) into sublibrary PRD2.DB2740. However, you should also refer to:

- The chapter “Customizing the DB2-Based Connector” in the manual *z/VSE e-business Connectors User's Guide*, SC33-8231, which provides detailed and step-by-step instructions.
- The Program Directory for the *DB2 Server for VSE*, which complements this information with the DB2 details needed to understand how DB2 is set up, and how it must be installed.
- The *DB2 System Administration* and the *DB2 Database Administration* manuals of the *DB2 Server for VSE*, which provide further details for setting up a DB2 environment.

1. Customize CICS TS so that it has access to the DB2 Server for VSE.
2. Customize TCP/IP so that your workstations are set up to use the DB2-based connector.
3. Customize the DB2 Server for VSE and the DB2 sample database for use by the DB2-based connector.

   Skeletons provided: **SKDB2VAR** (including member ARISIVAR.Z; refer to "Use of Installation Member ARISIVAR.Z" for details).
4. Set up your z/VSE environment for use with DRDA (Distributed Relational Database Architecture) support.

   Skeleton provided: **SKDB2VAR**.
5. Set up the DB2 Stored Procedure Server.

   Skeleton provided: **SKDB2SPS**.
6. Set up for DB2 Stored Procedures.

   Skeletons provided: see steps 7 and 8.
7. Customize the DB2-based connector for VSE/VSAM data access.

   Skeletons provided: **SKCRESTP, SKCPSTP, SKVSSAMP**.
8. Customize the DB2-based connector for DL/I data access.

   Skeletons provided: **SKDLISTP, SKDLICMP, SKDLISMP**.
9. Start DB2, and then start the DB2 Stored Procedure Server by placing the job DB2START in your VSE/POWER reader queue.

   Skeleton provided: **SKDB2STR**.
10. Establish a connection between your Web client and the z/VSE host, by installing and configuring DB2 Connect on your middle-tier platform.

Use of Installation Member ARISIVAR.Z

The parameters provided by ARISIVAR.Z determine the setup of your *DB2 Server for VSE* and the DB2 sample database provided. The DB2 Job Manager uses in addition job list control tables to control the installation process. Member ARISIVAR.Z has three parts:

- Preparation Steps

  The preparation steps are always required, for an installation as well as for a migration (where migration means that you already have installed an earlier DB2 version). The job list control table used is ARISITBP.
- Installation Steps

  The job list control table used for installation is ARISITBI.
- Migration Steps

  The job list control table used for migration is ARISITBM.

There are mandatory and optional steps. Depending on how you want to set up your *DB2 Server for VSE*, you may decide to not perform some of the optional steps. Note that this chapter and the *z/VSE e-business Connectors User's Guide* do not discuss migration steps. If you need to perform a migration, consult the *Program Directory* of the *DB2 Server for VSE.*

Implementation Details

ARISIVAR.Z is shipped in two formats which differ in content. The original ARISIVAR is shipped as Z-Book in PRD2.DB2740 and covers the whole range of DB2 definitions and functions available. In addition, VSE/ICCF library 59 provides skeleton **SKDB2VAR**. SKDB2VAR renames the original ARISIVAR.Z into ARISIVAR.ORIG and creates a new ARISIVAR.Z.

This new ARISIVAR.Z creates a **DB2 entry environment** for testing the connectors and the DB2 sample database provided. It is recommended to first create this less complex DB2 environment to ensure that the newly provided functions work correctly at your installation. Note that the skeleton assumes an IBM 3380 disk device for storage allocations. If you use a different disk device type, you must change these allocations accordingly (especially in case of FBA devices).

After successful testing, you may set up a more complex DB2 environment according to your requirements.

For a complete and detailed description of ARISIVAR.Z, its parameters and related control tables, refer to the *Program Directory* of the *DB2 Server for VSE*.

Submitting Skeleton SKDB2VAR

When submitted, skeleton SKDB2VAR performs the following major tasks:

- Defines for DB2 a separate disk volume (variable --V001--) and the related VSE/VSAM user catalog SQLCAT.
- Renames the original ARISIVAR.Z to ARISIVAR.ORIG.
- Catalogs the new ARISIVAR.Z into PRD2.DB2740.
- Places the startup job for the DB2 Job Manager (DB2JMGR) into the VSE/POWER reader queue.
- Places the startup job for DRDA (DB2DRDA) into the VSE/POWER reader queue.
- Places the startup job for the Stored Procedure Server (SPSERVER) into the VSE/POWER reader queue.

Starting and Processing ARISIVAR.Z

Note that the processing of ARISIVAR.Z is controlled by the **DB2 Job Manager**.

You start the Job Manager by releasing the job **DB2JMGR** in the VSE/POWER reader queue (placed there by skeleton SKDB2VAR). You must release the Job Manager once for each step: Preparation, Installation (or Migration).

As described in detail in the *Program Directory* for the *DB2 Server for VSE*, ARISIVAR.Z processes many parameters, globals and variables, to define DB2 characteristics and resources.

Major definitions include, for example:

- The DB2 sample database **SQLDS** which is defined on volume --V001-- (variable in skeleton SKDB2VAR).
- The **DB2 Server for VSE Help** component, the installation of which is controlled by the following variable:

  ```
  ARIS7ZJZ HELP      YES
  ```

  It is recommended that you install this component since it provides the help information for the *DB2 Server for VSE.* When this variable is processed, you are requested to mount the corresponding tape. This is the fourth tape (the extra tape) of the distribution tapes you received. You must replace the variable --V003-- with the cuu of the device on which you mount the tape.
- The creation of DB2 (work)files such as **BINDFILE, BINDWFILE**, and **SQLGLOB**.
- The setting of **CICS TS** parameters as required for a DB2 environment.

Overview of the VSAM Redirector Connector

The VSAM Redirector Connector enables VSE programs to access data on remote systems, in real-time.

Using the VSAM Redirector Connector:
- VSAM data can be migrated to other file systems or databases.
- Data can be synchronized on different systems with VSE VSAM data.
- VSE programs can work transparently with data on other file systems or databases.

A Java handler provides access to the specific file system or database on the remote system. For example, you can migrate your VSAM data into DB2 tables residing on a remote system, and your VSE programs will then work with this data, without requiring any changes to these VSE programs.

The VSAM Redirector Connector handles requests to VSAM datasets and redirects them to a different:
- Java platform (for example Linux on zSeries, Windows NT, Windows 2000, Windows XP).
- file system (for example DB2 or flat files).

Your existing z/VSE host programs that are:
- written in any language (COBOL, PL/I, ASSEMBLER)
- batch or CICS programs

can therefore work with migrated VSAM data without the need to amend and recompile these z/VSE host programs. The VSAM Redirector Connector manages all connections and data conversions.

The VSAM Redirector Connector consists of:
- The *VSAM Redirector Client* (installed on your z/VSE host).
- A *VSAM Redirector Server* installed on *each* Java platform.

VSAM request handlers (referred to simply as *request handlers*) are stored on the Java platform, and have a common interface. They are specific to the file system with which they work. For all connections, information about the file and the request are sent to the request handler.

For VSAM internal processing (such as the `POINT to END OF FILE`) changes have been made to VSAM so that the VSAM Redirector Client can perform its processing. The original VSAM cluster of a redirected file must, however, still exist on the z/VSE host.

For details of how to install and use the VSAM Redirector Connector, refer to the manual *z/VSE e-business Connectors User's Guide*, SC33-8231.

Overview of the VSE Script Connector

VSE Java Beans provide direct access to the z/VSE host from any kind of Java program (servlets, applets, EJBs, and so on) running on a *Java platform*. In addition, you can use the *VSE Script Connector* to access z/VSE host data from *non-Java* platforms. This is the main advantage of using the VSE Script Connector (although it can also be used to access z/VSE host data from Java platforms).

The VSE Script Connector is supplied as *part of* the Java-based connector. It can **only** be used in a 3-tier environment (explained in "Choosing Between 2-Tier and 3-Tier Environments" on page 139), and consists of:

- a VSE Script Client running on a Java or non-Java platform, and which can be either:
  - a user-written Java application (for example a Web-service).
  - a user-written non-Java application (for example a Windows C-program, a Windows CGI-program, or a COBOL application).
  - an office product, such as a word-processing or spreadsheet program (for example Lotus 1-2-3 or Lotus WordPro), where for example, a Visual Basic script is used to call a VSE Script.
- the VSE Script Server running on the middle-tier of a 3-tier environment, which interprets and executes VSE Script files.
- online documentation, including a programming reference manual.

The VSE Script Connector works in this general way:

1. The VSE Script Client calls a VSE Script, to make a request for data stored on the z/VSE host. These VSE Script (batch) files contain statements written using the VSE Script language, which is a special programming language. The VSE Script language can be used in any environment (even in Visual Basic scripts).
2. The VSE Script Server running on a Java-enabled middle-tier platform then reads, interprets, and translates, the VSE Script file statements into VSE Java Beans requests. The VSE Script Server uses the VSE Java Beans to connect to the VSE Connector Server running on the z/VSE host, and to forward the VSE Java Beans requests.
3. The VSE Connector Server accesses the required z/VSE data and functions, and sends the reply back to the VSE Script Server.
4. The VSE Script Server converts the data to the format that the VSE Script Client can use, and returns the data to the VSE Script Client.

For details of how to install and use the VSE Script Connector, refer to the manual *z/VSE e-business Connectors User's Guide*, SC33-8231.

Overview of z/VSE Support for Web Services and SOAP

SOAP is a standard, XML-based, industry-wide protocol that allows applications to exchange information over the Internet via HTTP.

XML is a universal format that is used for structured documents and data on the Web. It is independent of both the Web client's operating-system platform and the programming language used. HTTP is supported by all Internet Web browsers and servers.

SOAP combines the benefits of both XML and HTTP into one standard application protocol. As a result, you can send and receive information to/from various platforms.

Using Web browsers, you can view information contained on Web sites. However, using SOAP you can:

- combine the contents of *different* Web sites and services.
- generate a complete view of all the relevant information.

z/VSE supports the SOAP protocol and therefore allows you to implement Web services.

An example of using SOAP might be when a travel agent requires a combined view of the Web services covering hotel reservation, flight booking, and car rental. After the travel agent has entered the required data, all three Web services from the three different providers would be processed in one transparent step. This is an example of how a "Business-to-Business" (B2B) relationship can be implemented.

For details of how to implement SOAP in your z/VSE system, refer to the manual *z/VSE e-business Connectors User's Guide*, SC33-8231.

Summary of Available Connectors and Tools

Table 27 lists the connectors and tools you can use together with z/VSE. If you wish to download any of the tools that have a "Yes" entry in the column "Download From VSE Homepage?", you must start your Web browser and go to URL:

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

Table 27. Summary of z/VSE e-business Connectors and Tools

Tool	Description	Shipped With VSE?	Download From VSE Homepage?	Prerequisites
VSE Connector Client	Provides the VSE Java Beans class library, together with extensive online help, Javadoc, and coding samples for all kinds of Java programs (such as small applications, servlets, applets, and EJBs).	IESINCON.W in PRD1.BASE	Yes	VSE 2.5 or later, Java 1.3 or later
VSAM Redirector Server	Enables you to use your existing applications (e.g. COBOL programs) without any changes, to work with data on any Java-enabled platform. Provides a Java-based server which communicates with the VSAM Redirector Client on VSE.	IESVSMRD.W in PRD1.BASE	Yes	VSE 2.6 or later, Java 1.3 or later
VSE Navigator	A graphical user interface for use by the VSE systems programmer. Provides access to VSE file systems and VSE operator console. Comfortable mechanism for job creation, submission, and scheduling. Allows you to display system activity, monitor VSAM space usage, display label information, system tasks, and so on.	No	Yes	VSE 2.5 or later, Java 1.3 or later, VSE Connector Client
VSE Health Checker	A graphical user interface for use by the VSE systems programmer. Provides retrieval, display, and analysis, of VSE system data. You can use this tool to obtain a summary of the "health" of your VSE system.	No	Yes	VSE 2.6 or later, Java 1.4 or later, VSE Connector Client
VSE Virtual Tape Server	The server-part of the VSE Virtual Tape functionality. A virtual tape is represented by a file in AWSTAPE format, which can be either a VSE/VSAM file or a file on any Java-enabled platform.	VTAPESRV.W in PRD1.BASE	Yes	VSE 2.6 or later, Java 1.3 or later
VSE Script	Allows access to VSE functions and data from any kind of platform, Java and non-Java. VSEScript interprets and executes batch files (VSE script files), which contain commands to be executed on VSE.	No	Yes	VSE 2.7 or later, Java 1.3 or later, VSE Connector Client
VSE/VSAM JDBC Driver	A JDBC driver to issue SQL queries against VSAM files. Is part of the VSE Connector Client.	IESINCON.W in PRD1.BASE	Yes	Part of VSE Connector Client

Table 27. Summary of z/VSE e-business Connectors and Tools (continued)

Tool	Description	Shipped With VSE?	Download From VSE Homepage?	Prerequisites
VSE Web Services	Provides a SOAP server and a SOAP client, running in CICS on the basis of CICS Web Support. The server allows a Web service to be called that is implemented as CICS program from any kind of Web service client (e.g. Apache, AXIS, Microsoft .Net (C#, etc.). The client can call any Web service on, e.g., a WebSphere.	IESINCON.W in PRD1.BASE	Yes	Part of the VSE Connector Client
VSAM Maptool	Provides a Java-based dialog to generate VSAM data maps as they are used by the VSE/ESA e-business connectors and the IDCAMS function RECMAP. Maps can be either created by the user or imported from a VSE system, from an XML file, or from a COBOL copy book.	No	Yes	VSE 2.5 or later, Java 1.3 or later, VSE Connector Client
VSE Print Utility	A Java-based dialog to format and print VSE/POWER listings on PC or LAN-attached printers. Transfers the listings via AUTOFTP.	No	Yes	All VSE releases with TCP/IP Java 1.3 or later
TCP/IP for VSE/ESA Configuration Dialog	A Java-based dialog to create TCP/IP for VSE/ESA initialization and configuration members. Can also read existing IPINIT members.	IPNCFGE in PRD1.BASE	Yes	VSE 2.5 or later, Java 1.3 or later
JConVSE	Provides a Java-based dialog to automate your VSE jobs via an automated operator console. It is event or timer driven and is fully customizable.	No	Yes	VSE 2.5 or later, Java 1.3 or later, VSE Connector Client
JDataMig	Provides a Java application to read data from a VSAM file and write the data to a database, to spreadsheet format, or HTML.	No	Yes	VSE 2.5 or later, Java 1.3 or later, VSE Connector Client
Keyman/VSE	A VSE-specific SSL key management tool. It can directly upload generated RSA keys and certificates to VSE, and keep the server-side keyring in sync with client-side keyring files.	No	Yes	VSE 2.5 or later, Java 1.4 or later, VSE Connector Client

Chapter 11. TCP/IP, OSA, and HiperSockets Support

This chapter introduces the following z/VSE support:

- **TCP/IP for VSE/ESA**, described in "TCP/IP for VSE/ESA" below.
- **OSA Express**, described in "OSA Express Support" on page 156.
- **HiperSockets**, described in "HiperSockets Support" on page 160.
- **OSA-2**, described in "OSA-2 Support" on page 163.

TCP/IP for VSE/ESA

TCP/IP for VSE/ESA is a native implementation of Transmission Control Protocol/Internet Protocol (TCP/IP) for z/VSE. The product has been licensed from Connectivity Systems Incorporated.

TCP/IP for VSE/ESA provides the functions and the support described in detail in the manual *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*. This chapter provides an overview and planning information for setting up and using TCP/IP for VSE/ESA.

Function Overview

With TCP/IP for VSE/ESA you can, for example:

- Implement *Secure Sockets Layer (SSL)* security, including Client Authentication. This is described in detail in the manual *z/VSE e-business Connectors User's Guide*.
- Implement *HiperSockets* and *Hardware Crypto* support. For details, refer to "HiperSockets Support" on page 160.
- Transfer files to or from z/VSE using the File Transfer Protocol (FTP).
- Permit PC users access to server applications using TN3270.
- Logon to remote hosts using the Telnet protocol.
- Write (or port) TCP/IP applications to your z/VSE host using the socket interface.
- Print files from the VSE/POWER LST queue directly on any TCP/IP-attached printer or print workstation or LAN files on any VSE/POWER-attached printer.
- Use z/VSE as a Web server for corporate intranets or for the Internet.

Per default, TCP/IP for VSE/ESA is shipped enabled for **demonstration mode**, which allows a customer to try the whole product in a functionally very limited way. This mode is not suited for any production purposes.

To be able to use all the functions of TCP/IP for VSE/ESA, TCP/IP must be enabled **via a key** to be ordered from IBM. The complete set of TCP/IP functions is also referred to as **Application Pak**. For details on the functions provided, consult the following manual: *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*.

Installing TCP/IP for VSE/ESA

TCP/IP for VSE/ESA is automatically installed as part of z/VSE. After initial installation or FSU, TCP/IP for VSE/ESA resides in library PRD1.BASE and is available in **demonstration mode**.

If you finally decide not to use TCP/IP for VSE/ESA, run delete job DELTCPIP (provided in VSE/ICCF library 59) to remove the product from your system.

TCP/IP for VSE/ESA Startup

Partition Considerations

As shipped, z/VSE provides partition F7 with the storage values required for running TCP/IP for VSE/ESA:

```
ALLOC  SIZE  GETVIS

 16M    1M    15M    Environment A
 20M    1M    19M    Environment B
 32M    1M    31M    Environment C
```

TCP/IP for VSE/ESA can run in these predefined environments (A, B, and C). It can also run in a dynamic partition.

Startup Job Stream

The dialog described under "TCP/IP Configuration Dialog (Client Side)" creates a job stream for partition startup and includes statements similar to those shown below:

```
* $$ JOB JNM=TCPSTRT2,CLASS=7,DISP=K
* $$ LST CLASS=A DISP=D
    .
    .

// EXEC IPNET,SIZE=IPNET,PARM='ID=00,INIT=IPINIT00',DSPACE=2M
/&
* $$ EOJ
```

If DSPACE is not explicitly specified, a default of 1M is reserved for the VTAM application IPNET. For Telnet an additional 1M for VTAM is recommended.

Customizing TCP/IP for VSE/ESA

Installing the Product Key and Customer Number

You must install the product key and your customer number before you can use TCP/IP for VSE/ESA in **production mode**. You do this by running two assembler jobs of which examples are provided in the manual *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*.

TCP/IP uses this information during startup of the TCP/IP partition to enable those TCP/IP functions you are authorized to use according to your product key and customer number.

TCP/IP Configuration Dialog (Host Side)

The z/VSE dialog for configuring TCP/IP on the host side is described in detail in the manual *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*.

TCP/IP Configuration Dialog (Client Side)

z/VSE provides the *TCP/IP for VSE/ESA Configuration Dialog* to help customers to set up and configure TCP/IP for VSE/ESA for OS/2 and Windows clients. The dialog is shipped as part of TCP/IP and can be downloaded under OS/2 and Windows.

z/VSE provides library member IPINITnn.L which is called during startup by the // EXEC IPNET... statement included in the partition startup job stream. The dialog updates member IPINITnn.L by using selectively configuration data entered.

The configuration data to be entered includes:
- VTAM definitions (optional, for Telnet only)
- CICS definitions (optional, for Telnet only)
- TCP/IP definitions

  This includes the definition of:

  Links

  Daemons

  Routing Information

  General Information

The dialog creates output as follows:
- Updated (or newly created) member IPINITnn.L
- Partition startup job
- VTAM book with APPL statements (optional)
- Job for required CICS definitions (optional)
- An OS/2 or DOS batch file to upload the output files to the z/VSE host.

LE/VSE Support for TCP/IP Socket Interfaces

LE/VSE offers a C application programming interface (API) for TCP/IP sockets. This API for client/server communication across TCP/IP networks is compatible to the socket API of OS/390. The support is described in detail in the manual *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*.

Migration Considerations for TCP/IP for VSE/ESA

Refer to the manual *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information* for a detailed discussion of TCP/IP migration items.

OSA Express Support

The Open Systems Adapter (OSA) Express support is provided via the OSA Express adapter.

The OSA Express adapter is available for:
- IBM S/390 Parallel Enterprise Server G5 and G6 processors.
- IBM eServer zSeries processors.

Introduction

The OSA Express adapter is based on the QDIO (Queued Direct I/O) architecture and provides direct connectivity between z/VSE applications and other platforms on the attached network. The QDIO architecture allows a highly efficient data transfer since it eliminates the need for conventional I/O and interrupt processing. This results in accelerated TCP/IP data packet transmission. The OSA Express adapter consists of different features:
- Gigabit Ethernet
- Fast Ethernet
- ATM
- Token Ring

All features support the QDIO architecture. Additionally, the Fast Ethernet, ATM, and Token Ring feature can be configured to run either in QDIO or non-QDIO mode. To switch between the two modes, the *OSA/SF for VSE/ESA* program (which is part of z/VSE) is required.

The following table summarizes the OSA Express QDIO as well as the non-QDIO support. In a non-QDIO environment, either TCP/IP or SNA can be used as access method. As mentioned before, OSA/SF is needed if you plan to switch between the two modes.

Table 28. Overview of OSA Express QDIO and non-QDIO Support

Network	Mode	TCP/IP	SNA	OSA/SF required
Gigabit Ethernet	QDIO	yes	no	no
Fast Ethernet	QDIO	yes	no	no
Fast Ethernet	non-QDIO	yes	yes	yes
ATM (*)	QDIO	yes	no	no
ATM (*)	non-QDIO	yes	yes	yes
Token Ring	QDIO	yes	no	no
Token Ring	non-QDIO	yes	yes	yes

(*) z/VSE supports ATM in LAN Emulation mode only.

Invoking the Support

To use the OSA Express adapter, the following tasks must be performed:
1. Configuring IOCP
2. Defining OSA Express in z/VSE (IPL ADD statement)
3. Defining OSA Express in TCP/IP (DEFINE LINK statement).

IOCP Configuration

OSA Express is identified in the S/390 I/O configuration by its channel path identifier (CHPID). The channel type for QDIO is OSD.

```
CHPID PATH=FC,TYPE=OSD
CNTLUNIT CUNUMBR=D00,UNIT=OSA,PATH=FC
IODEVICE ADDRESS=(D00,3),CUNUMBR=D00,UNIT=OSA
IODEVICE ADDRESS=(D03,3),CUNUMBR=D00,UNIT=OSA
```

For non-QDIO, the TYPE would be OSE instead of OSD.

Note:

- When defining devices for OSD CHPIDs, it is important to consider the maximum number of subchannels per OSD CHPID. This means that the number of defined devices multiplied by the number of Logical Partitions (LPARs) that can access these devices must not exceed this maximum number. For example, if you have a 5-LPAR configuration and the maximum number of OSD devices is 240, this means you can define up to 48 OSD devices per OSD CHPID (48 x 5 = 240).
- Using device candidate lists can increase the number of devices that can be defined providing LPARs are excluded. For example, if the CHPID or device candidate list is limited to 3 LPARs, the maximum number of OSD devices is 80 (80 x 3 = 240). Also note that the candidate list should be specified explicitly, otherwise it defaults to all the LPARs defined on your RESOURCE statement in the IOCDS.

OSA Express Definition in z/VSE

To access the OSA Express adapter in QDIO mode, you need three OSA Express devices (read, write, and datapath). You must specify these devices in the IOCP generation macro with device type OSA.

For z/VSE, the corresponding new device type **OSAX** has been introduced. It must be used for all the devices specified in the IOCP generation macro with channel type OSD and can be specified for z/VSE with the hardware configuration dialog. For details of this dialog, refer to the section “Connecting a LAN to an OSA Express Device in QDIO Mode” in the manual *z/VSE Networking Support*, SC33-8235.

Following are examples of how to specify OSAX devices:

```
ADD D00:D02,OSAX
```

or

```
ADD D00,OSAX
ADD D01,OSAX
ADD D02,OSAX
```

All devices that are to be used later on must be added during IPL. If you want to specify a second DEFINE LINK within the same or a different TCP/IP partition, you have to add three more OSAX devices:

```
ADD D03:D05,OSAX
```

In case of non-QDIO, the device type for z/VSE must be OSA (as for the OSA-2 adapter). In non-QDIO mode, the OSA Express adapter behaves like an OSA-2 adapter and you have to specify only two devices of type OSA.

OSA Express Definition in TCP/IP

To use the OSA Express adapter in QDIO mode, specify the TCP/IP DEFINE LINK command as follows:

```
DEFINE LINK,ID=...,TYPE=OSAX,
       DEV=cuu1 (or DEV=(cuu1,cuu2)),
       DATAPATH=cuu3,
       IPADDR=addr,
       MTU=max. 9000,        (default: 1492)
       PORTNAME=(8byte)name,
       FRAGMENT={NO|YES}     (default: NO)
                             (YES not supported by OSA Express adapter)
       ROUTER={NONE|PRIMARY|SECONDARY}     (default: NONE)
```

Explanations:

1. cuu1,cuu2 must be an even/odd pair. If cuu2 is omitted, cuu1 + 1 is taken as default.
2. An IP address can only be used once per *physical* OSA Express adapter, that is, a second DEFINE LINK for the same physical OSA Express adapter must contain a different IP address.
3. When you specify PORTNAME, you assign a name to the port of the OSA Express adapter. The first user who initializes the adapter determines the name of the port. Subsequent users within the same or different operating systems must use the same name. Starting with a certain microcode level, the OSA Express adapter requires the specification of PORTNAME. It is therefore recommended to always specify PORTNAME. See also "Example 1" and "Example 2" below.

   If the DEFINE LINK fails with message 0S39I .... REASON=0032, the PORTNAME specified does not match the name specified initially.
4. The OSA Express adapter provides a routing facility that processes IP packets for an unknown IP address. The routing facility is activated via the ROUTER parameter.
   - If the OSA Express adapter receives IP packets for an unknown IP address, it will forward these packets to the link that has been defined as PRIMARY router.
   - If a PRIMARY router has not been defined, the OSA Express adapter will forward these IP packets to the link that has been defined as SECONDARY router.
   - If no router has been defined, the OSA Express adapter will discard the IP packets for the unknown address.

Example 1:

```
DEFINE LINK,ID=...,TYPE=OSAX,
       DEV=(D00,D01),
       DATAPATH=D02,
       IPADDR=9.164.155.90,
       MTU=9000,
       PORTNAME=OSAXPORT
```

Example 2:

```
DEFINE LINK,ID=...,TYPE=OSAX,
       DEV=D04,
       DATAPATH=D03,
       IPADDR=9.164.155.99,
       MTU=1492,
       PORTNAME=OSAXPORT
```

Further DEFINE LINK information:

- Several LINKs of type OSAX may be defined within one TCP/IP partition.
- The three OSA Express devices used for the DEFINE LINK must be unique within z/VSE.
- If running under VM, the three devices describing the OSAX link must be unique within VM.

Additional TCP/IP considerations:

- If you want to change the properties of an OSAX LINK you have to do a DELETE/DEFINE LINK. The MODIFY command is not supported.
- The DEFINE ADAPTER is not needed.

Partition resources required:

For each DEFINE LINK of an OSAX device, the TCP/IP partition requires 1050 KB partition GETVIS (ANY) space and 1050 KB for SETPFIX (ANY). It may therefore be necessary to adjust the TCP/IP startup procedure accordingly.

Further Documentation

The following manuals provide additional information:

- For general planning information and OSA/SF consult:

  The manual *Planning for the S/390 Open Systems Adapter Feature*, GC23-3870, and the *VSE/ESA Open Systems Adapter Support Facility User's Guide*, SC28-1946.
- For OSA Express information consult:

  The manuals *S/390 OSA Express Customer's Guide and Reference*, SA22-7403, and *zSeries 900 OSA Express Customer's Guide and Reference*, SA22-7476.

HiperSockets Support

z/VSE supports high-speed TCP/IP communication among logical partitions (LPAR) and virtual machines using HiperSockets. The HiperSockets support is available on:

- IBM eServer zSeries processors.
- IBM S/390 Parallel Enterprise Servers G5 and G6 processors (when running under z/VM 4.2 or later using z/VM's virtual HiperSockets).
- IBM Multiprise 3000 processors (when running under z/VM 4.2 or later using z/VM's virtual HiperSockets).

z/VM 4.2 or later supports *virtual HiperSockets* for use by guest operating systems which support HiperSockets (such as z/VSE 3.1). This virtual HiperSockets environment for guest systems is available on:

- IBM eServer zSeries processors.
- IBM S/390 Parallel Enterprise Server G5 or G6 processors.
- The IBM Multiprise 3000 processor.

HiperSockets provide the fastest TCP/IP communication between operating systems running in the LPARs (Logical Partitions) or virtual machines of a processor which provides Hipersockets support. Like the OSA Express adapter support, the Hipersockets support is based on the QDIO (Queued Direct I/O) architecture.

Support Overview

z/VSE supports up to four HiperSockets each of which acts like a TCP/IP network (LAN) within a processor. Each HiperSockets is defined by a HiperSockets channel path identifier (CHPID).

To use a HiperSockets connection, three HiperSockets I/O devices are required: a read control device, a write control device, and a data device for data exchange. This is basically the same as for an OSA Express adapter. Only the restriction to use an even/odd pair for the read/write control devices is dropped for HiperSockets. The sending and receiving I/O devices must be on the same CHPID. The processor maintains an IP address table for each HiperSockets. This table represents an emulated LAN. When a TCP/IP link is started, the link is registered in the table with its IP address and deleted when the TCP/IP link is closed.

Real LANs have a maximum frame size predefined by the architecture. With HiperSockets you can define the maximum frame size for each of the four possible HiperSockets yourself as shown under "IOCP Configuration."

Invoking the Support

To use HiperSockets, the following tasks must be performed:

1. Configure IOCP.
2. Define HiperSockets I/O devices in z/VSE (IPL ADD statement).
3. Define HiperSockets I/O devices/links in TCP/IP (DEFINE LINK statement).

IOCP Configuration

Each HiperSockets requires the definition of a channel path identifier (CHPID). The following rules and characteristics apply:

- You can define four HiperSockets (IQD CHPIDs) per processor and share them among LPARs.
- The CHPID type for a HiperSockets definition is IQD.
- You can define up to 16 control units on each IQD CHPID.
- You can connect up to 256 devices to an IQD control unit.
- You can define the maximum frame size for IQD CHPIDs with the OS parameter.

Relationship between OS parameter, frame size, and MTU (Maximum Transmission Unit):

1. OS = 00 (default)

   Maximum frame size/MTU = 16KB/8KB
2. OS = 40

   Maximum frame size/MTU = 24KB/16KB
3. OS = 80

   Maximum frame size/MTU = 40KB/32KB
4. OS = C0

   Maximum frame size/MTU = 64KB/56KB

IOCP Example:

```
CHPID PATH=(FC),SHARED,PARTITION=(...),TYPE=IQD,OS=40
CHPID PATH=(FD),SHARED,PARTITION=(...),TYPE=IQD
CNTLUNIT CUNUMBR=500,PATH=(FC),UNIT=IQD
CNTLUNIT CUNUMBR=600,PATH=(FD),UNIT=IQD
IODEVICE ADDRESS=(500,16),CUNUMBR=500,UNIT=IQD
IODEVICE ADDRESS=(600,3),CUNUMBR=600,UNIT=IQD
```

HiperSockets Device Definitions in z/VSE

To use a HiperSockets connection, three HiperSockets devices are required (a read, a write, and a data device). These devices must be specified in the IOCP definition with channel type **IQD**. For z/VSE, the corresponding device type is **OSAX** (IPL ADD command). To distinguish HiperSockets devices from OSA Express devices a **mode of 1** must be specified as shown in the example below:

```
ADD 500:515,OSAX,1
ADD 600,OSAX,1
ADD 601,OSAX,1
ADD 602,OSAX,1
```

Note that all required devices must be added during IPL. The “Configure Hardware” dialog (Fastpath 241) supports the definition of HiperSockets devices. For details, refer to the section “Defining a HiperSockets Device” in the manual *z/VSE Networking Support*, SC33-8235.

HiperSockets Device and Link Definitions in TCP/IP

To use a HiperSockets connection, specify device and link information in the TCP/IP DEFINE LINK command as follows:

```
DEFINE LINK,ID=...,TYPE=OSAX,
       DEV=cuu1,            (or DEV=(cuu1,cuu2))
       DATAPATH=cuu3,
       IPADDR=addr,
       MTU=xxxx,            (default: as specified in the OS parameter)
       FRAGMENT={NO|YES}    (default: NO)
                            (YES not supported by HiperSockets)
```

These definitions are very similar to those required for OSA Express, except that:

1. HiperSockets do not require a PORTNAME (as does OSA Express).
2. The MTU size must not exceed the MTU size specified in the OS parameter (CHPID definition). The default MTU size is the size specified in the OS parameter (CHPID definition).

Additional Information

Partition resources required:

For each DEFINE LINK of an OSAX device, the TCP/IP partition requires

- Partition GETVIS (ANY) space as follows:

  About 400KB when defining OS=40 and about 1050KB when defining OS=C0.
- SETPFIX (ANY) space as follows:

  About 400KB when defining OS=40 and about 1050KB when defining OS=C0.

It may therefore be necessary to adjust the TCP/IP startup procedure accordingly.

Error Messages:

In case of errors, message 0S39I is issued which includes two new reason codes for HiperSockets:

- REASON=x'0033'

  The HiperSockets device does not support HiperSockets features. This is most likely a hardware error.
- REASON=x'0034'

  The HiperSockets device could not be enabled. This is most likely a hardware error.

Further Documentation

The following Redbook offers a broad description of the HiperSockets architecture and microcode function. It provides planning and implementation information, as well as setup examples:

zSeries HiperSockets, SG24-6816

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

www.redbooks.ibm.com

For HiperSockets support under VM, consult the corresponding VM documentation.

OSA-2 Support

The terms listed below are used frequently in the following discussion of the z/VSE OSA support.

Terminology Used

APPN	Advanced Peer-to-Peer Networking
ATM	Asynchronous Transfer Mode
ENTR	EN stands for Ethernet, TR for Token Ring
FDDI	Fiber Distributed Data Interface
IOCP	Input/Output Control Program
SNA	Systems Network Architecture

z/VSE supports the S/390 Open Systems Adapter 2 (OSA-2) in **SNA and TCP/IP mode**. OSA-2 in SNA mode provides the connectivity to support clients on LANs that use Systems Network Architecture (SNA) and Advanced Peer-to-Peer Networking (APPN).

OSA-2 provides the ability to connect to a wide range of network protocols including Token Ring, Ethernet, FDDI and ATM.

- An ENTR OSA-2 has two ports, either of which can be attached to an Ethernet or a Token Ring LAN.
- An FDDI OSA-2 has one port.
- An ATM OSA-2 has one physical port, but can be used in LAN emulation mode as two logical ports.

The SNA mode services clients that are connected to a LAN that is directly attached to an OSA-2, or through an emulated Ethernet or Token-Ring LAN across an ATM-based network. An OSA-2 allows LAN-based clients or ATM-based LAN emulation clients, using the SNA/APPN network protocol, to access the host applications and facilities available through VTAM on the host. To VTAM, OSA-2 appears as an external communication adapter (XCA). OSA-2 is defined to VTAM as XCA major node and as Switched Network (SWNET) major node.

The OSA-2/TCP/IP configuration details are provided in the manual *TCP/IP for VSE/ESA Installation Guide*.

Defining an OSA-2 to Your System

The major tasks involved to define an OSA-2 to your system are outlined below. These tasks are described in detail in the manuals listed under "Further Information" on page 165:

1. Define each OSA-2 in the system hardware I/O configuration using IOCP statements.
2. Use the z/VSE *Configure Hardware* dialog to define:
   a. Each OSA-2 to z/VSE. The dialog creates ADD statements for the new device types OSA and OSAD and the corresponding VTAM books.

      There are two device types to be specified for an OSA-2 (with different addresses). Their meaning is as follows:

      OSA is the device type for data transfer.

      OSAD is the device type for use by OSA/SF.
   b. Each OSA-2 to VTAM as XCA and SWNET major node.

Further details about the dialog are provided in the manual *z/VSE Administration*.

Once you have defined an OSA-2 to your system, you can use the Open Systems Adapter Support Facility (OSA/SF) to customize it for SNA or TCP/IP with or without LAN emulation.

Introducing OSA/SF

z/VSE provides the **Open Systems Adapter Support Facility (OSA/SF)** for customizing and managing an OSA-2 environment. OSA/SF is shipped as part of z/VSE and is automatically installed into the z/VSE sublibrary PRD1.BASE.

OSA/SF requires an additional sublibrary, PRD2.OSASF, for storing and processing data and work files.

You can communicate with OSA/SF through:
- An OS/2 workstation
- OSA/SF commands

OS/2 Workstation

To use an OS/2 workstation, you must first download from the z/VSE host the necessary files to create the OSA/SF graphical user interface (GUI) on OS/2. You can use the GUI for any OSA-2 configuration and management task.

OSA/SF Commands

Instead of using the GUI, you can manage and customize an OSA-2 by submitting OSA/SF commands from the host (z/VSE). All the tasks done from the OSA/SF GUI can also be done by submitting OSA/SF commands.

Setting Up OSA/SF

Before you can use OSA/SF, you must prepare z/VSE so that you can run OSA/SF jobs in static or dynamic partitions of z/VSE. OSA/SF includes the following jobs (available in VSE/ICCF library 59):

OSA/SF Job:	Explanation:
IOAMAIN	This job must be active in order to use OSA/SF. It must be running when submitting OSA/SF commands but also when using the GUI.
IOACMD	This job runs the command EXEC for submitting OSA/SF commands from the host (z/VSE).
IOAINATM	This job is required for installing ATM parameters.
IOASNAVS	This job is required for installing an SNA image on an OSA-2.
IOAXHRUN	This job is started whenever the GUI becomes active. It establishes the host-GUI connection.

To use the jobs IOAMAIN, IOACMD, IOAINATM, and IOASNAVS, you must prepare them as follows:
1. Modify them as required for your environment.
2. Submit them to the VSE/POWER reader queue.
3. Release them as needed for an OSA/SF task.

Defining OAT Entries and Installing the SNA Image Using OSA/SF

A typical task for SNA-mode customization is the definition of entries in the OSA Address Table (OAT) and installing the SNA image on OSA-2.

Each OSA-2 maintains an OAT to track the source and destination of the data being transferred through each of its ports in each of its modes of operation. An OAT entry consists of a base segment and an extension. An SNA entry type has an SNA extension.

For the entry definition and SNA-image installation you can use the OSA/SF GUI or submit OSA/SF commands. The advantage of the GUI is that you are asked for the input required and need not directly update the OAT file.

Using OSA/SF Commands:

This requires that you release jobs IOAMAIN and IOACMD so that you can submit OSA/SF commands to z/VSE. As a final step, you would release job IOASNAVS to install the SNA image on the OSA-2.

Using the OSA/SF GUI:

This requires that you release job IOAMAIN so that you can use the GUI. Job IOAXHRUN is automatically started to establish the host-GUI connection. You can then do the OAT changes as guided by the panels but you may defer the installation of the SNA image to a later time.

As a final step, you would activate the SNA image on the OSA-2.

For a detailed task description and related planning information refer to the manuals listed under “Further Information”.

Further Information

Consult the following manuals if you plan to set up an OSA-2 environment for z/VSE:

- *Planning for the System/390 Open Systems Adapter Feature*, GC23-3870

  This manual includes planning information on hardware, IOCP, VTAM, TCP/IP, and OSA/SF. It discusses planning for OSA-2 in a z/VSE environment and includes checklists, requirements, and an example.
- *VSE/ESA Open Systems Adapter Support Facility User’s Guide*, SC28-1946

  This manual provides information on OSA/SF including setup procedures for its OS/2 graphical user interface (GUI), the syntax of the OSA/SF commands, and instructions for customizing OSA-2.

Chapter 12. Using Data Spaces and Virtual Disks

This chapter introduces and provides planning information for the capabilities and the support of:

Data Spaces

Virtual Disks

Note: The z/VSE operating system can execute in 31-bit mode only. It does not implement z/Architecture, and specifically does not implement 64-bit mode capabilities.

In general, these capabilities allow for more "data in memory" which reduces the number of physical input/output (I/O) operations to disk devices. This provides better response times for online applications and better elapsed times for applications running in batch. These advantages require, however, that **sufficient real storage** is available. Refer to "z/VSE Exploitation Overview" on page 260 for additional details about "data in memory".

Consult the manual *VSE/ESA Extended Addressability* for details and information about planning and writing application programs that make use of these capabilities.

This chapter contains these main sections:

Data Space Support

Requirements

The support for data spaces is based on the following:

- A zSeries or ESA/390 processor, as listed in "Processor Support" on page 4.
- The use of the **High Level Assembler** programming language for application programs making use of this support.

Data spaces are data-only spaces that can hold up to 2GB of data. A data space can be considered as an area in virtual storage like an address space. Unlike an address space, however, a data space can **only** contain data. It does not contain shared areas as the following diagram shows:

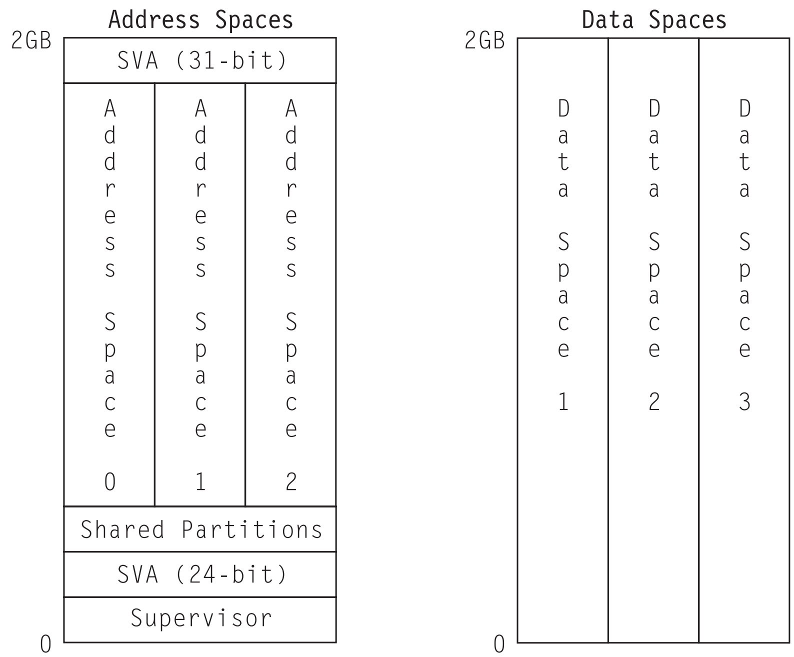


Figure 13. The Difference Between Address Spaces and Data Spaces

By using data spaces, you move data closer to a program since the data also resides in virtual storage and not on disk. Data Spaces are an extremely flexible solution to problems related to accessing large amounts of data. They also provide integrity and isolation for the data they contain in much the same way as address spaces provide integrity and isolation for code and data they contain. z/VSE includes macros allowing you to write programs to create, control, and delete data spaces as required.

Notes:

1. Only programs in 31-bit addressing mode can create data spaces and access data located above the 16MB line in a data space.
2. Programs in 24-bit addressing mode can access data located below the 16MB line in a data space.

z/VSE data spaces have the following characteristics:

- Program code cannot execute in a data space. But a program can reside in a data space as nonexecutable code.
- Data spaces cannot contain shared areas or control blocks.
- The owner of a data space is the task that creates the data space.
- Programs accessing data spaces must be in access register (AR) mode.

  Note that most supervisor services cannot be called in AR mode. For example, it is not possible to issue an I/O request from or to a data space.
- Data spaces improve data sharing capabilities for:
  - Selected users (SCOPE=ALL)
  - System wide (SCOPE=COMMON)
- Only key 0 programs can create data spaces of TYPE=ALL or COMMON.

z/VSE data spaces are a subset of the MVS data space support and are implemented via MVS compatible interfaces and concepts such as source code compatible macro expansions.

Virtual Disk Support

Requirements

With the virtual disk support it is possible to have (temporary) data reside in virtual storage (in a data space) and not on a real disk device. Each virtual disk resides in its own separate data space. A virtual disk in z/VSE has been implemented by emulating an FBA (Fixed Block Architecture) disk device. You can use it like a real FBA device, without being required to change code in your applications. In addition, applications running in different partitions on the same z/VSE system can share virtual disks like real disk devices.

Since data can be accessed at memory speed, response times for transactions and throughput for jobs and applications accessing such data may improve significantly.

With z/VSE, you can define up to 128 virtual disks.

Where to Use Virtual Disks

Since virtual disks are not permanent, they should be used for files that easily can be recovered in case of loss (because of a power failure, for instance). These include:

- **Temporary work files or test files**

  All types of temporary work files can be used on a virtual disk. These can be work files for Assembler programs, compilers, sort programs, and so on. See also skeleton SKWRKFIL.
- **VSE/VSAM space and user catalogs**

  As with a real disk, the space on a virtual disk can be split between VSE/VSAM space and other files. If you have VSE/VSAM files on a virtual disk, note that:
  1. A VSE/VSAM user catalog for files on a virtual disk *must* also reside on a virtual disk. This is also true for VSE/VSAM workfiles.
  2. A multi-volume file *must* be completely allocated either on virtual disks or on real disks. It cannot be allocated on both.
  3. You should use the BACKUP and RESTORE functions of VSE/VSAM to move VSE/VSAM objects between virtual disks and real disks, if, for example, VSE/VSAM test files are to be created on a virtual disk.
- **VSE libraries**

  You can place VSE libraries into VSE/VSAM or non-VSE/VSAM space on a virtual disk. To reduce real I/O requests to frequently-used applications or data, copy them to the virtual disk by using the librarian program LIBR. Refer also to "Creating a VSE Library that Resides on a Virtual Disk" on page 172.

"z/VSE Virtual Disks" on page 265 provides further details about when and where to use virtual disks.

Where Not to Use Virtual Disks

Since virtual disks are volatile, they cannot be used for data that must be kept if the system goes down or for data that is needed during system startup. Therefore, do **not** use a virtual disk to hold files such as the following:

- Control file
- Text repository file
- CSD file

- Page data sets
- System recorder file
- System history file
- Hardcopy file
- VSE/VSAM master catalog

A z/VSE virtual disk cannot be shared among z/VSE systems. It can, however, be shared by different partitions on the same z/VSE system.

Performance Considerations for Virtual Disks

Virtual disks in data spaces are subject to the same paging rules as virtual storage for address spaces. To benefit from virtual disks, you must have sufficient real (processor) storage available to avoid a possible paging bottleneck. If the real storage available is insufficient, I/O requests to real disks are replaced by I/O requests to the page data set.

Further Planning Information for Data Spaces and Virtual Disks

Storage Considerations

With the AR and JCL command **SYSDEF** you define the maximum amount of virtual storage that can be allocated to data spaces. However, the specified storage may not always be available for data spaces. The system may use it also for allocating partitions if required. The storage specified for SYSDEF is taken from the value specified for VSIZE. Consequently, you have to increase the **VSIZE** definition to accommodate the storage requirements for data spaces (and virtual disks) as follows:

- Add the size of all data spaces that might be used concurrently and needed besides those needed for virtual disks.

  Note: For each data space, virtual storage is allocated in multiples of 32KB. For example, if you define a data space of 40KB, the available VSIZE decreases by 64KB although the usable data space size is 40KB.
- Add the size of all virtual disks that might be used *concurrently*.

