IBM z/VSE

Administration

Version 5 Release 1

SC34-2627-01

IBM z/VSE

Administration

Version 5 Release 1

SC34-2627-01

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

This edition applies to Version 5 Release 1 of IBM z/Virtual Storage Extended (z/VSE), Program Number 5609-ZV5, and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC34-2627-00.

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 Research & Development 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

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

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 websites 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 websites specifically mentioned in this publication or accessed through an IBM website 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
Dept. M358
IBM-Allee 1
71139 Ehningen
Germany
```

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

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

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

IPv6/VSE is a registered trademark of Barnard Software, Inc.

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/VSE enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using Assistive Technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/VSE. Consult the assistive technology documentation for specific information when using such products to access z/VSE interfaces.

Documentation Format

The publications for this product are in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when you use the PDF files and want to request a web-based format for a publication, you can either write an email to s390id@de.ibm.com, or use the Reader Comment Form in the back of this publication or direct your mail to the following address:

```
IBM Deutschland Research & Development GmbH
Department 3282
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany
```

In the request, be sure to include the publication number and title.

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

About This Book

This manual mainly describes how to perform resource definition tasks for IBM® z/VSE 5.1.

Resource definition tasks are tasks which define the characteristics of your z/VSE system. System resources include items such as startup procedures, user profiles, files, libraries, and devices installed. This manual describes how to define and modify such resources.

Who Should Use This Book

This manual is mainly intended for the system administrator or persons performing resource definition tasks. Some of the information provided concerns programmers as well.

How to Use This Book

For many resource definition tasks, z/VSE provides dialogs via the Interactive Interface. The manual shows for the model system administrator (SYSA) how to access a dialog for a particular task. It shows (in boxes) the Fast Path and the default synonym, if available. In the synonym box, space is left to allow you to add your own synonym. For those tasks for which z/VSE provides skeletons, the manual describes the skeletons and shows when and how to use them.

Where to Find More Information

An overview of z/VSE 5.1 enhancements and changes is provided in the *z/VSE Release Guide*.

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 to VSE users.

You can find the z/VSE home page at

http://www.ibm.com/systems/z/os/zvse/

You can also find VSE User Examples (in zipped format) at

http://www.ibm.com/systems/z/os/zvse/downloads/samples.html

Summary of changes

These are the enhancements that have been made available via the *June 2013* Service Upgrade of z/VSE 5.1:

- A topic describing the use of LUNs has been included. See "Use of Logical Unit Numbers (LUNs) With SCSI Disks" on page 99
- The *TS1140 Tape Drive* is now supported. See "Tape Devices" on page 91 and "How Tape Libraries are Configured" on page 256.
- Cartridge media types CSTB, CSTC, and CSTD are now supported. See "Output File Produced by a Query Inventory Request" on page 259.

These are the new items and changes that were delivered *at General Availability of z/VSE 5.1:*

- Chapter 4, "Configuring an OSA Express Adapter," on page 75 and Chapter 5, "Configuring a HiperSockets Device," on page 81 are new. Some of this information was previously contained in the *Networking Support* manual.
- A description has been provided describing how to connect to an *Intra-Ensemble Data Network* (IEDN). An IEDN provides connectivity between an IBM zEnterprise® CEC and an IBM zBXs (System z Blade Center Extension), or between two zEnterprise CECs. See Chapter 6, "Participating in an Intra-Ensemble Data Network," on page 85.
- Since *preferred paths* are no longer supported for SCSI disks, the relevant information has been removed from "Using Multipathing to Access SCSI Disks" on page 110.
- A new parameter SCOPE=JOB enables you to limit the lifetime of a VTAPE definition to the lifetime of a submitted job. See "Starting, Stopping, and Cancelling Virtual Tapes" on page 246.
- z/VSE now supports the *Copy Export* feature, which is included in the Model TS7740 of the IBM TS7700 Virtualization Engine. It allows you to "copy export" selected logical volumes to an offsite location in case they are required for a disaster recovery. See "Using the Copy Export Facility for a Disaster Recovery" on page 261.
- The commands and output for displaying hardware crypto information have changed. See "Displaying Hardware Crypto Status Information Under z/VSE" on page 469.
- A new command has been included for displaying details about an Adjunct Processor. See "Using the APSTAT Command to Obtain Details about an AP" on page 475.
- RSA keys with length 4096 are now supported for the Encryption Facility for z/VSE OpenPGP. See Chapter 48, "Implementing the Encryption Facility for z/VSE OpenPGP," on page 569.

Note: For a summary of *all* the items that have been introduced with z/VSE 5.1, refer to the *z/VSE Release Guide*, SC34-2636.

Part 1. SYSTEM CUSTOMIZATION

Chapter 1. Using the Interactive Interface and Skeletons

The z/VSE Interactive Interface makes it easier for you to interactively use the facilities of z/VSE and its components. You select the task you want to perform from selection panels. A dialog requests input from you to complete the specific task. Some functions are not supported by dialogs but by skeletons. To perform a task with a skeleton, you edit (change) the skeleton and submit it for processing.

z/VSE Profiles

z/VSE provides model user profiles. A user profile defines a user to the z/VSE System. It includes a user-ID and password which you use to sign on to the system. The profile defines what is invoked after you sign on. The model profiles reflect different levels of authorization. The user-IDs and corresponding passwords provided with z/VSE are shown below.

Table 1. Model User Profiles of z/VSE

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)
CNSL	CNSL	Default CICS user with administrator authority for the internal master console and other internal consoles
FORSEC	FORSEC	Model system administrator (without VSE/ICCF)
$SRV	$SRV	Model for problem determination
VCSRV	VCSRV	Connector Server/Virtual Tape Data Handler partition user