Add the resulting value to your current VSIZE.

Once you have determined the total value for VSIZE, it may be necessary to increase the current size of your page data set (defined through the IPL **DPD** command).

You can change both, VSIZE and DPD values, through the *Tailor IPL Procedure* dialog.

Command and Macro Support

Data Space Commands

Consult the manual *z/VSE System Control Statements* under “SYSDEF”, “QUERY”, and “MAP” for a detailed description of the commands listed below.

- **SYSDEF** command

  With the SYSDEF command you define the maximum amount of virtual storage which may be allocated to data spaces (and virtual disks) if available. Optionally, it allows you to specify values such as the default size of a data space and the maximum number of data spaces.

- **QUERY** and **MAP** command
  - With the QUERY command you can display information about data spaces and their characteristics.
  - With the MAP command you can display the amount of virtual storage allocated to data spaces.

Data Space Macros

Consult the manual *z/VSE System Macros Reference* for a detailed description of the macros listed below.

- **ALESERV** macro

  A program uses the "ALESERV" macro to control the data space entries in the access list.
- **ATTACH ALCOPY** macro

  A program uses the "ATTACH ALCOPY" macro to transfer data space information to a subtask to be attached.
- **DSPSERV** macro

  A program uses the "DSPSERV" macro to create, delete, and control data spaces.
- **SDUMPX** macro

  A program uses the "SDUMPX" macro to dump storage from a data space.
- **SYSSTATE** macro

  A program uses the "SYSSTATE" macro to set and test the address space control (ASC) mode.

Virtual Disk Commands and Macros

Consult the manual *z/VSE System Control Statements* for a detailed description of the commands and the manual *z/VSE System Macros Reference* for a detailed description of the macro listed below.

- **ADD** command

  With the "ADD" command, you add a virtual disk (device type FBAV) to z/VSE in the same way as a real disk device.

  The *Hardware Configuration* dialog supports virtual disks through device type FBAV.
- **VDISK** command

  With the "VDISK" command, issued from the BG partition, you create the data space required for a virtual disk.
- **VOLUME** command

  With the "VOLUME" command, you can retrieve information about existing virtual disks.
- **GETVCE** macro

  A program can use the "GETVCE" macro to retrieve device characteristics of a virtual disk.
- **QUERY** command

  With the "QUERY" command, you can display data space limits and other data space characteristics.

Examples of Virtual Disk Usage

Following are two examples of virtual disk usage:

1. "Creating a VSE Library that Resides on a Virtual Disk" on page 172.
2. "Label Area on Virtual Disk" on page 172.

Creating a VSE Library that Resides on a Virtual Disk

The following steps are an example of how to create and use a VSE library on a virtual disk:

1. Define a virtual disk in the IPL startup procedure with the command:
   ```
   ADD 234,FBAV
   ```
   You can use the *Tailor IPL Procedure* dialog to add the command.
2. Catalog the following procedure (named VDISK) into a sublibrary so that you can run it any time required.

   The procedure defines a library on a virtual disk with address 234 and a size of 3000 blocks. As a next step, it copies macros (A-Books) into the virtual library. These macros, it is assumed, are accessed frequently by compile jobs.
   ```
   // VDISK UNIT=234,BLKS=3840
   // DLBL MYLIB,'MY.TEMP.LIBRARY'
   // EXTENT SYS019,,1,0,10,3000
   // ASSGN SYS019,234
   // EXEC LIBR
   DEF L=MYLIB
   DEF S=MYLIB.MYSUBLIB
   CONN S=PRD1.MACLIB:MYLIB.MYSUBLIB
   COPY MAP*.A
   /*
   ```
3. You can activate the procedure from the BG partition with:
   ```
   EXEC PROC=VDISK
   ```
4. It is recommended that the labels are not included in the standard label procedure (STDLABEL) and that temporary LIBDEF statements are submitted with the jobs accessing the virtual library.

Label Area on Virtual Disk

Notes:

1. Under this version of z/VSE, the label area resides automatically on virtual disk FDF.
2. Do not delete the ADD FDF,FBAV statement in the IPL procedure since it is required for system startup. The same applies to the VDISK statement in the $0JCL procedure:
   ```
   // VDISK UNIT=FDF,BLKS=2880,VOLID=VDIDLA,USAGE=DLA
   ```

Labels are used to identify files and libraries and are stored in the system's label information area. This area is located on a disk device (DOSRES). For each first-time access (OPEN) to a file or library a search for the corresponding label on disk is required. In great numbers, such search operations for label information can become time-consuming and degrade performance.

The search time for labels can be reduced considerably by moving the label area from disk into virtual storage. This is done by defining a **virtual disk** for holding the label area. Note that a virtual disk is a temporary storage media, that is, the data stored on it is lost in case of a system restart (IPL). Since the label area and its content are recreated with each startup, this should cause no problem. It requires, however, that the procedures run during startup for the label area reflect the latest status of the system's label information.

Planning Information

It is necessary to check whether your VSIZE value is large enough for the SYSDEF definition to accomodate the virtual disk. If you have to increase the VSIZE, it may be required to increase the size of the page data set as well. Page data set extents are defined through DPD commands. Both, the VSIZE specification and the DPD commands are part of the IPL startup procedure and you can modify them using the *Tailor IPL Procedure* dialog.

Since the layout of a virtual disk reflects an FBA disk device, the system allocates blocks (of 512 bytes each) for the label area. The allocation of a virtual disk is done in multiples of 960 blocks. This means that this value is also the minimum size possible for the label area. If the value specified for BLKS in the VDISK statement is equal or lower than 960, the system always allocates 960 blocks. 960 blocks provide space for about 3000 labels. This is an average value, however, which may be higher or lower depending on such parameters as the number of partitions and the number of labels stored in each subarea.

The maximum size supported for the label area is 2880 blocks.

Background Information for Label Area Size

The label processing routine uses an 90-byte table to maintain 2KB label subareas. These 2KB label subareas (also called label area segments) are the allocation units to store label information records. The table can thus handle 720 (8 times 90) 2KB subareas.

Each 2KB subarea is mapped to four 512-bytes FBA blocks. This results in a maximum of 2880 (4 times 720) FBA blocks.Two FBA blocks are required for the VOL1 label residing at the very beginning of the virtual disk. These two FBA blocks cannot be used to store label information – actually four FBA blocks, because label processing considers four consecutive FBA blocks as one allocation unit. Therefore, a maximum of 2876 (4 times 719) blocks are available to store label information. This value can be decreased by the VTOC which resides at the very end of the virtual disk. Eight FBA blocks are the minimum (and the default if USAGE=DLA has been specified in the VDISK command) to hold the VTOC.

Each 2KB allocation unit holds a maximum of 19 label records in case of sequential or VSE/VSAM files (or 24 label records in case of tape files). Hence, under the assumption that we deal with sequential or VSE/VSAM files and that each 2 KB label subarea is filled up completely, 13661 (19 times 719) labels can be stored.

However, this is a theoretical upper limit. The real upper limit depends on:

- The number of active partitions using partition labels (temporary, permanent, or free-usage).
- The number of labels stored for each label group (such as BG permanent labels).

If, for example, we have 100 dynamic partitions, each writing one temporary and one permanent partition label, then we have 200 labels occupying 200 label subareas (from a total of 712). If, on the other hand, we have 5 dynamic partitions, each writing 40 temporary partition labels, then we have 200 labels occupying only 15 label subareas.

9000 labels is a reasonable average value for a label area with the maximum size of 2880 blocks.

Theoretically, you can define a much larger virtual disk than the 2880 blocks supported for the label area and use the additional space for storing other data. However, this is not recommended and it is preferable to define a separate virtual disk for holding the label area only.

Where to Find Further Information

For a detailed description of the ADD, DLA, SYSDEF, and VDISK commands or statements, refer to the manual *z/VSE System Control Statements*.

For a detailed description of skeleton SKJCL0, refer to the manual *z/VSE Administration* under “Skeletons for Starting Up BG Partition”. This manual also describes the *Tailor IPL Procedure* dialog under “Tailoring the IPL Procedure”.

Improving Virtual Storage Management for Application Programs

Additional programming enhancements are available which further improve virtual storage management for application programs. The support is based on zSeries and ESA/390 processor functions and the use of the High Level Assembler.

These enhancements include:
- Linkage stack functions
- Callable cell pool services

“Linkage Stack Functions” and “Callable Cell Pool Services” in the manual *VSE/ESA Extended Addressability* describe these enhancements in detail. The following paragraphs provide introductory information.

Linkage Stack Functions

The linkage stack is an area of protected storage that the system gives to a program to save status information in case of a branch or a program call. Each VSE task has its own linkage stack available for all programs running under this task.

The saved status information includes the contents of the 16 general purpose registers (GRPs), the 16 access registers (ARs), the PSW (program status word), and other relevant system information. Instructions are available to branch and save and to return and restore the saved information, as well as to access the linkage stack entries and their contents.

The linkage stack support is required when using the callable cell pool services introduced below.

Callable Cell Pool Services

Cell pool services are available through the CALL macro interface. The services can be used to manage virtual storage located in the primary address space, any other address space, or in data spaces.

A cell pool is an area of virtual storage that is subdivided into fixed-sized areas of storage called **cells**. A cell pool contains an anchor, at least one extent, and any number of cells of the same size. You define these resources and their size by calling the cell pool services through the CALL macro.

You obtain the virtual storage for the cell pool through either the GETVIS or the DSPSERV macro.

Chapter 13. Dynamic Partition Support

This chapter provides planning information relevant if you want to use dynamic partitions at your installation. Before you read this chapter you should be familiar with the concept of dynamic partitions as described under "Static and Dynamic Partitions" on page 50.

Dynamic Partition Support Summary

This section is a summary of the system support provided by z/VSE for dynamic partitions. It mentions items such as dialogs, skeletons, and statements needed to implement, modify, or operate dynamic partitions. It refers to other manuals for details where appropriate.

This section mentions also the restrictions which apply to dynamic partitions with regard to static partitions. For example, not all system programs that run in static partitions can also be run in dynamic partitions.

System Support for Dynamic Partitions

The system support for dynamic partitions includes:

- Predefined Environment A and B which support dynamic partitions as described under "Predefined System Environments" on page 53. z/VSE provides a dialog (*Maintain Dynamic Partitions*) and skeletons to implement and tailor dynamic partitions. Refer to "Planning for Implementation" on page 177.
- Support of multiple dynamic class tables to easily change the setup of a dynamic partitions environment.
- Dialogs and commands for operating and controlling dynamic partitions:
  – *Display System Activity* dialog for displaying dynamic partition activities. Refer to the manual *z/VSE Operation* under "Displaying System Activity" for details.
  – *Display Storage Layout* dialog for displaying dynamic partition and dynamic GETVIS area layout. Refer to "Displaying System Status and Storage Information" in the manual *z/VSE Administration* for details.
  – Extended VSE/POWER commands for dynamic partition control:
    - PLOAD DYNC,ID=n (where n identifies the dynamic class table)
    - PDISPLAY DYNC
    - PDISPLAY STATUS
    - PVARY

    Refer to "Loading a Dynamic Class Table (Format 3)", "Format 8: PDISPLAY DYNC,ALL", "Format 3: Displaying Varied Status Information", and "PVARY: Dis-/Enabling Exit Routines, Task Trace, or Dynamic Classes" in the manual *VSE/POWER Administration and Operation* for details about the above commands.
- Label option CLASSTD for storing labels in the dynamic partitions label subarea. Refer to the the manual *z/VSE System Control Statements* under "OPTION" for details.
- The extended SDAID and DUMP programs for handling dynamic partitions. Refer to the manuals *z/VSE Diagnosis Tools* and *z/VSE Guide for Solving Problems* for details.

The following commands have either been extended for dynamic partition support or are restricted in their use for dynamic partitions. Refer to the manual *z/VSE System Control Statements* for details.

ASSGN
DVCDN/DVCUP
EXEC
HOLD
LIBDEF
LIBLIST
LISTIO
MAP
PRTY (affected by MSECS value)
PRTYIO
RSTRT
SET
START
STATUS
STOP
TPBAL

Which Programs Can Run in Dynamic Partitions?

The z/VSE programs listed in Table 6 on page 40 **can run** in dynamic partitions **except** for the following:

- ICKDSF
- VSE/POWER
- VSE/OLTEP

The z/VSE optional programs listed in Table 7 on page 43 **can run** in dynamic partitions **except** for the following:

- ACF/NCP
- ACF/SSP
- EP
- DL/I DOS/VS can run but the MPS restart support is not available.
- GDDM can run except for the batch print utilities.
- NetView FTP can run, but for file transfer with dynamic allocation the VSE partition component must run in a static partition.
- SDF II VSE

Applications of the following z/VSE base and optional programs run in dynamic partitions if they observe the general restrictions listed in Table 29 on page 177:

- CICS Transaction Server
- CICS/VSE
- VSE/VSAM
- VTAM
- DL/I DOS/VS
- DL/I VSE
- DB2 Server for VSE

Note: You should also consult the *Program Directory*, which is part of the z/VSE shipment, for the latest information about programs being able to run in dynamic partitions.

Restrictions of Dynamic Partitions

There are certain system restrictions to be observed when using dynamic instead of static partitions. They are shown in Table 29:

Table 29. Restrictions of Dynamic Partitions

Items	Static Partitions	Dynamic Partitions
Partition balancing	only for balanced partitions	always balanced within one dynamic class
Storage protection	unique storage keys (PIK=storage key)	unique storage keys within one address space
Real execution	YES	NO (virtual only)
Full unit record device support	YES	NO (VSE/POWER support spooling only)
SYSFIL support	YES	NO
OCR, Diskette support	YES	NO
Checkpoint/Restart	YES	NO
VMCF support	YES	YES
XECB support	YES (same address space)	NO

Programs can PFIX pages in static as well as in dynamic partitions.

Planning for Implementation

Dynamic Class Tables and Dynamic Classes

The characteristics and attributes of dynamic partitions are defined in one or more **dynamic class tables**. Such a table is stored in the system sublibrary IJSYSRS.SYSLIB under the member name DTR$DYNn.Z. **DTR$DYNC.Z** is the name of the default table shipped with z/VSE.

You activate a dynamic class table through the VSE/POWER PLOAD DYNC,ID=n command. VSE/POWER creates a dynamic partition if a job for a dynamic class has been submitted and retrieves the partition parameters defined for that class from the active table.

For dynamic partitions you can define up to 23 dynamic classes per table. For defining these classes you can use the letters C, D, E, and G through Z. Note that you cannot use the characters A, B, F, and 0 through 9. The maximum number of dynamic partitions that can be specified per class is 32.

Multiple Dynamic Class Tables

You can define up to 36 dynamic class tables to be able to easily change the setup of a dynamic partition environment. The name of a table must be DTR$DYNn.Z, where n can be any character from 0 - 9 to A - Z. z/VSE supports multiple dynamic class tables through the *Maintain Dynamic Partitions* dialog.

Number of Dynamic Partitions Supported

z/VSE supports up to **12 static partitions** and in addition you can define up to **150-200 dynamic partitions**. Following is some background information about the maximum number of partitions supported.

The supervisor supports a maximum of 255 VSE/Advanced Functions tasks. This limitation also exists with dynamic partitions. VSE/Advanced Functions tasks are classified into system tasks, maintasks and subtasks. A number of these tasks are reserved for z/VSE component programs.

A partition corresponds to a maintask. The available maintasks are the theoretical maximum number of partitions you can allocate. This value is calculated as follows:

```
Number of partitions =
Number of maintasks  = 255 minus
       1. Number of VSE/Advanced Functions tasks for z/VSE component programs
       2. Number of VSE/Advanced Functions subtasks for user applications
```

This results in a maximum value of about 150-200 for the number of dynamic partitions possible. But this is more or less a theoretical value and the *reasonable* number of allocated partitions depends also on your:

- Processor
- System configuration
- Customer environment
- Job profile

The maximum number of partitions (12 static plus xx dynamic) is defined with the NPARTS parameter of the IPL SYS command.

Customizing Dynamic Partition Support

All predefined environments **A**, **B**, and **C** contain dynamic partition support, as described under "Predefined Dynamic Partition Support" on page 180. No further tailoring effort is required.

You are recommended to select the environment A, B, or C which best suits your requirement for dynamic partition support. You can easily change the number of dynamic partitions by tailoring or creating an environment with dynamic partitions:

- Tailor the IPL Procedure
- Catalog the JCL Startup Procedure
- Tailor the VSE/POWER Startup Procedure
- Define or tailor a Dynamic Class Table
- Verify a Dynamic Class Table
- Activate a Dynamic Class Table

After completing these tasks and performing IPL, you can submit and process jobs in dynamic partitions. z/VSE provides the:

- *Maintain Dynamic Partitions* dialog for creating a dynamic class table.
- *Tailor IPL Procedure* dialog for tailoring the IPL procedure.
- Skeleton SKJCLDYN for cataloging the JCL startup procedure for dynamic partitions.
- Skeleton SKPWSTRT for tailoring the VSE/POWER startup procedure.

Instead of tailoring the VSE/POWER startup procedure, you can have the operator activate the dynamic partition support via the VSE/POWER *PLOAD* command.

- VSE/POWER PLOAD DYNC,ID=n,VERIFY command to "debug" a dynamic class table before it is actually loaded and used.

The manual *z/VSE Administration* describes in detail the dialogs and skeletons mentioned above. "Dynamic Partition Support" in the manual *VSE/POWER Administration and Operation* describes the dynamic partition support as implemented and supported by VSE/POWER including VSE/POWER commands. This includes information also important for making planning decisions.

Further Tailoring Information

Your modifications in the dynamic class table may result in further changes to environment A, B, or C, as outlined below.

- If you need more virtual storage, you have to increase the values for:
  - VSIZE (virtual storage size)
  - DPD (page data set extents)

  You may also have a need to increase the total number of partitions and thus change the value for:
  - NPARTS

  These are all parameters of the IPL procedure which you can tailor with the *Tailor IPL Procedure* dialog.
- Since VSE/POWER controls dynamic partitions, its storage requirements increase with the number of dynamic partitions to be handled. For each additional dynamic partition VSE/POWER requires:
  - About 2KB to be reserved through the SETPFIX command.
  - About 10KB to 15KB partition GETVIS space, assuming 3 spooled devices (1 RDR, 2 PRT/PUN) per dynamic partition and the default DBLK size.

  The VSE/POWER values depend mainly on
  - the number of spooled and actually used devices per dynamic partition.
  - the DBLK value of the VSE/POWER generation macro.

  For detailed values, refer to "Planning for VSE/POWER" in the manual *VSE/POWER Administration and Operation*.

In addition, at IPL time, for each dynamic partition defined in the NPARTS parameter, the system GETVIS area is extended by about 4KB. Since this must be real space, the total SETPFIX value for static partitions is reduced correspondingly.

Predefined Dynamic Partition Support

The Predefined Dynamic Class Table

The dynamic class table provided (named DTR$DYNC) defines 57 dynamic partitions:

- 9 partitions with 1 MB (class C)
- 32 partitions with 1 MB (class P)
- 3 partitions with 8 MB (class R)
- 2 partitions with 15 MB (class S)
- 8 partitions with 3 MB (class Y)
- 3 partitions with 5 MB (class Z)

Note that the predefined classes R and S are new with VSE/ESA 2.5.

The number of dynamic partitions that can be active at the same time is limited by two parameters:

- The VSIZE available for dynamic partitions.
  - For predefined environment A, the VSIZE is 150 MB; about 39 MB of this space is available for dynamic partitions.
  - For predefined environment B, the VSIZE is 264 MB; about 58 MB of this space is available for dynamic partitions.
  - For predefined environment C, the VSIZE is 2 GB; about 735 MB of this space is available for dynamic partitions.

  For further VSIZE details, refer to:
  - “Storage Layout of Predefined Environment A” on page 54
  - “Storage Layout of Predefined Environment B” on page 55
  - “Storage Layout of Predefined Environment C” on page 57
- The number of dynamic partitions which can be active at the same time.

  This depends on the NPARTS definition.
  - For predefined environment A, the IPL SYS command defines NPARTS=40 as system maximum; 12 of these are static partitions, leaving 28 for dynamic partitions.
  - For predefined environment B, the IPL SYS command defines NPARTS=60 as system maximum; 12 of these are static partitions, leaving 48 for dynamic partitions.
  - For predefined environment C, the IPL SYS command defines NPARTS=120 as system maximum; 12 of these are static partitions, leaving 108 for dynamic partitions.
  - Figure 3 on page 54 shows the storage layout for environment A, including dynamic partitions.
  - Figure 4 on page 56 shows the storage layout for environment B, including dynamic partitions.
  - Figure 5 on page 58 shows the storage layout for environment C, including dynamic partitions.

Note: Class P is intended for z/VSE workstation platform users. For performance reasons, class P has its own startup profile (PWSPROF) shown under “The Predefined Startup Profiles” on page 182.

In Figure 14 on page 181, the character **v** (or another character) identifies column beginning and end. All numeric values are right-bound, but text is left bound. The table also includes the addresses for the VSE/POWER spooled reader, printer, and punch devices for each dynamic class. For the reader only 1 device is allowed; for the printer and punch devices you can specify additional devices (up to 14 in one row and separated by commas).

```
          CLASS  ALLOC   SIZE   SP-GETV LUBS PROFILE  MAX-P   ENABLED
----------v------vvvv----vvvvv--vvvvv---vvv--vvvvvvvv-vv------v---------
CLASS=    C         1      500    128    50  STDPROF  09      Y
          POWER SPOOLED DEVICES
----------v---v---v---v---v---v---v---v---v---v---v---v---v---v---------
READER=   FEC
PRINTERS= FEE
PUNCHES=  FED
----------v------vvvv----vvvvv--vvvvv---vvv--vvvvvvvv-vv------v---------
CLASS=    P         1      512    128    50  PWSPROF  32      Y
          POWER SPOOLED DEVICES
----------v---v---v---v---v---v---v---v---v---v---v---v---v---v---------
READER=   FEC
PRINTERS= FEE
PUNCHES=  FED
----------v------vvvv----vvvvv--vvvvv---vvv--vvvvvvvv-vv------v---------
CLASS=    R         8     1024    128    50  STDPROF  03      Y
          POWER SPOOLED DEVICES
----------v---v---v---v---v---v---v---v---v---v---v---v---v---v---------
READER=   FEC
PRINTERS= FEE
PUNCHES=  FED
----------v------vvvv----vvvvv--vvvvv---vvv--vvvvvvvv-vv------v---------
CLASS=    S        15     1024    128    50  STDPROF  02      Y
          POWER SPOOLED DEVICES
----------v---v---v---v---v---v---v---v---v---v---v---v---v---v---------
READER=   FEC
PRINTERS= FEE
PUNCHES=  FED
----------v------vvvv----vvvvv--vvvvv---vvv--vvvvvvvv-vv------v---------
CLASS=    Y         3     1024    128    50  STDPROF  08      Y
          POWER SPOOLED DEVICES
----------v---v---v---v---v---v---v---v---v---v---v---v---v---v---------
READER=   FEC
PRINTERS= FEE
PUNCHES=  FED
----------v------vvvv----vvvvv--vvvvv---vvv--vvvvvvvv-vv------v---------
CLASS=    Z         5     1024    128    50  STDPROF  03      Y
          POWER SPOOLED DEVICES
----------v---v---v---v---v---v---v---v---v---v---v---v---v---v---------
READER=   FEC
PRINTERS= FEE
PUNCHES=  FED
```

Figure 14. Predefined Dynamic Class Table (DTR$DYNC.Z)

Note: When modifying the table it is recommended to use the dialog and not edit the existing table. If, for any reason, you edit the table you should be aware that when using the dialog later it may no longer work correctly.

The columns have the following meaning:

CLASS
Specifies the class which is also used to create the partition ID.

ALLOC
Specifies in MB the virtual storage of a partition within a class: *maximum program size + partition GETVIS area* and includes the dynamic space GETVIS area. The theoretical maximum is 2046 MB.

SIZE
Specifies in KB the maximum program size. This is the amount of contiguous virtual storage reserved for program execution.

SP-GETV Specifies in KB the size of the dynamic space GETVIS area which is located below 16 MB. The minimum required is 128KB.

LUBS Specifies the number of programmer logical units allocated to each dynamic partition within a dynamic class. The maximum is 255.

PROFILE Specifies the name of the JCL procedure processed when a dynamic partition is activated. (The name must follow the standard naming conventions for phases).

MAX-P Specifies the maximum number of dynamic partitions that can be activated within a class. It can range from 1 to 32.

ENABLED Specifies whether a dynamic partition can activated via the PLOAD DYNC, ID=n command of VSE/POWER.

The Predefined Startup Profiles

The default startup profiles (STDPROF and PWSPROF) for dynamic partitions are shown below. To maintain and catalog such a procedure, use skeleton SKJCLDYN.

```
CATALOG STDPROF.PROC   DATA=YES REPLACE=YES
// LIBDEF DUMP,CATALOG=SYSDUMP.DYN,PERM
// OPTION NODUMP
// EXEC PROC=LIBDEF
ASSGN SYSIN,FEC
ASSGN SYSPCH,FED
ASSGN SYSLST,FEE
ASSGN SYSLNK,DISK,VOL=DOSRES,SHR           SYSTEM LINK FILE
ASSGN SYS001,DISK,VOL=SYSWK1,SHR           SYSTEM WORK FILE 1
ASSGN SYS002,DISK,VOL=SYSWK1,SHR           SYSTEM WORK FILE 2
ASSGN SYS003,DISK,VOL=SYSWK1,SHR           SYSTEM WORK FILE 3
ASSGN SYS004,DISK,VOL=SYSWK1,SHR           SYSTEM WORK FILE 4
/+
```

Figure 15. Default Startup Profile STDPROF

For better performance, you may replace the LIBDEF statement in STDPROF with one that specifies less sublibraries as, for example, in profile PWSPROF. This profile is intended for dynamic class P to be used with the programmable workstation support.

```
CATALOG PWSPROF.PROC   DATA=YES REPLACE=YES
ASSGN SYSIN,FEC
ASSGN SYSPCH,FED
ASSGN SYSLST,FEE
/+
```

Figure 16. Default Startup Profile PWSPROF

Partition Standard Labels for Dynamic Partitions

Partition standard labels can also be specified for dynamic partitions (// OPTION PARSTD). They are searched instead of the CLASSTD labels, but will be automatically removed from the system when the dynamic partition is UNBATCHed at the end of the VSE/POWER job.

The // OPTION PARSTD statement for a dynamic partition can be specified only in the partition where the labels are used and not from the BG partition.

Chapter 14. Tailoring System Startup

z/VSE provides an *automated startup* facility for performing a fast and efficient system startup. This is sometimes also referred to as ASI (automated system initialization). In most cases, no operator intervention is required, except for IPL (initial program load). You may use the procedures and jobs of the startup facility as shipped by IBM, or you can tailor them to meet the needs of your installation.

Related Information

This chapter helps you plan startup requirements for your installation. For detailed information about startup procedures and jobs and how to tailor them, refer to "Tailoring IPL and System Startup" in the manual *z/VSE Administration*.

"Starting the System" in the manual *z/VSE Guide to System Functions* provides additional startup information about topics such as interrupting IPL processing for modifications, loading phases into the SVA, or creating a $ASIPROC master procedure.

Notes:

1. During initial installation, you must specify the predefined environment you want to use. Refer to "Predefined System Environments" on page 53 for details about the predefined environments provided. Careful planning of your system's environment may considerably reduce any tailoring effort needed later.
2. If you use z/VSE as shipped by IBM, the startup procedures provided with the system are processed during startup. These procedures create a system with the partition layout of the predefined environment selected.
3. If you wish, you may also create your own startup procedures instead of using or modifying the ones supplied by IBM.

Components of the Startup Facility

The main components of the Startup facility are:

- IPL procedure.
- JCL startup procedures and jobs.
- Startup program *DTRISTRT*.
- Procedures *CPUVARn* and *$COMVAR*.

IPL Procedure

The name of your system's original IPL procedure is determined by the disk device type on which you install z/VSE. For example, for a system that resides on an IBM 3390, z/VSE uses initially IPL procedure *$IPLE90*.

During initial installation, the IPL procedure selected is renamed to *$IPLESA* This is the name that appears on the screen when you use the *Tailor IPL Procedure* dialog, for example, to change IPL parameters.

JCL Startup Procedures and Jobs

After IPL successfully completes, the JCL startup procedure for the BG partition is called. For initial installation, the name is *$0JCL790* (for the example outlined under IPL Procedure before). The second digit identifies the partition and the last two digits the disk device type. During initial installation, the procedure is renamed to $0JCL.

The procedure calls various other procedures and releases jobs necessary for specific startup tasks. This includes the startup program *DTRISTRT*, which determines the startup mode used by the system (WARM, for example).

Startup Program DTRISTRT

The startup program DTRISTRT determines the startup mode for *each* static partition. This decision is based on system status analysis and the startup request (if any) made by the operator.

DTRISTRT is activated by the JCL startup procedure for the BG partition ($0JCL). At that time, the system variables of *CPUVARn* (see below) still reflect the status of the last shutdown, or the last startup if no shutdown was performed.

DTRISTRT analyzes the information stored in CPUVARn for each static partition. In addition, DTRISTRT issues messages that allow the operator to intervene if requested and select a particular startup mode. After deciding on the startup modes, DTRISTRT sets the system variables in CPUVARn accordingly. The JCL startup procedures retrieve these variables to initiate the correct startup procedures and jobs.

Procedure $COMVAR

$COMVAR is a CPU-independent procedure where the CPU ID and number are stored. It is used to determine the CPU number of the CPU where the startup is to be performed.

The default $COMVAR procedure is set to a single system environment. In a DASD sharing environment with two or more CPUs, you *must* modify $COMVAR to identify the different CPUs in use. In addition, each CPU requires its own CPUVARn procedure.

z/VSE provides skeleton SKCOMVAR for tailoring such an environment. The skeleton is described in detail in the manual *z/VSE Administration* under "Skeleton for Tailoring $COMVAR Procedure".

Procedure CPUVARn

A CPUVARn procedure consists of SETPARM statements that contain system variables. These variables describe the system and reflect each partition's status. For a each CPU, one procedure is required. **n** is the CPU number. The name of the default procedure shipped is CPUVAR1.

A CPUVARn procedure is used to save system information such as:

- **Use of Partitions –** This information shows which partitions are used and which programs run in the partitions.
- **Partition Status –** A partition can be either active or inactive. For example, if a partition shows status ACTIVE at startup time, it indicates that the previous shutdown was not successful.

- **Outstanding Requests –** z/VSE functions may request a certain partition startup mode. For example, a COLD start is requested for the VSE/POWER partition after the VSE/POWER queues have been extended.
- **Results of Status Analysis and Operator Requests –** The startup program DTRISTRT updates system variables according to these requests.
- **Synchronization Points –** Synchronization points can be used to define how partitions interact during startup. For example, a synchronization point in one partition may trigger further processing in another partition.
- **System Environment –** This is the environment that was selected at initial installation.

Note: As described in the manual *z/VSE System Utilities* under "DTRSETP Utility", the *DTRSETP* utility program helps you create and modify CPUVARn procedures.

Startup Modes

z/VSE provides several startup modes to be able to react properly to a specific system status. The startup modes are:

- **WARM –** A WARM startup is performed by the system if the last shutdown was completed successfully. The system is in normal, re-usable condition.
- **RECOV –** A RECOV (recovery) startup is performed by the system if the last shutdown of one or more partitions could not be completed successfully or if no shutdown was performed at all.
- **COLD –** A COLD startup is performed or should be selected by the operator if, for example, there is a need to reformat the VSE/POWER queues. This means, however, that the jobs in the VSE/POWER reader queue are lost unless they are saved and reloaded afterwards. The manual *z/VSE Operation* under "Offloading and Loading VSE/POWER Queues" describes how to save and reload user jobs. System jobs are reloaded automatically.
- **BASIC –** A BASIC startup is performed or should be selected by the operator when the system does not start as normally required due to an error condition like exhausted VSE/VSAM space or VTAM buffer space.

  A BASIC startup gives you a "basic" system with the original system tables active. *Your modifications are ignored*. During a BASIC startup, the operator is requested to specify up to three terminal addresses. These terminals are activated and allow interactive correction of errors.
- **MINI –** A MINI startup is performed or should be selected by the operator if library problems exist or library maintenance is necessary. MINI starts three partitions only, BG, F1 (VSE/POWER), and FB (Security Server).

z/VSE allows the operator to interrupt startup processing if requested via the IPL load parameter. The operator can then request a COLD, BASIC, or MINI start and override the startup modes selected by the system. *Startup modes WARM and RECOV can only be selected by the system*.

During a WARM or RECOV startup, a startup mode is selected individually for every partition of the system. This means that during startup, partitions may be started in different modes.

If startup mode is BASIC or MINI, only a certain set of partitions is started (all in BASIC or MINI mode). If the operator selects COLD, COLD is used as startup mode for all the partitions of the system.

Notes:

1. These five startup modes have been developed for z/VSE. They are not identical to the startup modes defined by CICS.
2. During startup, CICS selects its startup mode from the restart data set. Only specifying a BASIC or COLD startup mode forces CICS to perform a cold start.
3. The manual *z/VSE Guide for Solving Problems* under “System Startup Modes Available” describes system conditions that should be handled with COLD, BASIC and MINI starts.

Considerations for Tailoring System Startup

The following should be considered if you plan to tailor your system’s startup:

- The startup procedures and jobs are written in the job control language (JCL). Functions such as conditional job control, symbolic parameters, and nested procedures are used. These functions are described in detail in the *z/VSE Guide to System Functions* under “Controlling Jobs”.
- Procedures CPUVARn and $COMVAR mainly contain JCL *SETPARM* statements. For details about SETPARM statements, refer to “SETPARM” in the manual *z/VSE System Control Statements*.
- Table 30 on page 187 shows the skeletons that z/VSE provides in VSE/ICCF library 59 for startup tailoring tasks. The manual *z/VSE Administration* under “Tailoring IPL and System Startup” lists the contents of these skeletons and has detailed information about changing them.

  When you change startup skeletons, do *not* use the original members. Copy them to your primary VSE/ICCF library and apply your changes to the copied versions. Test these changed versions carefully. It is your responsibility that the changed startup is error free and runs correctly.
- You should *not* change the following:
  - $nJCLBSX and all related procedures and jobs for BASIC startup.
  - $nJCLMIN and all related procedures for MINI startup.

  This ensures that you can still perform a BASIC or MINI startup if problems arise. This may be your only chance to correct an error.

Tasks for Tailoring System Startup

Table 30 shows the tasks that are involved in tailoring your system's startup. You may perform some or all of these tasks, depending on the scope of your changes. "Tailoring IPL and System Startup" in the manual *z/VSE Administration* has detailed information about the tailoring tasks and instructions on how to perform them. That information is not repeated in this chapter.

Table 30. Tasks for Tailoring System Startup

TASK	SKELETON(S) USED
Modify IPL procedure	(See Note 1)
Modify allocations for static partitions (see Notes 2 and 3)	SKALLOCx
Modify BG partition startup (see Note 3)	SKJCL0 SKUSERBG
Modify VSE/POWER partition startup (See Note 3)	SKJCL1 SKPWSTRT
Modify other partition startups (see Note 3 for SKJCLx and SKLIBCHN)	SKJCLx SKLIBCHN SKVTAM SKCICS
Define CPU IDs in a DASD-sharing environment	SKCOMVAR
Add jobs to load list for VSE/POWER reader queue for a COLD start	SKCOLD
Create sample job for loading jobs into VSE/POWER reader queue	SKLOAD
Create a startup job for a dynamic partition	SKJCLDYN

Notes:

1. The Interactive Interface provides the ***Tailor IPL Procedure*** dialog for this task.
2. You usually modify just one of the skeletons available in VSE/ICCF library 59 for partition allocations:

   SKALLOCA = 12 static partitions (12 address spaces)

   SKALLOCB = 12 static partitions (12 address spaces)

   SKALLOCC = 12 static partitions (12 address spaces)

   These skeletons reflect the predefined environments provided. Refer to "Predefined System Environments" on page 53 for details.
3. After you have modified the startup skeleton(s) mentioned, use skeleton *SKENVSEL* to catalog them. SKENVSEL is also a member of VSE/ICCF library 59.

Chapter 15. Tailoring the Interactive Interface

Overview of the Interactive Interface

The Interactive Interface makes it easier to access and use system functions. Through *selection panels*, you can select dialogs for specific tasks that you want to perform (define files or backup libraries, for example).

Appendix C, “Dialogs of the Interactive Interface,” on page 299 lists the dialogs that z/VSE ships as part of the Interactive Interface.

z/VSE User Profiles

In z/VSE, a *user profile* defines a system user to the Interactive Interface. At initial installation, z/VSE creates the user profiles shown in Table 31.

User profiles are stored in the VSE.CONTROL.FILE.

Table 31. z/VSE Predefined User Profiles

User ID	Password	Function
SYSA	SYSA	Model system administrator
PROG	PROG	Model programmer
OPER	OPER	Model operator
POST	BASE	User to complete initial installation
CICSUSER	CICSUS	CICS default user
DBDCCICS	DBDCCI	CICS partition user (F2)
PRODCICS	PRODCI	CICS partition user (F8)
FORSEC	FORSEC	Model administrator (SECURITY=YES)
$SRV	$SRV	Model for problem determination
VCSRV	VCSRV	Connector Server partition user
CNSL	CNSL	Model for CICS TS console user

For security reasons, you are forced during the first logon to change the above passwords except for users DBDCCICS and PRODCICS.

You can use the user IDs **SYSA, PROG, and OPER** as models to define your own profiles for an administrator, programmer, or system operator. You should not use them for daily operation, because they may be affected if you later install a system refresh.

POST is a reserved user ID. It is used *only* during initial installation for performing special tasks. You should *not* use it for any other tasks.

CICSUSER is a default user ID required for CICS Transaction Server startup. It performs security checks for terminal users that are not signed on.

DBDCCICS and PRODCICS are partition user IDs required for the startup of the primary and secondary CICS Transaction Server.

FORSEC is a user ID required for system startup if SECURITY=YES has been selected during initial installation. This user ID is defined in DTSECTAB as well as in the VSE.CONTROL.FILE.

$SRV allows access to a default panel hierarchy for problem determination.

VCSRV is the partition user required for the Connector Server startup. It does not have an initial selection panel and can therefore not be used for sign-on.

CNSL is a console user ID for the CICS Transaction Server. This ID is defined with all security transaction classes, which allows all transactions to be executed on the system or Interactive Interface consoles.

The manual *z/VSE Administration* has further details on user profiles.

Interactive Interface Panels

The Interactive Interface has several types of panels:

- Selection Panels

  A selection panel displays up to nine options from which you can choose one. Each option represents either a dialog or another selection panel. The options on the panel are numbered. You make your selection by entering the appropriate number at the bottom of the panel.

  Figure 17 on page 191 shows an example of a selection panel.
- Data Entry Panels

  The dialogs use data entry panels to obtain information about the task you are performing. You enter the input in particular fields on the panel. For example, if you are restoring a library, you enter the tape unit address.
- Function Lists (FULISTs)

  A FULIST is a special type of data entry panel. It displays a list of items that you can process. It also displays options that you use to process the items. Each option corresponds to a specific number. You enter the number of the option you want to perform next to the item you want to process.

  Figure 18 on page 191 shows an example of a FULIST.
- HELP Panels

  From most panels, you can press **PF1** to display a HELP panel. The HELP panel provides an explanation about the task you are performing.

  A HELP panel often provides information about a message that the system displays. For example, if you enter incorrect data, the system displays a message. If you press **PF1**, the HELP panel may explain the error.

Panel Hierarchies for Predefined Users

A *panel hierarchy* refers to the complete set of selection panels that are available to a user. z/VSE supplies separate panel hierarchies for the predefined users SYSA, PROG, OPER, and $SRV which you can use as models for your own panel hierarchies.

z/VSE Application Profiles

z/VSE provides an *application profile* for each dialog in the Interactive Interface. This profile defines execution parameters for the dialog. The application profiles defined for Interactive Interface dialogs are important to you when creating your own selection panels and panel hierarchies. Appendix C, "Dialogs of the Interactive Interface," on page 299 lists the application profile names for the dialogs of the Interactive Interface.

Note: z/VSE also provides a number of applications that are not included in the Interactive Interface. You can also include these applications, which are listed in Appendix D, "Additional z/VSE Applications," on page 305, in selection panels that you create.

```
IESADMSL.IESEADM         z/VSE FUNCTION SELECTION
                                                      APPLID: DBDCCICS
  Enter the number of your selection and press the ENTER key:

        1  Installation
        2  Resource Definition
        3  Operations
        4  Problem Handling
        5  Program Development
        6  Command Mode
        7  CICS Supplied Transactions








PF1=HELP                    3=SIGN OFF                       6=ESCAPE(U)
                            9=Escape(m)

==>_
```

Figure 17. Example of a Selection Panel

```
 IESFILFL1             DISPLAY OR PROCESS A FILE         Page   1 of 3
 CATALOG:   VSESP.USER.CATALOG                                VSESPUC

 OPTIONS:  1 = SHOW    2 = SORT    3 = PRINT   4 = COPY   5 = DELETE
           6 = VERIFY  7 = LOAD

 OPT      FILE ID                               FILE NAME  FILE TYPE

  _       CICS.AUTO.STATS.A                     DFHSTM        B
  _       CICS.AUTO.STATS.B                     DFHSTN        B
  _       CICS.CSD                              DFHCSD        B
  _       CICS.DUMPA                            DFHDMPA       B
  _       CICS.DUMPB                            DFADMPB       B
  _       CICS.RSD                              DFHRSD        B
  _       CICS.TD.INTRA                         DFHNTRA       B
  _       DEFAULT.MODEL.ESDS.SAM                *NONE*        B
  _       DFHTEMP                               DFHTEMP       B
  _       DICKINS.SAM.ESDS                      *NONE*        B
  _       DOS.WORKFILE.SYS001.RECOVER.F2        *NONE*        B

 PF1=HELP       2=REFRESH    3=END        4=RETURN
                8=FORWARD    9=PREFIX

 LOCATE FILE ID ==> ________________________________________________
```

Figure 18. Example of a FULIST

Dialogs for Interactive Interface Tailoring

Although z/VSE ships three panel hierarchies, these may not accurately reflect your specific system environment. Through dialogs, you can change the appearance of the Interactive Interface so that it reflects your environment and your users.

The following dialogs, described in detail in the manual *z/VSE Administration*, are available for Interactive Interface tailoring:

1. ***Maintain User Profiles*** dialog to define users to the system.

   Every user of the Interactive Interface has a *user profile record* that specifies:
   - User ID and password
   - z/VSE information
   - CICS/VSE information
   - VSE/ICCF information

   Note: To maintain large numbers of user profiles, you can use the batch utility IESUPDCF described in the manual *z/VSE System Utilities* under "Batch Program IESUPDCF".
2. ***Maintain Selection Panels*** dialog to create your own selection panels.

   The system uses *selection panel records* to build and maintain selection panels. z/VSE ships records for each selection panel in the Interactive Interface. The record specifies the:
   - Name of the selection panel.
   - Choices (selections) on the panel.
   - Panel or application corresponding to each selection.
3. ***Maintain Application Profiles*** dialog to define your CICS applications to the system and access them from the Interactive Interface.

   An *application profile record* defines a CICS application to the Interactive Interface. The record contains execution information about the application such as:
   - The name of the application.
   - How the application is initiated.
   - The name you use to activate the application.
   - Specifications for terminal input and input data.
4. ***Maintain Synonyms*** dialog to define names (synonyms) for selection panels and applications.

   With the *Maintain Synonyms* dialog, you can also create character strings that users can enter for accessing dialogs and applications. Users can enter such synonyms (which are easier to remember than number strings) from any selection panel of the Interactive Interface or from selection panels that you create.

   The records for user profiles, selection panels, Interactive Interface applications, and synonyms are part of the z/VSE control file *(VSE.CONTROL.FILE)* on DOSRES (in the VSE/VSAM user catalog).

Note: The panel *User Interface Tailoring* offers in addition the following dialogs which are not directly related to Interactive Interface tailoring and which are described in the manual *z/VSE Administration*:

Maintain PRIMARY Sublibraries

Customize z/VSE Workstation Platform

General Planning Considerations for the Interactive Interface

After you install your system, you must define profiles for each user of the Interactive Interface. Remember that you can use the SYSA, PROG, and OPER profiles that z/VSE defines as models for various users. Do *not* use POST.

You should carefully consider who can access the dialogs for user interface tailoring. For example, only one person should maintain user profiles and have access to the *Maintain User Profiles* dialog. You may want some application programmers to be able to maintain application profiles.

You should probably use the Interactive Interface for some time before changing it. As you and other users work with the system, compare the default panels and panel hierarchies of the Interactive Interface with the distribution of tasks and responsibilities in your company. Remember that you can:

- Modify selection panels and thus the structure of the hierarchy.
- Change access rights in the programmer and operator profiles.
- Add applications to the structure.

If you decide to create your own selection panels and panel hierarchies, there are several things you should consider. You should *not* build a panel hierarchy of more than nine levels. Beyond the ninth level, the END key always returns you to the selection panel at level eight. You can have selection panels and applications in different places within the panel hierarchy. You do not necessarily have to recreate an entire set of panels. You can create an initial selection panel and display this panel when the user signs on. From this selection panel, you can invoke different Interactive Interface panels which the user needs.

Example of Profile-Driven System Views

When you create a user profile, you define what a specific user sees of the entire system. Figure 19 on page 194 is a simple example of how a system could look to several users, each of whom has a different profile. Note that all of these users sign on to the system from the same panel. What is defined in their user profiles, however, causes the system to show each of them something different.

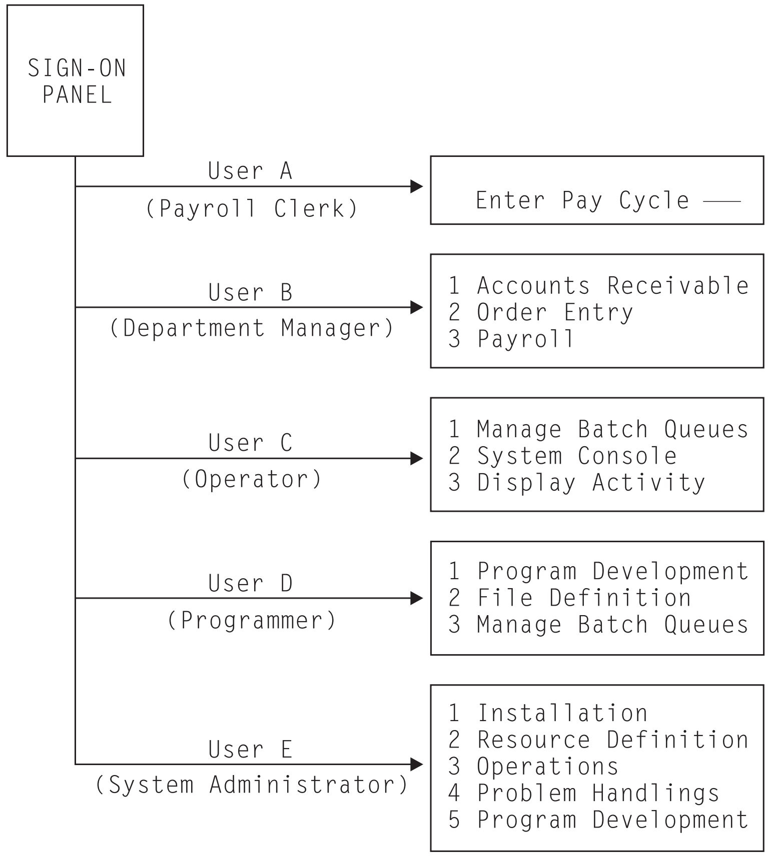


Figure 19. Profile-Driven System Views

The payroll clerk's profile specifies that an application is to be invoked when this user signs on. Access to other system functions is not possible. At sign on, the system finds the corresponding application profile and then invokes the application.