You can use the first three profiles as models to define your own user-IDs for an administrator, programmer, or operator. It is recommended that you do not change or delete the authorizations of SYSA, PROG, OPER, or FORSEC. They can be affected when you perform a Fast Service Upgrade.

After initial installation, you should define and use your own user-IDs using either:

- the *Maintain User Profiles* dialog.
- batch utility IESUPDCF.

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 CICS Transaction Server startup. CNSL is the default user used with the master console and other internal consoles (such as REXX consoles). These user-IDs do not have initial selection panels and can therefore not be used for sign-on.

Note: Do NOT delete the CICSUSER or CNSL user-IDs.

FORSEC is a model user-ID and password for system startup with security (access control) active. This user-ID is relevant only, if you have selected SECURITY=YES during initial installation of z/VSE or if you want to activate it later. For further details, refer to Chapter 32, “Customizing/Activating DTSECTAB-Based Security,” on page 391.

$SRV identifies a default panel hierarchy which provides access to a 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. For details about the dialogs available with $SRV, refer to the manual *z/VSE Guide for Solving Problems* under “Model User Profile for Problem Determination”.

VCSRV is the partition user required for the Connector Server and the Virtual Tape Data Handler startup. It does not have an initial selection panel and can therefore not be used for sign-on.

Chapter 54, “Fast Paths and Synonyms for Dialogs,” on page 649 provides an overview of the dialogs available and shows which model user-ID can access which dialog.

The panel hierarchies for SYSA, PROG, and OPER are shown in the foldouts at the back of the manual.

After initial installation you must change the passwords of SYSA, PROG, OPER, and $SRV during the first logon.

Types of Interactive Interface Panels

The Interactive Interface uses several types of panels:

Selection Panels

A selection panel displays up to nine options which you can select. The selections are numbered. You make your selection by entering the appropriate number at the bottom of the panel.

After you have entered the selection, either another selection panel is displayed or an application is invoked. This application may offer you a:

- Selection panel
- Data entry panel
- Function list
- Application sign-on panel

Data Entry Panels

The dialogs use data entry panels to obtain input about the task you are performing. You enter information in particular fields on the panel. For example, if you define a library, you must enter the library file name in the LIBRARY NAME field.

Function Lists

A Function List (FULIST) displays a list of items which you can process. The items can be, for example:

- Devices
- VSE/VSAM files
- VSE/ICCF library members
- VSE/POWER queue entries

Such a panel also displays options you can use to process these items. Options could be:

- Alter
- Show
- Print
- Copy
- Delete

Each option corresponds to a particular number. When using a FULIST, you simply enter the number of the option next to the item you want to process.

Help Panels

On most panels you can press PF1 to display a HELP panel. This provides additional information about the FULIST, selection panel, or data entry panel and the task you are performing.

Sometimes the system displays a message on your panel for which specific help information is available. For example, if you have entered incorrect data, the message informs you of the error. If you then press PF1 in the case of CICS panels, the HELP panel explains the error and how you can correct it. Some HELP panels display a list of topics which you can select for more information.

Using the Fast Path Facility

The Interactive Interface has a "Fast Path" facility which allows you to go directly to a dialog without going through the entire panel hierarchy to reach the dialog. You will use the Fast Path facility after having some experience and knowing the selection numbers required for a task.

To use a Fast Path, you enter all the numbers (on one selection panel) of the selections you would enter on the individual panels in the hierarchy. An example is given below to show you how a Fast Path works. After signing on with an administrator user-ID, the system displays the *z/VSE Function Selection* panel.

Suppose you want to access the *Maintain User Profiles* dialog. To reach the dialog by going through the entire hierarchy, you would do the following:

- In the *z/VSE Function Selection* panel, select:

  2 (Resource Definition)

- In the *Resource Definition* panel, select:
  1 (User Interface Tailoring)
- In the *User Interface Tailoring* panel, select:
  1 (Maintain User Profiles)

By using the Fast Path, you can go directly to the *Maintain User Profiles* dialog. In the *z/VSE Function Selection* panel, enter:

211

A Fast Path may be entered on any selection panel. It represents the selections starting from the current panel. If a Fast Path is preceded by the character '=', selection begins at the *z/VSEFunction Selection* panel. Note that if you use the Fast Path facility to access a dialog, PF3 returns to the point where the Fast Path started.

Chapter 54, "Fast Paths and Synonyms for Dialogs," on page 649 shows the Fast Paths that can be used to access dialogs.

Using the Synonym Function

This function allows you to use a private synonym for selecting a panel. For example, instead of specifying the Fast Path **211** for the *Maintain User Profiles* dialog you can also enter a word which you may remember better. **upm** is the default synonym provided by z/VSE for the *Maintain User Profiles* dialog.

Synonym models are shipped for the model users SYSA, PROG, and OPER. Refer to Chapter 54, "Fast Paths and Synonyms for Dialogs," on page 649 for the synonyms provided for these users. With the *Maintain Synonym* dialog, users can change their synonyms and create new ones. Refer to "Maintaining Synonyms" on page 142 for a description of the dialog. Note that if you use a synonym to access a dialog, PF3 returns to the initial function selection panel.

Note: For the dialogs described in this book, you will find two boxes showing the related Fast Path and the default synonym, if available. The synonym box has space left to allow you to add the synonym you created for yourself.

Signing on to the Interactive Interface

In order to use the Interactive Interface, you have to *sign on*. The sign-on procedure identifies you to the system and accesses the Interactive Interface. Before you can sign on, you need a user-ID and password. The system administrator is responsible for creating user-IDs.

If LDAP-authentication is *not* enabled, the user-ID is a 4 - 8 character name that identifies a user to the system. The password is a 3 - 8 character confidential code associated with the user-ID. You sign on to the Interactive Interface from the *z/VSE Online* panel by entering your user-ID and password. An example of the panel is shown in Figure 1 on page 10. The password is **not** displayed on the panel.

```
USER-ID....      xxxxxxxx
PASSWORD...      yyyyyyyy
```

The system checks the user-ID and password. If they are correct, it accesses the selection panel or application which is defined for that particular user-ID. "Password Expiration" on page 143 gives details on how to change a password.

If LDAP-authentication *is* enabled, an LDAP sign-on panel (a z/VSE LDAP panel containing a *long-user-ID* and a *long-password* field) is displayed:

```
IESADMSO3                          z/VSE SIGN ON
    5609-ZV5 and Other Materials (C) Copyright IBM Corp. 2011 and other dates


                                 ++
                                ++    VV    VV    SSSSS    EEEEEEE
                               ++     VV    VV   SSSSSSS   EEEEEEE
                 zzzzzz       ++      VV    VV  SS         EE
                 zzzzz       ++       VV    VV   SSSSSS    EEEEEE
                   zz       ++        VV    VV    SSSSSS   EEEEEE
                  zz       ++          VV  VV          SS  EE
                 zzzzzz   ++            VVVV     SSSSSSS   EEEEEEE
                zzzzzzz  ++              VV       SSSSS    EEEEEEE

        Your terminal is      and its name in the network is D3000001
        Today is 10/22/2011   To sign on to DBDCCICS  --  enter your:

USER-ID. ________________________________________________________________
PASSWORD

PF1=HELP       2=TUTORIAL   3=TO VM      4=REMOTE APPLICATIONS     6=ESCAPE(U)
                            9=Escape(m)  10=NEW PASSWORD           12=LOGON HERE
```

For details of how LDAP sign-on is used, see Chapter 27, “Maintaining User Profiles in an LDAP Environment,” on page 323.

Using Program Function (PF) Keys

The Interactive Interface supports Program Function (PF) keys to perform various functions. PF keys and the function they represent are displayed at the bottom of each panel.

Note: Some keyboards use 'F keys' (for example, F9) instead of PF keys. The function, however, is the same. This book uses the name 'PF' key.

Your terminal has either 10, 12 or 24 PF keys depending on the model of the terminal. If you have 24 keys, PF13 - PF24 correspond to the same functions as PF1 - PF12 within a VSE environment. This may not be true for some applications which may use all 24 PF keys.