Unlike the payroll clerk, the department manager needs access to several applications. This user's profile contains the name of a selection panel that offers options for these applications.

The operator, programmer, and system administrator have access to selected z/VSE system functions.

Chapter 16. Setting Up Your CICS Environment

Overview

z/VSE provides in partition F2 the **CICS Transaction Server for VSE/ESA** which replaces CICS/VSE of former VSE releases.

As shipped, z/VSE provides predefined support for the installation of a **second** (predefined) **CICS Transaction Server** in partition F8, and a CICS/VSE system in partition F4. Running the CICS Transaction Server together with a CICS/VSE system is referred to as **CICS Coexistence Environment**. This is further discussed under "Characteristics of a CICS Coexistence Environment" on page 198.

Both, the second CICS Transaction Server and CICS/VSE require that you select **predefined environment B or C** for initial installation.

CICS/VSE 2.3 is shipped on the Extended Base Tape. To install it, you must use the dialogs provided for the installation of optional programs. Refer to "Installing Programs from the Extended Base Tape" on page 77 for further details.

Major Characteristics of the CICS Transaction Server

The CICS Transaction Server provides new functions that improve application and system programming, system management, and the reliability and integrity of your system. Key benefits include:

- Exploitation of ESA/390 subsystem storage protection.
- Extensive virtual storage constraint relief with more space below 16MB for user applications.
- System management enhancements including Resource Definition Online (RDO) for files.
- Shared data tables for improved performance and availability.
- Starting with VSE/ESA 2.5, enhancements such as CICS Web Support (CWS), CICS 3270 Bridge, and REXX for CICS are available.
- Starting with VSE/ESA 2.6, enhancements such as SSL (Secure Sockets Layer) and ECI (External Call Interface) are available.

For details, refer to the corresponding CICS Transaction Server documentation.

Overview on CICS Skeletons and Tables

Table 32 on page 196 shows for the predefined CICS systems the available skeletons and tables. The figure allows you to identify quickly the differences and the requirements for the:

- Primary CICS Transaction Server which is available after initial installation and which is running in partition F2.
- Secondary CICS Transaction Server which must be installed separately (default partition F8) as described under "Installing a Second CICS Transaction Server" on page 200.
- CICS/VSE which must be installed separately (default partition F4) as described under "Installing CICS/VSE" on page 205.

The source code of all the skeletons, tables, and members provided is available in **VSE/ICCF library 59**.

Table 32. CICS Skeletons and Tables

Skeletons/Tables	Primary CICS TS (DBDCCICS)	Secondary CICS TS (PRODCICS)	CICS/VSE (OLDCICS)
Startup Skeleton	SKCICS	SKCICS2	SKCICSOL
Resource Definition Skeleton	—	SKPREPC2	SKPREPCO SKPREPSO (see Note 2)
Destination Control Table	DFHDCTSP	DFHDCTC2	DFHDCTCO
Monitoring/Statistics Table	DFHDMFSP	—	—
File Control Table	DFHFCTSP *	DFHFCTC2 *	DFHFCTCO
Journal Control Table	DFHJCTSP	—	DFHJCTCO
Processing Program Table	—	—	DFHPPTCO DFHPPTSO (see Note 2)
Program Control Table	—	—	DFHPCTCO DFHPCTSO (see Note 2)
Program List Table (startup)	DFHPLTPI	DFHPLTP2	DFHPLTPO
Program List Table (shutdown)	DFHPLTSD (see Note 1)	DFHPLTS2	DFHPLTSO
System Initialization Table	DFHSITSP	DFHSITC2	DFHSITCO
Temporary Storage Table	DFHTSTSP	—	DFHTSTCO
Terminal Control Table	—	—	DFHTCTCO
Transaction List Table	DFHXLTSP	DFHXLTSP	DFHXLTCO
Sign-on Table	—	—	DFHSNTCO

(*) These tables are migrated into the CSD during initial installation or FSU.

Notes:

1. DFHPLTSD includes a minimum support for statistics to be printed at shutdown. This support is invoked if DFH0STAT has been activated in DFHPLTSD.
2. In a CICS coexistence environment, the CSD (CICS System Definition) file can either be shared or the CICS Transaction Server and CICS/VSE have their own CSD file each. If the CSD file is to be shared, the skeletons SKPREPSO, DFHPPTSO, and DFHPCTSO must be used in case of CICS/VSE (instead of the skeletons with suffix CO).

   If the CSD file is shared, it means that the CSD file of the CICS Transaction Server is also used by CICS/VSE.

   For further details, refer to “CSD File Considerations for Coexistence Environment” on page 199.

Note that the CICS Transaction Server does not use TCT, PCT, and PPT tables (they are only used by CICS/VSE). The reason is that resource definition is done online via RDO with the CEDA DEFINE command or the DFHCSDUP utility of the CICS Transaction Server. This resource information is stored in the CSD file.

The following source books for CSD definitions are available in IJSYSRS.SYSLIB:

```
IESWPPT.Z     Program definitions for workstations
IESWPPTL.Z    Program definitions for workstations (NLS)
IESZPPT.Z     Program definitions for Interactive Interface
IESZPPTI.Z    Program definitions for Interactive Interface-VSE/ICCF
IESZPPTL.Z    Program definitions for Interactive Interface-VSE/ICCF (NLS)
IESWPCT.Z     Transaction definitions for workstations
IESZPCT.Z     Transaction definitions for Interactive Interface
IESZFCT.A     File definitions
IESZTCT.Z     Terminal and console definitions
IESCSEZA.Z    CICS Listener definitions
CEECCSD.Z     LE/VSE Base definitions
IBMCCSD.Z     PL/I Run-Time definitions
IGZCCSD.Z     COBOL Run-Time definitions
EDCCCSD.Z     C Run-Time definitions
```

The following source books for CSD definitions are available in PRD2.SCEEBASE:

```
CEETICSD.Z    Set USESVACOPY(YES) for LE/VSE Base
EDCTICSD.Z    Set USESVACOPY(YES) for C Run-Time
```

The following source book for CSD definitions is available in PRD1.BASE:

```
IPNCSD.Z      TCP/IP definitions
```

Autoinstall Exits

Autoinstall exits are used for VTAM terminal definitions according to predefined models.

The autoinstall exit member for the CICS Transaction Server is IESZATDX, for CICS/VSE it is IESZATCO. Both exit members are available as skeletons of the same name in VSE/ICCF library 59.

CICS Transaction Server User Profiles

A user profile defines a user to z/VSE. It includes user ID and password for signing on to the system. z/VSE includes the following predefined CICS user profiles for the CICS Transaction Server:

- DBDCCICS

  This is the user ID for the primary CICS Transaction Server. and is required for startup. This name is also used as CICS Region Prefix to identify this CICS Transaction Server and the transactions belonging to it (in security table DTSECTXN, for example).
- PRODCICS

  This is the user ID for the secondary CICS Transaction Server. and is required for startup. This name is also used as CICS Region Prefix to identify this CICS Transaction Server and the transactions belonging to it (in security table DTSECTXN, for example).
- CICSUSER

  This is a default user ID required for CICS Transaction Server startup. It performs security checking for users that are not signed on.
- CNSL

  This is a default user ID that allows all CICS transactions to run on the system console.

Refer to “z/VSE User Profiles” on page 189 for additional profile information.

CICS Transaction Server Monitoring and Statistics Support

There are significant changes concerning the monitoring and statistics support of the CICS Transaction Server compared to CICS/VSE.

Statistical and monitoring data is written to data sets of the Data Management Facility (DMF), rather than to CICS journals. This data can be used for subsequent processing by the new monitoring utility program, **DFHMNDUP**. DMF requires a partition of its own.

The statistics utility program **DFHSTUP** has been enhanced.

To show the different ways of how to use the commands EXEC CICS INQUIRE and EXEC CICS COLLECT STATISTICS, a new set of programs (**DFH0STAT**) for producing CICS statistics is part of the CICS Transaction Server. DFH0STAT provides CICS Transaction Server statistics at shutdown similar to those provided by CICS/VSE. Also refer to "DFH0STAT Considerations" on page 204.

Skeletons for Monitoring and Statistics Support

The following skeletons are provided for the monotoring and statistics support:

DFHMNDUP Monitoring support CICS Transaction Server.

DFHMOLS Monitoring support CICS Transaction Server.

SKDMFPR Print statistics CICS Transaction Server.

SKDMFST Startup monitoring partition CICS Transaction Server.

SKJOBDMF Job accounting (DMF records).

SKJOURN Format journal data sets for CICS Transaction Server.

Characteristics of a CICS Coexistence Environment

Figure 20 on page 199 shows the major differences of the support provided by the CICS Transaction Server and CICS/VSE in a coexistence environment.

After initial installation, the CICS Transaction Server is running in the F2 partition ready for use. CICS/VSE must be installed separately as described under "Installing CICS/VSE" on page 205.

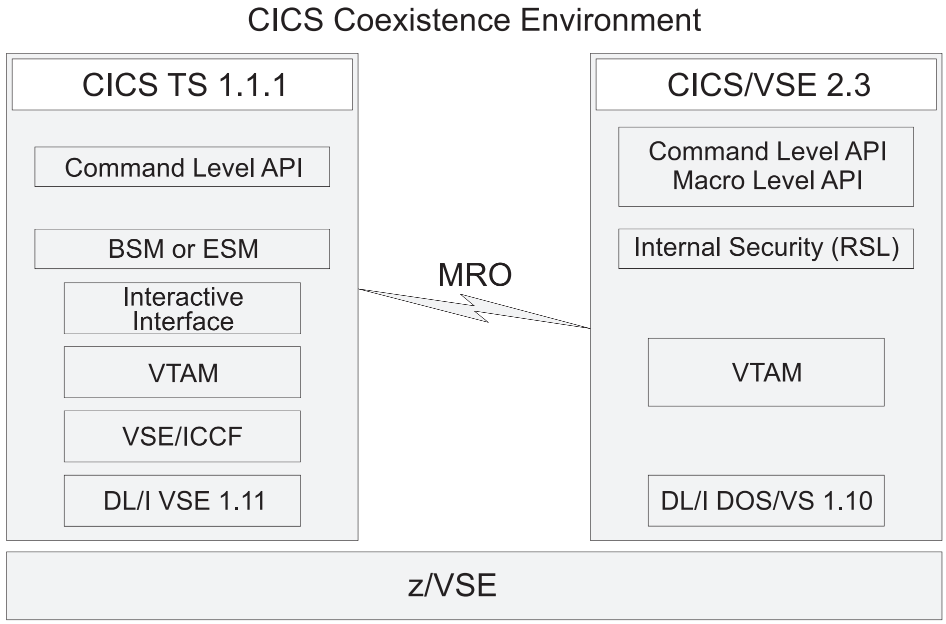


Figure 20. CICS Coexistence Environment

In a CICS Coexistence Environment you can:

- Run the CICS Transaction Server 1.1 and CICS/VSE 2.3 independently.
- Connect the CICS Transaction Server 1.1 and CICS/VSE 2.3 via MRO.
- Migrate CICS/VSE 2.3 applications to the CICS Transaction Server 1.1.

Under CICS/VSE 2.3, you can run DL/I DOS/VS 1.10. In addition, CICS/VSE 2.3 provides RSL security for CICS resources (where RSL stands for Resource Security Level). However, CICS/VSE 2.3 has no Interactive Interface and no VSE/ICCF support, and provides no OLPD (Online Problem Determination) support.

The CICS Transaction Server includes support for the Interactive Interface, VSE/ICCF, and the BSM (Basic Security Manager). It provides support for DL/I VSE 1.11 and for using an ESM (External Security Manager).

Since the CICS Transaction Server offers no RSL security for CICS resources as does CICS/VSE, you may consider the installation of an External Security Manager (ESM). Refer to Chapter 19, “Security Support,” on page 233 for further details on ESM.

CSD File Considerations for Coexistence Environment

After initial installation, only one CICS system, the primary CICS Transaction Server is available and running. It also includes a CSD (CICS System Definition) file. If you install in addition CICS/VSE to create a coexistence environment, CICS/VSE can:

- Have a CSD file of its own.

  In this case you must use skeleton **SKPREPCO** to define the resources for CICS/VSE. The skeleton defines among other resources a separate CSD file for CICS/VSE.
- Share the existing CSD file of the CICS Transaction Server.

In this case you must use skeleton **SKPREPSO** to define the resources for CICS/VSE. The skeleton defines resources for CICS/VSE but not a separate CSD file.

Information stored in the CICS/VSE tables is accessed directly with the exception of the DFHPCT and DFHPPT tables. Their information is stored in the CSD file and read from there. This is the reason that two versions of these tables (skeletons) are needed: for a shared CSD and for two separate (non-shared) CSDs.

In case of separate (non-shared) CSDs, the skeleton names are:

DFHPCTCO

DFHPPTCO

In case of a shared CSD, the skeleton names are:
- DFHPCTSO
- DFHPPTSO

It is recommended that the table entries in DFHSITCO show NO for PCT and PPT. The CICS suffix used is CO as for a separate CSD file. Refer to the manual *z/VSE Administration* for details. This manual also describes when and how to use which skeleton (table) for customization.

If you have transaction security on CICS/VSE (or on both CICS Transaction Server and CICS/VSE) you may include the correct transaction security keys for your CICS/VSE in the common CSD file, using the PF2 toggle key. This key allows you enable the *CICS/VSE compatibility mode*. In this mode, you can enter or change the TRANSEC and RSL classes.

Security Considerations

Security for the CICS Transaction Server is provided through the BSM (Basic Security Manager).

Security for CICS/VSE is provided through the SNT (Sign-on Table).

Installing a Second CICS Transaction Server

This section introduces the installation of a second CICS Transaction Server. For a detailed description of the required installation steps, refer to the manual *z/VSE Administration*.

The installation support provides the following skeletons in VSE/ICCF library 59:

SKCICS2 (for defining startup)

SKPREPC2 (for defining resources)

For an overview on the skeletons and tables provided, refer to "Overview on CICS Skeletons and Tables" on page 195.

Environment Characteristics

The support defines a secondary CICS Transaction Server, named PRODCICS, for use in a production environment. PRODCICS is defined to communicate with DBDCCICS, the primary CICS Transaction Server, via **MRO** (Multi-Region Option). Refer to the manual *z/VSE Administration* for the customization tasks involved.

The two CICS systems can also communicate via ISC (Inter-System Communication). In case of an ISC connection, the two CICS systems can run on the same processor or on different processors. For further ISC information, you may also refer to “Activating the APPC Support” on page 204.

Note that the second CICS Transaction Server can run in a dynamic partition as well.

The environment has the following characteristics:

- The two CICS Transaction Servers share the VSE.CONTROL.FILE. Therefore, the user definitions and applications stored in this file are known to and available for both CICS systems.
- The installation support provided assumes that you use the autoinstall function to define terminals and terminal printers for both CICS systems. For that reason, VTAM must be up and running.
- The second CICS Transaction Server will have an Interactive Interface without the services dependent on VSE/ICCF. This means that your use of the Interactive Interface is restricted to the following functions:
  - Sign on to and sign off from the system.
  - Select and control the processing of CICS applications.
  - Work with the Online Problem Determination (OLPD) tool.
  - Work with the Workstation File Transfer Support.
  - Work with the following dialogs:

    Inspect Message Log

    Display Channel and Device Activity

    Display System Activity

    Display Storage Layout

    Display Active Users/Send Message
  - Work with most of the functions of the *Manage Batch Queues*
  - Work with the dialogs for maintaining user profiles, selection panels, and application profiles. VSE/ICCF information, however, cannot be changed.

Planning for the Second CICS Transaction Server

Virtual Storage Requirements

During initial installation you **must** select either predefined **environment B or C**. Environment A does not allow you to run a second CICS Transaction Server.

These are the storage values defined for the second CICS Transaction Server, which is assumed to run in partition F8.

- For environment B:
```
ALLOC F8=50M
SIZE F8=2M
```
- For environment C:
```
ALLOC F8=512M
SIZE F8=2M
```

Note that these are default values. The values required depend on the size of the applications you are going to run. The minimum value required is 30M.

If required, adjust the partition values by using skeleton SKALLOCB or SKALLOCC. The SETPFIX value is set by the startup procedure provided (skeleton SKCICS2).

If you reduce the default value for ALLOC, you must also reduce the value for EDSALIM in the DFHSIT.

Disk Storage Requirements for System Files

While some system files are shared between the two CICS Transaction Servers, others must be defined additionally for the second CICS Transaction Server.

The following files are shared with DBDCCICS, the primary CICS Transaction Server:

```
CICS.CSD
CICS.DBDCCICS.DFHDMFA
CICS.DBDCCICS.DFHDMFB
PC.HOST.TRANSFER.FILE
VSE.CONTROL.FILE
VSE.TEXT.REPSTORY.FILE
VSE.MESSAGE.ROUTING.FILE
```

The system files to be defined for the second CICS Transaction Server are shown in Table 33 on page 203. Initially, you should reserve the amount of space shown.

z/VSE provides skeleton **SKPREPC2** to define these system files.

Table 33. System Files and Disk Space Needed for Second Predefined CICS Transaction Server

File Name	IBM-Provided File Identifier	File Type	Remarks
DFHRSD	CICS2.RSD	VSAM KSDS	
DFHNTRA	CICS2.TD.INTRA	VSAM ESDS	
DFHTEMP	CICS2.DFHTEMP	VSAM ESDS	
DFHGCD	CICS2.GCD	VSAM KSDS	
DFHLCD	CICS2.LCD	VSAM KSDS	
DFHDMPA	CICS2.DUMPA	VSAM ESDS	See Note 1
DFHDMPB	CICS2.DUMPB	VSAM ESDS	See Note 1
DFHAUXT	CICS2.AUXTRACE	VSAM ESDS	See Note 1
IESPRB	CICS2.ONLINE.PROB.DET.FILE	VSAM ESDS	

Approximate Amount of Disk Space Required				
	Number of FBA Blocks	Number of Tracks (full cylinders) IBM 3380	Number of Tracks (full cylinders) IBM 3390	Approximate Number of Megabytes (MB)
VSAM Space	24,000	390	360	12
Journal Files (see Note 2)	12,032	240	240	6

All files are ready for use after skeleton SKPREPC2 has been run.

Notes:

1. The file is created dynamically when required. It is released again (on user request) when it is no longer needed.
2. Possibly, your second CICS Transaction Server needs system journal files (DFHJCTxx). The size of a system journal file and of any CICS user journal file that you might want to use, is workload dependent. For guidance about journal file sizes, refer to the appropriate CICS documentation. Add the size of the planned journal files to your disk space requirements.

   For CICS journal files, refer also to ″Note on CICS Journaling″ under “Non-VSE/VSAM System Files” on page 108.

Startup Considerations for a Dynamic Partition

The startup program DTRISTRT processes and sets variables in CPUVARn (shipped as CPUVAR1) such as XSTATxx or XMODxx only for static partitions. This means, it ignores these parameters if your second CICS Transaction Server runs in a dynamic partition. However, you can use the concept of controlling startup via SETPARM variables in CPUVARn by coding your own CICS startup procedure.

CICS Control Tables

For the second CICS Transaction Server, z/VSE provides a set of tables as listed in Table 32 on page 196. In general, these predefined tables need not be modified except for adding your own applications and resources.Minor changes are to be made to the DFHSIT of the primary CICS (DBDCCICS) if the two CICS Transaction Servers are to communicate with each other via MRO or ISC as described in the manual *z/VSE Administration* under “Installation Tasks for a Second CICS Transaction Server”.

DFH0STAT Considerations

DFH0STAT uses EXEC CICS COLLECT STATISTICS to retrieve statistical information. You should be aware, however, of the following restrictions when using DFH0STAT:

1. If a CEMT PERF STAT RESET is issued or the ″statistics collection interval″ has expired or midnight is reached (all statistics information is written to DMF), the counter fields for most resources are reset. DFH0STAT cannot collect these numbers. Therefore, you should consider to set interval statistics collection STATRCD=OFF to avoid a counter reset.

   But be aware that by doing so you will loose any interval statistics information with DFHSTUP.
2. If an autoinstalled resource like a terminal or a program is removed from a CICS partition, all statistics information is written to the DMF before deleting the resource control blocks. Deletion of these control blocks means also deleting all counters for it. A subsequent usage of DFH0STAT will **not** show any counters for this resource. When running DFHSTUP against the DMF dataset, all of the information will be shown.
3. If an uncontrolled shutdown occurs, DFH0STAT in DFHPLTSD will **not** produce any output. DFHSTUP may not show the last activities but all the previous ones.

   Refer also to Note 1 for Table 32 on page 196.

Because of the restrictions outlined above, it may be the better solution for a production system to use DFHSTUP (instead of DFH0STAT) for collecting statistics information.

Definition of DFH0STAT, DFH$STAS and DFH$STCN

All three programs have to be defined with EXECKEY=CICS if DFH0STAT is put into the DFHPLTSD. Programs defined in DFHPLTSD run under CICS control in CICS mode.

DFHPLTSD is available as skeleton in VSE/ICCF library 59.

Activating the APPC Support

This support is available for the CICS Transaction Server as well as for CICS/VSE.

APPC (advanced program-to-program communication) is important for programs such as the VSE/ESA Distributed Workstation Feature. Among other customization tasks for activating the APPC support, the ISC (Inter-System Communication) parameter of CICS/VSE must be set to YES. This can be done in two ways:

- In the CICS/VSE startup job.
- In the CICS/VSE DFHSITxx table.

  DFHSITSP for the CICS Transaction Server must be re-assembled with ISC=YES. DFHSITC2, the initialization table for the second CICS Transaction Server is shipped with ISC=YES and need not be re-assembled.

ISC=YES enables the APPC connection between the CICS system in the host and a workstation with the VSE/ESA Distributed Workstation Feature installed.

For further details about the tasks involved in activating the APPC support, consult the manual *VSE/ESA Distributed Workstation Feature - User's Guide (OS/2 Support)* or the Windows version of this manual. The manuals are available online via the z/VSE home page.

Installing CICS/VSE

This section introduces the installation support z/VSE provides for installing CICS/VSE. CICS/VSE can only run together with the primary CICS Transaction Server DBDCCICS in a **CICS Coexistence Environment**.

Note: CICS/VSE is shipped on the extended base tape and therefore not part of your system after initial installation. You must first use the dialogs for optional program installation to make CICS/VSE part of your system.

It is recommended to install the generation part only if it is explicitly needed.

The installation support provides the following skeletons in VSE/ICCF library 59:

SKCICSOL (for defining startup)

SKPREPCO (for defining resources including a separate CSD file)

SKPREPSO (for defining resources and sharing the CICS TS CSD file)

For an overview on the skeletons and tables provided, refer to "Overview on CICS Skeletons and Tables" on page 195.

Environment Characteristics

The support defines a CICS/VSE system, named OLDCICS, for use in a production environment. OLDCICS is defined to communicate with DBDCCICS, the primary CICS Transaction Server, via **MRO** (Multi-Region Option). Refer to the manual *z/VSE Administration* for the customization tasks involved.

Note that CICS/VSE can also run in a dynamic partition.

The environment has the following characteristics:

- The support assumes that you use the autoinstall function of CICS to define terminals and terminal printers. For that reason, VTAM must be up and running. There is a separate exit (IESZATCO) for CICS/VSE autoinstall support.
- In case you have a need for using BTAM terminals, you must define them in the TCT table DFHTCTCO. This applies also to VTAM if you do not use the RDO autoinstall function.
- CICS/VSE does not provide Interactive Interface support.

Planning for CICS/VSE

Note: Because of storage key restrictions, do not install CICS/VSE in another partition than F4.

Virtual Storage Requirements

CICS/VSE requires that you selected predefined environment B for initial installation. These are the storage values required for CICS/VSE, which is assumed to run in partition F4.

- For environment B:

```
ALLOC F4=20M
SIZE F4=2M
```

- For environment C:

```
ALLOC F4=32M
SIZE F4=2M
```

If LE/VSE is not to be used, these values can be smaller.

If required, adjust the partition values by using skeleton SKALLOCB. The SETPFIX value is set by the startup procedure provided (skeleton SKCICSOL).

Disk Storage Requirements for System Files

The system files to be defined for the CICS/VSE are shown in Table 34. Initially, you should reserve the amount of space shown.

z/VSE provides skeletons **SKPREPCO** and **SKPREPSO** to define the system files.

Note: Job SKPREPSO will only work if there are any transactions included in skeleton DFHPCTSO. If skeleton DFHPCTSO is not used, the MIGRATE command in job DFHCSDSO should be removed.

Table 34. System Files and Disk Space Needed for CICS/VSE

File Name	IBM-Provided File Identifier	File Type	Remarks
DFHCSD	CICSO.CSD	VSAM KSDS	See Note 1
DFHRSD	CICSO.RSD	VSAM KSDS	
DFHNTRA	CICSO.TD.INTRA	VSAM ESDS	
DFHTEMP	CICSO.DFHTEMP	VSAM ESDS	
DFHSTM	CICSO.AUTO.STATS.A	VSAM SAM-ESDS	
DFHSTN	CICSO.AUTO.STATS.B	VSAM SAM-ESDS	
DFHDMPA	CICSO.DUMPA	VSAM ESDS	See Note 2
DFHDMPB	CICSO.DUMPB	VSAM ESDS	See Note 2
DFHAUXT	CICSO.AUXTRACE	VSAM ESDS	See Note 2

Approximate Amount of Disk Space Required

	Number of FBA Blocks	Number of Tracks (full cylinders) IBM 3380	Number of Tracks (full cylinders) IBM 3390	Approximate Number of Megabytes (MB)
VSAM Space	12,288	195	180	6
Journal Files (see Note 3)	12,288	240	240	6

All files are ready for use after skeleton SKPREPCO or SKPREPSO has been run.

Notes:

1. Needed only in case of a separate CSD.
2. The file is created dynamically when required. It is released again (on user request) when it is no longer needed.
3. Possibly, your CICS/VSE needs system journal files (DFHJCTCO). The size of a system journal file and of any CICS user journal file that you might want to use, is workload dependent. For guidance about journal file sizes, refer to the appropriate CICS/VSE documentation. Add the size of the planned journal files to your disk space requirements.

   For CICS journal files, refer also to ″Note on CICS Journaling″ under “Non-VSE/VSAM System Files” on page 108. See also skeleton SKJOURO in VSE/ICCF library 59.

Startup Considerations for a Dynamic Partition

The startup program DTRISTRT processes and sets variables in CPUVARn (shipped as CPUVAR1) such as XSTATxx or XMODxx only for static partitions. This means, it ignores these parameters if CICS/VSE runs in a dynamic partition. However, you can use the concept of controlling startup via SETPARM variables in CPUVARn by coding your own CICS startup procedure.

CICS/VSE Control Tables

For CICS/VSE, z/VSE includes a set of skeletons as listed in Table 32 on page 196. In general, these predefined tables need not be modified. Minor changes are to be made to the DFHSIT of the primary CICS Transaction Server (DBDCCICS) if CICS/VSE is to communicate with DBDCCICS via MRO. Refer also to"CSD File Considerations for Coexistence Environment" on page 199.

CICS/VSE provides the autoinstall exit member IESZATCO for VTAM terminal definitions. For BTAM terminals you require the TCT table DFHTCTCO.

For further details about the installation steps required for installing CICS/VSE, refer to the manual *z/VSE Administration*.

The CICS Report Controller

The CICS Report Controller is available with the **CICS Transaction Server** as well as with **CICS/VSE**.

It allows users to control and print information (*reports*) created by both online applications and batch programs. See the corresponding CICS documentation for details.

During system startup, the Report Controller it is automatically activated and made available through the *CICS Supplied Transactions* panel of the Interactive Interface.

The following manuals are available for the CICS Transaction Server:

CICS TS Report Controller Planning Guide

CICS TS Report Controller User

The following manuals are available for CICS/VSE:

CICS/VSE Report Controller Planning Guide

CICS/VSE Report Controller User's Guide

z/VSE's Implementation of the Report Controller

The system initialization table (DFHSITSP) that z/VSE provides for the *primary CICS Transaction Server* contains the following entry for the Report Controller:

```
SPOOL=(YES,A,A)
```

YES specifies that the Report Controller is part of the CICS system. **A,A** mean that its ID is *SYSCICSA* and its default class is *A*.

The DFHSITxx for a *secondary CICS Transaction Server* or *CICS/VSE*, includes:

```
SPOOL=(YES,B,A)
```

The DFHSITxx for a *third CICS Transaction Server* or *CICS/VSE*, includes:

```
SPOOL=(YES,C,A)
```

CICS/VSE Report Controller Specifics

This section discusses specifics of the CICS/VSE Report Controller support. This topic is of interest to you if you have installed CICS/VSE in addition to the CICS Transaction Server. Refer also to "Characteristics of a CICS Coexistence Environment" on page 198.

CICS/VSE Resource Security Levels (RSL) for Terminal Printers

Note: The following description of Resource Security Levels (RSL) applies to CICS/VSE only. The CICS Transaction Server does not support RSL and requires the installation of an ESM (External Security Manager) which provides such support.

When you install z/VSE, the initial configuration for your system will *not* contain a terminal printer. You must use the *Configure Hardware* dialog to add one of more of these devices to the configuration.

Terminal printers added using the dialog have the following default RSL:

- **SPRTRSL=1**

  Any user whose profile specifies RSL=1 can change the characteristics of this printer. For the Report Controller, such characteristics include the forms used for printing.

  In the pregenerated z/VSE system, only the profile for the user SYSA specifies RSL=1. PROG and OPER do not.
- **OPERRSL=1**

  The terminal printer will print reports that specify RSL=1. Such reports can either be output created by online applications or batch programs.

You can change these defaults by editing either *DTRTCTxL* (for a non-SNA terminal printer) or *DTRTCTxS* (for a SNA terminal printer). These members of VSE/ICCF library 51 have DFHTCT entries for local terminal printers that you add to your configuration via the *Configure Hardware* dialog.

After editing the library members, be sure to reassemble the TCT.

CICS/VSE Transactions CEMS and CEOS

CICS/VSE provides two transactions for the Report Controller, *CEMS* and *CEOS*. CEMS makes all functions of the Report Controller available to a user; CEOS a subset of those functions.

The Interactive Interface panel hierarchy defined for the user SYSA has a selection for CEMS. Through it, a user whose profile specifies RSL=1 can:

- Print, hold, and delete reports created by online applications or change their characteristics (priority, for example).
- Start and stop terminal printers or change their characteristics.
- Manipulate jobs in the batch queues. This includes releasing jobs for processing and printing job output.
- Create reports from data in transient data queues.

The panel hierarchies for users PROG and OPER have a selection for CEOS. These users can send reports to a printer and check their status. Because their user profiles do *not* specify RSL=1, however, they cannot change the characteristics of a report or printer. In addition, they cannot release batch jobs for processing or create reports from data in transient data queues.

CICS/VSE Transient Data Queues Used by the Report Controller

In z/VSE, the Report Controller makes use of two transient data queues:

- **CSPA**

  This queue holds audit information about:

  - Reports printed on a printer and
  - Changes made to the characteristics or status of reports.

  When CICS/VSE is shut down, this information is forwarded to SYSLST and automatically printed.

- **CSPW**

  Action messages for a print task (such as a forms change request) and severe error messages are routed to CSPW.

  As defined in DFHDCTSP, the final destination of these messages can either be a terminal or a file. The default for z/VSE is the system console.

  If you wish, you can route the messages to any other terminal by changing the terminal ID specified in **IEP1** of DFHDCTSP. You also can use **IEP2** to hold the messages in a file. This file then can be manipulated via option **4** (Transient data queue selection) of the selection panel for CEMS.

  Note: If you hold the messages in a file, be sure to print or delete them from time to time. If you do not, transient data space may gradually fill up.

CICS/VSE Resource Security Level for Batch Reports

z/VSE automatically assigns RSL=1 to batch reports. This matches the default OPERRSL for a terminal printer that you add to your configuration via the *Configure Hardware* dialog.

Defining CICS/VSE Resource Security Levels

Before you define RSLs for online transactions or change the RSL values that z/VSE predefines for you, you should read the information about security levels and keys in the manual *CICS/VSE Facilities and Planning Guide*. Also note that an RSL other than 1 for batch job output can be specified in a VSE/POWER $$ LST statement. In a z/VSE system, that statement must read:

```
* $$ LST CLASS=A,CICSDATA=n...
```

where **n** is the RSL for the job. **CICSDATA** is a VSE/POWER JECL output operand that is defined in the startup for VSE/POWER using the command:

```
DEFINE L,CICSDATA,3F00,1,255,*
```

Other CICS/VSE Facilities

Server Support for CICS/VSE Clients

A CICS/VSE client is not a full-function CICS/VSE system. But it contains code to enable it to access the services (such as transactions or programs) of a CICS/VSE system. CICS/VSE systems to which Clients are connected are known as CICS/VSE servers. In a z/VSE environment, z/VSE is the server and a CICS/VSE client may be an operating system such as OS/2 or Microsoft Windows.

The manual *CICS/VSE Server Support for CICS Clients*, SC33-1712, describes the support in detail.

Application Migration Aid

The CICS/VSE Application Migration Aid is designed to assist you in converting your CICS applications from the macro-level API (application programming interface) to the command-level API. The migration aid can be used for applications written in assembler, COBOL, or PL/I. Compared to macros, commands are easier to use, have more functions, offer virtual storage constraint relief, and ease migration to later CICS releases.

The manual *CICS/VSE Application Migration Aid Guide*, SC33-1901, describes the support in detail.

Security Migration Aid

CICS/VSE 2.3 includes the Security Migration Aid which helps you convert your existing CICS/VSE security definitions to the CICS Transaction Server. The file created by the migration aid serves as input for migrating CICS/VSE TRANSEC values into DTSECTXN entries for the CICS Transaction Server. The manual *z/VSE Administration* describes the migration of TRANSEC values in detail.

The manual *CICS/VSE Server Support for CICS Clients*, SC33-1406, describes the migration aid in detail.

Chapter 17. Console Support

Overview

The major items of the enhanced console support, introduced with VSE/ESA 2.1, include:

- Console integration.

  All functions provided today for the z/VSE system console are also available for a PS/2 based processor console Such a console is generally referred to as **Integrated Console**. The Integrated Console can serve as system console when installing z/VSE.

  For daily operation with high workloads and heavy message traffic the performance of an Integrated Console is not sufficient. For that reason, daily system operation from a 3270 terminal has been further enhanced as outlined next under ″Improved CICS/VSE console support″.
- Improved CICS/VSE console support.

  This support allows to use CICS/VSE terminals (independent of VSE/ICCF) as master or user consoles. The support provides all the advantages of the current Interactive Interface console support and improved performance.
- Improved 3270 system console support as outlined below.
- Enhancements in the areas of:
  - IPL system console handling
  - Communicating with the operator
  - VM/VSE linkage functions
  - Console message routing
  - Hardcopy file

Console Types and Functions Supported

The following types of consoles are supported by z/VSE:

Integrated Console
: An integrated console allows to operate z/VSE from the console used on processors to control and maintain the configuration of the processor.

3270 Console
: This is the existing z/VSE full-screen system console support based on a local non-SNA 3270 terminal. This console is optional when the Integrated Console is used in connection with one of the following consoles:

  Interactive Interface master console

  CMS master console

  NetView master console.

  The functions, presentation characteristics, and performance have been significantly improved.

Line Mode Console
: This is the existing Line Mode system console based on a virtual 3215 printer-keyboard. Its use is only meaningful in an environment where z/VSE runs under VM. The functions and presentation characteristics are unchanged.

Interactive Interface Consoles
These consoles replace the former VSE/ICCF-based consoles. Their capabilities, presentation characteristics, and performance have been improved in such a way that they can effectively replace the 3270 system console

CMS Consoles
This is the existing facility, supported by the VM/VSE Interface Routines to operate a z/VSE guest system from a CMS console of VM. This support has been extended to dynamic partitions and to optionally route master console traffic to a predefined CMS user ID (SYSECHO command).

Supported Console Functions

The functions supported by the various types of consoles are shown in Table 35.

Table 35. Supported Console Functions

Console Functions	Integrated Console	3270 System Console	Line Mode Console	Interactive Interface Console	CMS Console
Interact with IPL	yes	yes	yes	no	no
Receive messages	yes	yes	yes	yes	yes
Enter message replies	yes	yes	yes	yes	yes
Enter system commands	yes	yes	yes	yes	yes
Receive command responses	yes	yes	yes	yes	yes
PF-Keys	(Note 1)	yes	(Note 2)	yes	(Note 2)
Extended Attributes	(Note 1)	yes	(Note 2)	yes	(Note 2)
Redisplay Hardcopy File	yes	yes	yes	yes	yes
Message Explanation, Help	no	yes	no	yes	no
Multiple User Support	no	no	no	yes	yes

Notes:
1. Presentation features are controlled by the Service Processor implementation.
2. Presentation features are controlled by VM.

Console Types (Logical Level)

On a logical level, z/VSE distinguishes between system, master, and user consoles. Table 36 on page 213 shows the combinations that are possible.

System Console
The system console is identified by being assigned the logical unit name **SYSLOG**. The 3270 system console, the Line Mode console, or the Integrated Console may be used for this purpose. Its function is taken over automatically by the Integrated Console, if available, and if the SYSLOG device fails. The system console is used to IPL the system and operates after IPL like a master console (3270 or Line Mode) or as a console of the ″last resort″ (Integrated Console). It has master console authorization and a user can reply to all outstanding messages and enter all system commands.

Master Console
A z/VSE system may have several master consoles active. Each master console receives all system messages which are not routed exclusively to a specific console, and it can reply to all outstanding messages and enter all

system commands. After IPL, a 3270 system console or a Line Mode system console also operates as a master console.

User Console
User consoles differ from master consoles with respect to their input capability and/or to the scope of messages routed to them. The following cases are possible:

1. Limited output with full input capability.

   The console receives only messages related to jobs submitted with a matching ECHO/ECHOU option or to programs invoked directly from the console, but the console is authorized for all system commands and replies and has full REDISPLAY capability. The typical user is a system administrator.
2. Limited output with restricted input capability.

   The console receives limited output as in the first case, but may only enter a restricted set of system commands and only reply to the messages it receives. REDISPLAY is also limited to messages and inputs routed to or entered from the console. The typical user is a program developer.
3. Restricted input capability with full output.

   The console receives all messages that are also routed to master consoles and has full REDISPLAY capability, but may only enter a restricted set of system commands and only reply to messages that are specifically routed to it. This console is typical for users that need to monitor system activity as, for example, in a printer room.

The section "Console Selection and Assignment" on page 217 describes how system, master, and user consoles are assigned.

Table 36. Supported Console Types (Logical Level)

Console Type	Integrated Console	3270 System Console	Line Mode Console	Interactive Interface Console	CMS Console
System Console	yes	yes	yes	no	no
Master Console	no	yes	yes	yes	yes
User Console	no	no	no	yes	yes

3270 Console Support

3270-type terminals are supported as:

- System console
- Master and user consoles (Interactive Interface consoles)

For the 3270 system console, a local non-SNA terminal is needed. But since the Integrated Console can also be used for this purpose, the 3270 system console is optional.

The functional characteristics of all 3270 consoles are identical, and their presentation characteristics depend only on the capabilities of the terminal model used.

3270 System Console Activation

The 3270 system console becomes active at IPL time when the console is identified as IPL communication device, or as soon as the system recognizes an ASSGN SYSLOG,cuu for the console device.

Activation of Interactive Interface Consoles

An Interactive Interface console becomes active when a user selects the *Console* dialog. The console type is determined by the user type and by user profile options. Refer also to “Console Selection and Assignment” on page 217.

Dropped Console Printer Support

The support of a console printer, in connection with a 3270 system console, has been dropped.

Modes of Operation

You can operate 3270-type consoles in the following modes:

- Console mode (C)

  The messages are displayed in the order as they are being generated by the system.
- Redisplay mode (R)

  Messages no longer visible on the screen can be redisplayed. The messages can be retrieved selectively or displayed sequentially as stored on the Hardcopy file.
- Explanation mode (E)

  This mode allows to display z/VSE message explanations. The display format is identical to the format used in the *z/VSE Messages and Codes* manual.
- Help mode (H)

  Help information is available for explaining panel functions and local messages.

Console Definitions

Console definitions consist of definitions for:

PF keys
Panel data
Local messages
Operation characteristics

These definitions are predefined and shipped as phase $IJBEDEF in library IJSYSRS.SYSLIB. Table 37 shows the default PF-key settings for 3270-type consoles as shipped with z/VSE. PF-key settings are candidates for user modifications and are therefore discussed in more detail.

Table 37. Predefined PF-Key Settings

Type	Key	Mode	Text	Command / Variable
PFKEY	1	C	′1=HLP′	′%HELP′
PFKEY	2	C	′2=CPY′	′%COPY ′′?CL′
PFKEY	3	C	′3=END′	′%END′
PFKEY	4	C	′4=RTN′	′%RETURN′
PFKEY	5	C	′5=DEL′	′%DELETE ′′?CL′′,′′?IN′
PFKEY	6	C	′6=DELS′	′%DELETE ′′?CL′′,′′SYSTEM′
PFKEY	7	C	′7=RED′	′%REDISPLAY ′′?IN′

Table 37. Predefined PF-Key Settings (continued)

Type	Key	Mode	Text	Command / Variable
PFKEY	8	C	’8=CONT’	’%CONTINUE’
PFKEY	9	C	’9=EXPL’	’%EXPLAIN ’’?TK’
PFKEY	10	C	’10=HLD’	’%CHANGE ’’HOLD’
PFKEY	11	C	’ ’	’’
PFKEY	12	C	’12=RTRV’	’%RETRIEVE’
PFKEY	ENTER	C	’INPUT’	’?IN’
PFKEY	CLEAR	C	’CLEAR’	’%CLEAR’
PFKEY	1	R	’1=HLP’	’%HELP’
PFKEY	2	R	’2=CPY’	’%COPY ’’?CL’
PFKEY	3	R	’3=END’	’%REDISPLAY E’
PFKEY	4	R	’ ’	’’
PFKEY	5	R	’ ’	’’
PFKEY	6	R	’6=CNCL’	’%REDI C’
PFKEY	7	R	’7=BWD’	’%REDI ’’?CL’’;’’B,’’?IN’
PFKEY	8	R	’8=FWD’	’%REDI ’’?CL’’;’’F,’’?IN’
PFKEY	9	R	’9=EXPL’	’%EXPLAIN ’’?TK’
PFKEY	10	R	’10=INP’	’?IN’
PFKEY	11	R	’ ’	’’
PFKEY	12	R	’12=INFO’	’%CHANGE INFO’
PFKEY	ENTER	R	’REDISPLY’	’%REDI ’’?CL’’;’’?IN’
PFKEY	CLEAR	R	’CLEAR’	’%CLEAR’
PFKEY	1	E	’1=HLP’	’%HELP’
PFKEY	2	E	’2=CPY’	’%COPY ’’?CL’
PFKEY	3	E	’3=END’	’%END’
PFKEY	4	E	’ ’	’’
PFKEY	5	E	’ ’	’’
PFKEY	6	E	’ ’	’’
PFKEY	7	E	’7=BWD’	’%BACKWARD’
PFKEY	8	E	’8=FWD’	’%FORWARD’
PFKEY	9	E	’9=EXPL’	’%EXPLAIN ’’?TK’
PFKEY	10	E	’10=INP’	’?IN’
PFKEY	ENTER	E	’EXPLAIN’	’%EXPLAIN ’’?TK’
PFKEY	CLEAR	E	’CLEAR’	’%CLEAR’
PFKEY	1	H	’1=HLP’	’%HELP’
PFKEY	2	H	’ ’	’’
PFKEY	3	H	’3=END’	’%END’
PFKEY	4	H	’ ’	’’
PFKEY	5	H	’ ’	’’
PFKEY	6	H	’ ’	’’
PFKEY	7	H	’7=BWD’	’%BACKWARD’

Table 37. Predefined PF-Key Settings (continued)

Type	Key	Mode	Text	Command / Variable
PFKEY	8	H	'8=FWD'	'%FORWARD'
PFKEY	9	H	' '	''
PFKEY	10	H	'10=INP'	'?IN'
PFKEY	ENTER	H	'HELP'	'%HELP'
PFKEY	CLEAR	H	'CLEAR'	'%CLEAR'

Macro IJBDEF

Macro IJBDEF is used for defining and modifying console definitions. The macro can be of type:

DSECT Creates a DSECT of the definition table

PANEL Defines a panel entry in definition table

PFKEY Defines a PF-key entry in definition table

MSG Defines a message text entry in definition table

DEFAULT Selects operation characteristics and message colors

GEN Starts the table generation process

As shown in Table 37 on page 214, for type PFKEY the following parameters can be specified:

1. PF-key number (1 through 12), or ENTER, or CLEAR.
2. Mode (C=Console, R=Redisplay, E=Explanation, H=Help).
3. Descriptive text (up to 12 characters).
4. Command string which may include variables.

For PFKEY, the following command and variable characteristics apply:

- A command may be a local command (to invoke a local console function) or a VSE command to be sent to the system.
- Only one local command is allowed and it must be the first item in the string.
- Local commands are not echoed like system commands.
- Local commands may be entered from the input line. For example, requesting a message explanation:

  ```
  %EXPLain 1Q15
  ```
- The following variables can be used with local commands:

  ```
  ?CL  (insert screen message line number pointed to by cursor)
  ?TK  (insert token (message ID) pointed to by cursor)
  ?IN  (insert input from the input line without trailing blanks)
  ```
- Variables are only substituted when being defined with PFKEY (but not when entered in the input line).

Modifying PF-Key Settings

PF-key settings (and other console definitions) can be changed separately for the 3270 system console by assembling member IJBEDEF, and for all Interactive Interface consoles (not for individual users) by assembling member IESEDEF.

Under "Tailoring Console Definitions", the manual *z/VSE Administration* describes in detail macro IJBDEF, member IJBEDEF, and the tasks involved for modifying console definitions.

Modifying Color Settings

The support offers additional IJBDEF DEFAULT specifications for the colors of messages and input lines displayed on the console as follows:

```
IJBDEF DEFAULT,msgtype,color
```

msgtype may be:

IMSG Information message

RMSG Immediate command response

PMSG Prompt message, needing a reply

AMSG Action message

EMSG System failure message

LMSG RED L response (redisplay)

HMSG RED H response (redisplay)

EINP Echoed console input

color may be: 00, F0 through FF, with the same color conventions already supported for panel fields. Refer to the manual *z/VSE Administration* for details.

Currently reserved bytes in the default table generated by IJBDEF are used to store the new color specifications. These bytes are currently generated to contain X'00', and remain X'00' if the corresponding DEFAULT specification was omitted or specifies X'00'.

The implemented algorithm replaces the standard colors generated by the console routers, if the corresponding byte in the default table contains a non-zero value. This ensures that standard colors continue to be displayed for existing versions of the default table.