```
IESADMSO1                        z/VSE ONLINE
    5609-ZV5 and Other Materials (C) Copyright IBM Corp. 2011 and other dates


                                ++
                               ++    VV    VV    SSSSS    EEEEEEE
                zzzzzz        ++     VV    VV   SSSSSSS   EEEEEEE
                zzzzz        ++      VV    VV  SS         EE
                  zz        ++       VV    VV   SSSSSS    EEEEEE
                 zz        ++        VV    VV    SSSSSS   EEEEEE
                zzzzzz    ++          VV  VV         SS  EE
               zzzzzzz   ++            VVVV     SSSSSSS   EEEEEEE
                                        VV       SSSSS    EEEEEEE

        Your terminal is A000 and its name in the network is D3000001
        Today is 10/22/2011   To sign on to DBDCCICS  --  enter your:

       USER-ID........ ________   The name by which the system knows you.
       PASSWORD.......            Your personal access code.

PF1=HELP      2=TUTORIAL   3=TO VM      4=REMOTE APPLICATIONS     6=ESCAPE(U)
                           9=Escape(m) 10=NEW PASSWORD
```

Figure 1. z/VSE Online Panel

On a PC keyboard with only 10 PF keys, the following applies:
- PF9 corresponds to the PF11 key when pressed simultaneously with the shift key.
- PF10 corresponds to the PF12 key when pressed simultaneously with the shift key.

Some PF keys used by the Interactive Interface have the same function from every panel that uses them. Other PF key functions differ for different dialogs. Each panel shows the PF keys you can use and the functions to which they correspond. When you use a PF key, review the panel you are working with to know which function the PF key represents.

The following table shows the PF keys that have the same meaning on all panels of the Interactive Interface:

Table 2. Standard PF Key Usage

PF KEY	NAME	ACTION
1	HELP	Help information.
3	END	Quit and go back one level.
4	RETURN	Quit and go to top panel.
7	BACKWARD	Scroll (move) to previous page.
8	FORWARD	Scroll (move) to next page.

PF keys 2, 5, 6, 9, 10, 11, and 12 are used for different functions, but not for a function shown in Table 2.

Using Skeletons

Overview of How Skeletons Are Used

z/VSE provides skeletons to help you complete a number of tasks. A skeleton is a member in a VSE/ICCF library. You use it to create a job which completes a task. It contains variables and parameters which you change to reflect your requirements. After you make the changes, you submit the completed skeleton (job) to the system for processing.

z/VSE ships most skeletons in VSE/ICCF library 59. They are intended to be used more than once. Therefore, you should **copy** them to another library (usually your VSE/ICCF primary library) before you edit and change them. Keep the original skeleton in its library for future use.

A skeleton is replaced if it is affected by service such as FSU (Fast Service Upgrade) or when applying a PTF. Therefore, you should make sure you copy it to another library and only change the copied member.

Copying Skeletons

The Interactive Interface provides the *Program Development Library* dialog which you can use to copy VSE/ICCF members between libraries. This topic describes how you use the dialog to copy skeletons from the library in which they reside to your primary library.

When you access the *Program Development Library* dialog, the panel displays your default primary library in the PRIMARY field. This is the library to which you copy a skeleton. Access the library where the skeleton resides as the **secondary** library.