Modules IJBEDEF and IESEDEF, which contain standard console definitions for the system console and the Interactive Interface consoles, remain unchanged. The new DEFAULT definitions must be explicitly added to obtain the desired alternate colors.

Console Selection and Assignment

Overview

Initial Installation

The initial installation procedure contains an ADD command for a dummy system console with predefined device number:

```
ADD  FFF,CONS
```

IPL may thus select as the **system console** for installation the Integrated Console with device number FFF (for cuu), depending on the IPL load parameter.

Performing IPL

During IPL, the **system console** is assigned to the Integrated Console, or to the 3270 or Line Mode console. Which console is finally selected depends on the following:

- IPL load parameter.

- Device availability.
- Console device specification in the IPL procedure.
- I/O interrupt received from a console device.

For further details refer to "IPL System Console Selection."

Running System

After IPL, the **system console** may be re-assigned at any time by changing the SYSLOG assignment:

```
ASSGN SYSLOG,cuu
```

1. **Master consoles** can be requested and are assigned as follows:
   - For Interactive Interface users defined in their profiles as type 1 with full console output option or type 2 with full input and output options, by selecting the CONSOLE dialog. Refer to "Maintaining User Profiles" in the manual *z/VSE Administration* for details.
   - One CMS master consoles (for operating z/VSE from VM) through the SYSECHO ON command. Details about the command are provided under "SYSECHO Command" on page 221.
   - A NetView master console through the VSE/OCCF QLOGON command.

   Typically, most messages are then routed to one or more of these master consoles. The system console, when assigned to the Integrated Console, is then used for critical messages only which cannot be routed to any other master console. When the system console is assigned to a 3270 or Line Mode console, it continues to receive all message traffic not routed to NetView.
2. **User consoles** can be requested and are assigned as follows:
   - For Interactive Interface users defined in their profiles as type 1 with limited console output option or type 2 with limited input and/or output options, by selecting the CONSOLE dialog.
   - CMS user consoles by logging on to VM. Such a user has limited operating capabilities for z/VSE.

IPL System Console Selection

IPL supports the Integrated Console as system console (in addition to a 3270 or Line Mode system console). The basic selection rules are:

1. If a console type is specified in the IPL load parameter, then the system will route the messages to that device. This can be the Integrated Console or a local console. Local console means in this case 3270 or Line Mode console.
   a. If a local console is requested or the Integrated Console is not available, the system will route messages to the local console specified in the ASI IPL procedure.
   b. If that device is not available or not operational, the system will wait for an interrupt from a local console.
2. If a system is IPLed without specification of a device type in the IPL load parameter, the system will route messages to a local console (3270 or Line Mode console).
   a. The system selects first the console specified in the ASI IPL procedure.
   b. If this device is not operational, the system waits for an interrupt from a local console.

The IPL Load Parameter

To support the Integrated Console (of S/390 service processors), a new IPL load parameter which allows the specification of the following operands has been introduced:

Console type which specifies whether the messages are to be routed to an Integrated Console or to a local console.

IPL message suppression code which may be used to request message suppression and command logging during IPL.

IPL prompting code which may be used to request a prompting for IPL parameters (this is identical to the former IPLSTOP function). How to use this function is described in detail in the manual *z/VSE Guide to System Functions* under “Starting the System”.

Startup mode prompting code which may be used to request prompting for a startup mode such as BASIC or MINI.

For a detailed description of the available operands of the IPL load parameter refer to the manual *z/VSE System Control Statements* under “The IPL Load Parameter”.

Message Flow and Message Logging

Routing Rules

When multiple consoles are concurrently active, messages and DOM (Delete Operator Message) requests are routed to a selected subset of these consoles according to the following routing criteria:

- Messages or responses destined for a specific console are routed to that console if it is active. This is the only criteria for **user consoles** with limited input capability.
- Messages that are (also) destined for **master consoles** are routed to them based on matching routing codes, message level, and automation option.

  The automation option, activated for selected messages by VSE/OCCF for example, is taken into account only when at least one automation console, like the NetView console, is active and eligible for the message.
- **Master console** messages that cannot be delivered to any active master console are routed to the **Integrated Console** if active.
- DOM requests are routed to all eligible consoles, based on a matching DOM option defined for each console.

Note that the above routing rules do not apply for messages originating from Interactive Partitions if these messages are intercepted by VSE/ICCF and are displayed exclusively on a VSE/ICCF user terminal.

Recovery Conditions and Actions

Recovery conditions and actions are summarized below:

- When a task terminates, messages issued by the task and awaiting a reply or action are removed from the ″hold″ state, and a corresponding DOM request is routed to eligible consoles. Incomplete messages are suppressed. All consoles activated by the terminating task are deactivated.
- When a console is deactivated or suspended, the system does not wait any longer for messages to be received or replied to by that console. All messages

that are only waiting to be routed to that console are deleted but logged when applicable. Messages that are to be routed exclusively to that console and need a reply cause the originating task(s) to be canceled. Other console messages which require an operator action or a reply remain pending.
- When a console with full output capability is activated or resumed, it receives all pending messages eligible for it before delivery of current messages begins.

Logging Messages on the Hardcopy File

All normal message traffic is logged by default on the Hardcopy File. Exceptions are DOM and message explanation requests, as well as redisplay commands and responses. Optionally, logging may also be suppressed (through the HCLOG command) for messages that are routed exclusively to user consoles with limited input capability and for the replies to such messages. Hardcopy file records include message routing information and the name of the originating console for commands and replies.

Security Aspects

The console support ensures that **user consoles** with limited input capability can only reply to and redisplay messages that were routed to them. For other commands, console authorization checking is performed.

When z/VSE security (access control) is active, the system determines the user ID and the authority level of inputs submitted by a **master console** on behalf of other users (for example, CMS users) and uses this information for authorization checking.

Communicating with the Operator

The following macros are available for application programs to communicate with the operator. They allow to submit messages or command responses, receive replies and control the disposition of submitted messages.

WTO (WTOR)
The WTO (Write to Operator) and WTOR (Write to Operator with Reply) macros have been extended and support new functions such as:
- Specification of message text lines by addresses.
- Concatenation of multiple WTOs to a single logical message of arbitrary length.
- Routing to a specified console.
- Specification of a correlation token for command responses.
- Identification of the console replying to a WTOR.
- Writing of log-only messages.

Both macros are compatible with the MVS/ESA macros of the same name. Compatibility with the previous level of WTO and WTOR is also preserved.

DOM The DOM (Delete Operator Message) macro was introduced with VSE/ESA 2.1. It allows to delete a submitted message or to mark it as deletable when the condition that caused the message no longer exists.

For a detailed description of these macros, their syntax and their operands, refer to "WTO Macro", "WTOR", and "DOM Macro" in the manual *z/VSE System Macros Reference*. Under "WTO, WTOR, DOM Usage Examples", the manual *z/VSE System Macros User's Guide* provides coding examples exemplifying how to use these macros in application programs.

Command Processing

Command Authorization

Commands are classified into three categories:

- **Restricted commands** which can be entered from system or master consoles only.
- **Semi-restricted commands** which can be entered from user consoles with limited input capability depending on certain argument values as follows:
  1. Partition related commands that are accepted only when an ECHO or ECHOU option for the originating console is effective for the job currently running in the specified partition (ECHO scope).

     The ECHO or ECHOU option is to be specified in the VSE/POWER * $$ JOB statement.
  2. Commands that are accepted from user consoles with limited input capability in ″query format only″.
- **General-use commands** that may be entered from any console.

For example, most AR (Attention Routine) commands are restricted and can be entered only from a console with full input capability except for the commands listed below:

CANCEL	ECHO scope
GETVIS	General use
MAP	General use
MSG	ECHO scope
MSECS	Query only
PAUSE	ECHO scope
PRTY	Query only
PRTYIO	Query only
STATUS	General use
TPBAL	Query only
VOLUME	General use

Further Information

Consult the following manuals for command details including their classification:

z/VSE System Control Statements

VSE/POWER Administration and Operation

VSE/ICCF Administration and Operation

New and Changed Commands

Following is a summary of new and changed commands. The manual *z/VSE System Control Statements* describes these commands in detail.

SYSECHO Command

This is a **new** command for use in a VM/VSE environment. It allows a VM user to operate as a z/VSE master console.

The command can be issued from a system or a master console, or it can be included in the BG JCL startup procedure ($0JCL). The command is rejected when startup for z/VSE was performed with an IPL procedure including VMCF=NO in the IPL SYS command.

* CP Command

* CP is an Attention Routine (AR) command for the VM/VSE environment. It allows z/VSE operators from a system or master console to submit CP commands to VM and receive the associated responses.

MSG Command

The MSG command transfers control to an operator communications (OC) exit routine. The MSG command has been extended allowing to pass control information to the OC exit routine (STXIT OC macro with MSGPARM=YES). The data is saved in the OC exit save area in system GETVIS storage. Refer also to "STXIT OC Macro Extensions" on page 225.

OPERATE Command

The OPERATE command is used to query or change the system operating mode and the state of the system console (SYSLOG). The format of this command is unchanged (except for a new CONN option) but the resulting modes and states reflect the new console capabilities.

EXPLAIN Command

This is a new command to activate (ON) or deactivate (OFF) the online display of message explanations and help text. This information is stored in the file VSE.MESSAGES.ONLINE on DOSRES. The command can be issued from the system or a master console, or it can be included in the startup procedure for the BG partition ($0JCL). After initial installation of z/VSE, the status is ON.

HCLOG Command

HCLOG is a **new** Attention Routine (AR) command that allows to control message logging on the Hardcopy file.

REDISPLAY Command

The REDISPLAY command has been extended and replaces the former D command. It offers additional options such as filter functions for selectively retrieving messages stored on the Hardcopy file. On a user console, only messages that were destined for this console can be redisplayed. On a master console, the message traffic of all consoles can be redisplayed.

VM Linkage Commands

Details about changes to VM Linkage commands are provided under "VM/VSE Linkage Functions."

VM/VSE Linkage Functions

Operating z/VSE from a CMS Console

As in previous releases, a CMS user of VM can submit jobs to z/VSE guests through the **SUBVSE** interface routine. The **VSECMD** interface routine for operating z/VSE guests from VM has been generalized.

The interface routines VSEREP, VSECP and VSEMSG have been dropped since their functions are also covered by VSECMD now.

Under "VSECMD: Send Commands and Replies to the VSE/ESA Virtual Machine", the manual *z/VSE Operation* describes in detail the command level of the interface routines.

Requirements

To use the VM/VSE linkage functions, the following requirements must be met:

- The VM/VSE interface routines must first be installed through skeleton SKVMVSE. This task is described in detail in the manual *z/VSE Installation* under "How to Install the VM/VSE Interface".
- The IPL startup procedure of a z/VSE guest must include VMCF=YES in the IPL SYS command.

Planning Details

1. **SUBVSE Interface/Command**

   SUBVSE allows a CMS user to submit jobs to z/VSE guests. The record length is restricted to a maximum of 80 characters.

   For the ECHO function the following applies:

   - ECHO is now supported as an option in the VSE/POWER * $$ JOB statement and must be coded explicitly like other output routing options such as LDEST or PDEST. The FOR and ECHO arguments for SUBVSE have therefore been dropped.
   - If the ECHO or ECHOU option is specified for a submitted job, all messages related to the processing of that job are routed to the console of the ECHO user ID. If no z/VSE console is active for that user ID, the messages are routed to the CMS console with that user ID (typically the job submitter).

     Messages that are sent to a CMS user and need a reply can then be answered with VSECMD provided the user is authorized for the reply.

     Master authorization is required for selected messages that are also routed to master consoles.

2. **VSECMD Interface/Command**

   VSECMD allows CMS users to operate z/VSE guests and is the only way of submitting console input (command or reply) to a z/VSE guest, including redisplay commands for the Hardcopy file.

   To ensure that VSECMD works properly, z/VSE user IDs and CMS user IDs of VM should be unique. If identical user IDs exist, they must identify the same person and cannot be used concurrently to access z/VSE console functions from both, CMS and z/VSE.

   To ensure z/VSE integrity and security, master console authority for CMS users should be implemented as follows:

   - Perform IPL of z/VSE with SEC=YES in the IPL SYS command.
   - Define CMS users requiring z/VSE master console authority in the z/VSE access control table DTSECTAB with MCONS=YES and/or AUTH=YES. Under "Defining Entries in DTSECTAB", the manual *z/VSE Guide to System Functions* describes DTSECTAB and its parameters in detail.

   Implementing the system as described above ensures that:

   - A user with "master" authority can enter any command and reply to any outstanding message (whether or not it was routed to the own CMS user ID).
   - A user without "master" authority can only enter a subset of commands and reply to most messages that were routed to the own CMS user ID, based on the ECHO or ECHOU option of a submitted job.

Since not all console traffic is recorded on the Hardcopy file (as described under "Logging Messages on the Hardcopy File" on page 220), some previous user inputs or z/VSE messages may be missing in the redisplay response. The CMS console log may be used to maintain a complete record of all CMS sessions, including all message traffic with z/VSE guests.

3. Two additional VM/VSE functions are provided through commands:
   - The SYSECHO command can be used to route all z/VSE master console messages to a CMS user ID specified in the command. This support can be exploited to automate z/VSE operations via a CMS program.
   - The * CP command can be used to allow z/VSE operators with master authority to submit VM CP commands and receive the associated responses.

   "SYSECHO" and "* CP" in the manual *z/VSE System Control Statements* describe in detail the commands SYSECHO and * CP.

Using REXX/VSE

REXX/VSE offers a feature called REXX/VSE Console Automation. The support helps to automate console operation by handling one or more z/VSE console sessions from a REXX program. z/VSE console commands can be imbedded into a REXX program and commands can be retrieved.

The following types of commands are supported:
- Attention routine (AR) commands
- Console redisplay commands
- VSE/POWER, VSE/ICCF, CICS, VTAM, and SQL/DS commands.

The manual *REXX/VSE Reference* provides a detailed description of the REXX/VSE Console Automation.

Miscellaneous Changes

VSE/POWER Changes

Command Authorization

All VSE/POWER commands are accepted from the system or a master console. From a user console, only selected commands or command variations are accepted. The manual *VSE/POWER Administration and Operation* provides details about VSE/POWER commands and their usage with the various console types.

ECHO Support for User Consoles

New ECHO or ECHOU parameters have been introduced for the VSE/POWER * **$$ JOB** statement. They allow to specify whether job-related system console messages are to be routed to a given user console.

The system checks first for an active z/VSE user console matching the specified user ID. If no such console exists and z/VSE runs under VM, an attempt is made to route the message to a CMS console with that user ID. If this is not successful, the message is ignored or, if a user reply is required, the job is canceled.

The former ECHO parameter of the JCL // JOB statement is no longer supported.

Message Prefix Changes

A job related message shows the prefix of the partition in which the job was actually executed. In former releases, such messages showed the prefix of the VSE/POWER partition (F1).

VSE/ICCF Command Authorization

All VSE/ICCF commands are accepted from a master console. From a user console, only the following commands are accepted:

/DISPLAY

/MAP

/USERS

JCL Command Authorization

The following JCL commands are accepted from a master console only:

ALLOC

DVCDN

DVCUP

HOLD

JCLEXIT

MSECS

NPGR

PRTY

ROD

SET

SIZE

START

STOP

UCS

UNBATCH

Security Enhancements

Master Console Capability

For security and integrity reasons the assignment of **master console** authority to CMS users can be controlled by assigning only to selected users this authority level. The following steps are required:

1. Perform IPL of z/VSE with SEC=YES in the IPL SYS command.
2. Define master console users with MCONS=YES or AUTH=YES in the VSE.CONTROL.FILE via the *Maintain User Profile* dialog. Refer to the *z/VSE Administration* manual for details on the dialog.

A user with MCONS=YES is authorized to submit input which requires master console authority.

STXIT OC Macro Extensions

The OC (operator communications) exit of the STXIT macro has been extended by the operand MSGPARM.

MSGPARM indicates that control information (routing and correlation parameters) is accepted from the MSG command. They are to be passed to the save area of the OC exit. For a detailed description of the macro, refer to the manual *z/VSE System Macros Reference* under “STXIT Macro”.

SPLEVEL Macro Extensions

The level of z/VSE macro expansions can be controlled with the SPLEVEL macro. The value of the global symbol &SYSSPLV determines the level used for expansion:

1 Expansion for VSE 1.1/1.2

2 or 3 Expansion for VSE 1.3

4 Expansion for VSE 2.1/2.2/2.3/2.4/2.5/2.6/2.7/3.1

Migration Considerations Related to Console Support

Initial Program Load (IPL)

IPL Load Parameter

To support the Integrated Console, the format and contents of the IPL load parameter has changed. The old format (IPLSTOP) is no longer supported. Refer to the manual *z/VSE System Control Statements* under “The IPL Load Parameter” for details about the format of the IPL load parameter.

IPL Communication Device

If your system console is an **Integrated Console**, the source of your IPL commands may be your IPL ASI procedure, or you may interactively enter the IPL commands at the Integrated Console.

The support for a diskette or a card reader as IPL communication device is no longer available.

Interrupting and Restarting IPL

When your system console is an **Integrated Console**, you must use the IPL load parameter if you want to interrupt and restart IPL (for modifications). Refer to the manual *z/VSE Guide to System Functions* under “Interrupt IPL Processing for Modifications” for details.

WTO and WTOR Macros

Both source and object level compatibility is preserved for programs using the WTO and WTOR macro version of earlier z/VSE releases. However, the VSE/ESA 2.1/2.2 default expansion is backward incompatible. If none of the parameters unique with VSE/ESA 2.1/2.2 is used, a backward compatible expansion suitable for execution on all z/VSE releases is possible for the VSE/ESA 2.1/2.2 WTO/WTOR. It can be achieved by issuing SPLEVEL SET=n (with n < 4) prior to the WTO/WTOR invocation. Refer also to “SPLEVEL Macro Extensions.”

VM Linkage Functions

VM/VSE Interface Routines

The interface routines VSEREP, VSECP, and VSEMSG are no longer available. Their functions are covered by VSECMD now. Refer to “VM/VSE Linkage Functions” on page 222 for further details.

ECHO Support

The ECHO parameter for the current SUBVSE support in the JCL // JOB statement is no longer supported. Instead, the ECHO parameter can be specified in the VSE/POWER * $$ JOB statement. This means that jobs must be submitted via VSE/POWER, if job messages are to be routed to a CMS user console.

Message Prefix Changes

The reply ID field of the message prefix added by the system to all console messages is extended from 3 to 4 digits Programs that process console messages (for automation purposes, for example) and depend on the length of the reply ID must be adapted accordingly.

IPL uses for its messages the prefix 0 (prefix of the BG partition).

Chapter 18. Data Compression Support

Data compression provides advantages such as the saving of disk space, reduced number of I/O operations, and reduced channel utilization. In a network it means faster data transmission and lower line costs.

z/VSE provides support for data compression (and expansion) through:

1. The hardware instruction CMPSC which is available on all the processors that are listed in "Processor Support" on page 4.
2. The CSRCMPSC macro which will use the hardware instruction if available; otherwise it simulates the CMPSC instruction. The CSRCMPSC macro is compatible with the MVS macro of the same name.

It is transparent to an application which of the two functions actually performs compression. IPL detects whether a processor supports instruction CMPSC and IPL message 0J76I informs about the compression support available.

A CMPSC application can only run if the hardware instruction is supported. A CSRCMPSC application, however, runs with the CMPSC hardware instruction if supported; if not supported the CSRCMPSC macro simulates the CMPSC instruction.

With z/VSE, you can exploit data compression as follows:

1. Using the data compression support provided by **VSE/VSAM** for ESDS, KSDS, and VRDS files. This support is available on the VSE/VSAM command level as well as through the VSE/VSAM dialogs of the Interactive Interface. Further details are provided below.
2. Using the **VTAM** data compression support as introduced below.
3. Writing or using CSRCMPSC applications as introduced below.
4. Writing or using CMPSC applications as described in the manual *ESA/390 Data Compression* (SA22-7208).

VSE/VSAM Data Compression Support

VSE/VSAM uses the CSRCMPSC interface to provide compression support for KSDS, VRDS, and ESDS files. To be eligible for compression, a VSE/VSAM file should at least have a size of 5MB. The average record length of such a file should be at least 40 bytes. For a KSDS file the record length (minus key length and key offset) must be greater than 40 bytes.

Data compression takes place on the record level. When a record is loaded into a file it is compressed; when the record is read from the file it is expanded again.

You request data compression for a VSE/VSAM cluster (file) in the IDCAMS command DEFINE CLUSTER with the parameter:

    COMPRESS

Control information for compression and expansion is kept in a dictionary. This dictionary is built by VSE/VSAM when loading the cluster and added to the CCDS (Compression Control Data Set). Any VSE/VSAM catalog including clusters that are to be compressed must have a CCDS defined. The reserved name for a CCDS is VSAM.COMPRESS.CONTROL.

The VSE/VSAM dialog *Define a New File* supports data compression parameters and the dialog *Define a New User Catalog* supports the creation of a CCDS. The dialogs are described in detail in the manual *z/VSE Administration* under "Managing VSE/VSEAM Files and Catalogs".

Consult the manuals *VSE/VSAM User's Guide and Application Programming* and *VSE/VSAM Commands* for further details about the VSE/VSAM data compression support and its related parameters.

VSE/VSAM Backup/Restore

Currently, VSE/VSAM Backup/Restore supports the COMPACT option. It creates a compressed backup copy of a file via software compaction.

It is not recommended to specify the COMPACT option for the BACKUP run of a cluster defined with the COMPRESS parameter. There would be little gain with regard to compression but considerable consumption of CPU time.

Compression Prediction Tool

A measure for compressibility is the *compression ratio*, which is the ratio of the length of the data in uncompressed format to the length in compressed format.

The compression prediction tool IKQCPRED can compute the anticipated compression ratio for one or more data sets residing in the same VSE/VSAM catalog. Further details about the tool are provided in the manual *VSE/VSAM User's Guide and Application Programming*.

VTAM Data Compression Support

VTAM data compression enables you to reduce the amount of data being exchanged between LUs in multiple-domain environments (and APPN environments), thus improving response time for LU-LU sessions and reducing traffic over the lines. VTAM supports static and dynamic compression algorithms. With the CMPMIPS start option it is possible to control the amount of CPU cycles spent for compression and better balance CPU time and compression effectiveness.

Refer to the VTAM manuals *VTAM Network Implementation Guide* and *VTAM Resource Definition Reference*for further details.

Introducing the CSRCMPSC Programming Interface

Macro CSRCMPSC is available for writing applications making use of the z/VSE data compression support. The algorithm used is the Ziv-Lempel method where frequently occurring substrings are represented by index entries in a dictionary. This dictionary must be passed to z/VSE with the CSRCMPSC macro call.

Macro CSRCMPSC has two control parameters:

- CBLOCK

  Specifies a 36-byte input/output area which contains parameter information for compression (and expansion) such as compression dictionaries, source and target areas, and associated lengths.
- RETCODE

  Specifies a fullword output variable or register for storing the return code.

Refer to the manuals *z/VSE System Macros Reference* under “CSRCMPSC Macro” and *z/VSE System Macros User's Guide* under “Compressing and Expanding Data” for further details about this support.

Chapter 19. Security Support

This chapter discusses the security changes introduced with VSE/ESA 2.4. For details on how to implement security at your installation, refer to the Chapter "Protecting Resources" in the manual *z/VSE Administration*.

Through TCP/IP, z/VSE provides SSL (Secure Sockets Layer) support. SSL is a security protocol and allows Internet servers and clients to authenticate each other and to encrypt the data flowing between them. The SSL security support is described in detail in the *z/VSE e-business Connectors User's Guide*.

z/VSE Security Changes

The security changes introduced with VSE/ESA 2.4 were mainly due to the replacement of CICS/VSE by the *CICS Transaction Server for VSE/ESA*. The *CICS Transaction Server* does not support the integrated security functions that are available with CICS/VSE.

z/VSE offers two ways of implementing security at a customer's installation:
- Using the Basic Security Manager (BSM) included in z/VSE.
- Installing an External Security Manager (ESM).

The **BSM** provides **basic security support** and is part of z/VSE. It is ready for customization after initial installation. Note that apart from implementation changes and the support of transactions, this support is to a great extent identical with the access control support provided in releases prior to VSE/ESA 2.4.

In addition, z/VSE provides an **interface** for the use of an **ESM**. An ESM is usually a priced vendor product and must be installed separately. An ESM is for users who require more functionality and flexibility in their security implementation than provided by the BSM.

The major characteristics of an ESM are:
- It maintains its own user and resource data base for security checking.
- It is activated during startup with the ESM parameter in the IPL SYS command.

An ESM is described in its own set of manuals.

The following sections outline the basic security support as provided by the BSM. They mention the ESM if a function or support discussed relates to an ESM as well.

System Authorization Facility (SAF)

The centralized control of z/VSE security processing is done by the SAF. Vendors or customers that want to use the SAF directly find an introduction under Chapter 23, "System Authorization Facility (SAF) and its External Security Interface (RACROUTE)," on page 287.

Changes to the IPL SYS Command

The IPL SYS command has been extended. In addition to the existing security parameters

```
SEC=NO
SEC=YES
SEC=(YES,NOTAPE)
```

The following security parameters were introduced with VSE/ESA 2.4:

```
ESM=name
SERVPART=partition
SEC=RECOVER
```

The new parameters are introduced in this chapter. For a detailed description of the IPL SYS command, refer to the manual *z/VSE System Control Statements*.

Changes Implemented for the Basic Security Support

This section summarizes the major security changes that were implemented with VSE/ESA 2.4 for the basic security support, and points out the differences to former VSE/ESA releases.

- Security-related user profile information is no longer stored in DTSECTAB but in the VSE.CONTROL.FILE.

  Note: This file must be unique in the system. It is not compatible with a control file of an earlier system.
- Except for the predefined users FORSEC and DUMMY, security table DTSECTAB includes resource definitions only.
- Accesses to the VSE.CONTROL.FILE (for user information) are through the newly implemented Security Server. The Security Server runs per default in the FB partition.
- CICS transaction security is provided via a new access control table: DTSECTXN. It defines CICS transactions and their security class for which access is allowed.

  DTSECTXN is required since the *CICS Transaction Server* does not include the transaction security support (access control for transactions) as provided by CICS/VSE.
- The *Maintain User Profiles* dialog has been enhanced for defining additional security-related information in user profiles such as access control classes and access control rights. In former VSE/ESA releases, this information was included in DTSECTAB.
- VSE/POWER security has been enhanced as outlined under “Access Control for VSE/POWER Spool Entries” on page 242.

Activating the Security Support

Basic Security Manager (BSM):

Note that the BSM is always activated during startup (independent of the SEC setting in the IPL SYS command) in order to provide

- Sign-on security (signing on via the Interactive Interface or the CICS Transaction Server), and
- CICS transaction security (DTSECTXN).

Note, that for this support the DFHSIT must include SEC=YES (which is the default). If you reset it to SEC=NO, you will have no CICS security at all, not even sign-on security.

During initial installation you are asked whether you want to run your system with ″security on″. If you respond with YES, this will set in the IPL SYS command SEC=(YES,NOTAPE) which provides in addition

- Access control for resources defined in DTSECTAB.

External Security Manager (ESM):

After installing an **ESM**, you must define the name of the ESM initialization routine in the ESM parameter of the IPL SYS command. If the ESM you have installed requires a server partition, specify the partition ID in the SERVPART parameter of the IPL SYS command or use the default partition FB.

Consult in addition your ESM documentation for further implementation and customization steps that may be required on the z/VSE side.

z/VSE always checks for the ESM setting first. If the parameter is specified in the IPL SYS command, the ESM is activated, otherwise the BSM.

Implementation Hints

The following items should be considered prior to implementing security at your installation:

- Initial Installation

  If you did respond with NO during initial installation and want to activate security later, you can use the *Tailor IPL Procedure* dialog to change SEC=NO into SEC=YES or SEC=(YES,NOTAPE). Security activation requires additional z/VSE modifications to be done manually. To avoid this effort and if you definitively plan to implement security later, it is recommended to proceed as follows:

  – Respond with YES to the security question during initial installation. This prepares and initializes the system for security activation and avoids later customization work.

  – You can now use the IPL SYS command and specify SEC=NO if you want to run your system without DTSECTAB security for the time being. With SEC=YES you can at any time switch security on. No further customization effort is required except including your own entries in DTSECTAB.

- Security Server Partition

  Note that an ESM may also use a Security Server (as does the BSM). A Security Server runs in the partition specified in the SERVPART parameter of the IPL SYS command. Default is the FB partition.

  BSSINIT is the common security initialization routine for the BSM or an ESM. BSSINIT initializes and starts partition FB as BSM server partition unless another partition than FB has been specified in the IPL SYS command (SYS SERVPART=xx).

  To avoid any problems and additional customization work, it is recommended **not** to switch the predefined partition (FB) in which the Security Server runs after installation. If there is a need for a switch, you should plan it very carefully. You must select a static partition which is not controlled by VSE/POWER and modify the VSE/POWER startup procedure accordingly. The partition used must have the same priority as the default partition FB.

Refer also to "Security Server Commands."

- Parameter AUTH no longer available

  The parameter AUTH for identifying a user as security administrator is no longer available. Instead, when defining a user profile for a Type 1 user (system administrator, SYSA), this user has automatically ″AUTH authorization″. This means that such a user has access right ALT (Alter) and can access all protected resources defined in DTSECTAB.

  It is therefore essential from a security point of view, to consider carefully who and how many persons are to be defined as system administrator.

- Problem Solving in a Security Environment

  In case of problem recovery, for example during the implementation and test phase, there may be a need to run your system without the security support active. If such a situation occurs, you can specify SEC=RECOVER in the IPL SYS command. Note, however, that this switches off security completely (including sign-on and CICS transaction security) leaving your system unprotected.

- No RSL Security available with the *CICS Transaction Server*

  With RSL, which stands for *Resource Security Level*, you can control the access to CICS resources such as files, programs, transactions, temporary storage, terminals, and destinations.

  The *CICS Transaction Server* does not include RSL security as does CICS/VSE. RSL is also not supported by the BSM.

  If RSL security is required, the installation of an ESM should be considered.

Security Server Commands

The Security Server provides a set of commands to control its operation and display server status information. These commands can be entered from the system console through:

```
MSG xx,DATA=command
```

where **xx** indicates the selected server partition (default FB) and **command**, the command to be used. Refer to the manual *z/VSE Operation* for details.

Concept of Basic Security Support

This section provides an overview of the z/VSE basic security support. Details on how to implement this support is provided in the *z/VSE Administration* manual.

The basic security support provided by the BSM can be grouped as follows:

- Sign-on (log-on) security
- CICS transaction security (DTSECTXN)
- Access control for resources defined in DTSECTAB.

Figure 21 on page 237 shows, in a simplified form, the basic security support as provided by the BSM.

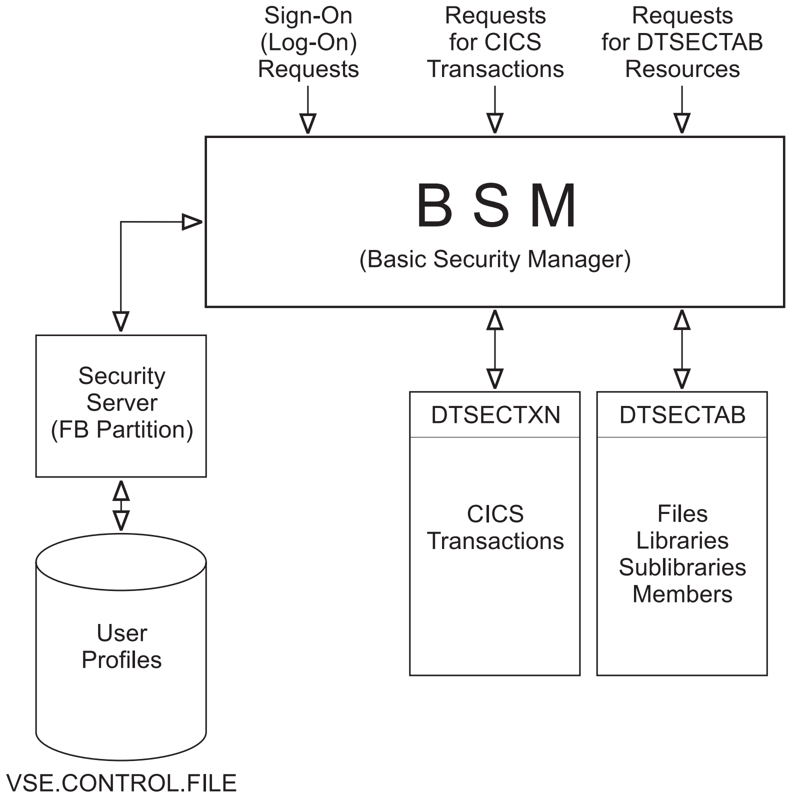


Figure 21. Logical Structure of Basic Security Support

Sign-On (Log-On) Security

This security check is related to user ID and password. It is also referred to as user identification and user authentication. This check is done when:

- Signing-on to z/VSE via the Interactive Interface (you must enter user ID and password).

  This check is always done independent of the security setting in the IPL SYS command, with the exception of IPL SYS=RECOVER in which case no password checking is done.
- Submitting a batch job.

  This checking requires that security for DTSECTAB is on: SEC=YES in the IPL SYS command.

  User ID and password are submitted with either the VSE/POWER JOB statement:

  ```
  * $$ JOB ... SEC=(userid,password)
  ```

  or the JCL ID statement:

  ```
  // ID USER=userid,PWD=password
  ```

The valid user ID and password are stored in the user's profile in the VSE.CONTROL.FILE. Security checking compares the stored user ID and password with the data entered or submitted.

Sign-on and log-on attempts to z/VSE by unauthorized users are logged on the system console or are saved by CICS.

CICS Transaction Security

CICS transaction security is part of the basic security support and protects CICS transactions from unauthorized access if they are specified in table DTSECTXN. This check is always active in case of the BSM and is independent of the SEC setting in the IPL SYS command.

Access attempts to CICS transactions by unauthorized users are logged on the system console.

Security Table DTSECTXN

DTSECTXN includes entries of those CICS transactions which are to be access-protected. Each transaction must be defined with a security class. 1 to 64 security classes are available. To access a transaction, the user's profile must include the corresponding security key. Figure 22 on page 241 shows the panel of the *Maintain User Profiles* dialog in which to define the security keys for a user.

If two or more *CICS Transaction Servers* run under z/VSE, a **CICS Region Prefix** can be used to identify a particular CICS system. This prefix is identical with the user ID (as shipped, the user ID is identical with the appl ID) of a particular CICS system:

```
DBDCCICS for the primary CICS Transaction Server in F2, and
PRODCICS for a second CICS Transaction Server installed in F8.
```

In order to use these prefixes, the SECPRFX=NO entry in the DFHSIT must be set to YES.

As shipped, z/VSE provides a predefined DTSECTXN including CICS-provided transactions that should be access-protected. Entries in DTSECTXN look as follows:

```
DTSECTXN NAME=IES1,TRANSEC=(61)
DTSECTXN NAME=IES6,TRANSEC=(1)
DTSECTXN NAME=IES8,TRANSEC=(61)
```

An entry with a CICS Region Prefix has the following layout:

```
DTSECTXN NAME=PRODCICS.IES1,TRANSEC=(61)
```

As shipped, the meaning of the security keys is as follows: 1 (general access), 61 (access to Interactive Interface resources). Refer also to Figure 22 on page 241.

DTSECTXN can be defined in two ways: through macro DTSECTXN or the *Define Transaction Security* dialog. The manual *z/VSE Administration* provides details on both the macro and the dialog.

Note: A transaction that is not defined in DTSECTXN cannot be accessed. Any transaction is assumed to be access-protected and must thus be defined in DTSECTXN.

It is also possible to use generic names for CICS transactions in DTSECTXN. Refer to the *z/VSE Administration* for details.

Migrating TRANSEC Values from CICS/VSE

In CICS/VSE, a transaction is protected against unauthorized access with the TRANSEC parameter in the Program Control Table (PCT). z/VSE provides migration support to convert TRANSEC values from CICS/VSE into entries in the transaction security table DTSECTXN of the CICS Transaction Server.

z/VSE provides three REXX/VSE procedures for migration. These are SKSECTXS which uses as input existing DFHPCT assembler source members, SKSECTX2 which uses as input the VSE/VSAM file created by the CICS/VSE Security Migration Aid (refer also to "Security Migration Aid" on page 210), and SKSECTX3 which uses as input the definition statements used by the CICS/VSE DFHCSDUP utility.

The manual *z/VSE Administration* describes in detail the migration support provided and how to use it.

BSM Post-Processing Installation Exit BSTXX01

If using BSM, you might need to extend the security checking of the BSM using your own logic. Therefore, you might wish to change the:

- Return code triple of the RACROUTE request.
- Contents of result areas as provided by EXTRACT requests.

The BSM provides the installation exit BSTXX01 to allow modifications immediately before returning to SAF. To use it, you have to build a phase with the name BSTXX01 and catalog it in IJSYSRS.SYSLIB.

For related information, see also Chapter 23, "System Authorization Facility (SAF) and its External Security Interface (RACROUTE)," on page 287.

These are the contents of the BSTXX01 interface:

```
AMODE 31
RMODE ANY

Input:
  R1 : address of area that contains:
        +0  address of SAFP
        +4  R1 contents to be returned to SAF
            (e.g. result area address)
        +8  R15 contents to be returned to SAF

Output:
  The customer created phase BSTXX01 is responsible for
  restoring the registers.

  Register
    0,2-13  as on entry
         1  value from input area or explicit return value
            from customer BSTXX01 code
        15  if the customer wants to change the return code
            then the new return code, else the value from
            input area
```

Access Control for Resources Defined in DTSECTAB

This support is also known as **z/VSE Access Control Function**, a term used in manuals describing the security support available prior to VSE/ESA 2.4.

You can run a batch job or program in a z/VSE partition or a VSE/ICCF interactive partition. If such a job or program accesses resources, these resources can be protected by defining them in DTSECTAB. The origin of the user ID and password is either the VSE/POWER JOB statement or the JCL ID statement. Refer to "Sign-On (Log-On) Security" on page 237 for further details.

The following resources can be defined in DTSECTAB:

- Files
- VSE Libraries
- VSE Sublibraries
- VSE Members

Access control for such resources is active if the security setting in the IPL SYS command is SEC=YES and the resource to be accessed is defined in DTSECTAB.

Access Control Table DTSECTAB

Note: The access control concept for resources defined in DTSECTAB has not changed.

User entries, however, are no longer included in DTSECTAB. All security-related user information is part of the user's profile stored in the VSE.CONTROL.FILE. An exception are the predefined users FORSEC and DUMMY. They are required by z/VSE for startup processing and are the only user definitions included in DTSECTAB.

Following is an example of a typical DTSECTAB resource entry - in this case for a library:

```
DTSECTAB TYPE=LIB,
         NAME=PRIVATE.TEST.LIB1
         ACC=(1-8),
         UACC=CON,
         LOG=(1-8)
```

All users can access library PRIVATE.TEST.LIB1 with access right CON (Connect). In addition, any user who has specified one or more of access control classes 1-8 can access the library with the access right (Read, Update, Alter) assigned to a class.

As shipped, z/VSE provides a predefined DTSECTAB which includes definitions of system-provided resources that need to be protected. The DTSECTAB source is available as skeleton DTSECTRC in VSE/ICCF library 59. Use this skeleton to modify DTSECTAB as required for your installation.

Notes:
1. You define entries in DTSECTAB with the DTSECTAB macro which is described in detail in the *z/VSE Administration* manual.
2. To modify DTSECTAB, copy skeleton DTSECTRC from VSE/ICCF library 59 to your private VSE/ICCF library. Use always this copy if there is a need to change DTSECTAB. You create a new (updated) DTSECTAB by submitting skeleton DTSECTRC from your private VSE/ICCF library.

User Profile Definitions

User profiles are stored in the VSE.CONTROL.FILE.

You can use the *Maintain User Profiles* dialog or the batch utility IESUPDCF to create or maintain user profiles.

Dialog: Maintain User Profiles

The major security parameters you can define with this dialog are:
- User ID
- Password

- Transaction security keys (1 to 64)

  Allows access to a CICS/VSE transaction defined in DTSECTXN if the security class of the transaction matches with one of the security keys defined for the user. Figure 22 shows the panel of the *Maintain User Profiles* dialog for specifying transaction security keys.
- Access control classes (1-32) for the resources that can be defined in DTSECTAB:

  Files

  Libraries

  Sublibraries

  Members

  Access rights for the 32 access control classes:

  1 = Connect

  2 = Read

  3 = Update

  4 = Alter

  Note that a blank (no specification) means **no access**.

  Allows access to batch resources defined in DTSECTAB (files, libraries, sublibraries, and members) on the level assigned as access right for a particular access control class in the user's profile. Figure 22 shows the panel of the *Maintain User Profiles* dialog for specifying these access rights.

Other security-related specifications for a user on this panel are READ

```
IESADMUPR1          ADD OR CHANGE RESOURCE ACCESS RIGHTS
Base     II       CICS   ResClass  ICCF

     Place an 'X' next to the transaction security keys for user YYYY
   01 X   02 _   03 _   04 _   05 _   06 _   07 _   08 _   09 _   10 _   11 _
   12 X   13 _   14 _   15 _   16 _   17 _   18 _   19 _   20 _   21 _   22 _
   23 _   24 X   25 _   26 _   27 _   28 _   29 _   30 _   31 _   32 _   33 _
   34 _   35 _   36 _   37 _   38 _   39 _   40 _   41 _   42 _   43 _   44 _
   45 _   46 _   47 _   48 _   49 _   50 _   51 _   52 _   53 _   54 _   55 _
   56 _   57 _   58 _   59 _   60 _   61 X   62 _   63 _   64 _
     Specify the access rights for 1-32 DTSECTAB access control classes
     ( _=No access, 1=Connect, 2=Read, 3=Update, 4=Alter )
   01 2   02 2   03 2   04 2   05 2   06 2   07 2   08 2   09 _   10 _   11 _
   12 _   13 _   14 _   15 _   16 _   17 _   18 _   19 _   20 _   21 _   22 _
   23 _   24 _   25 3   26 3   27 3   28 3   29 3   30 3   31 3   32 3

   READ DIRECTORY..... 1   User can read directory with Connect (1=yes, 2=no)
   B-TRANSIENTS....... 2   User can manipulate B-Transients     (1=yes, 2=no)

PF1=HELP                3=END                    5=UPDATE
PF7=BACKWARD   8=FORWARD
```

Figure 22. Panel for Defining CICS Security Keys and Batch Access Rights

DIRECTORY and B-TRANSIENTS.

The panel USER AUTHORIZATION (of the *Maintain User Profiles* dialog) allows further specifications that are related to security. They allow the system administrator to limit or expand the authorization a user has when working with the system and its resources. For the specifications possible, refer to the manual *z/VSE Administration* which describes the dialog in detail.

Access Control for VSE/POWER Spool Entries

VSE/POWER security has been improved by providing spool access protection for entries in the RDR, LST, PUN, and XMT queues. The support requires the specification of SEC=YES in the IPL SYS command and SECAC=SYS in the SET statement of the VSE/POWER startup procedure. Spool access protection can be further controlled at the entry level with the SECAC operand in the JOB, LST, and PUN statements. The support is available with the BSM as well as with an ESM.

Refer to the manual *VSE/POWER Administration and Operation* for further details.

Support for Logging and Reporting

The z/VSE optional program VSE/Access Control-Logging and Reporting (VSE/ACLR) is a flexible logging and reporting tool for the basic security support. It can log authorized and non-authorized accesses to the system and the resources defined in DTSECTAB but not accesses to CICS transactions which are defined in DTSECTXN. Access violations concerning CICS transactions are shown on the system console. The manual *z/VSE Administration* provides further details about the use of the VSE/ACLR program.

An ESM provides its own logging and reporting facilities.

Using Hardware Cryptographic Support

Hardware crypto support in z/VSE makes use of the following two hardware functions:

- Support of PCI Cryptographic Accelerator (PCICA) and Crypto Express2 (CEX2) cards. PCICA cards are available for all IBM zSeries processors, Crypto Express2 cards are available for zSeries processors z890, z990 and higher.
- Support of the CP Assist for Cryptographic Function (CPACF), which is part of the hardware of a z990 or higher.

PCICA and Crypto Express2 cards provide encryption-assist support, and can help to increase the throughput in a TCP/IP network using SSL (Secure Sockets Layer). CPACF provides hardware support for symmetric cryptographic algorithms, like DES, Triple-DES, and SHA-1.

Secure Sockets Layer (SSL) has become the dominant technique for enterprises to communicate securely with their customers via Internet browsers. SSL uses cryptography both for authentication of clients and servers, and for data confidentiality. SSL is a public key cryptography-based extension to TCP/IP networking.

z/VSE supports the IBM Crypto Express2 and PCI Cryptographic Accelerator (PCICA) cards which provide encryption assist support for increased Secure Sockets Layer (SSL) throughput. The support is based on functions provided by *TCP/IP for VSE/ESA 1.5*.

SSL support of *TCP/IP for VSE/ESA* was first introduced with VSE/ESA 2.6 and supported software encryption only. From z/VSE 3.1 onwards, SSL transparently uses Crypto Express2 and PCICA cards if available. There is no need to change any applications already using SSL. For example, existing applications that use SSL such as CICS Web Support (CWS) and the VSE e-business connectors *automatically* benefit from this transparent use of Crypto Express 2 and PCICA cards.

If the Hardware Crypto support is not available, *TCP/IP for VSE/ESA 1.5* provides software encryption.

Support Characteristics

The PCICA card supports the RSA encryption/decryption algorithm. It supports **RSA1024** and **RSA2048**. A key length of 512 is not directly supported by the hardware but is simulated by 1024 requests.

The PCICA card is plugged into an Adjunct Processor (AP) which has to be seen as an extension to the CPU. Once plugged into the system, the PCICA card is identified as follows:

- An eight-character serial number.
- A two-digit Adjunct Processor (AP) number.
- A CHPID number.

  The CHPID number is not known and is not relevant to VSE/ESA, but is used for the hardware management of the PCICA card. A CHPID number assigned to a PCICA card is no longer available for normal CHPID use and cannot be used in an IOCDS definition. If there are CHPID definitions which conflict with the IOCDS, that is, if a CHPID is defined in the IOCDS with a number that matches the CHPID number assigned to the PCICA card because of plugging rules, a system power-on-reset will cause both, the already IOCDS-defined CHPID and the PCICA, not to come online.

Since the PCICA card is to be seen as an extension to the CPU, rather than as a new channel attached device, the PCICA card requires no configuring and thus
- no device type
- no ADD statement
- no IOCDS definition.

During system initialization (IPL), z/VSE senses the hardware and recognizes the Crypto support (PCICA card) if installed. The following messages are issued on the console by the Security Server partition (usually FB) which activates the Hardware Crypto support via startup job SECSERV:

Crypto support available:

```
FB 0011 // JOB SECSERV
        DATE 12/12/2004, CLOCK 13/16/38
FB 0011 ID (PARAMETERS SUPPRESSED)
FB 0094 1J023I FOUND A CRYPTO EXPRESS2 CARD AT DEVICE INDEX 20
FB 0094 1J005I HARDWARE CRYPTO ENVIRONMENT INITIALIZED SUCCESSFULLY.
FB 0094 1J006I USING CRYPTO DOMAIN 11
FB 0094 1J022I CPU CRYPTOGRAPHIC ASSIST FEATURE AVAILABLE.
```

Crypto support not available:

```
FB 0011 // JOB SECSERV
        DATE 08/09/2002, CLOCK 13/16/38
FB 0011 ID (PARAMETERS SUPPRESSED)
FB 0095 1J017I CRYPTO HARDWARE NOT INSTALLED OR NOT DEFINED.
```

Using the Crypto Support with a z/VSE Guest under z/VM

The Hardware Crypto support requires **z/VM 4.2** or higher.

When z/VSE runs as a guest under z/VM, the Crypto support must be defined to the system in the *VM Directory Entry* with the following statement:

```
CRYPTO APVIRT
```

z/VM provides special commands for the Crypto support as shown below:

1. The installed Crypto hardware can be queried with the following CP command:

   Q CRYPTO

   The command provides output similar to the one shown below:

   ```
   q crypto
    00: Processor 00 Crypto Unit 0 usable
    00: Processor 01 Crypto Unit 1 usable
    00: There is no user enabled for PKSC Modify
    00: All users with directory authorization are enabled for key entry
    00: Crypto Adjunct Processor is installed
   ```

2. With the following CP command you can check the currently assigned Crypto domain and device number of your z/VSE guest:

   Q VIRTUAL CRYPTO

   The command provides output similar to the one shown below:

   ```
   q virtual crypto
    00: No CAM or DAC Crypto Facilities defined
    00: AP 0E Queue 08 shared
   ```

   In the above example, Crypto domain 08 is used and device 0E is available for this particular z/VSE guest.

3. Using the following CP command, you can view the available AP queues in z/VM. If there are multiple AP queues available, z/VM will automatically balance the workloads, and displays *one* AP queue only to the z/VM guest. In the example below, there are two Crypto Express2 cards in use, where each card has two queues. However, z/VSE will access one "virtual" queue only.

   ```
   * cp q crypto ap
   AR 0015 AP 03 CEX2C Queue 15 is installed
   AR 0015 AP 04 CEX2C Queue 15 is installed
   AR 0015 AP 05 CEX2C Queue 15 is installed
   AR 0015 AP 06 CEX2C Queue 15 is installed
   AR 0015 1I40I  READY
   ```

4. A domain can be dedicated to one particular guest. For example:

   CRYPTO DOMAIN 5

5. With the appropriate authority, the settings can be queried and updated in CMS:

   DIRM CRYPTO

z/VM assigns the AP (Adjunct Processor) queue numbers randomly, so it is normal for the guest to see a different queue number each time the guest is started. This is independent of the domain number. CP will not provide hardware Crypto support for third-level guests (VM2 as a second-level guest of VM1, with z/VSE as a guest on VM2). CP will not provide V=R guest survival support for the Crypto support.

Consult the corresponding z/VM manuals for further details about the z/VM Crypto support for guest systems.

Displaying Hardware Crypto Status Information Under z/VSE

If you use the Basic Security Manager (BSM), you can display the status of your hardware crypto support on your z/VSE console. To do so, you use the following console command:

```
msg fb,data=status=cr
```

The output from this command looks like following:

```
AR 0015 1I40I  READY
FB 0011 BST223I CURRENT STATUS OF THE SECURITY TRANSACTION SERVER:
FB 0011 ADJUNCT PROCESSOR CRYPTO SUBTASK STATUS:
FB 0011   AP CRYPTO SUBTASK STARTED .......... : YES
FB 0011   MAX REQUEST QUEUE SIZE ............. : 0
FB 0011   MAX PENDING QUEUE SIZE ............. : 0
FB 0011   TOTAL NO. OF AP REQUESTS ........... : 0
FB 0011   NO. OF POSTED CALLERS .............. : 0
FB 0011   AP CRYPTO WAIT TIME ................ : 7
FB 0011   AP CRYPTO TRACE LEVEL .............. : 3
FB 0011   NO. OF AVAIL. APQS: PCICC / PCICA .. : 0 / 0
FB 0011   NO. OF AVAIL. APQS: CEX2 ........... : 1
FB 0011     AP 20 : CEX2
FB 0011       STATUS : DEV_ONLINE
FB 0011       QDEPTH : 8
FB 0011   AP CRYPTO DOMAIN ................... : 14
FB 0011 CPU CRYPTOGRAPHIC ASSIST FEATURE:
FB 0011   CPACF AVAILABLE .................... : YES
```

The above output shows the availability of:
- One Crypto Express2 card in Crypto domain 14.
- The CPACF feature.

In addition to the status of crypto cards, the status output shows the availability of the CPU Cryptographic Assist feature (CPACF) which provides symmetric crypto functions (such as DES, triple-DES, and SHA-1 hashing).

CPACF is not available on all zSeries processors. Where it is available, it is used *transparently* in SSL sessions.

If you are not using BSM, for guidance you should refer to the section "Using the Crypto Support and an External Security Manager."

Using the Crypto Support and an External Security Manager

If you use an External Security Manager (and not the Basic Security Manager) the following implementation details of the Hardware Crypto support are important and must be observed.

The Hardware Crypto support is activated by the startup job SECSERV (Security Server) which is part of the Basic Security Manager and which runs in partition FB by default. If SECSERV is not started (because you are using an External Security Manager), the Hardware Crypto support is **not** available. However, the Hardware Crypto task can be started manually in any partition with a job stream such as the following:

```
* $$ JOB JNM=HWCRYPTO,DISP=D,CLASS=R
// JOB HWCRYPTO
// EXEC IJBCRYPT
/*
/&
* $$ EOJ
```

To activate the Hardware Crypto support, proceed as follows:
1. Start the above job stream (or a similar one).
2. Shutdown TCP/IP and your TCP/IP applications (TCP/IP runs in partition F7 by default).
3. Restart TCP/IP and your TCP/IP applications.

Chapter 20. Application Development Support

z/VSE provides a number of programming functions to help you develop online and batch applications. These include functions for creating and editing source code, translating, compiling, and testing it.

The functions listed below, described in other chapters of this manual or documented in other z/VSE manuals, reflect the broad range of the z/VSE application development support:

- Chapter 12, "Using Data Spaces and Virtual Disks," on page 167 for details on data spaces and virtual disk support.
- *z/VSE Guide to System Functions* under "Linking Programs" for Linkage Editor and under "The Librarian Program" for Librarian details.
- Manual *z/VSE Administration* for details about related dialogs such as the the *Create Application Job Streams* dialog.
- Manuals *z/VSE System Macros User's Guide* and *z/VSE System Macros Reference* for the system support provided through assembler macros.
- z/VSE programming and workstation support described in the manual:

  VSE/ESA Programming and Workstation Guide

Programming Language Support

For programming you can use any programming language supported by z/VSE. The High Level Assembler for VSE, REXX/VSE, and LE/VSE (IBM Language Environment for VSE/ESA) are part of the z/VSE base.

For introductory information on REXX/VSE, refer to "Support of REXX/VSE" on page 250. For introductory information on LE/VSE, refer to "Considerations for LE/VSE and the VSE C Language Run-Time Support" on page 78.

The following programming languages are available as z/VSE optional programs:

COBOL for VSE/ESA
C for VSE/ESA
PL/I for VSE/ESA
DOS/VS RPG II

Notes:

1. DOS/VS RPG II supports CICS/VSE online applications only.
2. VS FORTRAN is also supported but it is not a z/VSE optional program.
3. COBOL for VSE/ESA, C for VSE/ESA, and PL/I for VSE/ESA provide together with LE/VSE an advanced COBOL, C, and PL/I programming environment.

   For all three languages the following features are available:

   - Alternate Function feature comprising the compiler only.
   - Full Function feature comprising the compiler and interactive debugger. Note that the debugger covers all three languages and is needed only once in the system.

- Enterprise Workstation feature which provides the workstation development environment IBM VisualAge for COBOL/PL/I Professional Edition (no offering available for C for VSE/ESA).

   LE/VSE and its programming languages C, COBOL, and PL/I do not support VSE/ICCF interactive partitions. It is generally recommended to use dynamic partitions for programs written in these languages.
4. For the High Level Assembler for VSE the optional feature High Level Assembler Toolkit for VSE is available. It provides:
   - An interactive debug facility.
   - A disassembler converting object code into source statements.
   - A set of structured programming macros to simplify coding and better understand complex control flows.
5. The new z/VSE optional program CICS Command Level Conversion Aid for VSE (CCCA/VSE) helps you migrate your online and batch DOS/VS COBOL applications into COBOL for VSE/ESA applications.

Integrating Online Applications into the Interactive Interface

Be aware that you can integrate an online application into z/VSE's Interactive Interface. For details about the required tailoring of the Interactive Interface, refer to Chapter 15, "Tailoring the Interactive Interface," on page 189, and also to "Adding a User-Written Application to CICS" on page 255.

Tailoring Compile Skeletons

z/VSE provides compile skeletons for the following programming languages:

High Level Assembler for VSE
COBOL for VSE/ESA
C for VSE/ESA
PL/I for VSE/ESA
DOS/VS RPG II
VS FORTRAN

In addition, z/VSE provides skeletons for the relational database management system SQL/DS and DB2. Note that from z/VSE 3.1 onwards, the *DB2 Server for VSE* is provided as a base program.

These skeletons must first be tailored to reflect your programming environment. After tailoring, you can use them to compile online programs, BMS (Basic Mapping Support) map definitions, batch programs, and batch subroutines. z/VSE uses the tailored job streams if you select the **COMPILE** option of the *Program Development Library* dialog to submit a VSE/ICCF library member for assembly or compilation.

The system administrator should plan who will be allowed to use the compile skeletons stored in VSE/ICCF library 2. The administrator can do one of two things:

1. Give some or all application programmers the skeletons and have them tailor them for their own needs. In this case, the administrator would copy skeletons from VSE/ICCF library 2 into a VSE/ICCF library which programmers can access.

2. Leave the skeletons in VSE/ICCF library 2 and tailor them for the entire system. This allows the administrator to set standards for compile jobs and have every programmer use them.

When the COMPILE option is selected, the system searches for the correct compile skeleton in the following order:

1. User's primary library.
2. User's secondary library.
3. VSE/ICCF library 2.

Table 38 lists the name and function of each compile skeleton provided by z/VSE. The naming convention is as follows:

C$$xxyyy and **C$Qxxyyy**

where Q identifies the skeletons provided for use with DB2 Server for VSE & VM.

xx can be:

CN (for C for VSE/ESA)
CV (for COBOL for VSE/ESA)
PV (for PL/I for VSE/ESA)
AS (for High Level Assembler for VSE)
RP (for RPG II -- CICS/VSE 2.3 only)
FO (for VS FORTRAN)

yyy can be:

ONL for online program
BAT for batch program
SUB for batch subroutine
MAP for BMS map definition

Starting with VSE/ESA 2.6, HTML templates can be created from BMS maps. Maps of type HTML are stored in sublibary PRD2.DFHDOC.

Under "Tailoring Compile Skeletons", the manual *z/VSE Administration* has examples of compile skeletons and describes the modifications that are required.

Table 38. Compile Skeletons in VSE/ICCF Library 2

Member Name	Function
C$$ASONL	High Level Assembler for VSE online program.
C$$ASBAT	High Level Assembler for VSE batch program.
C$$ASSUB	High Level Assembler for VSE batch subroutine.
C$$ASMAP	High Level Assembler for VSE BMS map definition.
C$$CNONL	C for VSE/ESA online program.
C$$CNBAT	C for VSE/ESA batch program.
C$$CNSUB	C for VSE/ESA batch subroutine.
C$$CNMAP	C for VSE/ESA BMS map definition
C$$COMAP	COBOL for VSE/ESA BMS map definition.
C$$CVONL	COBOL for VSE/ESA online program.
C$$CVBAT	COBOL for VSE/ESA batch program.

Table 38. Compile Skeletons in VSE/ICCF Library 2 (continued)

Member Name	Function
C$$CVSUB	COBOL for VSE/ESA batch subroutine.
C$$FOBAT	VS FORTRAN batch program.
C$$FOSUB	VS FORTRAN batch subroutine.
C$$PLMAP	PL/I for VSE/ESA BMS map definition.
C$$PVONL	PL/I for VSE/ESA online program.
C$$PVBAT	PL/I for VSE/ESA batch program.
C$$PVSUB	PL/I for VSE/ESA batch subroutine.
C$$RPONL	DOS/VS RPG II online program (CICS/VSE only).
C$$RPBAT	DOS/VS RPG II batch program.
C$$RPMAP	DOS/VS RPG II BMS map definition.
C$$RPSUB	DOS/VS RPG II batch subroutine.
C$QASONL	High Level Assembler for VSE online program (for DB2 applications).
C$QASBAT	High Level Assembler for VSE batch program (for DB2 applications).
C$QASSUB	High Level Assembler for VSE batch subroutine (for DB2 applications).
C$QCNONL	C for VSE/ESA online program (for DB2 applications).
C$QCNBAT	C for VSE/ESA batch program (for DB2 applications).
C$QCNSUB	C for VSE/ESA batch subroutine (for DB2 applications).
C$QCVONL	COBOL for VSE/ESA online program (for DB2 applications).
C$QCVBAT	COBOL for VSE/ESA batch program (for DB2 applications).
C$QCVSUB	COBOL for VSE/ESA batch subroutine (for DB2 applications).
C$QFOBAT	VS FORTRAN batch program (for DB2 applications).
C$QFOSUB	VS FORTRAN batch subroutine (for DB2 applications).
C$QPVONL	PL/I for VSE/ESA online program (for DB2 applications).
C$QPVBAT	PL/I for VSE/ESA batch program (for DB2 applications).
C$QPVSUB	PL/I for VSE/ESA batch subroutine (for DB2 applications).

Support of REXX/VSE

Introduction

REXX/VSE is a partial implementation for z/VSE of the Level 2 SAA (Systems Application Architecture) REXX language.

REXX is a very versatile programming language. A REXX program consists of REXX language instructions that the REXX Interpreter translates directly. A program can also contain commands that the host environment executes.

One advantage of the REXX language in general is its similarity to ordinary English. This similarity makes it easy to read and write a REXX program. REXX has simple syntax rules and you can use uppercase, lowercase, or mixed case. There are no restrictions about columns in which you can type.

Note: Before a REXX program can be executed, you must store it into a VSE sublibrary with member type PROC (procedure). REXX programs can run in any partition of z/VSE.

Initialization

During startup, REXX/VSE phases are loaded into the SVA with the statements:

```
SET SDL
LIST=$SVAREXX
```

These statements are part of the BG startup procedure $0JCL. This procedure also initializes REXX/VSE by activating the startup program ARXLINK.

Characteristics of REXX/VSE

REXX/VSE provides:

- A selected subset of REXX functions of REXX/MVS.
- REXX/MVS compatible programming and customization services.
- z/VSE unique external functions and host command environments.

REXX/VSE offers both the

- REXX Interpreter that provides interpretation of REXX programs, and the
- REXX/370 Library that provides execution of compiled REXX programs.

Note: z/VSE does not support the compilation of REXX programs. This must be done on either a VM or OS/390 system where the compiler for SAA REXX/370, Program Number 5695-013, is required.

Benefits of the REXX Language

Ease of Use

REXX is a general purpose programming language that has the usual structured-programming instructions such as IF, SELECT, DO WHILE, and LEAVE. This allows programs and algorithms to be written in a clear and structured way.

REXX is easy to use by both computer professionals and by ″casual″ general users. It is a language that provides powerful character and arithmetical abilities in a simple framework. The values that REXX manipulates are in the form of character strings. There are no data types to be declared. In order to manage the data as naturally as you would manipulate words on a page or with a text editor, REXX has a rich set of character manipulation functions and built-in parsing with many options.

New System Management Capabilities for z/VSE

REXX offers capabilities as a language for general use, as well as for z/VSE system command programming. REXX/VSE can be used to tailor the z/VSE operating system by using the REXX language instead of the z/VSE conditional job control language. You can use REXX/VSE in the z/VSE batch environment for:

- z/VSE operation automation
- Substitution and parameterization for job execution
- Direct communication to the z/VSE system console
- Input/output (I/O) operations to z/VSE libraries and sequential data sets
- Dynamic creation and execution of z/VSE job streams
- VSE/POWER job submission and controlling

- VSE/POWER queue element manipulation
- VSE/POWER command execution
- Automated console operation.

REXX/VSE is particularly suitable for:

- Command procedures
- Rapid batch program development
- Prototyping
- Personal utilities.

Performance Improvements Due to Including the REXX/370 Library

A REXX program may be interpreted line-for-line, which provides the advantage of easy debugging. When a REXX program has been completed, it could be compiled on a VM or OS/390 system. The REXX/370 Library provides significant benefits when executing compiled programs on z/VSE. The benefits include improved performance, reduced system load, and portability of compiled programs.

REXX/VSE Application Enabling Platform

Accessing z/VSE Resources

REXX/VSE provides the EXECIO command which is compatible with the EXECIO command of REXX/MVS. You may use EXECIO to control input/output (I/O) operations to and from:

- VSE library members
- Sequential files
- SYSIPT (input only)
- SYSLST (output only).

EXECIO can read and write data on the program stack or in REXX variables (for example into a stem) directly. You can use EXECIO for I/O tasks such as copying information to and from a data set, to add, delete or update information. A REXX program can read information from a data set to the data stack for serialized processing or to a list of REXX variables for random processing. A program can write information from the data stack or from a list of REXX variables.

With VSE/ESA 2.3, two new operands are available for the EXECIO command: BYTES and STRTBYTE. These operands allow to process members of type ″string″ not only as a single record but to break them up into smaller units if required. For a detailed description of this support, refer to the manual *VSE/ESA Enhancements*.

z/VSE provides the VSAMIO command to access VSAM data from REXX/VSE. Refer to the *REXX/VSE Reference* manual for further details.

VSE/POWER Interface

You may switch to the VSE/POWER host command environment via ADDRESS POWER to retrieve and store VSE/POWER queue entries. You may also send VSE/POWER commands and retrieve the corresponding output via the OUTTRAP function.

The retrieval and storing of data in the LST and PUN queues of VSE/POWER includes options to:

- Handle print control characters for LST queue entries.
- Retrieve all parts of a segmented output entry.

- Retrieve only a certain part of an entry.
- Append records to an existing output queue entry.
- Specify disposition, priority, user information, the number of copies, and user-defined output operands for entries stored in the LST or PUN queue.

Creating and Executing z/VSE Job Streams Dynamically

By using the character string manipulation functions of REXX, you can dynamically create z/VSE job streams and store them into a VSE sublibrary, a sequential file, or into the program stack. In the latter case, when the REXX program exits and returns to job control, job control executes all JCL statements that have been left in the program stack.

Submitting and Controlling VSE/POWER Jobs

By using the VSE/POWER host command PUTQE, you may submit a VSE/POWER job to another partition and optionally wait a specified number of seconds until the job has been started and completed. The OUTTRAP function provides you with a job completion message and maximum or last return code and other useful information about the submitted job. The VSE/POWER job to be submitted may reside in a VSE sublibrary or in a list of REXX variables.

Issuing JCL Commands

A JCL command issued by a REXX batch program is immediately executed and a return code is provided. Input/output data can be exchanged via REXX ″compound variables″, called stems. Thus, REXX can loop on JCL commands and make very efficient use of the JCL conditional job control support.

Calling Batch Programs

The following functions are available:

- The called batch program can **read** input data from REXX. Within REXX the data is supplied using ″compound variables″, called stems. This is transparent to the batch program and can be used to replace previous data input via JCL.
- A batch program can also **write** output data to REXX stems using a service routine. This increases the possibilities for subsequent processing of output data created by user programs.
- Parameters can be passed in the same way as previously done via JCL. In addition, REXX provides a standard parameter list for COBOL, PL/I, and C/370. With it, multiple parameters can be passed and modified by the called program.

Issuing z/VSE Librarian and VSE/VSAM IDCAMS Commands

Such commands are executed immediately and return codes are provided. Input/output data can be exchanged using REXX stems.

Communicating with User Consoles

For messages and replies the REXX instructions SAY and PULL are available allowing interactive communication in line mode.

REXX/VSE Console Automation

REXX/VSE console automation is based on the console support of z/VSE. The support helps to automate console operation and provides an easy-to-use z/VSE console command environment that allows to activate or deactivate one or more z/VSE console sessions. z/VSE console commands can be imbedded into a REXX program to issue z/VSE console commands and retrieve command responses. The following types of commands are supported:

- Attention routine (AR) commands.
- Console redisplay commands.

- VSE/POWER, VSE/ICCF, CICS/VSE, VTAM, and SQL/DS commands.

A *Console Application Framework* is included to demonstrate how to exploit REXX/VSE console automation in the most beneficial way.

A REXX CPU monitor checks for critical performance values in z/VSE partitions and issues console messages if user-defined limits have been exceeded.

Performing Data Stack Services

REXX/VSE provides a data stack and related commands compatible to REXX/MVS. You can use these commands to create or delete data stacks or query information about the stack, for example, the number of elements in the stack.

REXX/VSE Socket API

REXX/VSE provides the function SOCKET to implement the SOCKET API for communication between network applications. With VSE/ESA 2.7, the SOCKET function has been enhanced to write SSL-enabled socket applications in REXX/VSE.

Extending the Programming Capabilities of REXX/VSE

You can write your own external functions and subroutines to extend the programming capabilities of REXX/VSE. You can write such functions or subroutines in REXX or in any programming language that supports the system-dependent interface of REXX/VSE. You can also group frequently used external functions and subroutines into a function package. This allows quicker access but they must all be link-edited into a phase. Interfaces to external functions, subroutines, and functions packages are compatible with REXX/MVS.

Running a REXX Program

A REXX program is a **program written in the REXX language**. A REXX program runs in the dynamic or static partition of the invoking job stream. REXX/VSE itself is mostly SVA (shared virtual area) resident and is shared by all z/VSE partitions.

A REXX program may reside in any z/VSE sublibrary. The name of that library member is the program name and the member type must be PROC (procedure); for example, MYPROG.PROC. The sublibrary in which the program is stored must be included in the current LIBDEF chain.

Job Control Extension to Invoke a REXX Program

A new operand, named **REXX**, has been introduced for the job control EXEC statement/command to invoke REXX programs. For example:

```
// EXEC REXX=prog_name,SIZE=size_operand,PARM='parm_list'
```

For a detailed syntax and operand description, refer to the EXEC statement/command in the manual *z/VSE System Control Statements* under “EXEC”.

Usually, a VSE/POWER job must be submitted to invoke a REXX program. However, you can also invoke a REXX program by using the programming services of REXX/VSE. These services let application programs written in assembler or high-level languages interface to REXX/VSE.

Leaving Data on the Stack

If you call a REXX program (through an EXEC statement), you can leave z/VSE JCL statements on the stack and z/VSE can then process these statements. This means you can insert JCL statements or data into the current job stream.

Compatibility of REXX/VSE

Compatibility exists in the following areas:

- The REXX language is consistent across the SAA environments.
- REXX/VSE provides a subset of functions compatible with REXX/MVS.
- EXECIO and stack services are compatible to REXX/MVS.
- The function provided by the REXX/370 Library is identical on z/VSE, OS/390, and z/VM.

Further Information

The following manuals describe REXX/VSE and its functions in detail:

- *REXX/VSE Reference*
- *REXX/VSE User's Guide*
- *IBM Compiler and Library for SAA REXX/370, User's Guide and Reference*.

The REXX/VSE homepage can be accessed via:

`http://www-1.ibm.com/servers/eserver/zseries/zvse/support/rexx/rexxhome.html`

Note on VSE/POWER Application Programs

Application programs which communicate with VSE/POWER via the SPOOL-macro support must run in the same private address space as VSE/POWER or in a partition in the shared area. The macros XECBTAB, XPOST, and XWAIT belong to the SPOOL-macro support.

However, it is recommended to use the XPCC macro support instead because application programs using this support can run in any private address space.

Adding a User-Written Application to CICS

This section summarizes the tasks required to add a user-written, online application (transaction) to your z/VSE system. Note that both the system administrator and the programmer each perform several of the tasks. *To ensure system integrity, a programmer should not be allowed to do all of them*.

In the following list:

- The tasks which the programmer performs are based on the default programmer profile (PROG) that is shipped with z/VSE.
- It is assumed that the source code for the application resides in a VSE/ICCF library.

Adding a user-written **online application** to CICS normally involves:

1. Defining a new VSE sublibrary for the program and maps (**Administrator**).

   This task is optional. An existing sublibrary can also be used. If so, the next two steps can be skipped.
2. Copying the skeleton for the compiler from VSE/ICCF library 2 to your primary VSE/ICCF library. Refer to Table 38 on page 249 for a list of the skeletons provided (**Administrator** or **Programmer**).
3. Tailoring the compile skeleton to include the name of the new sublibrary for the program and maps (**Administrator** or **Programmer**).
4. Compiling/assembling the program and maps using the Interactive Interface. (**Programmer**).

5. Reviewing the VSE/POWER LST queue to see if the compile/assemble was successful (**Programmer**).
6. Defining any required VSE/VSAM data files (**Administrator**).
7. If necessary, loading the new files (**Programmer**).
8. Defining the programs, transactions, and any maps to CICS using Resource Definition Online commands. The transaction ID for RDO is CEDA (**Administrator**).

   Note that for program testing, the administrator can activate transactions, programs, and maps using the command:

   ```
   CEDA INSTALL GROUP(name)
   ```

   Once the program has been tested, the administrator can modify the system startup to include the application by entering the command:

   ```
   CEDA ADD GROUP(name) LIST(VSELIST)
   ```

   VSELIST is the name of the group list shipped by z/VSE. To use RDO (transaction CEDA) in your system, you need to add the following to DFHSITSP:

   ```
   GRPLIST=VSELIST
   ```
9. Defining the data files to CICS by modifying the FCT in VSE/ICCF library 59 (**Administrator**).
10. Assembling the new FCT (**Administrator**).
11. Changing the CICS startup job stream to include the new user sublibrary that was defined in Step 3 (**Administrator**). This can be done by:
    a. Copying the skeleton *SKCICS* from VSE/ICCF library 59,
    b. Making the change, and
    c. Submitting the skeleton for processing.
12. Activating the FCT by shutting down and then restarting CICS with a COLD start or use CEDA to define the FCT entry (**Administrator**).
13. Using CEMT to check that the data files are open and available. (**Administrator**).
14. Defining the program to the Interactive Interface using the application profile function (**Administrator**).
15. Creating a selection panel that includes the program as a selection (**Administrator**).

    Note that at this time any HELP panels for the program could be added via the Interactive Interface.
16. Defining a profile for a user who works with the new program (**Administrator**).
17. Signing on as the new user to test the user and application profiles and the new selection panel (**Administrator** or **Programmer**).

Redefining IDs for System Transactions

z/VSE provides several transaction IDs that may conflict with your own transaction ID names. If you want to use your own transaction IDs, you can override the following IDs of z/VSE:

- OLPD
- USER
- HELP
- PF1

- PF3
- PF13
- PF15

OLPD activates the Online Problem Determination function. USER displays active users. The HELP and PF key transactions return a user from native CICS/VSE mode to the Interactive Interface. To use any of these IDs for your own transactions, you can do one of the following:

1. Select the *Invoke CEDA* dialog from the *CICS Supplied Transactions* panel, and redefine the z/VSE definition of the transaction ID.
2. If you use DFHPCT to define transaction IDs, change IESZPCT.Z and run DFHCSDUP.

The DFHPCT defines the transaction IDs used by z/VSE. This includes the seven transaction IDs listed above. Do *not*, however, override any of the other IDs defined by z/VSE. If you do, the Interactive Interface may not work correctly, and functions that you use could have unpredictable results.

In addition, do *not* override transaction IDs that are reserved for CICS.

Chapter 21. Storage and Tuning Recommendations

This chapter provides general information concerning:

- The performance analysis process.
- The exploitation of z/VSE features, including Virtual Storage Constraint Relief and Data In Memory.
- Other factors that influence performance.

Performance information is also available on the Internet via the z/VSE home page at:

`http://www.ibm.com/servers/eserver/zseries/zvse/documentation/performance.html`

Note: The information provided in this chapter about z/VSE performance aspects is of a more general nature. For more specific details, contact your IBM representative who will be glad to provide further information about z/VSE and its performance characteristics.

Performance Analysis Process

Whenever a performance problem is encountered, a structured and logical analysis is important to solve the problem as fast as possible. To this end, you should follow the outline provided on the following pages and ask the following questions:

- Where in the system does the problem occur?
- What has been changed (if the problem did not occur before)?
- When and with which application does the problem occur?
- What actions do I need to take for I/O related problems?

Where in the system does the problem occur?

1. For hardware and software related problems:

   Are the performance related ECs (Engineering Changes) and PTFs (Program Temporary Fixes) installed?
2. For microcode and software setup:

   Are the VM assists active?

   Are the VM guest parameters properly set?

   Are all traces switched off?

   Is DEBUG off?
3. For problems related to the system configuration:

   What is the CPU utilization?

   What is the paging rate?
4. For I/O related problems:

   Use performance monitoring tools for a detailed problem analysis.

What has been changed (if the problem did not occur before)?

- Was there any change in the system, workload, or partition setup?
- Has any new software release from IBM or from another supplier been installed?
  - Which new release was installed?

- Do PTFs for this release exist?
- Can I temporarily deactivate the software or even remove it?
- Can I reproduce the problem?

When and with which application does the problem occur?

- Already in single thread?
- Only under heavy system load?
- Only when specific other applications are running?
- Which particular job step or transaction is affected?
- Did the problem exist before? If not, what has been changed?

What actions do I need to take for I/O related problems?

- Did the number of I/Os increase compared to the former situation?
- Did the duration of the I/Os increase?
- Check the capacity guidelines for device and channel utilizations and the number of actuators and paths.

  Consider that z/VSE supports dynamic I/O handling, which includes dynamic path select (DPS) and dynamic path reconnect (DPR).
- Tune the software with regard to I/O as follows:
  - Check the VSE/VSAM file definitions.

    Use more or larger I/O buffers, especially for a z/VSE system with more private space.
  - Are other disk device types used without re-checking the following?

    The VSE/VSAM CI (control interval) sizes for data and index. The layout of the VSE/VSAM LSR buffers.
- Increase the multiprogramming level (I/O concurrency). This can be done by using additional partitions.
- Use faster I/O subsystems such as the IBM RAMAC disk family, or the IBM TotalStorage ESS (Enterprise Storage Server) family.

  If you emulate an IBM 3380 or an IBM 3390 disk device on an IBM RAMAC or IBM TotalStorage ESS, ensure that your ADD statement specifies:

  ```
  ADD cuu,ECKD
  ```
- Enable and use all the IBM 3990-3/3990-6 RAMAC I/O subsystem or IBM TotalStorage ESS basic caching functions.

z/VSE Exploitation Overview

By supporting 31-bit addressing, z/VSE provides significant performance benefits. These benefits include primarily:

- **Virtual Storage Constraint Relief** below 16MB
- **Data In Memory**

Virtual Storage Constraint Relief (VSCR)

VSCR designates all features and activities that provide space within a partition below the 16MB line. This space can be used for example to:

- Increase the throughput of a CICS partition.
- Support additional 24-bit applications.

VSCR for the private space below 16MB is provided by keeping VSE/POWER and VTAM in private partitions and through the following items which may reside above the 16MB line:

- CICS 31-bit application programs
- VSE/VSAM buffers and major internal control blocks and parameter lists
- DL/I 31-bit applications and DL/I index components
- Selected CICS/VSE internal areas

In addition to these items, page management tables have been moved out of shared space, allowing further growth of a z/VSE system with regard to virtual and real storage.

Note that the amount of storage relief you get, for example, for your CICS production partition is determined not only by the size of the partition below 16MB, but also, essentially, by the amount of virtual areas moved above the 16MB line.

Also, it is of importance that you do not give space away which z/VSE requires for segment rounding (1MB segments in MODE=ESA). Use this shared space of up to 1MB to load up to 1MB of CICS phases into the (enlarged) SVA. This provides more private space for your CICS production transactions. Figure 6 on page 61 provides further details about segment boundaries.

Starting with **VSE/ESA 2.4**, additional VSCR is provided by the **CICS Transaction Server**, where most of the CICS nucleus and control blocks have been moved above the 16MB line.

Data In Memory (DIM)

Data In Memory is a proven concept (introduced first with MVS) to keep more data in virtual (and real) storage. By reducing I/O operations to disk devices, Data In Memory can provide significant reductions in the number of physical I/O operations and thus improve transaction response times besides a reduction in CPU time. The number of reduced I/O operations also allows to drive an ESA-capable processor for a given transaction response time to higher CPU utilization, if this is preferred. z/VSE supports Data In Memory by:

- The ability to assign large VSE/VSAM buffer areas above the 16MB line. This can include up to 15 CICS and 16 batch VSE/VSAM LSR buffer pools.
- CICS data tables
- z/VSE data spaces
- Virtual disk support

CICS data tables (available with the CICS Transaction Server and with CICS/VSE) are a valuable means to also reduce the required CPU-time of a given transaction by keeping a rapidly accessible copy of records in storage thus bypassing VSE/VSAM record management.

To really benefit from Data In Memory, it is mandatory that the appropriate amount of processor (real) storage is available. Otherwise, file I/Os for a partition are replaced by paging-I/Os with potentially big disadvantages for CICS production environments.

Increased Overall Capacity

Increased z/VSE system capacity is made possible by the following enhancements:

- Dynamic partitions

- More address spaces
- More total virtual storage
- Larger real storage
- Virtual Storage Constraint Relief
- Data In Memory

Recommendations for the CICS Transaction Server

With z/VSE you can freely select the number of CICS partitions you want to use. There is enough real storage supported for any number of CICS partitions. On the other hand, you are no more forced to split up your CICS production system for capacity reasons into several CICS partitions and use CICS MRO transaction routing and/or function shipping. In addition, the CICS Transaction Server provides shared data tables. Refer to "CICS Data Tables" on page 265 for details.

Unchanged **24-bit CICS applications** benefit from the fact that the system moves VSE/VSAM control blocks and VSE/VSAM buffers above the 16MB line thus freeing space below 16MB. Specifying more data buffers or using CICS data tables may further improve the response times, if I/O operations to disk devices can be avoided.

31-bit CICS applications will effectively no longer be storage constrained by the size of programs. Together with the other enhancements such as VSE/VSAM buffer restructuring, the efficient exploitation of ESA-capable processors is possible.

Starting with **VSE/ESA 2.4**, the **CICS Transaction Server** provides additional VSCR since the majority of the CICS nucleus code and control blocks has been moved above the 16MB line. Also, the 31-bit storage is now fully managed by the CICS Transaction Server dynamic space management via Dynamic Storage Areas (DSAs).

Through the increased capacity of CICS for all types of applications, the cost of today's complex communication configurations with several CICS partitions and MRO or ISC can now be reduced in certain cases.

To benefit from the higher capacity and from more Data In Memory to reduce I/Os, it is definitely required that the appropriate amount of processor (real) storage is provided, in order not to replace file I/Os with page I/Os.

For debugging applications, you may consider to set storage protection ON and increase the size of the trace area.

Setting Up More Batch Partitions

To obtain higher batch throughput, you can set up more batch partitions. This may require a rearrangement of batch jobs to exploit possible parallelism. In any case, you have to check that no I/O bottleneck is created, especially that the batch I/Os are spread across enough disk devices. By using the PRTYIO command, make sure that your online production gets higher priority in the I/O subsystem.

With z/VSE, **batch programs** will especially benefit from the reduced numbers of I/Os to disk devices by using virtual disks for workfiles. the batch throughput increase for such type of jobs will be up to 30% and more, depending for example on the degree of I/O activity to the workfiles.

Recommendations for z/VSE Exploitation

With z/VSE and an ESA-capable processor, you can gain performance benefits through VSCR as discussed above and you can:

- Save I/Os by implementing the concept of Data In Memory.
- Speed up those I/Os by using fast cached I/O subsystems, such as the IBM TotalStorage disk family, or the Internal Disk of the IBM Multiprise 3000 processors.
- More overlap of I/Os by setting up more partitions and tasks concurrently.

The above list covers the opportunities z/VSE offers to reduce the dependency from the I/O speed. In order to fully exploit z/VSE, check your installation by using the checklist provided under Table 39.

Table 39. Checklist for Exploiting z/VSE Capabilities

Item	I/O savings for	CPU-time savings for
More/larger VTAM buffers	–	Buffer expansions
Larger CICS/VSE DSA below 16MB	–	CICS/VSE SOS processing
Related terminals/applications/ files in same CICS/VSE	–	CICS/VSE transaction routing and function shipping
More/larger VSE/VSAM buffers (NSR, LSR) for permanent user data on disk devices	READs	Setup of I/O, I/O Interference, LSR search
CICS Data Tables (VSE/VSAM LSR KSDS) CICS/VSE CICS Transaction Server	 READs (full key) (imprecise key)	Setup of I/O, I/O Interference, VSE/VSAM code
Virtual disks for temporary/work data	READs WRITEs	I/O interference
Using data spaces	READs WRITEs	I/O interference
More applications resident	READs	Setup of I/O, I/O interference, program load code
Applications more generous with regard to virtual storage	READs WRITEs (temp)	Setup of I/O, I/O interference, application code

VSE/VSAM Multiple LSR Pools

Multiple LSR pools are a big step forward in improving VSE/VSAM file accesses. Up to 15 LSR pools can be allocated for CICS (and 16 for batch partitions). They provide tuning improvements, such as:

- Separation of heavily used files.

  Such files may be "unfriendly" to each other (stealing buffers or dominating a subpool during BROWSE, for example).
- Grouping of files; for example by application, work shift, or usage frequency.
- Separation of data and index-CIs (of different files) so that data and index CIs do not compete in the same subpool.
- Full freedom to select optimal data and index-CI sizes (without regarding CI-sizes of other LSR files).
- Smaller subpools for faster searches are possible. This avoids long subpool searches (CPU-time) with a low chance of a hit ratio increase if:
  - Shared with many other files.
  - More subpools with less buffers per subpool (uses less CPU-time).

  Overall, you get the following performance benefits:

  Reduction of VSE/VSAM I/Os (with the same subpool sizes), or even a higher reduction of the VSE/VSAM I/Os (with larger subpool sizes) at the cost of some CPU-time.

VSE/VSAM supports up to 15 (sub)pools both for data and index buffers, which is exploited by the **CICS Transaction Server** and must be specified via Resource Definition Online (RDO).

z/VSE customers can exploit very large LSR buffer pools through VSE/VSAM buffer hashing. For details, refer to the manual *VSE/VSAM User's Guide and Application Programming*.

Tuning of Multiple LSR Pools

In order to properly tune multiple LSR pools (in the CICS/VSE DFHFCT TYPE=SHRCTL and the VSE/VSAM BLDVRP definitions) you should be aware of the following:

Independent of the LSR buffers being able to reside above the 16MB line, the VSE/VSAM control blocks per LSR pool are above the 16MB line:

```
Size in partition GETVIS per LSR pool

   = 2128 + 72 x no. of subpools + 108 x no. of buffers in pool
          + (920 + max. key length) x STRNO
```

The optimal settings for the performance relevant parameters are characterized by:

- LSR pools: Define only as many as required.
- STRNO: Do not largely oversize STRNOs.

  For a single LSR pool, the maximum STRNO was and remains 255. Assuming the same workload, you may for all your pools require only a few more STRNOs (since they are shared less) compared to the number you had before for a single pool.
- KEYLEN: Specify correctly.

  If you specify as KEYLEN the maximum of all the KSDS files in the pertinent pool, CICS need not determine that value at startup time.

Furthermore, use the provided CICS LSR hit ratio statistics with the I/Os and the number of hits per CI-size (subpool) for each pool to optimize the number of buffers per subpool as follows:

Increase number until hit ratio does no more increase sensibly.

Decrease number until hit ratio does no more decrease sensibly.

For further details refer to "Buffers for VSE/VSAM Files Accessed from a CICS/VSE Partition" on page 274.

CICS Data Tables

As described in the CICS documentation, CICS data tables can be viewed as a data look-aside within CICS virtual storage for full-key READs of VSE/VSAM LSR defined KSDS files. In addition, the CICS Transaction Server also supports imprecise (generic) keys as shown in the following table:

CICS/VSE	CICS Transaction Server
Full-key READs only.	Also imprecise key (GENERIC, GTEQ, BROWSE).
Not directly sharable between CICS/VSE partitions.	Directly sharable between CICS TS partitions.
Remote access from other CICS systems only possible via function shipping.	Function shipping only required for updates from other CICS TS systems.

For the CICS Transaction Server, normal VSE/VSAM LSR services are only required in the following cases:

- If a READ record is not in the table.
- For UPDATE requests.

The performance benefits of CICS data tables are the fast and efficient access to data, caused by:

- Shorter path length compared to access records via VSE/VSAM.
- Less I/Os to disk devices if sufficient number of records are loaded into virtual storage (and enough processor storage is available).

A noticeable reduction in CICS DSA (dynamic storage area) use is caused by the decrease in transaction residency time. All factors together result in improved transaction response times and throughput.

Note: Use CICS data tables before you increase VSE/VSAM LSR subpools to a much higher number than you used before.

z/VSE Virtual Disks

Consult also Chapter 12, "Using Data Spaces and Virtual Disks," on page 167 when planning for virtual disks.

z/VSE virtual disks are a versatile means to redirect I/O requests for disk devices to virtual storage (z/VSE data spaces). Thus, if enough processor (real) storage is available, the number of physical I/O accesses to disk devices can be significantly reduced.

Naturally, this concept can be applied only to data which need not reside on a permanent storage medium.

The following table provides an overview of such types of "workfiles" in the widest sense, ranging from "real" workfiles, via test files, via reproducible, short living files, up to copies of permanent data on real disk devices, such as read only, or even read intensive copies of user data.

All items shown in Table 40 apply in principle for both, batch and CICS partitions, although batch is the primary usage area.

Table 40. z/VSE Virtual Disk Opportunities

Compiler work areas Link-edit work areas (option CATAL) SORT work areas DTSANALS work areas CSP work files	IJSYS01 to IJSYS07 SYSLNK SORTWK1 to SORTWK9 IJSYS02 CSP.USER.WORK.VSAM
Test/work SAM and VSE/VSAM files Test/work libraries	
Reproducible input/output files across jobs/job steps. For example, for: - Batch production - DL/I reorganization	SYSIPT, SYSPCH
Read only user files (see Note below) Read only libraries. For example, for: - PSF/VSE fonts - RPG II CICS/VSE applications	Copied from real disk Copied from real disk Cannot be made resident
Read intensive user files Read intensive libraries	Copied from real disk, real disk updates by applications Copied from real disk, real disk updates by applications

Note: Apart from the read intensive items, all opportunities shown in Table 40 do **not** require any change in the application itself. You only have to take care for the updates on the real disk device, and to initially load the data onto the virtual disk. Put CICS TEMP STORAGE AUX (if used in spite of MAIN) on virtual disk only if the applicability has been evaluated (for example, if TS AUX is only used for recreatable data).

Virtual Disk Usage Hints

In order to optimally exploit the z/VSE virtual disk support, consider the following usage rules:

- Do not grossly overcommit your real storage with virtual disks.
- When selecting the type of files for virtual disks:
  - Prefer READ intensive over WRITE intensive files.
  - Prefer those files with the smallest I/O blocking.
- Define your virtual disks as small as possible.
- Reuse virtual disk extents as much as possible.

- If the data is no more required, release a complete virtual disk via DVCDN cuu and VDISK UNIT=cuu,BLKS=0. This avoids that in the long run unused data is unnecessarily paged out and paged in again.
- Do not load large virtual disks (for example, with VSE/VSAM Backup/Restore) during production. Loading virtual disks larger than the unused real storage will cause page-outs.
- Under z/VM, the VSE Lock file should be on a VM virtual disk or at least on a VM minidisk.

General Release Transition Hints

When migrating to a new z/VSE release, consider the following general performance rules:

- Adapt your hardware resources (processor power, real storage) to the intended degree of:
  - Increased concurrency of active partitions/tasks.
  - Data In Memory exploitation.

  Increase both items in a controlled manner, in order to gain most benefit of both for your specific environment.
- Start with moving VSE/VSAM buffers for 24-bit transactions above 16MB. Before going further, observe the results for storage relief.
- Continue by exploiting Data In Memory:
  - Enlarged VSE/VSAM buffers above 16MB.
  - CICS data tables.
  - Virtual disks.

    If batch is running side by side with online production, prefer to tune virtual disks **after** you have tuned your CICS environment.
  - Other usage of data spaces in addition to virtual disks.
  - Relink/Recompile your application programs for 31-bit addressing.

    Refer also to the LE/VSE performance documentation.
  - Check your assembler programs for 31-bit conformance before compiling for 31-bit with the High Level Assembler.

IPL Commands

The following IPL commands influence storage consumption and can affect performance:

- ADD
  - SHR (DASD sharing by multiple CPUs)
  - ECKD (when emulating 3380/3390 disk devices on an ESS or the Internal Disk of a Multiprise 3000)
- DPD (location and size of the page data sets)
- SYS (system parameters)
  - JA
  - DASDFP
  - CHANQ
  - NPARTS
  - PASIZE
  - SDSIZE

- SPSIZE
- BUFSIZE
- SEC

For details about these commands refer to “ADD”, “DPD”, and “SYS” in the manual *z/VSE System Control Statements*.

Notes:

1. Do not define a disk device as shared (SHR) unless you use DASD sharing.
2. Specifying TYPE=NORMAL in the ASI ($ASIPROC) *master procedure* avoids the overhead of device sensing during IPL. For information about this parameter, refer to the manual *z/VSE Guide to System Functions* under “The ASI Master Procedure”.

NPARTS Definition

You should not define much more partitions for NPARTS than you actually need. The upper limit that you should specify is the maximum number of concurrently active partitions (static plus dynamic). Specifying a higher value for NPARTS requires one additional task and 4KB shared space for each dynamic partition.

JCL Statements and AR Commands

You should consider the following statements and commands in terms of performance or storage consumption:

- **ALLOC**
- **SIZE**
- **PRTY** (set partition priorities)
- **PRTYIO** (set I/O priorities per partition)
- **MSECS** (partition balancing time)
- **SIZE** (parameter on EXEC card)
- **LIBDEF**
- **GETVIS**

General Recommendations for Allocating Virtual Storage

Adhere to the following rules when allocating virtual storage:

- All allocations should be multiples of:

  4KB for SIZE, ALLOC R, and SETPFIX

  64KB for VPOOL, VIO, and ALLOC
- Do not specify a greater value for VSIZE and PASIZE than actually needed. This helps you keep the tables for system management as small as possible.
- Specify SPSIZE as small as possible. This shared space can only be used for shared partitions and should not be higher than the sum of all required shared partition sizes.
- Specify VPOOL=64K in the IPL procedure.

  This recommended value applies for VSE/ESA Version 2: The VPOOL is the address area for the (larger) VIO area. To estimate the size of the necessary VIO area (native or under VM) use the following values:

  Linkage Editor 64KB

Additional CICS/VSE 128KB

The VIO value influences the total size of the page data set and must thus be reflected in the DPD command.

- Specify SIZE=AUTO in the EXEC statement whenever possible. As a result, a value is taken needed to load the largest phase with the same first 4 characters of the phase specified in the EXEC statement.

  You cannot specify SIZE=AUTO for VSE programs which dynamically expand program storage into dynamic storage without using GETVIS requests. Such programs are the Linkage Editor, the Librarian and IBM Compilers.

  Use the GETVIS command to display the GETVIS size available.

SVA Considerations

Place only those phases into the SVA-24 which are effectively used in a reentrant way by several tasks and partitions. This is especially true for user phases placed into the SVA via a SET SDL command. SVA-31 is not critical since sufficient GETVIS and PSIZE storage is available.

VTOC Considerations

ECKD Devices

After initial installation (automatic) the size of the VTOC is as follows for the ECKD disk devices IBM 3380 and 3390:

```
DASD:  Tracks:

3380     4       (provides space for 212 entries)
3390     4       (provides space for 206 entries)
```

The number of tracks can be increased up to 15 (1 cylinder) without the danger that an overlap occurs. You specify the number of tracks in the **INIT** command of the *Device Support Facilities* program (ICKDSF). You must subtract 11 tracks from the current VTOC beginning and start initialization there to get a VTOC size of 15 tracks.

FBA Devices

After initial installation (automatic) the size of the VTOC is as follows for an FBA disk device:

```
DASD:  CISIZE:

FBA     8192     (provides space for 228 entries)
```

The CISIZE of 8192 provides a performance optimum. It is at the same time the maximum and cannot be increased. This means that if a larger VTOC is required, an extension at that location (in the middle of the disk) is not possible since it would cause an overlap.

The manual *z/VSE Installation* shows the format of the INIT commands and the DOSRES and SYSWK1 layouts for the disk devices supported for installation under "z/VSE Disk Layouts".

VSE Libraries

You should consider the following for performance reasons:

1. Periodically backing up and restoring VSE libraries improves search performance. Note that reorganization of scattered members is only done when *entire* libraries are restored. A MINI startup can be used for this task and allows a restore of the entire library.
2. The Restore and Copy/Move functions of the librarian are faster if the target library is uniquely assigned.
3. Check the LIBDEF sequence. To optimize sublibrary access, the most frequently accessed sublibraries should come first.
4. After the installation of z/VSE, it is recommended to increase the size of the dump library and move it to a different volume. For the library definitions, refer to LIBRDEFS.Z in IJSYSRS.SYSLIB.

System File Placement

The following recommendations regard file placement:

1. Consider splitting the Page Data Set and placing the extents (of about equal size) on different volumes.
2. In case of high SPOOL utilization, you should move the VSE/POWER data file away from the system disk onto a private disk device.

   You should also consider defining the data file with more than one extent and place each extent on a different actuator.
3. A lock file is needed *only* when sharing disk devices. It should be on a disk device that has low usage and contains no other system files.
4. Concurrently active data files should be placed adjacent to one another, if (for any reason) they are located at the same address.

Parallel Page I/O and Page Data Set Extents

Within page management, there exists a device queue for every page data set device. This allows the overlapping of page I/O operations if the extents are on different disk devices.

The virtual storage of a partition is distributed across all available page data set extents. If you modify the page data set layout, you should distribute your page data set across several extents on separate disk devices. To optimize segment distribution, select extents of about equal size.

If your system uses the system-defined page data set, move it for performance reasons to a different volume.

Recommendations for Using DASD Sharing and Lock Files

When you share disk devices (DASD sharing), you can reduce the overhead as far as possible by adhering to the following rules:

- Put non-shared data on non-shared disk devices.

  This is a general rule which does not only bring performance benefits, but also is reasonable for non-performance reasons.
- Avoid, whenever possible, to add disk devices as cuu,SHR.
- Only specify as many CPUs as needed in the DLF NCPU parameter, in order to speed up Lock File I/Os.

  Increasing the total size of the Lock File (NCYLS) to 2 or more cylinders may reduce the I/Os by increased “hashing hits”.

- Lock File location:

  Put the z/VSE Lock File, if possible, on a separate volume (native). Under VM, use a VM virtual disk, or at least a separate VM minidisk for the Lock File. Place only other data (if at all) on the Lock File if it is rarely accessed. If possible, select a string that is not highly utilized. The reason is that:
  - RESERVE/RELEASE is being used by the VSE lock manager.
  - RESERVE/RELEASE locks a complete volume.
- Lock File volume:

  If all z/VSE systems sharing the same Lock File run under the same VM systems, you can use one of the following as Lock File volume:
  - A VM virtual disk
  - A separate VM minidisk

  Instead of using the above and when running on different processors, you may use as Lock File volume one of the following:
  - A cached IBM 3380
  - A cached IBM 3390
- Placing the Lock File on a SCSI FCP-attached FBA Disk:

  If you want to allocate a lock file on a SCSI disk, you must install a unique FCP adapter for each VSE system that shares the lock file. Each VSE system must access the lock file via its unique FCP adapter. This is necessary because the hardware does not reserve the SCSI disk (using RESERVE command) per FCP cuu, but instead per FCP adapter.

  Warning!: If the VSE systems sharing the lock file also share the same FCP adapter to access the SCSI disk, the lock file and the data to be protected may be destroyed.

  Notes:
  1. You cannot define the lock file on an FCP SCSI-attached DOSRES or SYSWK1 disk, or on another IPL device. The IPL checks for the IPL device.
  2. You should not establish a multipath connection to an FCP SCSI-attached device on which the lock file resides. If more than one connection to this device exists, the IPL will reject the DLF command for this SCSI device. If a DEF/SYSDEF command for the lock file device is entered after the DLF command, the FCP device driver will reject this command.
  3. If the z/VSE system enters hardwait, a release of the lock file is attempted in order to prevent other VSE systems being affected that share the lock file.
  4. During an IPL, the SCSI FCP device where the lock file resides is released unconditionally.

File Placement Specifics

For the following types of files it is performance-wise beneficial to place them (if non-shared) on non-shared volumes:

VSE/VSAM files and catalogs

VSE libraries

Apart from standard device contention, it is irrelevant from a performance point-of-view where the following files (if non-shared) are located, be it on non-shared or shared devices (without the Lock File):

SAM, DAM, and ISAM files

VSE/POWER files

In other words, it is performance-wise not beneficial to move such a file from a shared to a non-shared volume (except the shared volume would also contain the Lock File or the shared device would have too much device contention).

For further information on DASD sharing, refer to the manual *z/VSE Guide to System Functions* under “DASD Sharing with Multiple VSE Systems”.

Recommendations for VSE/POWER

The following recommendations apply only for experienced users which are able to perform these steps outside the daily production work. As shipped, VSE/POWER is running in partition F1.

- Select SETPFIX as small as possible.

  During VSE/POWER shutdown, you get statistics (representative workload needed) about the maximum number of pages fixed. During normal processing, you can use the D STATUS command to get statistics.

  Use the value provided and reduce SETPFIX for F1 to that size.
- In general, try to reduce the partition GETVIS size and the SIZE allocation.

For further information refer to “Processor- and Virtual Storage Requirements” in the manual *VSE/POWER Administration and Operation*.

Scheduling VSE/POWER Jobs in Dynamic Partitions

If a job **can** be processed in a dynamic partition, then it is recommended that you run the job in a dynamic partition. There is only a minimal overhead for a VSE/POWER job if it is run in a dynamic partition instead of a static partition.

If you follow this recommendation, then you will “conserve” your static partitions, which are limited in number. Remember that with static partitions, you can:

- Use the PRTYIO command for explicitly setting and changing I/O priorities.
- Run all types of programs.

Also note that if you have extremely short VSE/POWER jobs that you frequently run, it is recommended to run them in static partitions.

VSE/POWER DBLK Size

The maximum DBLK size for VSE/POWER is 65024 bytes. The number of SIOs to the VSE/POWER data file depends on the DBLK size specified. The affected buffers are allocated as follows:

- One for each data file extent in real space (SETPFIX) of the VSE/POWER partition.
- One for each spooled device (used by spooled partitions) in the partition GETVIS area of the VSE/POWER partition.

The default value of DBLK used by VSE/POWER is about 7500 bytes. You can specify another value using the DBLK operand of the VSE/POWER generation macro. For details refer to “Regenerating VSE/POWER” on page 279. For an overview on DBLK sizes, refer to Table 21 on page 105.

In general, a DBLK size of half the track size is recommended. This is possible with z/VSE 3.1, since VSE/POWER 7.1 allows the VSE/POWER Queue File virtual storage copy to be above the 16MB line.

Recommendations for VTAM

The following recommendation apply for the partition in which VTAM is running. As shipped, VTAM is running in partition F3.

- Use SIZE=AUTO.
- Try to reduce the VTAM buffer sizes.

  To find out about buffer sizes use the command D NET,BFRUSE after a peak hour of operation. The size of the buffer areas can be reduced as long as MAX USED is lower than the values allocated during VTAM startup. The storage for VTAM buffers is taken from the data space defined in the startup job VTMSTRT (skeleton SKVTAM). Adapt the DSPACE parameter in the // EXEC ISTINCVT... statement accordingly.

  Since VTAM buffer handling has changed with VSE/ESA Version 2, consult also the manual *VTAM Migration Guide for VSE*.

  For more information about buffers and their sizes, refer to the manual *VTAM Resource Definition Reference*, and the diskette *Estimating Storage for VTAM*.

Recommendations for the CICS Transaction Server

For information on how to tune the CICS Transaction Server, refer to the *CICS TS Performance Guide*, SC33-1667.

Recommendations for CICS/VSE

The following recommendations apply for a partition in which CICS/VSE is running. As shipped, partition F4 is prepared for CICS/VSE which must be installed from the extended base tape.

- Use PGSIZE=2048 in any CICS system initialization table (SIT) to minimize fragmentation of virtual storage for nonresident CICS applications or phases. This is the recommended value of allocation for the CICS DSA (Dynamic Storage Area), also called a ″CICS page″.
- Select the smallest version of a CICS/VSE module which satisfies your requirements. For example, often an E$ module is sufficient instead of an S$ module for TCP/ZCP (Terminal Control Programs).
- Make only those programs resident in CICS which really need to be there. During CICS shutdown or on request, you get statistics which tell you how often a program has been fetched from the library.
- If many non-resident, small programs are loaded frequently, you should consider making them resident and have them densely packed (via ALT) to avoid wasting virtual storage.
- The use of VSAM LSR (Local Shared Resources) is beneficial (at the cost of some CPU-time) to reduce I/Os and virtual storage. No general rules can be given since it is workload dependent. But you should follow LSR rules to avoid using more virtual storage than actually needed; for each opened LSR VSE/VSAM file only 3.5KB will be needed instead of 7KB for each NSR file.
- Remove obsolete, user-created table entries especially in the TCT and FCT tables.
- Reduce the number of DL/I buffers (to be controlled via the DL/I statistics) as well as the number of subpools.

Buffers for VSE/VSAM Files Accessed from a CICS/VSE Partition

Using the Local Shared Resources (LSR) facility of VSE/VSAM is the default. However, the number of buffers set up by CICS and the default sizes of these buffers may not be adequate to achieve an acceptable performance. You may want to do some performance optimization in this area. As a first step, analyze the CICS shutdown statistics after a peak production period with defaulted number of buffers and their sizes. The section `VSAM SHARED RESOURCES STATISTICS` provides the information you need. A high number of I/O requests waiting for a buffer of a specific size indicates that the number of available buffers of this size is too low. To achieve an acceptable performance, you can do the following:

1. Adjust the sizes of the control intervals (CIs) of your files. Check your LISTCAT output to see which CI sizes VSE/VSAM selected for the index component of your file. If necessary, redefine your file and select CI sizes such that they are close to or identical with the standard CI sizes as listed below:

512	
1,024	
2,048	
4,096	(or any multiple of 4,096 up to 32,768)

2. Try to keep the number of different control-interval sizes small. The smaller this number is, the better will be your performance.
3. Define the number and sizes of buffers, the maximum key length and the total string number for VSE/VSAM LSR (local shared resources). Use skeleton SKSHRCTL for this task and follow the description provided by the skeleton.
4. Reassemble and catalog the file control table (skeleton DFHFCTSP in VSE/ICCF library 59) of your CICS subsystem after having activated the /INCLUDE SKSHRCTL statement.

For a detailed description of the DFHFCT macro with `TYPE=SHRCTL`, refer to the *CICS Resource Definition Guide*.

Chapter 22. Regenerating the Supervisor and Other z/VSE Components

This chapter discusses the regeneration of:

- The supervisor
- VSE/POWER
- VSE/ICCF and the DTSFILE

For the regeneration of a supervisor, it is necessary that you first install the *Generation Feature*. You can do this during the installation of z/VSE or later using the dialog provided for it.

For the regeneration of VSE/POWER and VSE/ICCF, z/VSE provides skeletons.

Installing the Generation Feature

The z/VSE distribution tape(s) contain source code that provides generation capability for the **supervisor** modules. *Installation of this code (called the Generation Feature) is optional*. You need it if the options of the pregenerated system are not adequate for your needs. "Installing VSE Generation Feature" in the manual *z/VSE Installation* provides details for installing the Generation Feature.

Regenerating the Supervisor

z/VSE provides two pregenerated supervisors as shown under "Pregenerated z/VSE Supervisors." You may change any of the supervisor default values provided and generate your own supervisor if there is a need.

Pregenerated z/VSE Supervisors

Two supervisors are shipped with z/VSE ($$A$SUPI and $$A$SUPX). They are both MODE=ESA which is the only supervisor mode supported since VSE/ESA 2.1. The MODE generation option of the SUPVR macro has therefore been dropped. Refer to "Migration Items of Previous Releases" on page 94 for further details.

Following is a list of the generation options used. The macros used for specifying these options are: SUPVR, FOPT, and IOTAB.

1. Supervisor $$A$SUPI is used by default. It supports up to 1024 I/O devices compared to the 254 of $$ASUPX. The following options are set:
   ```
   SUPVR   ID=I      FOPT   TRKHLD=12           IOTAB   IODEV=1024
                            USERID=Z.VSE.SUPI           NPGR=3060
   ```
2. If you have defined less than 254 attached devices, you can save storage by using the Supervisor $$A$SUPX. The following options are set :
   ```
   SUPVR   ID=X      FOPT   TRKHLD=12           IOTAB   IODEV=254
                            USERID=Z.VSE.SUPX           NPGR=3060
   ```

Skeleton SKSUPASM

z/VSE provides in VSE/ICCF library 59 skeleton SKSUPASM for supervisor generation.

Supervisor Generation Macros

Coding Sequence

The macros and their operands have to be coded for the assembly of a supervisor in the following sequence: SUPVR, followed by FOPT, followed by IOTAB. This chapter discusses the macros and their operands in this sequence. The operands of a generation macro may be specified in any order.

Conventions for Format Descriptions

The conventions for showing the format of the generation-macro operands are:
- The default value for a specification, if applicable, is underscored. The assembler uses this default value if you omit the operand in question or specify an incorrect value.
- Braces ({ }) indicate that you must select one of the enclosed values. The values from which you can choose are separated by a vertical line (|).
- Uppercase characters and equal signs (=) must be coded as shown.
- Lowercase characters represent values which you must (or may) supply.

Available generation macros and their operands are:

```
SUPVR Macro     FOPT Macro      IOTAB Macro
-----------     ----------      -----------
 ID              TRKHLD           IODEV
                 USERID           NPGR
```

The former TTIME operand of the FOPT macro is no longer supported.

SUPVR Generation Macro

The operands for the macro are:

ID={<u>1</u>|c}
: For c, specify an identifying character unique to your system if you plan to use two or more supervisors on the system. As an identifying character, use any alphameric character (from A through Z and from 1 through 9).

  The assembler adds the specified character to the string $$A$SUP and uses this new string as the name of your supervisor. For example: if you specify ID=A, the assembler assigns $$A$SUPA as name to your supervisor.

FOPT Generation Macro

Use the FOPT (Optional Function) generation macro to specify optional functions that you want to be included in your supervisor.

The operands for the macro are:

TRKHLD={<u>NO</u>|n}
: Specify the number (n) of hold requests that the system should allow to be active at any one time if the hold function is to be supported.

  When processing a request with a hold for an update to a file on disk, the system prevents any other task using the track hold function from accessing the same data.

  The maximum number of hold requests you can specify to be active at a time is 255. If your specification is invalid (non-numeric or greater than 255), the generated support allows 10 hold requests to be active concurrently.

Certain program products may require the track hold support to be included in your supervisor.

USERID=identifier
Specify an identifier for a supervisor if you want this identifier to be printed as part of the IPL COMPLETE message.

You may specify an identifier of up to 16 characters in length. If you specify a longer identifier, the assembler truncates your specification to 16 characters at the right end. If you specify less than 16 characters, the assembler pads the identifier with blanks on the right.

Do not specify the identifier within quotes; imbedded blanks are not allowed.

IOTAB Generation Macro

The IOTAB (Input/Output Tables) generation macro defines the space needed by the system for device tables.

The operands for the macro are:

IODEV={254|n}
Specify the number (n) of I/O devices attached to your system's processing unit. The maximum you can specify is 1024, the minimum value is 4. Each unit that you define to the system by an IPL ADD command must be accounted for in your specification for the IODEV= operand.

NPGR=n
Specify the number (n) of programmer logical units you need to have available for all of your partitions at any point in time. The highest and lowest values you can specify for n are:

```
Highest      255 x p
Lowest        30 x p (this value is also the default)
```

where p = number of static partitions (maximum = 12)

Note: Regardless of what you specified as NPGR value, the number of allocated programmer logical units per partition defaults to 30. You can change this number at a later time with the NPGR job control command described in the manual *z/VSE System Control Statements* under "NPGR". The defaults are reset via the allocation procedures (see SKALLOCA, SKALLOCB, and SKALLOCC).

Supervisor Generation Example

Supervisor generation is dependent on functions of the High Level Assembler. A job stream example is provided below. In the example, it is assumed that you want to have the number of I/O devices set to 512 (IODEV parameter). You may also use skeleton SKSUPASM provided in VSE/ICCF library 59.

High Level Assembler Example

```
* $$ JOB JNM=SUPGEN,DISP=D,CLASS=X
* $$ LST CLASS=X,DISP=D
// JOB SUP GENERATION
// OPTION NOSYSDUMP
// LIBDEF *,SEARCH=(PRD1.MACLIB,PRD2.GEN1)
// LIBDEF *,CATALOG=IJSYSRS.SYSLIB
// OPTION CATAL,DECK
// EXEC ASMA90,PARM='MAX,(ABOVE)'
*PROCESS FLAG(NOCONT,NOSUBSTR),USING(WARN(0))
     TITLE 'VSE/AF VERSION 7 REL.1.0 STANDARD SUPERVISOR T'
         SUPVR                                                  X
               ID=T
         FOPT                                                   X
               TRKHLD=12,                                       X
               USERID=Z.VSE.SUPT.710
         IOTAB                                                  X
               IODEV=512,                                       X
               NPGR=3060
         END
/*
// IF $MRC GT 4 THEN
// GOTO NOLNK
// EXEC LNKEDT,PARM='MSHP'
/. NOLNK
/&
* $$ EOJ
```

Notes:

1. In the statement

   ```
   * $$ JOB JNM=SUPGEN,DISP=D,CLASS=X
   ```

   CLASS=X provides a partition of 60 MB size.

   The assembler Release 5.1 no longer uses workfiles. Instead, it uses partition GETVIS storage. This means, for a supervisor generation you must define a dynamic partition of at least 60 MB size.
2. The statement

   ```
   *PROCESS FLAG(NOCONT,NOSUBSTR),USING(WARN(0))
   ```

   prevents the excessive generation of warnings by the High Level Assembler for VSE. This may happen when assembling old (existing) programs.

Regenerating VSE/POWER

Skeleton **SKPWRGEN** defines the options for VSE/POWER generation. It reflects the values which were used to generate the supplied VSE/POWER phase **IPWPOWER** (identified by **--V100--**).

The skeleton is shipped in VSE/ICCF library 59. If you use the skeleton, copy it first to your VSE/ICCF primary library and edit the copied skeleton. Figure 23 on page 280 shows the skeleton. Comments included in the skeleton are not shown. You can change the operands of the POWER macro. Refer to "POWER Generation Macro" in the manual *VSE/POWER Administration and Operation* for a description of the POWER macro and operands.

In the skeleton, each operand is on a separate line. When you edit the file, **do not delete the continuation characters (*) in column 72**.

In the POWER statement, replace **--V100--** with your own VSE/POWER phase name. z/VSE uses the pregenerated VSE/POWER phase IPWPOWER. Do **not** use the name IPWPOWER since IPWPOWER is serviced together with VSE/POWER. If you generate your own VSE/POWER, you must also tailor skeleton SKPWSTRT which calls IPWPOWER. In SKPWSTRT, change the statement

```
// EXEC IPWPOWER
```

and replace IPWPOWER with the phase name you specified for **--V100--**. For details about skeleton SKPWSTRT, refer to the manual *z/VSE Administration* under "Skeletons for Starting Up VSE/POWER".

```
* $$ JOB JNM=POWERGEN,CLASS=0,DISP=D
* $$ LST CLASS=Q
// JOB POWER GENERATION
// LIBDEF *,SEARCH=(PRD2.GEN1,PRD1.MACLIB)
// LIBDEF PHASE,CATALOG=PRD2.CONFIG
// OPTION CATAL
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
PWR      TITLE 'VSE/POWER - IPWPOWER  GENERATION '
         EJECT
         SPACE 3
--V100-- POWER                                                         *
               ACCOUNT=YES,                                            *
               CLRPRT=YES,                                             *
               COPYSEP=YES,                                            *
               DBLKGP=8,                                               *
               DBLK=0,                                                 *
               FEED=NO,                                                *
               JLOG=YES,                                               *
               JSEP=(0,0),                                             *
               LTAB=(10,00,05,10,15,20,25,30,35,40,45,50,56),          *
               MEMTYPE=P,                                              *
               MRKFRM=YES,                                             *
               MULT12=NO,                                              *
               NTFYMSG=100,                                            *
               PAUSE=NO,                                               *
               PRI=3,                                                  *
               RBS=(0,0),                                              *
               SECNODE=AAAA,                                           *
               SHARED=NO,                                              *
               STDCARD=(0,0),                                          *
               STDLINE=(0,0),                                          *
               SPLIM=90,                                               *
               SPOOL=YES
         EJECT
*/INCLUDE SKPWRBSC
           .
           .
*/INCLUDE SKPWRSNA
         END
/*
// EXEC LNKEDT,PARM='MSHP'
/&
* $$ EOJ
```

Figure 23. Skeleton SKPWRGEN (VSE/POWER Generation)

Note: The statement

```
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
```

calls the High Level Assembler. Refer to Chapter 6, “Migrating From Earlier Releases,” on page 91 for details about calling the High Level Assembler.

You may add the following parameters to the skeleton (after the line --V100-- POWER shown in Figure 23):

- If you want to use a master password, add:

  ```
  MPWD=--V200--,                                       *
  ```

  The variable --V200-- is the master password.
- If you want to activate VSE/POWER PNET, add:

  ```
  PNET=--V101--,                                       *
  ```

The variable --V101-- is the name of your PNET phase. This is the name of the network definition table as specified for the first PNODE macro with LOCAL=YES.

- If you have SNA workstations attached to your system, add:

```
   SNA=YES,                                                      *
```

For the connection between VSE/POWER and VTAM, the VTAM APPLID *POWER* is provided in the VTAM application startup book. The same APPLID is used, if you add SNA=YES to the skeleton.

- If you want to use user-written exit routines, add:

```
JOBEXIT=--V102--,                                                *
  or
JOBEXIT=(--V102--,--V103--),                                     *

NETEXIT=--V104--,                                                *
  or
NETEXIT=(--V104--,--V105--),                                     *

OUTEXIT=--V106--,                                                *
  or
OUTEXIT=(--V106--,--V107--),                                     *

XMTEXIT=--V108--,                                                *
  or
XMTEXIT=(--V108--,--V109--),                                     *
```

Where:

--V102-- is the name of the user-written job exit routine.

--V103-- is the number of bytes reserved as work area.

--V104-- is the name of the user-written PNET receiver exit routine.

--V105-- is the number of bytes reserved as work area.

--V106-- is the name of the user-written output exit routine.

--V107-- is the number of bytes reserved as work area.

--V108-- is the name of the user-written PNET transmitter exit routine.

--V109-- is the number of bytes reserved as work area.

If you use the PNET, SNA, or exit parameters, make sure that you have a continuation character (*) in column 72.

If you have VSE/POWER RJE (Remote Job Entry) definitions for **BSC** work stations, you should include the skeleton SKPWRBSC. Remove the asterisk (*) in front of the statement:

```
*/INCLUDE SKPWRBSC
```

You must also tailor the SKPWRBSC skeleton. This is described in the manual *z/VSE Networking Support* under “VSE/POWER BSC Skeleton SKPWRBSC”.

If you used the remote configuration dialogs to define **SNA** workstations, you should include the skeleton SKPWRSNA. Remove the asterisk (*) in front of the statement:

```
*/INCLUDE SKPWRSNA
```

SKPWRSNA contains a predefined set of SNA workstations which can be configured with the remote configuration dialogs. The dialog generates the VTAM line, PU, and LU definitions. Under “VSE/POWER SNA Skeleton SKPWRSNA”, the manual *z/VSE Networking Support* describes the SKPWRSNA skeleton and the remote configuration dialogs.

After making the changes, run the DTRSEXIT macro. This macro deletes specific comments from the skeleton. You should do this before you file the skeleton. On the command line, enter:

```
@DTRSEXIT
```

After the macro finishes, file the job. You can then submit it to the system for processing.

Regenerating VSE/ICCF

z/VSE provides two phases for VSE/ICCF: DTSIGEN and DTSIGENM. They can both be used without regeneration. DTSIGEN is the default VSE/ICCF while DTSIGENM provides larger interactive partitions.

If you need to regenerate VSE/ICCF because you want different options to be set, proceed as follows:

- Define and create new phase via skeleton SKICFGEN.
- Shutdown VSE/ICCF.
- Restart VSE/ICCF to activate the new phase.

The **SKICFGEN** skeleton defines the options for VSE/ICCF generation.

The skeleton is shipped in VSE/ICCF library 59. If you use the skeleton, copy it first to your VSE/ICCF primary library and edit the copied skeleton.

Figure 24 shows the skeleton. Comments included in the skeleton are not shown. You can edit and change the generation operands for the DTSOPTNS macro. Under “VSE/ICCF Tailoring Options (DTSOPTNS Macro)”, the manual *VSE/ICCF Administration and Operation* describes the DTSOPTNS macro and its operands.

In the skeleton, each operand is on a separate line. When you edit the skeleton, **do not delete the continuation characters X in column 72**.

You can also change, add, or delete statements and operands, if required. You should **not** change the following operands. These are required for z/VSE.

- ALTSEC
- COMLIB
- CRJE

```
* $$ JOB JNM=ICCFGEN,CLASS=S,DISP=D
* $$ LST CLASS=Q
// JOB ICCF GENERATION
LIBDEF PHASE,CATALOG=PRD2.CONFIG
// OPTION CATAL
   PHASE DTSIGEN,*
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
         DTSOPTNS ALTSEC=NO,                                           X
               ATN2741=YES,                                            X
               CANKEY=PA2,                                             X
               CISIZE=2048,                                            X
               COMLIB=2,                                               X
               CRJE=(YES,Q,A,D,A),                                     X
               DISPKEY=PA3,                                            X
               DYNSPC=NO,                                              X
               EDFLAG=73,                                              X
               EDEND=72,                                               X
               FILEVER=NO,                                             X
               HCLINE=132,                                             X
               INTCOMP=YES,                                            X
               INTRVAL=1,                                              X
               KATAKAN=NO,                                             X
               LOADPRT=YES,                                            X
               NBUFS=20,                                               X
               NRECS=22,                                               X
               NUSRS=30,                                               X
               NPARTS=5,                                               X
               NTASKS=4,                                               X
               PGMRINP=5,                                              X
               PGMRLST=6,                                              X
               PGMRPCH=7,                                              X
               PGMRPIN=8,                                              X
               PGMRLOG=9,                                              X
               PSIZE=256,                                              X
               PARTN=(1,1024,4,I,                                      X
               2,384,4,A,                                              X
               3,384,4,A,                                              X
               4,512,4,BA,                                             X
               5,512,4,BA),                                            X
               PARTX=,                                                 X
               RDR=FFC,                                                X
               RDR2=FFA,                                               X
               PCH=FFD,                                                X
               PRT=FFE,                                                X
               SPOOL=250,                                              X
               TIOA40=600,                                             X
               TIOA00=600,                                             X
               TCTOFS=8
         DTSIGEN
         END
/*
// EXEC LNKEDT,PARM='MSHP'
/&
* $$ EOJ
```

Figure 24. VSE/ICCF Generation (SKICFGEN Skeleton)

Note: The statement

```
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
```

calls the High Level Assembler. Refer to Chapter 6, “Migrating From Earlier Releases,” on page 91 for details about calling the High Level Assembler.

After making the changes, run the DTRSEXIT macro. This macro deletes specific comments from the skeleton. You should do this before you file the skeleton. On the command line, enter:

```
@DTRSEXIT
```

After the macro finishes, file the job. You can then submit it to the system for processing.

VSE/ICCF DTSFILE Generation Parameters

Pregenerated Libraries and User IDs

The VSE/ICCF DTSFILE generated by z/VSE defines 199 libraries and 199 VSE/ICCF user ID records. Table 24 on page 116 shows the VSE/ICCF libraries and their use. Note that some libraries are reserved for z/VSE. The members that z/VSE ships in these libraries take up approximately 20% of the space reserved for the DTSFILE.

Skeleton SKICFFMT

With the skeleton *SKICFFMT*, you can *reformat* the DTSFILE to create up to 99 user ID records and up to 9999 libraries. This skeleton, which is a member of VSE/ICCF library 59, has the original values that z/VSE specifies for the file. Figure 25 shows the FORMAT and ADD statements used for the DTSFILE. The ADD statements define the libraries shown in Table 24 on page 116. Under “Reformatting the VSE/ICCF DTSFILE”, the manual *z/VSE Administration* describes skeleton SKICCFMT in detail.

z/VSE requires that you define all VSE/ICCF libraries with the DATE option. For detailed information about the FORMAT and ADD statements, refer to “Formatting the Library File or Changing its Size” and “Adding or Changing a User Library” in the manual *VSE/ICCF Administration and Operation*.

Note: If necessary, you can extend the DTSFILE. You can do this by defining a larger extent on SYSWK1 or by defining extents on several volumes. For information on how to use the skeleton *SKDTSEXT* to extend the DTSFILE, refer to the manual *z/VSE Administration* under “Using Skeleton SKDTSEXT”.

```
FORMAT LIBRARIES(199) USERS(199)
*  ADD LIBRARY 1  . . .
ADD LIBRARY FREESPACE(40) DATE
*  ADD LIBRARY 2  . . .
ADD LIBRARY FREESPACE(10) DATE
*  ADD LIBRARIES 3,4,5, AND 6  . . .
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE NOCOMMON PUBLIC
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE NOCOMMON PUBLIC
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE NOCOMMON PUBLIC
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE NOCOMMON PUBLIC
*  ADD LIBRARIES 7 THRU 49  . . .
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
⋮
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
*  ADD LIBRARIES 50 THRU 68  . . .
ADD LIBRARY DATE NOCOMMON PUBLIC
ADD LIBRARY DATE NOCOMMON PUBLIC
⋮
ADD LIBRARY DATE NOCOMMON PUBLIC
ADD LIBRARY DATE NOCOMMON PUBLIC
*  ADD LIBRARIES 69 THROUGH 199
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
⋮
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
```

Figure 25. Code Example for Formatting the DTSFILE

Chapter 23. System Authorization Facility (SAF) and its External Security Interface (RACROUTE)

System Authorization Facility (SAF)

Programming Interface Information

The System Authorization Facility (SAF) of z/VSE provides centralized control for security processing through a system service called the SAF router. The resource manager components and subsystems (for example, the CICS Transaction Server for VSE/ESA) call the SAF router as part of the security decision-making functions in their processing, such as access control checking and authorization-related checking.

To use the SAF router, a resource management component or subsystem issues a RACROUTE call (macro) which invokes the SAF router. Once invoked, the SAF router first calls an optional installation exit routine and then the Basic Security Manager (BSM) or an External Security Manager (if one is installed and active on the system).

The SAF provided with z/VSE was ported from OS/390. Although z/VSE and OS/390 are based on the same S/390 platform, there are differences. The following section points out the z/VSE SAF deviations from the OS/390 SAF and its interfaces.

End of Programming Interface Information

z/VSE Related SAF Changes

Programming Interface Information

The external interfaces of the SAF are documented in the following OS/390 books:

- *Security Server External Security Interface (RACROUTE) Macro Reference*, GC28-1922
- *Security Server (RACF) Data Areas*, SY27-2640

These books are included as part of the z/VSE documentation on CD-ROM.

The RACROUTE macro is the interface to RACF (Resource Access Control Facility) or another security manager for OS/390 and z/VSE resource managers. The macro descriptions in the manual listed above define this interface. This does not imply, however, that both the OS/390 and z/VSE operating systems support all the functions allowed by the interface. Rather, it defines the macros and keywords that are available for OS/390 and z/VSE resource managers to implement security for data and other resources.

The main deviations of z/VSE from OS/390 are summarized below:

- The independent RACF system macros are not supported by z/VSE.
- The RACF SVCs are not part of the z/VSE RACROUTE support.
- No cross-memory mode is supported; use only HASN=PASN=SASN.

- z/VSE does not know APF libraries. The APF authorization is mapped to valid subsystems and z/VSE authorizations.
- The SAF router exit ICHRTX01 is not supported by z/VSE.
- The SAF support for OpenEdition MVS is not provided with z/VSE. The callable service router and its exit IRRSXT00 are also not part of the z/VSE SAF.
- The ACEE address fields TCBSENV and ASXBSENV are not available with z/VSE. For external security products the services

  MODFLD FIELD=ACEEPTR, and

  GETFLD FIELD=ACEEPTR

  are provided to support an equivalent behavior.
- IHAACEE has been changed to indicate z/VSE specific authorizations.
- To use the z/VSE SAF service, RELEASE=1.9 up to 2.2 has to be specified for the RACROUTE requests.

Note: It is strongly recommended that only those SAF related macros are used which are included in library PRD2.GEN1 if the Generation Feature has been installed.

End of Programming Interface Information

Planning to Use the SAF Router Exit ICHRTX00

Programming Interface Information

The SAF router exit ICHRTX00 can be used if a security exit is required. The exit has to be cataloged as member ICHRTX00.PHASE in system library IJSYSRS.SYSLIB.

For further details refer to the OS/390 books listed under “z/VSE Related SAF Changes” on page 287 and be aware of the z/VSE specific changes outlined in this chapter.

End of Programming Interface Information

Appendix A. Startup Procedures

List of IPL, JCL, and Label Procedures

This appendix lists the names of the IPL, JCL, and label procedures shipped with z/VSE in system library IJSYSRS.SYSLIB.

Table 41. IPL, JCL, and Label Procedures for Initial Installation

```
                IPL            JCL            Label
             Procedures:    Procedures:    Procedures:

               $IPLE80        $0JCL780       STDLAB80
               $IPLE90        $0JCL790       STDLAB90
               $IPLEGF        $0JCL7GF       STDLABGF

                              $1JCL780
                                  .
                                  .
                              $2JCL780
                                  .
                                  .
                              $3JCL780
                                  .
                                  .
                              $4JCL780
                                  .
                                  .
                              $5JCL780
                                  .
                                  .
                              $xJCL780
```

Note: z/VSE uses these procedures only once for initial installation.

The name of a procedure is determined by the disk device type used for DOSRES and SYSWK1. The following naming conventions apply for the procedures:

- The last two characters identify the type of disk device on which DOSRES and SYSWK1 reside. Under “z/VSE Disk Layouts”, the manual *z/VSE Installation* shows the layout of DOSRES and SYSWK1 for each disk device type supported for initial installation. Procedures are provided for initial installation for the following disk device types:

  ```
  80 = IBM 3380
  90 = IBM 3390
  GF = GFBA (Minidisks and SCSI)
  ```

  The characters GF indicate that this is the generalized layout for FBA disk devices (GFBA), which are used by minidisks under z/VM, and SCSI disks.
- In a running system, after installation, the procedures have been renamed as follows:
  – IPL procedure: $IPLESA
  – JCL procedures: $0JCL, $1JCL, $2JCL, and so on.
  – Label procedure: STDLABEL

  Procedure STDLABEL calls procedures STDLABUP and STDLABUS.

It might be useful to have printouts available of those procedures which are relevant for your system. Such a printout can help you identify the original contents of a procedure in case of a system startup problem because of user modifications. This should be done right after initial installation.

You can use the VSE/ICCF **LIBRP** command to copy a procedure from the system library IJSYSRS.SYSLIB to your primary VSE/ICCF library. From it, print the procedure using the *Program Development Library* dialog.

Example of an IPL Procedure

Figure 26 shows the statements of IPL procedure $IPLE90, required for the initial installation of a z/VSE system on an IBM 3390 disk device. After initial installation, z/VSE completes the procedure and renames it to $IPLESA.

```
01F $$A$SUPI,VSIZE=150M,VPOOL=64K,VIO=512K
ADD FDF,FBAV VIRTUAL DISK FOR LABEL AREA
ADD FEC,3505
ADD FFC,3505 ICCF DUMMY DEVICE DON'T DELETE
ADD FFA,3505 ICCF DUMMY DEVICE DON'T DELETE
ADD FED,2520B2
ADD FFD,2520B2 ICCF DUMMY DEVICE DON'T DELETE
ADD FEE,PRT1
ADD FEF,PRT1
ADD FFE,PRT1 ICCF DUMMY DEVICE DON'T DELETE
ADD FFF,CONS DEDICATED CONSOLE DON'T DELETE
DEF SYSCAT=DOSRES,SYSREC=SYSWK1
SYS DASDFP=YES
SYS JA=YES
SYS SPSIZE=0K
SYS NPARTS=40
SYS PASIZE=30M
DLA VOLID=DOSRES,CYL=60,NCYL=3,DSF=N,NAME=AREA1
DPD VOLID=DOSRES,CYL=398,NCYL=36,DSF=N
DPD VOLID=DOSRES,CYL=434,DSF=NO
SVA PSIZE=(652K,6M),SDL=700,GETVIS=(768K,6M)
```

Figure 26. Example of an IPL Procedure

$$A$SUPX is the name of the supervisor required for initial installation.

Appendix B. z/VSE Disk Layouts (DOSRES, SYSWK1)

The figures in this appendix show the space that is reserved on DOSRES and SYSWK1 for z/VSE.

Note that the UNUSED SPACE and END.VTOC may be **different**, depending on the DASD model you use. For information on the exact disk capacity of your DASD model, refer to the hardware description of your respective DASD model.

Please remember that it is recommended to place the VTOC in the middle of the disk device.

Note that MCAT means that this space belongs to the VSE/VSAM master catalog, UCAT that it belongs to the VSE/VSAM user catalog.

Layout Depending on Environment Selected

In the following tables, the space reserved for holding the page data sets is open ended. To avoid overlaps, especially with the page data sets, you should not put any files in the reserved space area.

If you select predefined environment B during initial installation, an **enlarged** page data set of 264 MB (compared to 150 MB for predefined environment A) is active. The UNUSED SPACE area will thus be reduced by 114 MB.

If you select predefined environment C during initial installation, an **enlarged** page data set of 1898 MB (compared to 150 MB for predefined environment A) is active. The UNUSED SPACE area will thus be reduced by 1748 MB.

The minimum number of blocks (as rounded values) for the three environments are:

	DOSRES	SYSWK1
Environment A	625,000	1,510,000
Environment B	860,000	1,510,000
Environment C	4,525,000	1,510,000

IBM 3380 Disks

DOSRES -------- IBM 3380 Disk

Table 42. DOSRES Layout for All Models Except Model A or Equivalent Disk Sizes

START TRACK	NUMBER TRACKS	START CYL	START HEAD	FILE IDENTIFICATION
1	959	0	1	VSE.SYSRES.LIBRARY
960	45	64	0	DOS.LABEL.FILE.CPUID.AREA1
1005	8	67	0	VSE.POWER.QUEUE.FILE
1010	7	67	8	U N U S E D S P A C E
1020	120	68	0	VSAM.MASTER.CATALOG
1140	2010	76	0	VSAM.DATA.SPACE.DOSRES(UCAT)
				%DOS.WORKFILE.SYSLNK
				VSE.CONTROL.FILE
				VSE.TEXT.REPSTORY.FILE
				VSE.ONLINE.PROB.DET.FILE
				VSE.MESSAGE.ROUTING.FILE
				VSE.PRIMARY.LIBRARY
				%DOS.WORKFILE.SYS001.RECOVER
				CICS.AUTO.STATS.A
				CICS.AUTO.STATS.B
				CICS.TD.INTRA
				DFHTEMP
				CICS.CSD
				CICS.RSD
				CICS.GCD
				CICS.LCD
				CICS.DBDCCICS.DFHDMFA
				CICS.DBDCCICS.DFHDMFB
				PTF.FILE
3150	11	210	0	RESERVED FOR VTOC (UNUSED)
3161	4	210	11	RECOMMEN.VTOC
3165	2725	211	0	VSAM.DATA.SPACE.DOSRES (MCAT)
				VSE.PRD1.LIBRARY
				VSE.PRD2.LIBRARY
				VSE.MESSAGES.ONLINE
5895	60	393	0	CICS.SYSTEM.LOG.A
5955	60	397	0	CICS.SYSTEM.LOG.B
6015	60	401	0	CICS.USER.JOURNAL.A
6075	60	405	0	CICS.USER.JOURNAL.B
6135	75	409	0	VSE.SYSTEM.HISTORY.FILE
6210	660	414	0	PAGING.DATA.SET.ONE
6870	3195	458	0	PAGING.DATA.SET.TWO (ENV A)
6870	6120	458	0	PAGING.DATA.SET.TWO (ENV B)
6870	51975	458	0	PAGING.DATA.SET.TWO (ENV C)
12990	13065	866	0	U N U S E D S P A C E
26055	60	868	0	CICSO.SYSTEM.LOG.A
26115	60	872	0	CICSO.SYSTEM.LOG.B
26175	60	876	0	CICSO.USER.JOURNAL.A
26235	60	880	0	CICSO.USER.JOURNAL.B
26295	60	885	0	CICS2.SYSTEM.LOG.A
26355	60	889	0	CICS2.SYSTEM.LOG.B
26415	60	893	0	CICS2.USER.JOURNAL.A
26475	60	897	0	CICS2.USER.JOURNAL.B
26535	15	1769	0	RESERVED FOR END.VTOC

Note: If you wish to use the environment C, you must use a large DASD. In addition, if you wish to keep the page dataset on the DOSRES, *for CICS 2.3 or a second CICS Transaction Server you must define your journaling files on another volume!* For details of how to define journaling files, see the skeletons SKJOUR2 and SKJOURO in ICCF library 59.

SYSWK1 -------- IBM 3380 Disk

Table 43. SYSWK1 Layout for All Models Except Model A or Equivalent Disk Sizes

START TRACK	NUMBER TRACKS	START CYL	START HEAD	FILE IDENTIFICATION
1	959	0	1	SYS.NEW.RES
960	75	64	0	WORK.HIST.FILE
1035	2715	69	0	VSAM.DATA.SPACE.SYSWK1 (MCAT)
3750	150	250	0	VSESP.USER.CATALOG
3900	975	260	0	VSAM.DATA.SPACE.SYSWK1 (UCAT)
				%DOS.WORKFILE.SYS001
				to
				%DOS.WORKFILE.SYS007
				%DOS.WORKFILE.SYS002.RECOVER
				%DOS.WORKFILE.SYS001.SORT
				%WORK.FILE.N11
				to
				%WORK.FILE.N54
				CICS.DUMPA
				CICS.DUMPB
				CICS.AUXTRACE
4875	11	325	0	RESERVED FOR VTOC (UNUSED)
4886	4	325	11	RECOMMEN.VTOC
4890	1800	326	0	ICCF.LIBRARY
6690	2025	446	0	VSE.POWER.DATA.FILE
8715	90	581	0	VSE.POWER.ACCOUNT.FILE
8805	15	587	0	VSESP.JOB.MANAGER.FILE
8820	75	588	0	VSE.HARDCOPY.FILE
8895	60	593	0	VSE.RECORDER.FILE
8955	75	597	0	VSE.ALT.HISTORY.FILE
9030	3	602	0	INFO.ANALYSIS.DUMP.MGNT.FILE
9033	12	602	3	U N U S E D S P A C E
9045	1	603	0	INFO.ANALYSIS.EXT.RTNS.FILE
9046	14	603	1	U N U S E D S P A C E
9060	30	604	0	VTAM.TRACE.FILE
9090	180	606	0	CU37XX.LOAD.FILE
9270	2820	618	0	VSAM.DATA.SPACE.SYSWK1 (MCAT)
12090	1170	806	0	VSE.DUMP.LIBRARY
13260	360	884	0	U N U S E D S P A C E
13620	2115	908	0	VSAM.DATA.SPACE.SYSWK1 (MCAT)
15735	10800	1049	0	U N U S E D S P A C E
26535	15	1769	0	RESERVED FOR END.VTOC

Note: There are several DASD models of this disk device. For all of them, the space requirement and file location for z/VSE are the same. Environments B and C include an enlarged **Page Data Set** (264 MB total size for environment B, and 2048 MB total size for environment C). UNUSED SPACE and END.VTOC depend on the model of the disk device you use and on the size of the Page Data Set. Please remember that it is recommended to place the VTOC in the middle of the disk device.

IBM 3390 Disks

DOSRES -------- IBM 3390 Disk

Table 44. DOSRES Layout for an IBM 3390-1 Disk

START TRACK	NUMBER TRACKS	START CYL	START HEAD	FILE IDENTIFICATION
1	899	0	1	VSE.SYSRES.LIBRARY
900	45	60	0	DOS.LABEL.FILE.CPUID.AREA1
945	7	63	0	VSE.POWER.QUEUE.FILE
952	8	63	7	U N U S E D S P A C E
960	120	64	0	VSAM.MASTER.CATALOG
1080	2040	72	0	VSAM.DATA.SPACE DOSRES (UCAT) %DOS.WORKFILE.SYSLNK VSE.CONTROL.FILE VSE.TEXT.REPSTORY.FILE VSE.ONLINE.PROB.DET.FILE VSE.MESSAGE.ROUTING.FILE VSE.PRIMARY.LIBRARY %DOS.WORKFILE.SYS001.RECOVER CICS.AUTO.STATS.A CICS.AUTO.STATS.B CICS.TD.INTRA DFHTEMP CICS.CSD CICS.RSD CICS.GCD CICS.LCD CICS.DBDCCICS.DFHDMFA CICS.DBDCCICS.DFHDMFB PTF.FILE
3120	11	208	0	RESERVED FOR VTOC (UNUSED)
3131	4	208	11	RECOMMEN.VTOC
3135	2520	209	0	VSAM.DATA.SPACE.DOSRES (MCAT) VSE.PRD1.LIBRARY VSE.PRD2.LIBRARY VSE.MESSAGES.ONLINE
5655	60	377	0	CICS.SYSTEM.LOG.A
5715	60	381	0	CICS.SYSTEM.LOG.B
5775	60	385	0	CICS.USER.JOURNAL.A
5835	60	389	0	CICS.USER.JOURNAL.B
5895	75	393	0	VSE.SYSTEM.HISTORY.FILE
5970	540	398	0	PAGING.DATA.SET.ONE
6510	2685	434	0	PAGING.DATA.SET.TWO (ENV A)
6510	5115	434	0	PAGING.DATA.SET.TWO (ENV B)
6510	43500	434	0	PAGING.DATA.SET.TWO (ENV C)
11625	4575	775	0	U N U S E D S P A C E
16200	60	1080	0	CICS2.SYSTEM.LOG.A
16260	60	1084	0	CICS2.SYSTEM.LOG.B
16320	60	1088	0	CICS2.USER.JOURNAL.A
16380	60	1092	0	CICS2.USER.JOURNAL.B
16440	60	1096	0	CICSO.SYSTEM.LOG.A
16500	60	1100	0	CICSO.SYSTEM.LOG.B
16560	60	1104	0	CICSO.USER.JOURNAL.A
16620	60	1108	0	CICSO.USER.JOURNAL.B
16680	15	1112	0	RESERVED FOR END.VTOC

Note: If you wish to use the environment C, you must use a large DASD. In addition, if you wish to keep the page dataset on the DOSRES, *for CICS 2.3 or a second CICS Transaction Server you must define your journaling files on another volume!* For details of how to define journaling files, see the skeletons SKJOUR2 and SKJOURO in ICCF library 59.

SYSWK1 -------- IBM 3390 Disk

Table 45. SYSWK1 Layout for an IBM 3390-1 Disk

START TRACK	NUMBER TRACKS	START CYL	START HEAD	FILE IDENTIFICATION
1	899	0	1	SYS.NEW.RES
900	75	60	0	WORK.HIST.FILE
975	2550	65	0	VSAM.DATA.SPACE.SYSWK1 (MCAT)
3525	150	235	0	VSESP.USER.CATALOG
3675	840	245	0	VSAM.DATA.SPACE.SYSWK1 (UCAT) %DOS.WORKFILE.SYS001 to %DOS.WORKFILE.SYS007 %DOS.WORKFILE.SYS002.RECOVER %DOS.WORKFILE.SYS001.SORT %WORK.FILE.N11 to %WORK.FILE.N54 CICS.DUMPA CICS.DUMPB CICS.AUXTRACE
4515	11	301	0	RESERVED FOR VTOC (UNUSED)
4526	4	301	11	RECOMMEN.VTOC
4530	1800	302	0	ICCF.LIBRARY
6330	1920	422	0	VSE.POWER.DATA.FILE
8250	90	550	0	VSE.POWER.ACCOUTN.FILE
8340	15	556	0	VSESP.JOB.MANAGER.FILE
8355	60	557	0	VSE.HARDCOPY.FILE
8415	60	561	0	VSE.RECORDER.FILE
8475	75	565	0	VSE.ALT.HISTORY.FILE
8550	3	570	0	INFO.ANALYSIS.DUMP.MGNT.FILE
8553	12	570	3	U N U S E D S P A C E
8565	1	571	0	INFO.ANALYSIS.EXT.RTNS.FILE
8566	14	571	1	U N U S E D S P A C E
8580	30	572	0	VTAM.TRACE.FILE
8610	180	574	0	CU37XX.LOAD.FILE
8790	3960	586	0	VSAM.DATA.SPACE.SYSWK1 (MCAT)
12750	1425	850	0	VSE.DUMP.LIBRARY
14175	525	945	0	U N U S E D S P A C E
14700	1980	980	0	VSAM.DATA.SPACE.SYSWK1 (MCAT)
16680	15	1112	0	RESERVED FOR END.VTOC

Note: There are several DASD models of this disk device. For all of them, the space requirement and file location for z/VSE are the same. Environments B and C include an enlarged **Page Data Set** (264 MB total size for environment B, and 2048 MB total size for environment C). UNUSED SPACE and END.VTOC depend on the model of the disk device you use and on the size of the Page Data Set. Please remember that it is recommended to place the VTOC in the middle of the disk device.

IBM FBA (and SCSI) Disks

Note!
This layout is also used for FBA devices defined as VM virtual disks. These VM virtual disks are used under z/VM.

DOSRES -------- IBM FBA (and SCSI) Disks

Table 46. DOSRES Layout for an FBA (and SCSI) Disk

START BLOCK	NUMBER of BLOCKS	FILE IDENTIFICATION
2	59390	VSE.SYSRES.LIBRARY
59392	512	DOS.LABEL.FILE.CPUID.AREA1
59904	1024	VSE.POWER.QUEUE.FILE
60928	3072	VSAM.MASTER.CATALOG
64000	61440	VSAM.DATA.SPACE.DOSRES (UCAT)
		%DOS.WORKFILE.SYSLNK
		VSE.CONTROL.FILE
		VSE.TEXT.REPSTORY.FILE
		VSE.ONLINE.PROB.DET.FILE
		VSE.MESSAGE.ROUTING.FILE
		VSE.PRIMARY.LIBRARY
		%DOS.WORKFILE.SYS001.RECOVER
		CICS.AUTO.STATS.A
		CICS.AUTO.STATS.B
		CICS.TD.INTRA
		DFHTEMP
		CICS.CSD
		CICS.RSD
		CICS.GCD
		CICS.LCD
		CICS.DBDCCICS.DFHDMFA
		CICS.DBDCCICS.DFHDMFB
		PTF.FILE
125440	64	VTOC
125504	448	UNUSED SPACE
125952	172032	VSAM.DATA.SPACE.DOSRES (MCAT)
		VSE.PRD1.LIBRARY
		VSE.PRD2.LIBRARY
		VSE.MESSAGES.ONLINE
297984	3072	CICS.SYSTEM.LOG.A
301056	3072	CICS.SYSTEM.LOG.B
304128	3072	CICS.USER.JOURNAL.A
307200	3072	CICS.USER.JOURNAL.B
310272	5120	VSE.SYSTEM.HISTORY.FILE
315392	50176	PAGING.DATA.SET.ONE
365568	258048	PAGING.DATA.SET.TWO (ENV A)
365568	492544	PAGING.DATA.SET.TWO (ENV B)
365568	4158464	PAGING.DATA.SET.TWO (ENV C)
623616		U N U S E D S P A C E (ENV A)
858112		U N U S E D S P A C E (ENV B)
4524032		U N U S E D S P A C E (ENV C)

SYSWK1 --------- IBM FBA (and SCSI) Disks

Table 47. SYSWK1 Layout for an FBA (and SCSI) Disk

START BLOCK	NUMBER of BLOCKS	FILE IDENTIFICATION
2	59390	SYS.NEW.RES
59392	5120	WORK.HIST.FILE
64512	168960	VSAM.DATA.SPACE.SYSWK1 (MCAT)
233472	3072	VSESP.USER.CATALOG
236544	69632	VSAM.DATA.SPACE.SYSWK1 (UCAT)
		%DOS.WORKFILE.SYS001
		to
		%DOS.WORKFILE.SYS007
		%DOS.WORKFILE.SYS002.RECOVER
		%DOS.WORKFILE.SYS001.SORT
		%WORK.FILE.N11
		to
		%WORK.FILE.N54
		CICS.DUMPA
		CICS.DUMPB
		CICS.AUXTRACE
306176	64	VTOC
306240	448	UNUSED SPACE
306688	69632	ICCF.LIBRARY
376320	89088	VSE.POWER.DATA.FILE
465408	2048	VSE.POWER.ACCOUNT.FILE
467456	128	VSESP.JOB.MANAGER.FILE
467584	5248	VSE.HARDCOPY.FILE
472832	1024	VSE.RECORDER.FILE
473856	5120	VSE.ALT.HIST.FILE
478976	192	INFO.ANALYSIS.DUMP.MGNT.FILE
479168	64	INFO.ANALYSIS.EXT.RTNS.FILE
479232	1024	VTAM.TRACE.FILE
480256	9216	CU37XX.LOAD.FILE
489472	468992	VSAM.DATA.SPACE.SYSWK1 (MCAT)
958464	139264	VSAM.DATA.SPACE.SYSWK1 (UCAT)
1097728	409600	VSE.DUMP.LIBRARY
1507328		U N U S E D S P A C E

Note: There are several DASD models of this disk device. For all models of this disk device the space requirement and file location for z/VSE are the same. Environments B and C include an enlarged **Page Data Set** (264 MB total size for environment B, and 2048 MB total size for environment C). UNUSED SPACE and END.VTOC depend on the model of the disk device you use and on the size of the Page Data Set. Please remember that it is recommended to place the VTOC in the middle of the disk device.

Appendix C. Dialogs of the Interactive Interface

This appendix lists the dialogs that are available from selection panels of the Interactive Interface. If you wish, you can invoke these dialogs from selection panels that you create and add to the Interactive Interface.

In Table 48 on page 300:

- **z/VSE Selection** is the name that is used for a dialog in a selection panel of the Interactive Interface.
- **Default for** shows where the dialog appears in the default panel hierarchies:
  - **A** = hierarchy for **SYSA** (type 1 user)
  - **O** = hierarchy for **OPER** (type 2 user)
  - **P** = hierarchy for **PROG** (type 2 user)
  - **S** = hierarchy for **$SRV** (type 2 user)

  A, O, and P identify the default panel hierarchies available for a system administrator (SYSA), an operator (OPER), and a programmer (PROG). S, which stands for service ($SRV), identifies a default panel hierarchy which provides access to a selected set of standard dialogs for problem determination. This panel hierarchy is mainly intended for IBM personnel doing remote problem determination for a user site via a data link connecting the user installation with an IBM Support Center, for example. But the $SRV panel hierarchy can also be used for local problem determination.
- **Application Profile** is the name you would specify for the dialog when creating your own selection panels.
- **VSE/ICCF Partition** shows if a particular dialog uses a VSE/ICCF interactive partition. If so, the requirements for the partition's class (A, B, and I) and size are shown. If a dialog does not require a VSE/ICCF interactive partition, all user types (1, 2, or 3) can access it. If a dialog requires a partition, both type 1 and type 2 users can access it, unless otherwise indicated. Type 3 users (general users) cannot access the dialog.
- **Concurrent Execution** indicates how many users can execute a dialog at the same time. For example:
  - Any number of users can concurrently execute the *Command Mode* dialog.
  - Only one user at a time can execute the *Configure Hardware* dialog.
  - Two users can concurrently execute the *Back Up a File* dialog.

    Note that the number of allowed concurrent users is based on the initial partition layout for VSE/ICCF. You can increase the number of class A or B partitions to allow concurrent access for more users.

Table 48. Dialogs of the Interactive Interface

z/VSE Selection	Default for	Application Profile	VSE/ICCF Partition	Concurrent Execution
Analyze and Apply PTFs	A	IESS$PHT	I, 1024K	1 user only
Apply PTFs	A	IESS$PHJ	I, 1024K	1 user only
Archive All ICCF Libraries on Tape	A, P, O	IESS$SAR	B, 512K	2 users
Back Up a File	A, P, O	IESU$DMF	B, 512K	2 users
Back Up a User Catalog to Disk	A, O	IESC$BUD	B, 512K	2 users
Back Up a User Catalog to Tape	A, O	IESC$BUT	B, 512K	2 users
Back Up a Volume	A, P, O	IESU$DMV	B, 512K	2 users
Back Up History File	A, O	IESS$BAC	B, 512K	2 users
Back Up the DTSFILE (all ICCF Libraries)	A, P, O	IESS$SAL	B, 512K	2 users
Back Up the Master Catalog to Disk	A, O	IESC$BMD	B, 512K	2 users
Back Up the Master Catalog to Tape	A, O	IESC$BMT	B, 512K	2 users
Back Up VSAM File	A, P, O	IESD$BAC	B, 512K	2 users
Back Up VSE Library on Tape	A, O	IESS$LSV	B, 512K	2 users
Catalog Printer UCB	A	IEST$UCB	B, 512K	2 users
Change Nicknames	A	IESS$CNN	I, 1024K	1 user only
Command Mode (use ICCF in native mode)	A, P	IESNICCF		Any number of users
Configure Hardware	A	IESA$HDW	I, 1024K	1 user only
Console (see Note 6)	A, P, O, S	IESDC		Any number of users
Copy a File	A, P, O	IESU$FCF	B, 512K	2 users
Copy a Volume	A, P, O	IESU$FCV	B, 512K	2 users
Create Application Job Stream	A, P	IESP$EXE	B, 512K	2 users
Create Network Tape	A	IESC$CNT	B, 512K	2 users
Create Stand-Alone Dump Program on Tape	A, P, S	IESP$AIA	B, 512K	2 users
Create Stand-Alone Dump Program on Disk	A, P, S	IESP$AIK	B, 512K	2 users
Customize z/VSE Workstation Platform	A	IESWPM		1 user only
Define Transaction Security	A	IESC$SEC	I, 1024K	1 user only
Display Active Users/Send Message (see Note 6)	A, P, O, S	IESUSER		Any number of users
Display Channel and Device Activity	A, O	IESDS		Any number of users
Display CICS TS Storage Layout	A, O	IESDSA		Any number of users
Display Storage Layout	A, O	IESSTRGL		Any number of users
Display System Activity (see Note 6)	A, P, O	IESDA		Any number of users
Display VTOC	A	IESLVTOC	B, 512K	Any number of users
Down-Level Check (system refresh installation)	A	IESS$FSD	I, 1024K (see Note 1)	1 user only

Table 48. Dialogs of the Interactive Interface (continued)

z/VSE Selection	Default for	Application Profile	VSE/ICCF Partition	Concurrent Execution
Enter News	A, P, O	IESNEWS		1 user only
Export-Disconnect a User Catalog	A, O	IESC$EXP	B, 512K	2 users
Export ICCF Library Members to Tape	A, P, O	IESS$EXP	B, 512K	2 users
Export VSAM File	A, P, O	IESD$EXP	B, 512K	2 users
File and Catalog Management (see Note 2)	A, P	IESVSAM	B, 512K (see Note 1)	Any number of users
FlashCopy VSAM Catalog/Files	A, O	IESC$SNP	I, 1024K	1 user only
Format ICCF Dump Data	A, P, S	IESP$FDM	B, 512K	2 users
FSU Installation (system refresh installation)	A	IESS$FSI	I, 1024K	1 user only
FSU Preparation	A	IESS$FSR	I, 1024K	1 user only
Import-Connect a User Catalog	A, O	IESC$IMP	B, 512K	2 users
Import ICCF Library Member	A, P, O	IESS$IMP	B, 512K	2 users
Import VSAM File	A, P, O	IESD$IMP	B, 512K	2 users
Inspect Dump Management Output	A, P, S	IESDUMPV		Any number of users
Inspect Message Log	A, P, S	IESMLOG		Any number of users
Install Generation Feature	A	IESI$GLI	I, 1024K	1 user only
Install Network Tape	A	IESC$NET	B, 512K	2 users
Install Program(s) from Tape	A	IESI$OPI	I, 1024K	1 user only
Install Programs - V1 Format	A	IESI$ODI	I, 1024K	1 user only
Invoke CEDA	A	IESCEDA		Any number of users
Invoke CEMS (Report Controller – administrator functions)	A, O	IESCEMS		Any number of users
Invoke CEOS (Report Controller – programmer subset)	P	IESCEOS		Any number of users
Invoke CEMT	A, P, O	IESCEMT		Any number of users
List and Process User Files in Host Transfer File	A	IESIWS0		Any number of users
Look Up PTF/APAR (see Note 3)	A, P, S	IESS$RLP	I, 1024K	1 user only
Maintain Application Profiles (see Note 4)	A	IESAPM/M2		1 user only
Maintain Certificate - User ID List	A	IESC$AUT	I, 1024K	1 user only
Maintain Dynamic Partitions	A	IESC$DYN	I, 1024K	1 user only
Maintain Printer FCB	A	IESU$FCB	I, 1024K	1 user only
Maintain PRIMARY Sublibraries	A	IESSTM		1 user only
Maintain Selection Panels (see Note 4)	A	IESSPM/M2		1 user only
Maintain Synonyms	A, P, O	IESSYN		Any number of users

Table 48. Dialogs of the Interactive Interface (continued)

z/VSE Selection	Default for	Application Profile	VSE/ICCF Partition	Concurrent Execution
Maintain User Profiles (see Note 4)	A	IESUPM	I, 1024K	1 user only
Maintain VTAM Application Names	A	IESC$APP	I, 1024K	1 user only
Maintain VTAM Startup Options	A	IESC$ACT	I, 1024K	1 user only
Maintain Workstation Platform	A	IESWPM		Any number of users
Manage Batch Queues	A, P, O, S	IESBQU		Any number of users
Move Files from Host Transfer File to ICCF	A	IESIWS5		Any number of users
Move Files from Host Transfer File to VSAM	A	IESIWS3		Any number of users
Move ICCF Members to Host Transfer File	A	IESIWS4		Any number of users
Move VSAM Files to Host Transfer File	A	IESIWS2		Any number of users
Online Problem Determination	A, P, S	IESOLPD		Any number of users
PC File Transfer	A	IESIWS7		Any number of users
Personalize History File	A	IESA$LB	I, 1024K	1 user only
Prepare for Installation	A	IESI$PRI	I, 1024K	1 user only
Print IPL Diagnostics	A, P, S	IESP$AII	B, 512K	2 users
Print SDAID Tape	A, P, S	IESP$AID	B, 512K	2 users
Print Service Document	A	IESS$LST	I, 1024K	1 user only
Program Development Library (see Note 5)	A, P, O	IESLIBXX	A, 384K	4 users
Remove PTF Records from History File	A	IESS$PHR	I, 1024K	1 user only
Remove Standalone Dump Program from Disk	A, P, S	IESP$AIR	B, 512K	2 users
Restore a File	A, P, O	IESU$RSF	B, 512K	2 users
Restore a Member of an ICCF Library	A, P, O	IESS$RME	B, 512K	2 users
Restore a User Catalog from Disk	A, O	IESC$RUD	B, 512K	2 users
Restore a User Catalog from Tape	A, O	IESC$RUT	B, 512K	2 users
Restore a Volume	A, P, O	IESU$RSV	B, 512K	2 users
Restore History File	A, O	IESS$RHS	B, 512K	2 users
Restore One ICCF Library	A, P, O	IESS$RON	B, 512K	2 users
Restore the DTSFILE (all ICCF Libraries)	A, P, O	IESS$RAL	B, 512K	2 users
Restore the Master Catalog from Disk	A, O	IESC$RMD	B, 512K	2 users
Restore the Master Catalog from Tape	A, O	IESC$RMT	B, 512K	2 users
Restore VSAM File	A, P, O	IESD$RES	B, 512K	2 users
Restore VSE Library from Tape	A, O	IESS$LRS	B, 512K	2 users

Table 48. Dialogs of the Interactive Interface (continued)

z/VSE Selection	Default for	Application Profile	VSE/ICCF Partition	Concurrent Execution
Retrace (retrace entire System History File, see Note 3)	A, P, S	IESS$RRP	I, 1024K	1 user only
Retrace APARs	A, P, S	IESS$RAP	I, 1024K	1 user only
Retrace Component ID	A, P, S	IESS$ROP	I, 1024K	1 user only
Retrace Components	A, P, S	IESS$RCP	I, 1024K	1 user only
Retrace Members	A, P, S	IESS$RMP	I, 1024K	1 user only
Retrace Products	A, P, S	IESS$RPP	I, 1024K	1 user only
Retrace PTFs	A, P, S	IESS$RFP	I, 1024K	1 user only
Retrieve Files from Another System	A, P	IESFR	A, 384K	4 users
Retrieve Message	A, P, O, S	IESIMSG		Any number of users
Scan Dump Files on Tape	A, P, S	IESP$SCT	B, 512K	2 users
Scan Dump Files on Disk	A, P, S	IESP$SCD	B, 512K	2 users
Scan VSE Library Backup Tape	A, O	IESS$SRS	B, 512K	2 users
Storage Dump Management	A, P, S	IESP$IDH	I, 1024K	1 user only
Submit a Job to Another System	A, P	IESSS	A, 384K	4 users
Tailor IPL Procedure	A	IEST$MAS	I, 1024K	1 user only
TCP/IP Configuration	A	IESC$TCP	I, 1024K	1 user only
Transfer Files and Jobs to Another System	A, P	IESFT	A, 384K	4 users
Verify Location of Involved Serviced Files	A	IESS$LOC	I, 1024K	1 user only

Notes:

1. CICS builds and uses a temporary VSE/ICCF partition for this dialog.
2. If a user has authority to define VSE/VSAM files and manage catalogs, the application IESVSAM provides access to 6 functions:
   a. ***Display or Process a File***
   b. ***Define a New File***
   c. ***Define a Library***
   d. ***Define an Alternate Index or Name***
   e. ***Display or Process a Catalog, Space***
   f. ***Define a New User Catalog***

   The last two functions are not shown to a user who has no authority to manage catalogs. Selections 2 through 4 are not accessible to users with no authority to define VSE/VSAM files.
3. The ***Look Up PTF/APAR*** and ***Retrace*** dialogs listed in this appendix are accessed via the ***Retrace History File*** panel.

   The panel hierarchy for an administrator also allows access to the dialogs via the ***IBM Service*** panel. These dialogs are identical in function to those accessed from the ***Retrace History File*** panel. The only difference is that their application names end in ***S***, not ***P*** (IESS$RAS instead of IESS$RAP, for example).

   If you wish to include an option for a ***Retrace*** dialog in a selection panel that you create, you can specify either application name for the dialog.

4. To work with this dialog, a user must have special authorization to be defined in the user profile.
5. The following functions of the ***Program Development Library*** dialog use a VSE/ICCF interactive partition: SUBMIT, PRINT, and COMPILE.
6. These dialogs have no timeout set due to conversational programming.

Appendix D. Additional z/VSE Applications

z/VSE provides a number of applications that are not included as options in selection panels of the Interactive Interface. If you wish, you can invoke one or more of these applications from selection panels that you create.

In Table 49 on page 306:

- **Function** lists the function provided by an application.
- **Application Profile** is the name you would specify when adding an application to your own selection panels.
- **VSE/ICCF Partition** shows if a particular dialog uses a VSE/ICCF interactive partition. If so, the requirements for the partition's class and size are shown.

  If a dialog does not require a VSE/ICCF interactive partition, all user profile types (1, 2, or 3) can access it.

  If a dialog requires a partition, both type 1 and type 2 users can access it, unless otherwise indicated. Type 3 users cannot access the dialog.
- **Concurrent Execution** indicates how many users can execute an application at the same time.

  Note that the number of concurrent users shown for these applications is based on the initial partition layout for VSE/ICCF. You can increase the number of class B partitions to allow concurrent access for more users.

If you use an application which requires an optional program:

- The optional program must be installed.
- If the optional program has a user ID and password for sign-on, you should define the Interactive Interface user ID and password to the optional program.

  For example, if USER1 is the Interactive Interface user ID which uses the SDF/CICS application, USER1 and its password must be known to SDF/CICS.

Additional z/VSE Applications

Table 49. Additional z/VSE Applications

Function	Application Profile	VSE/ICCF Partition	Concurrent Execution
CEMT Perform Shutdown	IESCMT01		Any number of users
CEMT Perform Shutdown Immediate	IESCMT02		Any number of users
CEMT Perform Shutdown Immediate Dump	IESCMT03		Any number of users
Display Channel and Device Activity (see Note 1)	IESLS		Any number of users
Display LST Queue	IESLST		Any number of users
Display PUN Queue	IESPUN		Any number of users
Display RDR Queue	IESRDR		Any number of users
Display System Activity (see Note 1)	IESLA		Any number of users
Display XMT Queue	IESXMT		Any number of users
Interactive Structured Query Language (ISQL)	IESISQL		Any number of users
Shut Down ISQL Resource Manager	IESSQLT		Any number of users
Start ISQL Resource Manager	IESSQLS		Any number of users
Invoke CEDB (Resource Definition Online)	IESCEDB		Any number of users
Invoke CEDC (Resource Definition Online)	IESCEDC		Any number of users
Invoke DITTO/ESA for VSE (see Note 2)	IESDITTO		Any number of users
Invoke Librarian in an Interactive Partition	IESLIBR	B, 512K	2 users
Query Management Facility (QMF)	IESQMF		Any number of users
QMF / Trace	IESQMFT		Any number of users
Screen Definition Facility (SDF/CICS)	IESSDF		Any number of users

Notes:

1. When creating your own selection panels, specify IESLS or IESLA for the application name if you want to change the system defaults for refreshing the screen. If you do not want to change the defaults, use IES**D**S or IES**D**A instead. The z/VSE dialogs are listed in Table 48 on page 300.
2. This dialog has no timeout set due to conversational programming.

Appendix E. Reserved Names of the Interactive Interface

z/VSE uses certain prefixes when assigning names for the following resources:

- VSE libraries, sublibraries, and members
- VSE/ICCF
  - Library members
  - Files (in DLBL statements)
- CICS
  - Programs
  - Maps and map sets
  - Transaction IDs
  - Temporary storage queue IDs
  - Transient data queues
  - Files
- Selection panels
- Application profiles
- Message prefixes

When you name any of these items for your own use, you should not use the reserved prefixes listed below to avoid duplicate names.

The following prefixes are reserved:

```
I$
IE
INW
DTR
DTS
VSE
```

In addition, all names with $ in the second or fourth position are reserved:

```
X$X$XXX
```

CICS uses the following transaction IDs:

```
HELP
OLPD
USER
ICCF
PF1
PF3
PF13
PF15
```

Appendix F. z/VSE Skeletons and REXX/VSE Procedures

As the name implies, skeletons are usually incomplete jobs that you cannot immediately submit for execution. Instead, they contain parameters and variables that you must specify to create jobs for your specific system. They also contain comments that describe JCL statements and variables and explain how you make the necessary changes.

To use a skeleton, you should copy it from its original VSE/ICCF library into a VSE/ICCF library that you own. You can then edit the copy and make the necessary changes.

This appendix also includes"REXX/VSE Procedures" on page 317.

Startup Skeletons

Table 50. Startup Skeletons

Member Name	Library	Use
SKALLOCA	59	Allocate static partitions for Environment A.
SKALLOCB	59	Allocate static partitions for Environment B.
SKALLOCC	59	Allocate static partitions for Environment C.
SKCOLD	59	Add jobs to the load list for the RDR queue for a COLD startup.
SKCOMVAR	59	Define CPU IDs in a DASD-sharing environment.
SKENVOVW	59	Overview of procedures for tailoring system startup.
SKENVSEL	59	Include jobs created by startup skeletons into a single job.
SKJCL0	59	Start BG partition.
SKBGSTRT	59	Startup for BG.
SKJCL1-SKJCLB	59	Start partitions F1 through FB.
SKJCLDYN	59	Start dynamic partition (dynamic partition profile).
SKLIBCHN	59	Define library search chains.
SKLOAD	59	Create job for loading jobs into VSE/POWER reader queue.
SKPWSTRT	59	Startup job for VSE/POWER.
SKTCPSTR	59	Startup job for TCP/IP.
SKUSERBG	59	Release jobs CICSICCF and VTAMSTRT.
SKVCSSTJ	59	Startup job for Connector Server.
SKVTAM	59	Startup job for VTAM.
SKVTASTJ	59	Startup job for Virtual Tape Data Handler.

Skeletons for e-business Connectors Support

Table 51. Skeletons for e-business Connectors Support

Member Name	Library	Use
SKCPSTP	59	Compile and link sample stored procedure.

Table 51. Skeletons for e-business Connectors Support (continued)

Member Name	Library	Use
SKCRESTP	59	Create sample stored procedure DB2 connector.
SKDB2SPS	59	Catalog stored procedure server job.
SKDB2STR	59	DB2 server startup job.
SKDB2VAR	59	Define and initialize the DB2 sample database.
SKDLICMP	59	Compile sample stored procedure for DL/I access.
SKDLISMP	59	Define and initialize DL/I sample database.
SKDLISTP	59	Create sample stored procedures to access DL/I data.
SKRDCFG	59	Set up VSE/VSAM Redirector.
SKVCSCAT	59	Connector Server, catalog VCS configuration members.
SKVCSCFG	59	Connector Server, main configuration member.
SKVCSLIB	59	Connector Server, library configuration member.
SKVCSPLG	59	Connector Server, plugin configuration member.
SKVCSSSL	59	Connector Server, SSL configuration member.
SKVCSSTJ	59	Connector Server, startup job.
SKVCSUSR	59	Connector Server, user security configuration member.
SKVSSAMP	59	Define VSAM connector sample data.

Skeletons for CICS Transaction Server

Refer also to Table 32 on page 196; it shows the major skeletons and tables for a CICS environment.

Table 52. Skeletons for CICS Transaction Server

Member Name	Library	Use
DFHAUXPR	59	Print CICS Transaction Server AUXTRACE file.
DFHCNV	59	Conversion table for CICS Web Support.
DFHDCTSP	59	Sample destination control table.
DFHDMFSP	59	Sample monitoring/statistics table.
DFHFCTSP	59	Sample file control table.
DFHJCTSP	59	Sample journal control table.
DFHMNDUP	59	Monitoring support for CICS Transaction Server.
DFHMOLS	59	Monitoring support for CICS Transaction Server.
DFHPLTPI	59	Sample program list table (startup).
DFHPLTSD	59	Sample program list table (shutdown).
DFHSITSP	59	Sample system initialization table.
DFHTSTSP	59	Sample temporary storage table.
DFHWBEP	59	Sample exit for CWS (CICS Web Support).
DFHXLTSP	59	Sample transaction list table.
DFH0STAT	59	Sample program print statistics.
IESZATDX	59	Autoinstall exit for CICS Transaction Server.
IESZNEP	59	Node error program for CICS Transaction Server.

Table 52. Skeletons for CICS Transaction Server (continued)

Member Name	Library	Use
IESZNEPS	59	Sample error program for CICS Transaction Server.
IESZNEPX	59	Sample node error program for CICS Transaction Server.
SKCICMD	59	Sample program issue GETVISxx,RESET
SKCICS	59	Startup job for CICS Transaction Server with IUI and ICCF.
SKCICSLI	59	Define CICS Listener file.
SKCICCLD	59	CICS cold start, clearing LCD and GCD.
SKCIDTRA	59	Print internal trace from dump in dump library.
SKCITMP	59	Redefine and/or increase CICS/VSE temporary storage (DFHTEMP) or correct it if corrupted.
SKCONSII	59	Define console for Interactive Interface.
SKCSDFC2	59	Update CSD file for secondary CICS Transaction Server.
SKCICS2	59	Startup job for a secondary CICS Transaction Server in partition F8.
SKPREPC2	59	Preparation job for a secondary CICS Transaction Server in partition F8.
DFHJCTC2	59	Sample Journal Control Table for a secondary CICS Transaction Server.
SKCSDFIL	59	Perform an upgrade of the CICS CSD file.
SKDFHDUP	59	Print transaction dump of CICS Transaction Server.
SKDMFPR	59	Print statistics for CICS Transaction Server.
SKDMFST	59	Startup monitoring partition.
SKEXCIAS	59	HLASM translation sample for batch client.
SKEXCICN	59	C translation sample for batch client.
SKEXCICV	59	COBOL translation sample for batch client.
SKEXCIPV	59	PL/I translation sample for batch client.
SKEXITDA	59	CICS/VSE sample user exit program for saving activity data.
SKEXIT1	59	Sample sign-on exit.
SKGCDFIL	59	Global catalog redefinition.
SKJOURN	59	Define/format journal data sets.
SKJOUR2	59	Define/format journal data sets for second CICS Transaction Server.
SKLCDFIL	59	Local catalog redefinition.

Skeletons for CICS/VSE

Refer also to Table 32 on page 196; it shows the major skeletons and tables for a CICS environment.

Table 53. Skeletons for CICS/VSE

Member Name	Library	Use
DFHAUXOL	59	Print CICS/VSE AUXTRACE file.
DFHDCTCO	59	Sample destination control table.
DFHFCTCO	59	Sample file control table.
DFHJCTCO	59	Sample journal control table.
DFHPCTCO	59	Sample program control table for (CICS coexistence environment).

Table 53. Skeletons for CICS/VSE (continued)

Member Name	Library	Use
DFHPCTSO	59	Sample program control table (a second CICS).
DFHPLTPO	59	Sample program list table (startup).
DFHPLTSO	59	Sample program list table (shutdown).
DFHPPTCO	59	Sample processing program table.
DFHPPTSO	59	Sample processing program table.
DFHSITCO	59	Sample system initialization table.
DFHSNTCO	59	Sample sign-on table.
DFHTCTCO	59	Sample terminal control table.
DFHTSTCO	59	Sample temporary storage table.
IESZATCO	59	Autoinstall exit for CICS/VSE.
IESZNEPO	59	Node error program for CICS/VSE.
SKCICSOL	59	Startup skeleton.
SKCIDUMP	59	Print formatted CICS/VSE dump.
SKJOURO	59	Define/format journal data sets.
SKPREPCO	59	Resource definition skeleton.
SKPREPSO	59	Resource definition skeleton.
SKSECTXS	59	Migrate TRANSEC values to CICS Transaction Server from program control table (PCT).
SKSECTX2	59	Migrate TRANSEC values to CICS Transaction Server from migration dataset.
SKSECTX3	59	Migrate TRANSEC values to CICS Transaction Server from DFHCSDUP definition.

Skeletons for Job Accounting

Table 54. Skeletons for Job Accounting

Member Name	Library	Use
SKJADACC	59	Extended job accounting.
SKJADOFF	59	Extended job accounting offload.
SKJADPRT	59	Extended job accounting reporting.
SKJOBACC	59	Standard job accounting.
SKJOBDMF	59	Extended job accounting reporting routine.

Skeletons for Libraries

Table 55. Skeletons for Libraries

Member Name	Library	Use
LIBRSTAT	59	Librarian STATE request.
SKLIBDEF	59	Define VSE user library in non-VSAM managed space.
SKLIBDEL	59	Delete VSE user library in non-VSAM managed space.
SKLIBEXT	59	Extend VSE user library in non-VSAM managed space.

Skeletons for VSE/POWER

Table 56. Skeletons for VSE/POWER

Member Name	Library	Use
SKPWRBSC	59	Define POWER BSC lines and work stations.
SKPWRDAT	59	Dynamic extension of VSE/POWER data file (which permits a VSE/POWER warm start).
SKPWRGEN	59	Generate VSE/POWER.
SKPWRNDT	59	Create PNET network definition table (SSL support).
SKPWRSNA	59	Define SDLC work stations.
SKPWSTRT	59	Startup job for VSE/POWER.
SKPWREXT	59	Extend or relocate VSE/POWER files.
SKPWRDMP	59	Create dump of the VSE/POWER spool file.

Skeletons for VSE/ICCF

Table 57. Skeletons for VSE/ICCF

Member Name	Library	Use
SKDTSEXT	59	Extend VSE/ICCF DTSFILE.
SKICFFMT	59	Format VSE/ICCF DTSFILE.
SKICFGEN	59	Generate VSE/ICCF.
SKICFRST	59	Selective restore of DTSFILE from the z/VSE installation tape or cartridge.
SKREORG	59	VSE/ICCF file reorganization.

Skeletons for Network Definitions

Table 58. Skeletons for Network Definitions

Member Name	Library	Use
SKAPPN	59	Define VSE/VSAM clusters for VTAM APPN.
SKDTRNET.B	IJSYSRS.SYSLIB	Create resource definitions for installing and operating a VSE/ESA system in a network[1].
SKNCPLD	59	Load 37xx from CKD device using ACF/SSP.
SKNCPCLF	59	Create NCP load module file.
SKNCPCST	59	Set up NCP configurable station.
SKNCPSAL	59	Define the Subarea Link.
SKNCPWKF	59	Create NCP workfile.
SKVTMMOD	59	Create VTAM Logon Mode table.
SKVTMUSS	59	Create VTAM USS table.

[1] Use the LIBRP command to punch a copy of SKDTRNET into a VSE/ICCF library.

Delete Skeletons

Table 59. Delete Skeletons

Member Name	Library	Use
DELCICS	59	Delete CICS Transaction Server.
DELCOBII	59	Delete VS COBOL II.
DELDB274	59	Delete DB2 Server for VSE 7.4.
DELDIT	59	Delete DITTO for VSE.
DELHLASM	59	Delete High Level Assembler.
DELLECOB	59	Delete LE/VSE COBOL.
DELLEPLI	59	Delete LE/VSE PL/I.
DELREXX	59	Delete REXX/VSE.
DELTCPIP	59	Delete TCP/IP for VSE/ESA.
DELVTM	59	Delete VTAM.

Skeletons for Debugging

Table 60. Skeletons for Debugging

Member Name	Library	Use
SKDMPINI	59	Initialize Info/Analysis work files.
SKDMPMGR	59	Sample dump file manager procedure.
SKPWRDMP	59	Generate a dump of the VSE/POWER spool files.
SKSDBRA	59	Run SDAID branch trace with area definition.
SKSDBRJ	59	Run SDAID branch trace with jobname definition.
SKSDGTVA	59	Run SDAID GETVIS trace with area definition.
SKSDGTVJ	59	Run SDAID GETVIS trace with jobname definition.
SKSDINSA	59	Run SDAID instruction trace with area definition.
SKSDINSJ	59	Run SDAID instruction trace with jobname definition.
SKSDIOA	59	Run SDAID I/O interrupt trace with area definition.
SKSDIOJ	59	Run SDAID I/O interrupt trace with jobname definition.
SKSDPGCA	59	Run SDAID program check trace with area definition.
SKSDPGCJ	59	Run SDAID program check trace with jobname definition.
SKSDPGMA	59	Run SDAID program load trace with area definition.
SKSDPGMJ	59	Run SDAID program load trace with jobname definition.
SKSDSTA	59	Run SDAID storage alteration trace with area definition.
SKSDSTJ	59	Run SDAID instruction trace with jobname definition.
SKSDSVCA	59	Run SDAID SVC trace with area definition.
SKSDSVCJ	59	Run SDAID SVC trace with jobname definition.
SKSUPDMP	59	Sample for suppressing dumps.
SKSYSDMP	59	Dump SYSWK1 volume.
SKVSDMPD	59	Run utility program DOSVSDMP to create a standalone dump program on a CKD or FBA disk device.
SKVSMSNP	59	Activate VSE/VSAM SNAP dump.

Table 60. Skeletons for Debugging (continued)

Member Name	Library	Use
SKVTAMBU	59	Run SAID VTAM buffer trace.
SKVTAMIO	59	Run SDAID VTAM I/O trace.
TPRINT	59	Print VTAM trace.

Resource Definition and other Skeletons

Table 61. Resource Definition and other Skeletons

Member Name	Library	Use
DTSECTRC	59	Create access control table DTSECTAB.
IESBLDUP	59	Migrate user profiles.
IESELOGO	59	Change *z/VSE Online* panel and related functions.
IESUPDCF	59	Batch program for user profile maintenance.
JOBEXIT	59	Sample for dummy job exit.
JOBXSAMP	59	Sample job exit program.
SKARCHIV	59	Create user product-ID and component-ID entry for System History file.
SKBOOTST	59	Create IPL bootstrap records for IJSYSTR1 on volume SYSWK1.
SKCCERT	59	Sample client certificate.
SKEDECKX	59	Setting up an exit for the High Level Assembler for VSE to process E-Decks.
SKIESBLD	59	Create access control table DTSECTAB.
SKIOCPCN	59	IOCP configuration sample.
SKJMGRIN	59	Job manager file initialization.
SKLE370	59	Update the CICS/VSE CSD file for the VSE C Language Run-Time Support.
SKOMERST	59	Selective restore of online message file from z/VSE installation tape or cartridge.
SKREFRE	59	Restore member DTRIHIST.Z from z/VSE installation tape.
SKRSTRFL	59	Restore member IESTRFL.V from z/VSE installation tape.
SKRXPRTF	59	REXX printlog automation sample, first usage.
SKRXPRTL	59	REXX printlog automation sample, repeated usage.
SKSHRCTL	59	VSE/VSAM shared resource definition for DFHFCT.
SKSRVMOD	59	Sample APAR for object deck.
SKSRVPHS	59	Sample APAR for a phase.
SKSRVSCN	59	Sample job for scanning a phase.
SKSRVSRC	59	Sample APAR for source code.
SKSSLKEY	59	Catalog keyring set into CRYPTO.KEYRING sublibrary (SSL support).
SKSUPASM	59	Regenerate the supervisor.
SKTERSE	59	Sample for using TERSE utility.
SKUNDO	59	Remove user correction from a phase.
SKVMVSE	59	Punch VM/VSE modules to VM user.
SKVSAMDC	59	Define VSE/VSAM compression control data set.

Table 61. Resource Definition and other Skeletons (continued)

Member Name	Library	Use
SKVTAPE	59	Define VSE/VSAM file for virtual tape support.
SKWRKFIL	59	Define system work files on a Virtual Disk.
STDLABUS	59	Create standard labels for your files.
TLSDEF	59	Customization options for 3494 TLS (Tape Library Support).

Skeletons for JCL Analyzer

Table 62. Skeletons for JCL Analyzer

Member Name	Library	Use
ARDWINST	59	Install the JCL Analyzer for VSE.
ARDWREAD	59	JCL Analyzer for VSE User's Guide.
ARDWSVAL	59	Load the pre-open exit into the SVA.
ARDWVER	59	Verify installation of the JCL Analyzer for VSE.
ARDWVSAM	59	Create and load the VSAM log database.
ARDW101R	59	Run the JCL Analyzer data collection.
ARDW104	59	JCL Analyzer EXEC to be used on VM.

Skeletons for Workstation File Transfer Support

Table 63. Skeletons for Workstation File Transfer Support

Member Name	Library	Use
INWAMSA	59	Sample program (Assembler) for application interface to Host Transfer File.
INWAPIA	59	Sample program (Assembler) for application interface to Host Transfer File.
INWDAT1	59	File used by the Move Utility exit routines.
INWDAT2	59	File used by the Move Utility exit routines.
INWDESC1	59	Descriptor record used for Move Utility exit routines.
INWDESC2	59	Descriptor record used for Move Utility exit routines.
INWEXTA1	59	Sample exit routine (Assembler) for Move Utilities.
INWEXTA2	59	Sample exit routine (Assembler) for Move Utilities.
INWEXTC1	59	Sample exit routine (COBOL) for Move Utilities.
INWPATOE	59	Procedure for converting ASCII to EBCDIC.
INWPENG3	59	Translation table Euro support.
INWPETOA	59	Procedure for converting EBCDIC to ASCII.
INWPFRC3	59	Translation table Euro support.
INWPGER3	59	Translation table Euro support.
INWPITL3	59	Translation table Euro support.
INWPMS	59	Translate PC file transfer messages.
INWPMSXX	59	Translation of prefix.
INWPSPN3	59	Translation table Euro support.

Table 63. Skeletons for Workstation File Transfer Support (continued)

Member Name	Library	Use
INWPUK3	59	Translation table Euro support.
INWSAMA1	59	Sample program (Assembler) for a transfer operation with a SEND command.
INWSAMA2	59	Sample program (Assembler) for a transfer operation with a RECEIVE command.
INWSAMC1	59	Sample program (COBOL) for a transfer operation with a SEND command.
INWSAMC2	59	Sample program (COBOL) for a transfer operation with a RECEIVE command.
INWSAMP1	59	Sample program (PL/I) for a transfer operation with a SEND command.
INWSAMP2	59	Sample program (PL/I) for a transfer operation with a RECEIVE command.
INWSCRA	59	Map for program defined by INWSAMA1.
INWSCRC	59	Map for program defined by INWSAMC1.
INWSCRP	59	Map for program defined by INWSAMP1.
SKIWSTF	59	Create the Host Transfer File.

REXX/VSE Procedures

Table 64. REXX/VSE Procedures

Member Name	Library	Use
DMPMGR.PROC	IJSYSRS.SYSLIB	Dump file manager (Librarian, VSE/POWER).
IESWAITR.PROC	IJSYSRS.SYSLIB	Program synchronization (VSE/POWER, Console).
REXDFHDU.PROC	IJSYSRS.SYSLIB	Print routine for CICS transaction dumps.
REXXPRTL.PROC	IJSYSRS.SYSLIB	Hardcopy file manager (Console, Job Control).
REXXCO.PROC	PRD1.BASE	Monitoring console (Console).
REXXDOM.PROC	PRD1.BASE	Delete operator messages (Console automation).
REXXSCAN.PROC	PRD1.BASE	Scan job messages (Console automation).
REXXJMGR.PROC	PRD1.BASE	Job manager.
REXXEVNT.PROC	PRD1.BASE	Event handler (Console automation).
REXXCPUM.PROC	PRD1.BASE	Monitoring system load (Console).
REXXSAA.PROC	PRD1.BASE	Interactive REXX (Console).
REXXWAIT.PROC	PRD1.BASE	Start VSE/POWER job (VSE/POWER).
REXXSPCE.PROC	PRD1.BASE	Serialize VSE/POWER jobs (Console automation).
REXXSTOP.PROC	PRD1.BASE	Stop REXXCXIT (Console automation).
REXXASM.PROC	PRD1.BASE	Compile, link, and start assembler program (Job Control, Linkage Editor).
REXXSSDL.PROC	PRD1.BASE	Load phase and display load information.
REXXVSAM.PROC	PRD1.BASE	Open VSAM file and write records (Console automation).
SKRXPRT	59	Allows the hardcopy file to be managed and printed.

Appendix G. Devices Supported

This chapter provides an overview of the IBM devices supported by z/VSE. The devices are listed by type and number. Model information is given only if it is of significance. Almost all of the devices listed in this chapter can be defined to z/VSE by using the *Hardware Configuration* dialog. Devices which are compatible to other devices may not show up in the *Hardware Configuration* dialog. Such a device must be defined by selecting the device it is compatible with.

You may be interested in devices that are not listed but which may be used with z/VSE. Whenever you have a question of this kind or require further details about a listed device, please contact your IBM marketing representative.

Note that IBM may have announced other devices as supported by z/VSE after this manual was printed. Please contact your IBM marketing representative for the latest list of devices supported.

For details on the **processors** supported by z/VSE, refer to "Processor Support" on page 4.

Disk Devices

z/VSE supports the following types of disk devices:

FBA (Fixed Block Architecture)

ECKD (Extended-Count-Key-Data)

You can use a combination of FBA and ECKD disk devices in your hardware configuration.

For initial installation, you need two disk devices (volumes) of the same device type and with the same capacity.

The following table shows all disk devices supported by z/VSE. ISAM is not supported for any of these devices.

CKD/ECKD Devices
IBM TotalStorage DS8000 Series
IBM TotalStorage DS6000 Series
IBM TotalStorage Enterprise Storage Server (ESS)[4]
IBM 3390 [1]
IBM 3380 [2]
IBM RAMAC Array Family [3]
Internal Disk (IBM Multiprise 3000 processors) [4]

FBA Devices
Generic FBA Devices[5]
SCSI FCP-Attached Devices on IBM TotalStorage Enterprise Storage Server

1 Dependent on the model, the IBM 3390 is supported in 3390 (ECKD) mode and in 3380 track compatibility mode. It can be attached to a parallel channel or to an ESCON channel through an IBM 3990 Model 2, 3, or 6 control unit. When running in 3380 track compatibility mode, it cannot be attached to an ESCON channel.

2 The IBM 3380 Model A is supported as data device but not as system device for initial installation.

3 For details about IBM RAMAC disk devices refer to “Support for the IBM RAMAC Array Family” on page 14 and “IBM RAMAC Array DASD/Subsystem” on page 14.

4 z/VSE supports the ESS and the Internal Disk of the MP3000 either as an IBM 3380 or IBM 3390 disk device.

5 DAM is not supported for this device type.

Tape Devices

If you frequently back up and restore large amounts of data, it is recommended that you have at least one tape device that supports 6250 bytes per inch or one tape cartridge device.

IBM 3420	9 Track (Reel-to-Reel)

IBM 3480	Cartridge
IBM 3490	Cartridge
IBM 3490E	Cartridge
IBM 3590	Cartridge
IBM 3592	Cartridge

For details on the IBM 3590 refer to “Support for the IBM TotalStorage 3590 Tape Subsystem” on page 24.

IBM TotalStorage 3494 Tape Library

IBM 3494
IBM 3494 with Virtual Tape Server (VTS)

The IBM 3494 is an intermediate tape library dataserver. It automates the retrieval, storage, and control of IBM 3490E, IBM 3590, and IBM 3592 cartridge tapes. Further details are provided under “Support for the IBM TotalStorage 3494 Tape Library” on page 27.

Printers

Channel-Attached Printers

IBM 3800 Models 3, 6, 8
Note: The IBM 3800 Model 8 is only supported in Asia Pacific countries.

SNA LU 6.2 Attached Printers

These printers are supported via the optional licensed programs PSF/VSE.

IBM 3820	
IBM 3820	Remote PrintManager and PSF/2 Distributed Print Function
IBM 3935	

Advanced Function Printing Family, SNA-Attached

These printers are supported via the z/VSE optional program PSF/VSE. All of them have the Remote PrintManager and PSF/2 Distributed Print Function installed.

IBM 3825 IBM 3835 Models 001 and 002 IBM 3827 IBM 3828 IBM 3829 IBM 3900

Advanced Function Printing Family, LU1

IBM 3812	
IBM 3816	
IBM 4019	PSF/2 Distributed Print Function
IBM 4224	Models 201, 202, 2C2, 2E2, 2E3 (AA001 and higher)
IBM 4028	
IBM 4029	PSF/2 Distributed Print Function
IBM 4230	
IBM 4234	Model 011 (A0030 and higher)
IBM 3912	Model NS1
IBM 3916	Model NS1
IBM 3930	Models 02S and 02D

Advanced Function Printing Family, SNA-Token Ring Attached

IBM 3935

Advanced Function Printing Family, Channel-Attached

These printers are supported via the z/VSE optional program PSF/VSE.

IBM 3800 Models 3, 6, 8 IBM 3825 IBM 3835 Models 001 and 002 IBM 3827 IBM 3828 IBM 3829 IBM 3831 IBM 3900 IBM 3935

Terminal Printers and Channel-Attached Line Printers

The ″C″ indicates a channel-attached line printer.

A local terminal printer can function like a system printer via the CICS/VSE Report Controller Feature. When used in this manner, the terminal printer should have a minimum speed of 300 lines per minute.

IBM 1403 (C)	IBM 3289 (both)
IBM 3200	IBM 4019
IBM 3203 (C)	IBM 4224
IBM 3211 (C)	IBM 4234
IBM 3262 (C)	IBM 4245 (C)
IBM 3268	IBM 4248 (C)
IBM 3284	IBM 5553
IBM 3286	IBM 5557
IBM 3287	IBM 6252

IBM 3288 (supported like an IBM 3800)	IBM 6262 (C)

Notes:

1. The printers 3284, 3286, 3287, 4224, 4234, and 4245 are no longer supported as console printers.
2. The following Pennant printers are supported by the optional program PSF/VSE 2.2.1:

   3112, 3116, 3130, 3160, 6400, 6408, and 6412.

Display Stations

A selection of display stations supported by z/VSE is shown below. The *Hardware Configuration* dialog displays the characteristics a display station must have in order to be defined via the Interactive Interface.

Note: Local display stations are supported by VTAM or BTAM-ES, whichever you choose during initial installation.

IBM 3178	IBM 3277
IBM 3179	IBM 3278
IBM 3180	IBM 3279
IBM 3191	IBM 3472
IBM 3192	IBM 5540
IBM 3193	IBM 5555
IBM 3194	IBM 5560
IBM 3197	

Communication Control Units

z/VSE supports devices that can be attached to IBM processors via their integrated communication adapters or subsystem controllers/adapters or to the following local and remote communication control units.

For control units supported by TCP/IP for VSE/ESA, refer to the manual *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*.

IBM 3172	Interconnect Controller
IBM 3174	Local and Remote Terminal Control Unit
IBM 3274	Terminal Control Unit
IBM 3276	Terminal Control Unit
IBM 3745	Communication Controller
IBM 3791	Local Communication Controller

Communication Adapters

z/VSE supports the communication adapters listed below.

OSA-Express2	Open Systems Adapter Express 2
OSA-Express	Open Systems Adapter Express
OSA-2	Open Systems Adapter 2

For details of the z/VSE support for the above adapters, see Chapter 11, “TCP/IP, OSA, and HiperSockets Support,” on page 153.

Reader/Punch Devices

z/VSE supports the following Reader/Punch devices:

IBM 2520	Needed for VSE/POWER without blank card
IBM 2540	Needed for VM as default device
IBM 3505	Needed for VSE/POWER with blank card (or for VM virtual addresses)
IBM 3525	Needed for VSE/POWER as virtual device

Personal Computers

3270 Terminal Emulation

z/VSE supports any workstation as a **3270 display station**, provided that the workstation supports 3270 terminal emulation via a suitable adapter and the corresponding emulation control program. The workstation is connected and defined to z/VSE like a normal 3270 display station: as a local or as a remote SNA or non-SNA workstation.

Workstation File Transfer Support

In addition to 3270 terminal emulation, z/VSE provides file transfer support between workstations and a z/VSE host system. This support is available for any workstation that runs with one of the *emulation programs* listed in Table 65.

Table 65. 3270 Emulation Programs for Workstation File Transfer Support

3270 Emulation Program Name	Supported Version	Operating System	Mode
IBM OS/2 Communication Manager	any	OS/2	DFT
IBM Personal Communications	any	Windows	DFT
IBM AIX 3270 Host Connection Program for AIX	1.4 and later	AIX 3.2 and later	DFT
IBM OS/2 Communication Manager (Japanese Version)	any	OS/2 (Japanese Version)	DFT
IBM PS/55 Japanese 3270 Personal Computer Series	any	Windows (Japanese Version)	DFT

Table 65. 3270 Emulation Programs for Workstation File Transfer Support (continued)

3270 Emulation Program Name	Supported Version	Operating System	Mode
Note: • For information about required emulation adapter cards and possible host connections, refer to the appropriate programmable workstation and emulation program documentation or contact your IBM representative. • The IBM AIX 3270 Host Connection Program for AIX uses the XFER command rather than SEND/RECEIVE commands for file transfer. • z/VSE's Workstation File Transfer function supports file transfer buffer sizes from 2KB to 32KB with the following exception: For local non-SNA connections the maximum file transfer buffer size is 31KB (CICS restriction). • For workstations operating in DFT mode, the Extended Data Stream (EXTDS) feature, also known as "query bit", must be specified for each device associated with a host session.			

More information

Refer to the following IBM manuals:

- The manuals that come along with your hardware devices.
- *VSE/ESA Programming and Workstation Guide* for details on Workstation File Transfer Support.
- *AIX RISC System/6000 Command Reference* for a description of the XFER command.

Glossary

If you do not find the term you are looking for, refer to the index of this book or to the *IBM Glossary of Computing Terms* at:

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

The glossary includes definitions with:
- Symbol * where there is a one-to-one copy from the IBM Dictionary of Computing.
- Symbol (A) from the *American National Dictionary for Information Processing Systems*, copyright 1982 by the Computer and Business Equipment Manufacturers Association (CBEMA). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- Symbols (I) or (T) from the *ISO Vocabulary - Information Processing* and the *ISO Vocabulary - Office Machines,* developed by the International Organization for Standardization, Technical Committee 97, Subcommittee 1. Definitions of published segments of the vocabularies are identified by the symbol (I) after the definition; definitions from draft international standards, draft proposals, and working papers in development by the ISO/TC97/SC1 vocabulary subcommittee are identified by the symbol (T) after the definition, indicating final agreement has not yet been reached among participating members.

*** Access Control - Logging and Reporting.** An IBM licensed program used to log access to protected data and to print selected formatted reports on such access.

access control table (DTSECTAB). A table used by the system to verify a user's right to access a certain resource.

access method. A program, that is, a set of commands (macros), to define files or addresses and to move data to and from them; for example VSE/VSAM or VTAM.

*** account file.** A direct access file maintained by VSE/POWER to hold the accounting information it generates and the programs that it controls.

*** ACF.** Advanced Communications Function.

ACF/NCP. See *NCP*.

ACF/SSP. See *SSP*.

ACF/VTAM. See *VTAM*.

addressing mode (AMODE). A program attribute that refers to the address length that a program is prepared to handle on entry. Addresses may be either 24 bits or 31 bits in length. In 24-bit addressing mode, the processor treats all virtual addresses as 24-bit values; in 31-bit addressing mode, the processor treats all virtual addresses as 31-bit values. Programs with an addressing mode of ANY can receive control in either 24-bit or 31-bit addressing mode.

address space. A range of up to two gigabytes of contiguous virtual storage addresses that the system creates for a user. Unlike a data space, an address space contains user data **and** programs, as well as system data and programs, some of which are common to all address spaces. Instructions execute in an address space (not a data space). Contrast with *data space*.

*** Advanced Communications Function (ACF).** A group of IBM licensed programs, principally VTAM programs, TCAM, NCP and SSP that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

Advanced Function Printing (AFP). A group of IBM licensed programs that support APA printers.

*** advanced peer-to-peer networking (APPN).** Data communications support that routes data in a network between two or more APPC systems that do not need to be adjacent.

advanced program-to-program communication (APPC). A synonym for logical unit (LU) 6.2 and its implementations.

*** AFP.** Advanced Function Printing.

*** American National Standard Code for Information Interchange (ASCII).** The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

AMODE. Addressing mode.

*** APA.** All points addressable.

*** APAR.** Authorized program analysis report.

*** APPC.** Advanced program-to-program communications.

* **application profile.** A control block in which the system stores the characteristics of one or more application programs.

* **application program.** A program written for or by a user that applies directly to the user's work, such as a program that does inventory control or payroll. See also *batch program* and *online application program*.

* **APPN.** Advanced peer-to-peer networking.

* **ASCII.** American National Standard Code for Information Interchange.

ASI (automated system initialization) procedure. A set of control statements which specifies values for an automatic system initialization.

* **assemble.** To translate an assembly language program into an object program. (T)

* **assembler.** A computer program that converts assembly language instructions into object code.

assembler language. A programming language whose instructions are usually in one-to-one correspondence with machine instructions and allows to write macros.

attention routine (AR). A routine of the system that receives control when the operator presses the Attention key. The routine sets up the console for the input of a command, reads the command, and initiates the system service requested by the command.

* **authorized program analysis report (APAR).** A request for a correction of a problem caused by a defect in a current unaltered release of a program.

* **automated system initialization (ASI).** A function that allows control information for system startup to be cataloged for automatic retrieval during system startup.

background partition. An area of virtual storage in which programs are executed under control of the system. By default, the partition has a processing priority lower than any of the existing foreground partitions.

backup copy. A copy, usually of a file or a library member, that is kept in case the original file or library member is unintentionally changed or destroyed.

batch processing. Serial processing of computer programs. Pertaining to the technique of executing a set of computer programs such that each is completed before the next program of the set is started. (A)

batch program. A program that is processed in series with other programs and therefore normally processes data without user interaction.

bits per second (bps). In serial transmission, the instantaneous bit speed with which a device or channel transmits a character.

block. Usually, a block consists of several records of a file that are transmitted as a unit. But if records are very large, a block can also be part of a record only. On an FBA disk, a block is a string of 512 bytes of data.

* **bps.** Bits per second.

* **BSC.** Binary synchronous communication.

* **cache.** A high-speed buffer storage that contains frequently accessed instructions and data; it is used to reduce access time.

cache storage. A random access electronic storage in selected storage controls used to retain frequently used data for faster access by the channel. For example, the IBM 3990 Model 3 contains cache.

* **catalog.** A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors. (I) (A) To store a library member such as a phase, module, or book in a sublibrary.

See also *VSE/VSAM catalog*.

* **cataloged procedure.** A set of control statements placed in a library and retrievable by name.

cell pool. An area of virtual storage obtained by an application program and managed by the callable cell pool services. A cell pool is located in an address space or a data space and contains an anchor, at least one extent, and any number of cells of the same size.

Central Processing Complex. A segment of the physical resources of a system configuration.

central processing unit (CPU). The hardware component that interprets and executes instructions. Synonym for *processor*.

* **channel adapter.** A communication controller hardware unit used to attach the controller to a data channel.

* **channel-attached.** Pertaining to attachment of devices directly by data channels (I/O channels) to a computer. Contrast with *link-attached*. Synonymous with *locally attached*.

channel-to-channel connection (CTC). A function that allows data to be exchanged (1) under the control of VSE/POWER between two virtual VSE machines running under VM or (2) under the control of VTAM between two processors.

channel subsystem. A feature of Enterprise Systems Architecture that provides extensive additional channel (I/O) capabilities over the System/370.

CICS Transaction Server for VSE/ESA. This is the successor system to CICS/VSE.

* **CICS/VSE.** Customer Information Control System/VSE.

CKD device. Count-key-data device.

class. In VSE/POWER, a group of jobs that either come from the same input device or go to the same output device.

* **CMS.** Conversational monitor system.

COBOL. Common business-oriented language.

common business-oriented language (COBOL). A high-level programming language based on English used primarily for business application programs.

* **communication adapter.** A circuit card with associated software that enables a processor, controller, or other device to be connected to a network.

* **communication controller.** A device that directs the transmission of data over the data links of a network; its operation may be controlled by a program executed in a processor to which the controller is connected or it may be controlled by a program executed within the device. (T) A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit. It manages the details of line control and the routing of data through a network.

* **compile.** To translate a source program into an executable program (an object program). See also *assembler*.

compiler. A program used to compile.

control program. A program to schedule and supervise the running of programs in a system.

control unit. See *communication controller*. Synonymous with *controller*.

* **conversational monitor system (CMS).** A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities.

* **corrective service.** The installation of a PTF or an APAR fix that corrects a specific problem.

count-key-data (CKD) device. A disk device that stores data in the record format: count field, key field, data field. The count field contains, among others, the address of the record in the format: cylinder, head (track), record number and the length of the data field. The key field, if present, contains the record's key or search argument. CKD disk space is allocated by tracks and cylinders. Contrast with *FBA disk device*. See also *extended count-key-data device*.

* **CPU.** Central processing unit.

CTC. Channel-to-channel connection.

Customer Information Control System/VSE (CICS/VSE). A VSE/ESA base program that controls online communication between terminal users and a database. Transactions entered at remote terminals are processed concurrently by user-written application programs. The product includes facilities for building, using, and servicing databases.

The successor system is called CICS Transaction Server for VSE/ESA.

CUT (control unit terminal) mode. IBM PCs operating in CUT mode make not use of the extended data stream facility. Contrast with *DFT mode*.

* **DASD.** Direct access storage device.

DASD sharing. An option that lets independent computer systems use common data on shared disk devices.

database. A set of data available online that is organized by a common system and used for a common purpose.

* **data block (DBLK).** In VSE/POWER, the unit of transfer for spooling job input and job output.

Data Interfile Transfer, Testing and Operations/ESA for VSE (DITTO/ESA for VSE). A z/VSE base program that provides file-to-file services for card I/O, tape, and disk devices.

Data Language/I (DL/I). A database access language used with CICS/VSE and CICS TS.

* **data management.** A major function of the operating system. It involves organizing, storing, locating, and retrieving data.

data security. See *access control*.

data set. See *file*.

data space. A range of up to two gigabytes of contiguous virtual storage addresses that a program can directly manipulate through ESA/370 instructions. Unlike an address space, a data space can hold only user data; it does not contain shared areas, system data or programs. Instructions do not execute in a data space, although a program can reside in a data space as non-executable code. Contrast with *address space*.

* **DBLK.** Data block.

* **dedicated (disk) device.** A device that cannot be shared among users.

default value. A value assumed by the program when no value has been specified by the user.

* **device address.** The identification of an input/output device by its channel and unit number. In

data communication, the identification of any device to which data can be sent or from which data can be received.

* **device class.** The generic name for a group of device types; for example, all display stations belong to the same device class. Contrast with *device type*.

* **Device Support Facilities (DSF).** See *ICKDSF*.

* **device type code.** The four- or five-digit code to be used for defining an I/O device to a computer system.

DFT mode. Distributed function terminal (DFT) mode.

* **dialog.** In an interactive system, a series of related inquiries and responses similar to a conversation between two people. For z/VSE, a set of panels that can be used to complete a specific task; for example, defining a file.

direct access. Accessing data on a storage device using their address and not their sequence. This is the typical access on disk devices as opposed to magnetic tapes. Contrast with *sequential access*.

* **Direct access storage device (DASD).** A device in which access time is effectively independent of the location of the data.

directory. A table of identifiers and references to the corresponding items of data. (I) (A) In VSE, specifically, the index for the program libraries. See also *library directory*.

disk operating system residence volume (DOSRES). The disk volume on which the system sublibrary IJSYSRS.SYSLIB is located including the programs and procedures required for system startup.

disk sharing. An option that lets independent computer systems use common data on shared disk devices.

display station. A display screen with attached keyboard for communication with the system or a network. See also *terminal*.

disposition. A means of indicating to VSE/POWER how job input and output is to be handled. A job may, for example, be deleted or kept after processing.

distributed function terminal (DFT) mode. IBM PCs operating in DFT mode make use of the extended data stream facility allowing simultaneous host sessions by using screen windows.

* **distribution tape.** A magnetic tape that contains, for example, a preconfigured operating system like z/VSE. This tape is shipped to the customer for program installation.

* **DITTO/ESA for VSE.** Data Interfile Transfer, Testing and Operations/ESA for VSE.

DL/I. Data Language/I.

DOSRES. Disk operating system residence volume.

double-byte character set (DBCS). A character set which allows Korean, Japanese, and Chinese languages to be internally represented by two bytes per character.

* **DSF.** Device Support Facilities.

dummy device. A device address with no real I/O device behind it. Input and output for that device address are spooled on disk.

* **dump.** Data that has been dumped. (T) To record, at a particular instant, the contents of all or part of one storage device in another storage device. Dumping is usually for the purpose of debugging. (T)

* **duplex.** Pertaining to communication in which data can be sent and received at the same time.

dynamic class table. Defines the characteristics of dynamic partitions.

dynamic partition. A partition created and activated on an 'as needed' basis that does not use fixed static allocations. After processing, the occupied space is released. Contrast with *static partition*.

* **dynamic partition balancing.** A VSE facility that allows the user to specify that two or more or all partitions of the system should receive about the same amount of time on the processing unit.

EBCDIC. Extended binary-coded decimal interchange code.

ECKD device. Extended count-key-data device.

emulation. The use of programming techniques and special machine features that permit a computer system to execute programs written for another system or for the use of I/O devices different from those that are available.

* **emulation program (EP).** An IBM control program that allows a channel-attached 3705 or 3725 communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, or an IBM 2703 Transmission Control.

Enterprise Systems Architecture (ESA). See *ESA/390*.

Environmental Record Editing and Printing (EREP) program. A z/VSE base program that makes the data contained in the system recorder file available for further analysis.

EREP program. Environmental Record Editing and Printing program.

ESA/390. IBM Enterprise Systems Architecture/390. The latest extension to the IBM System/370 architecture which includes the advanced addressability feature and advanced channel architecture.

ESCON Channel (Enterprise Systems Connection Channel). A serial channel, using fiber optic cabling, that provides a high- speed connection between host and control units for I/O devices. It complies with the ESA/390 I/O Interface.

ESDS. Entry-sequenced data set. See *entry-sequenced file*.

exit. A routine, normally user-supplied, that receives control from the system when a certain event occurs (abnormal-end exit, for example).

extended addressability. See *31-bit addressing*. The ability of a program to use virtual storage that is outside the address space in which the program is running. Generally, instructions and data reside in a single address space - the primary address space. However, a program can have data in address spaces other than the primary or in data spaces. (The instructions remain in the primary address space, whilst the data can reside in another address space or in a data space.) To access data in other address spaces, a program must use access registers (ARs) and execute in access register mode (AR mode).

extended binary-coded decimal interchange code (EBCDIC). A coded character set consisting of 8-bit coded characters.

extended count-key-data (ECKD) device. A disk storage device that has a data transfer rate faster than some processors can utilize. A specialized channel program is needed to convert ordinary CKD channel programs for use with an ECKD device.

FASTCOPY. See *VSE/Fast Copy*.

fast service upgrade (FSU). A service function of z/VSE for the installation of a refresh release without regenerating control information such as library control tables.

FBA disk device. Fixed-block architecture disk device.

* **FCB.** Forms control buffer.

* **FCOPY.** See *VSE/Fast Copy*.

* **file.** A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

fixed-block architecture (FBA) disk device. A disk device that stores data in blocks of fixed size. These blocks are addressed by block number relative to the beginning of the file. Contrast with *CKD device*.

* **foreground partition.** A space in virtual storage in which programs are executed under control of the system. By default, a foreground partition has a higher processing priority than the background partition.

* **forms control buffer (FCB).** In the 3800 Printing Subsystem, a buffer for controlling the vertical format of printed output.

FSU. Fast service upgrade.

FULIST (FUnction LIST). A type of selection panel that displays a set of files and/or functions for the choice of the user.

* **GB.** Gigabyte.

* **generate.** To produce a computer program by selecting subsets of skeletal code under the control of parameters. (A)

generation. See *macro generation*.

generation feature. An IBM licensed program order option used to tailor the object code of a program to user requirements.

* **GETVIS space.** Storage space within a partition or the shared virtual area, available for dynamic allocation to programs.

gigabyte (GB). 1 024MB or 1 073 741 824 bytes.

guest system. A data processing system that runs under control of another (host) system.

* **half-duplex.** In data communication, pertaining to transmission of data in only one direction at a time. Contrast with *duplex*.

hardcopy file. A system file on disk, used to log all lines of communication between the system and the operator at the system console, to be printed on request.

hard wait. The condition of a processor when all operations are suspended. System recovery from a hard wait is impossible without performing a new system startup.

* **hardware.** All or part of the physical components of an information processing system, such as computers or peripheral devices. (T) (A) Contrast with *software*.

* **help panel.** A display of information provided by the system in response to a user's help request.

High Level Assembler for VSE. A programming language providing enhanced assembler programming support. It is a base program of z/VSE.

host mode. In this operating mode, a PC can access a VSE host. For programmable workstation functions, the Move Utilities of VSE can be used.

* **host processor.** In a network, the processing unit in which resides the access method for the network. In an SNA network, the processing unit that contains a system services control point (SSCP).

*** host system.** The controlling or highest level system in a data communication configuration.

host transfer file (HTF). Used by the Workstation File Transfer Support of z/VSE as an intermediate storage area for files that are sent to and from IBM Personal Computers.

ICCF. See *VSE/ICCF*.

ICKDSF (Device Support Facilities). A z/VSE base program that supports the installation, use, and maintenance of IBM disk devices.

index. In data management, a table used to locate the records of a file.

*** initial program load (IPL).** The process of loading system programs and preparing the system to run jobs.

*** input/output control system (IOCS).** A group of routines provided by IBM for handling transfer of data between main storage and auxiliary storage devices.

integrated console. In z/VSE, the service processor console available on ES/9000 processors that operates as the z/VSE system console. The integrated console is typically used during IPL and for recovery purposes when no other console is available.

intelligent workstation. See *programmable workstation*.

interactive. A characteristic of a program or system that alternately accepts input and then responds. An interactive system is conversational, that is, a continuous dialog exists between user and system. Contrast with *batch*.

interactive interface. A system facility which controls how different users see and work with the system by means of user profiles. When signing on, the interactive interface makes available those parts of the system authorized by the profile. The interactive interface has sets of selection- and data-entry panels through which users communicate with the system.

interactive partition. An area of virtual storage for the purpose of processing a job that was submitted interactively via VSE/ICCF.

Interactive User Communication Vehicle (IUCV). Programming support available in a VSE supervisor for operation under z/VM. The support allows a user to communicate with other users or with CP in the same way he would with a non-preferred guest.

interface. A shared boundary between two hardware or software units, defined by common functional or physical characteristics. It might be a hardware component or a portion of storage or registers accessed by several computer programs.

I/O (input/output). See *input* and *output*.

IOCS. Input/output control system.

*** IPL.** Initial program load.

IUCV. Interactive User Communication Vehicle.

*** JCL.** Job control language.

JECL. Job entry control language.

job accounting. A system function that lists how much every job step uses of the different system resources.

*** job accounting interface.** A function that accumulates accounting information for each job step that can be used for charging usage of the system, planning new applications, and supervising system operation more efficiently.

*** job accounting table.** An area in the supervisor where accounting information is accumulated for the user.

job control language (JCL). A language that serves to prepare a job or each job step of a job to be run. Some of its functions are: to identify the job, to determine the I/O devices to be used, set switches for program use, log (or print) its own statements, and fetch the first phase of each job step.

job control statement. A particular statement of JCL.

job entry control language (JECL). A control language that allows the programmer to specify how VSE/POWER should handle a job.

job stream. The sequence of jobs as submitted to an operating system.

Kanji. A set of symbols used in Japanese ideographic printing. Every symbol is represented by two bytes.

KB. Kilobyte.

key-sequenced file. A VSE/VSAM file whose records are loaded in key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in the file in key sequence.

*** kilobyte (KB).** 1024 bytes.

KSDS. Key-sequenced data set. See *key-sequenced file*.

label. An identification record for a tape, disk, or diskette volume or for a file on such a volume. In assembler programming, a named instruction generally used for branching.

label information area. An area on a disk to store label information read from job control statements or commands. Synonymous with *label area*.

*** LAN.** Local area network.

*** language translator.** A general term for any assembler, compiler, or other routine that accepts statements in one language and produces equivalent statements in another language.

*** librarian.** The set of programs that maintains, services, and organizes the system and private libraries.

library. See *VSE library* and *VSE/ICCF library*.

*** library block.** A block of data stored in a sublibrary.

*** library directory.** The index that enables the system to locate a certain sublibrary of the accessed library.

*** library member.** The smallest unit of data that can be stored in and retrieved from a sublibrary.

*** licensed program.** A separately priced program and its associated materials that bear an IBM copyright and are offered to customers under the terms and conditions of the IBM Customer Agreement (ICA).

*** line printer.** A device that prints a line of characters as a unit. (I) (A)

*** linkage editor.** A program used to create a phase (executable code) from one or more independently translated object modules, from one or more existing phases, or from both. In creating the phase, the linkage editor resolves cross references among the modules and phases available as input. The program can catalog the newly built phases.

linkage stack. An area of protected storage that the system gives to a program to save status information in case of a branch or a program call.

link-edit. To create a loadable computer program by having the linkage editor process compiled (assembled) source programs.

local area network (LAN). A data network located on the user's premises in which serial transmission is used for direct data communication among data stations. (T)

local shared resources (LSR). A VSE/VSAM option activated by three extra macros to share control blocks among files.

*** lock file.** In a shared disk environment under VSE, a system file on disk used by the sharing systems to control their access to shared data.

*** logging.** The recording of data about specific events.

logical partition. In LPAR mode, a subset of the processor unit hardware that is defined to support the operation of a system control program.

logical record. A user record, normally pertaining to a single subject and processed by data management as a unit. Contrast with *physical record* which may be larger or smaller.

logical unit (LU). A name used in programming to represent an I/O device address.

*** logical unit name.** In programming, a name used to represent the address of an input/output unit.

logical unit 6.2. A SNA/SDLC protocol for communication between programs in a distributed processing environment. LU 6.2 is characterized by

1. a peer relationship between session partners,
2. efficient utilization of a session for multiple transactions,
3. comprehensive end-to-end error processing, and
4. a generic

application program interface (API) consisting of structured verbs that are mapped into a product implementation.

logo. A trademark or other art work that is associated with a firm or product. A logo often appears as the first screen of an interactive program.

LPAR mode. Logically partitioned mode. The CP mode that is available on the Configuration (CONFIG) frame when the PR/SM feature is installed. LPAR mode allows the operator to allocate the hardware resources of the processor unit among several logical partitions.

LSR. Local shared resources.

*** LU.** Logical unit.

macro (macro instruction). In assembler programming, a user-invented assembler statement that causes the assembler to process a set of statements defined previously in the macro definition. A sequence of VSE/ICCF commands defined to cause a sequence of certain actions to be performed in response to one request.

macro generation. An assembler operation by which a macro instruction gets replaced in the program by the statements of its definition. It takes place before assembly.

*** main task.** The main program within a partition in a multiprogramming environment.

*** maintain system history program (MSHP).** A program used for automating and controlling various installation, tailoring, and service activities for a VSE system.

major node. In VTAM, a set of minor nodes that can be activated as a group. See *node* and *minor node*.

master console. In z/VSE, one or more consoles that receive all system messages, except for those that are directed to one particular console. Contrast with the *user console* which receives only those messages that are specifically directed to it, for example messages issued from a job that was submitted with the request to echo its messages to that console. The operator of a master console can reply to all outstanding messages and enter all system commands.

* **MB.** Megabyte.

* **megabyte (MB).** 1 024 KB or 1 048 576 bytes.

* **member.** The smallest unit of data that can be stored in and retrieved from a sublibrary. See also *library member*.

message. In VSE, a communication sent from a program to the operator or user. It can appear on a console, a display terminal or on a printout. In telecommunication, a logical set of data being transmitted from one node to another.

* **migrate.** To move to a changed operating environment, usually to a new release or version of a system.

* **minor node.** In VTAM programs, a uniquely-defined resource within a major node. See *node* and *major node*.

mode. A method of operation. See also *ESA mode*.

* **module.** A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from an assembler, compiler, linkage editor, or executive routine. (A)

* **MSHP.** Maintain system history program.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture.

* **NCP.** Network Control Program.

* **nest.** To incorporate a structure or more structures of one kind into a structure of the same kind. For example, to nest one loop (the nested loop) within another loop or to nest one subroutine (the nested subroutine) within another subroutine. (T)

NetView. A z/VSE optional program used to monitor a network, manage it, and diagnose its problems.

network. An arrangement of nodes (data stations) and connecting branches. The assembly of equipment through which connections are made between data stations.

* **network address.** In SNA, an address, consisting of subarea and element fields, that identifies a link, link station, or NAU. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See also *network name*.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is ACF/NCP.

networking. Making use of the services of a network program.

network name. 1. In SNA, the symbolic identifier by which users refer to a NAU, link, or link station. See also *network address*. 2. In a multiple-domain network, the name of the APPL statement defining a VTAM application program. This is its network name which must be unique across domains.

* **node.** In SNA, an end point of a link or junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. In VTAM programs, a point in a network defined by a symbolic name. Synonymous with *network node*. See *major node* and *minor node*.

* **node name.** In VTAM programs, the symbolic name assigned to a specific major or minor node during network definition.

* **node type.** In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) it can contain.

nonprogrammable workstation (NPWS). A workstation that does not have processing capability and that does not allow the user to change its functions. Contrast with *programmable workstation*.

NPWS. Nonprogrammable workstation.

* **object code.** Output from a compiler or assembler which is itself executable machine code or is suitable for processing to produce executable machine code. (A)

object module (program). A program unit that is the output of an assembler or compiler and is input to a linkage editor.

OLTEP. See *VSE/OLTEP*.

online application program. An interactive program used at display stations. When active, it waits for data. Once input arrives, it processes it and sends a response to the display station or to another device.

online processing. Processing by which the input data enters the computer directly from a display station and the output data is transmitted directly to the display station.

* **operating system.** Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output

control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

optical reader/sorter. A device that reads hand written or machine printed symbols on a voucher and, after having read the voucher, can sort it into one of the available stacker-select pockets.

optional program. An IBM licensed program that a user can install on z/VSE by way of available installation-assist support.

OS/2 (Operating System/2). An IBM operating system that runs on IBM personal computers with appropriate microprocessors.

page. In a virtual storage system, the unit of code or data or both which is transferred between processor storage and the page data set (PDS) as needed for processing. To transfer pages between processor storage and the page data set.

page data set (PDS). One or more extents of disk storage in which pages are stored when they are not needed in processor storage.

* **page fixing.** Marking a page so that it is held in processor storage until explicitly released. Until then, it cannot be paged out.

* **page pool.** The set of page frames available for paging virtual-mode programs.

panel. The complete set of information shown in a single display on a terminal screen. Scrolling back and forth through panels is like turning manual pages. See also *selection panel*.

partition. A division of the virtual address area available for running programs. See also *dynamic partition*, *static partition*.

* **partition balancing, dynamic.** A VSE facility that allows the user to specify that two or more or all partitions of the system should receive about the same amount of time on the processor.

* **PDS.** Page data set.

* **phase.** The smallest complete unit of executable code that can be loaded into virtual storage.

* **physical record.** The amount of data transferred to or from auxiliary storage. Synonymous with *block*.

* **PL/I.** A programming language that is designed for use in a wide range of commercial and scientific computer applications.

PNET. Programming support available with VSE/POWER; it provides for the transmission of selected jobs, operator commands, messages, and program output between the nodes of a network.

POWER. See *VSE/POWER*.

* **preventive service.** The installation of one or more PTFs on a VSE system to avoid the occurrence of anticipated problems.

Print Services Facility (PSF)/VSE. An access method that provides support for the advanced function printers.

priority. A rank assigned to a partition or a task that determines its precedence in receiving system resources.

private area. The part of an address space that is available for the allocation of private partitions. Its maximum size can be defined during IPL. Contrast with *shared area*.

* **private library.** A user-owned library that is separate and distinct from the system library.

* **private partition.** Any of the system's partitions that are not defined as shared. See also *shared partition*.

procedure. See *cataloged procedure*.

* **processing.** The performance of logical operations and calculations on data, including the temporary retention of data in processor storage while this data is being operated upon.

* **processor.** In a computer, a functional unit that interprets and executes instructions. A processor consists of at least an instruction control unit and an arithmetic and logic unit. (T)

Processor Resource/System Manager. A feature of certain processors that allows the processor storage and resources to be divided into multiple logical processors.

processor storage. The storage contained in one or more processors and available for running machine instructions. Synonymous with *real storage*.

profile. A description of the characteristics of a user or a computer resource.

* **programmable workstation.** A workstation that has some degree of processing capability and that allows the user to change its functions. Contrast with *nonprogrammable workstation*.

* **programmer logical unit.** A logical unit available primarily for user-written programs. See also *logical unit name*.

program service. The customer- or program-related IBM service of correcting design or implementation errors via APARs and PTFs.

program temporary fix (PTF). A solution or by-pass of one or more problems documented in APARs. PTFs are distributed to IBM customers for preventive service to a current release of a program.

protocol. In SNA, the set of rules for requests and responses between communicating nodes that want to exchange data.

PR/SM. Processor Resource/Systems Manager.

PSF/VSE. Print Services Facility/VSE.

* **PTF.** Program temporary fix.

* **PU.** Physical unit.

* **queue.** A line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a network. To arrange in, or form a queue.

* **queue file.** A direct access file maintained by VSE/POWER that holds control information for the spooling of job input and job output.

real address. The address of a location in processor storage.

real mode. A processing mode in which a program may not be paged. Contrast with *virtual mode*.

real storage. See *processor storage*.

record. A set of related data or words, treated as a unit. See *logical record*, *physical record*.

* **remote job entry (RJE).** Submission of jobs through an input unit that has access to a computer through a data link.

residency mode (RMODE). A program attribute that refers to the location where a program is expected to reside in virtual storage. RMODE 24 indicates that the program must reside in the 24-bit addressable area (below 16 megabytes), RMODE ANY indicates that the program can reside anywhere in 31-bit addressable storage (above or below 16 megabytes).

* **restore.** To write back onto disk data that was previously written from disk onto an intermediate storage medium such as tape.

REXX/VSE. A general-purpose programming language which is particularly suitable for command procedures, rapid batch program development, prototyping, and personal utilities. It is part of VSE Central Functions, a base program of z/VSE.

* **RJE.** Remote job entry.

RMODE. Residency mode.

* **routine.** A program, or part of a program, that may have some general or frequent use. (T)

* **RPG II.** A commercially oriented programming language specifically designed for writing application programs intended for business data processing.

SCSI (Small Computer System Interface). A standard hardware interface that enables a variety of peripheral devices to communicate with one another.

SDL. System directory list.

security. See *access control*.

* **selection panel.** A displayed list of items from which a user can make a selection. Synonymous with *menu*.

sense. Determine, on request or automatically, the status or the characteristics of a certain I/O or communication device.

sequential access. The serial retrieval of records in their entry sequence or serial storage of records with or without a premeditated order. Contrast with *direct access*.

sequential access method (SAM). A data access method that writes to and reads from an I/O device record after record (or block after block). On request, the support performs device control operations such as line spacing or page ejects on a printer or skip a certain number of tape marks on a tape drive.

* **sequential file.** A file in which records are processed in the order in which they are entered and stored.

shared area. An area of storage that is common to all address spaces in the system. z/VSE has two shared areas:
1. The shared area (24 bit) is allocated at the start of the address space and contains the supervisor, the SVA (for system programs and the system GETVIS area), and the shared partitions.
2. The shared area (31 bit) is allocated at the end of the address space and contains the SVA (31 bit) for system programs and the system GETVIS area.

shared disk option. An option that lets independent computer systems use common data on shared disk devices.

* **shared partition.** A partition allocated for a program that provides services for and communicates with programs in other partitions of the system's virtual address spaces.

* **shared spooling.** A function that permits the VSE/POWER account file, data file, and queue file to be shared among several computer systems with VSE/POWER.

* **shared virtual area (SVA).** A high address area that contains a system directory list (SDL) of frequently

used phases, resident programs that can be shared between partitions, and an area for system support.

* **skeleton.** A set of control statements, instructions, or both, that requires user-specific information to be inserted before it can be submitted for processing.

* **SNA.** Systems Network Architecture.

* **SNA network.** The part of a user-application network that conforms to the formats and protocols of SNA.

source member. A library member containing source statements in any of the programming languages supported by VSE.

* **source program.** A program that a particular translator can accept. (T) Contrast with *object module*.

* **source statement.** A statement written in symbols of a programming language.

* **spool access support.** A function of VSE/POWER that allows user programs or subsystems running on VSE system to access the spool files of VSE/POWER.

* **spool file.** 1. A file that contains output data saved for later processing. 2. One of three VSE/POWER files on disk: queue file, data file, and account file.

* **spooling.** The use of disk storage as buffer storage to reduce processing delays when transferring data between peripheral equipment and the processors of a computer. In VSE, this is done under the control of VSE/POWER.

SQL/DS. Structured Query Language/Data System.

stand-alone program. A program that runs independently of (not controlled by) the VSE system.

startup. The process of performing IPL of the operating system and of getting all subsystems and application programs ready for operation.

static partition. A partition, defined at IPL time and occupying a defined amount of virtual storage that remains constant. Contrast with *dynamic partition*.

storage dump. See *dump*.

submit. A VSE/POWER function that passes a job to the system for processing.

* **subsystem.** A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system. (T)

* **supervisor.** The part of a control program that coordinates the use of resources and maintains the flow of processor operations.

supervisor mode. See *ESA mode*.

* **SVA.** Shared virtual area.

SYSRES. System residence file.

* **system console.** A console, usually equipped with a keyboard and display screen for control and communication with the system.

system directory list (SDL). A list containing directory entries of frequently-used phases and of all phases resident in the SVA. The list resides in the SVA.

* **system file.** A file used by the operating system, for example, the hardcopy file, the recorder file, the page data set.

system logical unit. A logical unit available primarily for operating system use. See also *logical unit name*.

* **system recorder file.** The file used to record hardware reliability data. Synonymous with *recorder file*.

system residence file (SYSRES). The z/VSE system sublibrary IJSYSRS.SYSLIB that contains the operating system. It is stored on the system residence volume DOSRES.

system sublibrary. The sublibrary that contains the operating system. It is stored on the system residence volume (DOSRES).

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

* **tailor.** A process that defines or modifies the characteristics of the system.

* **task.** The basic unit of synchronous program execution. A task competes with other tasks for system resources such as processing time and I/O channels.

* **TCT.** Terminal control table.

* **telecommunication.** Transmission of data between computer systems over telecommunication lines and between a computer system and remote devices.

* **terminal.** A point in a system or network at which data can either enter or leave. (A) Usually a display screen with a keyboard.

terminal control table (TCT). A control block in which the system stores information about the characteristics and modes of operation of the terminals defined to the system.

token ring. A network configuration where series of attaching devices are connected by unidirectional transmission links to form a closed path. Tokens are passed from adapter to adapter.

trace. To record a series of events as they occur. A record of specified events during the run of a program. A program to produce such a record.

* **track.** A circular path on the surface of a disk or diskette on which information is recorded and from which recorded information is read. Smallest unit of physical disk space.

* **transaction.** In a batch or remote batch entry, a job or job step. In CICS/VSE, one or more application programs that can be used by a display station operator. A given transaction can be used concurrently from one or more display stations. The execution of a transaction for a certain operator is also referred to as a task. A given task can relate only to one operator.

UCB. Universal character set buffer.

user console. In z/VSE, a console that receives only those system messages that are specifically directed to it. These are, for example, messages that are issued from a job that was submitted with the request to echo its messages to that console. Contrast with *master console*.

* **user exit.** A programming service provided by an IBM software product that may be requested during the execution of an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

* **utility program.** A computer program in general support of computer processes; for example, a diagnostic program, a trace program, or a sort program. (T) Synonymous with *service program*. A program designed to perform an everyday task such as copying data from one storage device to another. (A)

virtual address. An address that refers to a location in virtual storage. It is translated by the system to a processor storage address when the information stored at the virtual address is to be used.

virtual address area. The virtual range of available program addresses.

* **virtual address space.** A subdivision of the virtual address area available to the user for the allocation of private, nonshared partitions.

virtual disk. A range of up to two gigabytes of contiguous virtual storage addresses that a program can use as workspace. Although the virtual disk exists in storage, it appears as a real FBA disk device to the user program. All I/O operations directed to a virtual disk are intercepted and the data to be written to, or read from, the disk is moved to or from a data space.

Like a data space, a virtual disk can hold only user data; it does not contain shared areas, system data or programs. Unlike an address space or a data space, data is not directly addressable on a virtual disk. To manipulate data on a virtual disk, the program has to perform I/O operations.

* **virtual I/O area (VIO).** An extension of the page data set used by the system as intermediate storage, primarily for control data.

* **virtual machine (VM).** A functional simulation of a computer system and its associated devices.

* **virtual mode.** The operating mode of a program which may be paged.

virtual storage. Addressable space image for the user from which instructions and data are mapped into processor (real) storage locations.

z/VM. z/Virtual Machine.

volume. A data carrier that is mounted and demounted as a unit, for example, a reel of tape or a disk pack. (I) Some disk units have no demountable packs. In that case, a volume is the portion available to one read/write mechanism.

volume ID. The volume serial number, which is a number in a volume label assigned when a volume is prepared for use by the system.

volume table of contents (VTOC). A table on a disk volume that describes every file on it.

VRDS. Variable-length relative-record data set.

VSAM. See *VSE/VSAM*.

VSE (Virtual Storage Extended). A system that consists of a basic operating system and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware it controls form a complete computing system. Its current version is called z/VSE.

VSE/Advanced Functions. A program that provides basic system control and includes the supervisor and system programs such as the Librarian and the Linkage Editor. It is part of VSE Central Functions, a base program of z/VSE.

VSE/ESA Turbo Dispatcher. A facility of z/VSE that allows to exploit multiprocessor systems (also called CEC: Central Electronic Complexes). Each CPU within such a CEC has access to the shared virtual areas of z/VSE: supervisor, shared areas (24-bit), and shared areas (31-bit). The CPUs have equal rights which means that any CPU may receive interrupts and work units are not dedicated to any specific CPU.

* **VSE/Fast Copy (VSE/Fast Copy Data Set program).** A program for fast copy data operations from disk to disk and dump/restore operations via an intermediate dump file on magnetic tape or disk. It is part of VSE Central Functions, a base program of z/VSE.

*** VSE/ICCF (VSE/Interactive Computing and Control Facility).** A program that serves as interface, on a time-slice basis, to authorized users of terminals linked to the system's processor. It is part of VSE Central Functions, a base program of z/VSE.

VSE/ICCF library. A file composed of smaller files (libraries) including system and user data which can be accessed under the control of VSE/ICCF.

VSE library. A collection of programs in various forms and storage dumps stored on disk. The form of a program is indicated by its member type such as source code, object module, phase, or procedure. A VSE library consists of at least one sublibrary which can contain any type of member.

*** VSE/OLTEP (VSE/Online Test Executive Program).** A program that controls the activities on the online-test system and provides communication with the operator. This test system can be used to test I/O devices, control units, and channels while programs are running. It is part of VSE Central Functions, a base program of z/VSE.

*** VSE/POWER.** A program primarily used to spool input and output. The program's networking functions enable a VSE system to exchange files with or run jobs on another remote processor. It is part of VSE Central Functions, a base program of z/VSE.

VSE/VSAM (VSE/Virtual Storage Access Method). An IBM access method for direct or sequential processing of fixed and variable length records on disk devices. It is part of VSE Central Functions, a base program of z/VSE.

VSE/VSAM catalog. A file containing extensive file and volume information that VSE/VSAM requires to locate files, to allocate and deallocate storage space, to verify the authorization of a program or an operator to gain access to a file, and to accumulate use statistics for files.

*** VSE/VSAM managed space.** A user-defined space on disk placed under the control of VSE/VSAM.

VTAM (Virtual Telecommunications Access Method). A z/VSE base program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability; it supports application programs and subsystems (VSE/POWER, for example).

*** VTAM application program.** A program that has opened an ACB to identify itself to VTAM and can now issue VTAM macro instructions.

*** VTOC.** Volume table of contents.

workstation. See *programmable* and *nonprogrammable workstation*.

Workstation File Transfer Support. Enables the exchange of data between IBM Personal Computers (PCs) linked to a z/VSE host system where the data is kept in intermediate storage. PC users can retrieve that data and work with it independently of z/VSE.

X.25. In data communication, a recommendation of the CCITT that defines the interface between data terminal equipment and packet switching networks.

X.25 NCP Packet Switching Interface (NPSI). An IBM licensed program that allows SNA programs to communicate with SNA equipment or non-SNA equipment over packet-switched data networks. In addition, the product may be used to attach native X.25 equipment to SNA host systems without a packet network. See also *Recommendation X.25 (Geneva 1980)*.

z/Virtual Machine (z/VM). An IBM program providing operating system support. Among other services, it manages the resources of a single computer so that multiple computing systems seem to exist. Each of those virtual machines is the functional equivalent of an IBM computer system.

z/VSE (z/Virtual Storage Extended). The most advanced VSE system currently available.

31-bit addressing. Provides addressability for address spaces and data spaces of up to 2 gigabytes. (The maximum amount of addressable storage in previous systems was 16 megabytes.)

Index

Special characters

Numerics

A

B

C

D

Q

R

S

W

X

Z

Readers' Comments — We'd Like to Hear from You

IBM z/VSE
Planning
Version 3 Release 1

Publication No. SC33-8221-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name	Address
Company or Organization	
Phone No.	

Readers' Comments — We'd Like to Hear from You

SC33-8221-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

File Number: S370/S390-34
Program Number: 5609-ZVS

Printed in USA

SC33-8221-00

Spine information:

z/VSE | Planning | Version 3 Release 1 | SC33-8221-00