```
SECONDARY .... 59__      (Enter library number; 59, for example)
PREFIX ....... _______   (Optionally, enter a prefix for skeleton
                          names)
OPTION ....... 2         (Enter 2 for secondary library)
```

Note: Library member names (as for skeletons) consist of up to eight characters. The PREFIX field accepts up to seven characters.

The *Secondary Library* panel is displayed after pressing ENTER. Locate the skeleton name. Copy the skeleton to your primary library by entering **4** in the OPT column. You must specify a member name in the NEW NAME column. Enter the character “=” if the new name is to be the same as the original name.

Press **PF3** to return to the *Program Development Library* panel. Access your VSE/ICCF primary library to edit the copied skeleton.

Chapter 2. Tailoring IPL and System Startup

This chapter contains these main topics

Note: From z/VSE 5.1 onwards, CICS/VSE (Program Number 5686-026) is no longer shipped on the Extended Base Tape. Therefore, all references to CICS/VSE have been removed from this chapter.

Related Topics:

Before you modify the procedures for IPL (initial program load) and system startup, you should be familiar with the information provided in the chapter "System Organization and Concepts" of the manual *z/VSE Planning*.

You should also be familiar with the IPL and startup information provided in the manual *z/VSE Guide to System Functions* under "Starting the System" which provides details on topics such as the following:

- The ASI master procedure ($ASIPROC).
- Establishing the communication device for IPL.
- Interrupt IPL processing for modifications.
- Loading phases into the SVA.

The description of IPL and system startup in this chapter follows the sequence (and uses the names) of a z/VSE system as shipped by IBM.

Note: Terminology! The process of IPL and system startup is also referred to as ASI (automated system initialization). In this context, startup procedures and functions are also referred to as ASI procedures and functions. This chapter uses the term ASI only if technical or terminology reasons make it advisable to do so.

Initiating System Startup

Before starting up z/VSE, the operator has to perform IML (Initial Microprogram Load) for hardware (processor) initialization.

The operator can then initiate startup for z/VSE by performing IPL. After the programs required for IPL have been loaded the appropriate IPL procedure will be processed. An IPL procedure defines specific system parameters needed during IPL processing.

The name of the IPL procedure used during initial installation depends on the disk device type on which z/VSE is to reside (DOSRES). For example, for a system residing on an:

- IBM 3390 disk device, z/VSE selects IPL procedure $IPLE90 for initial installation.
- IBM FCP-attached FBA-SCSI disk device, z/VSE selects IPL procedure $IPLEGF for initial installation.

For an example of the contents of an IPL procedure and the names of the IPL procedures shipped with z/VSE, refer to the *z/VSE Planning*.

During initial installation, the IPL procedure is modified according to the customer's environment and is renamed to **$IPLESA**. In a running system (after installation), the IPL procedure appears under this name, for example, when working with the *Tailor IPL Procedure* dialog. You can use this dialog to add or delete an IPL procedure or modify its values.

Using a $ASIPROC Procedure

When the operator initiates IPL, z/VSE does one of the following to get the correct IPL and JCL procedure names for startup:

1. Retrieves the names from $ASIPROC, if a $ASIPROC master procedure exists. For a detailed description of creating a $ASIPROC master procedure refer to the manual *z/VSE Guide to System Functions* under "The ASI Master Procedure". As shipped, z/VSE includes a $ASIPROC for initial installation only.
2. Uses the default names: $IPLESA and $$JCL.
3. In addition, z/VSE allows the operator to interrupt startup processing. The operator can then enter the procedure names to be used by the system. This is described in detail in the manual *z/VSE Guide to System Functions* under "Interrupt and Restart the IPL Process".

Tailoring the IPL Procedure

When you tailor an IPL procedure, be aware of the predefined disk layouts used by z/VSE and shown in the manual *z/VSE Planning* under "z/VSE® Disk Layouts". There you can find the location of all the system files and libraries used by z/VSE and the free space still available on DOSRES and SYSWK1. *Do not use the areas defined as reserved*.

Be aware that:

- Predefined environment B (a medium environment) contains an enlarged page data set of 512 MB (compared to 256 MB for predefined small environment A). The free space is reduced accordingly.
- Predefined environment C (a large environment) contains an enlarged page data set of 1792 MB. The free space is reduced accordingly.

To access the *Tailor IPL Procedure* dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

4 (Hardware Configuration and IPL)

2 (Tailor IPL Procedure)

The panel displayed lists the IPL procedures defined for your system. **ADD**, **ALTER**, and **DELETE** are the options you can select. Enter the option number in the OPT column next to the procedure you want to process.

Adding or Altering an IPL Procedure

If you add a procedure, the procedure you select is used as a model. The values defined for the model are used as defaults for the new procedure. You are requested for a name of the new IPL procedure. This is the name you must include in $ASIPROC or the operator must enter it during IPL.

IPL Parameters You Can Modify

If you add or alter an IPL procedure, you can modify the IPL parameters listed below. Refer to the manual *z/VSE System Control Statements* for a detailed description of the IPL commands and parameters or use the help text provided by the dialog via PF1.

- **Supervisor**

  Used to modify parameters such as the physical address of the IPL console and virtual storage options (VSIZE, VIO, VPOOL, IODEV). Figure 2 on page 17 shows the panel layout and the parameters that can be specified.
- **SYS**

  Used to modify various system parameters displayed on two panels:

  Panel TAS$ICM1

  ```
  BUFLD
  CHANQ
  DASDFP
  SUBLIB
  VMCF
  SEC
  ESM
  SERVPART
  TRKHLD
  ```

  For details of the security parameters SEC, ESM, and SERVPART, refer to "Using the Tailor-IPL-Procedure Dialog to Tailor Security Parameters" on page 278.

  Panel TAS$ICMA

  ```
  BUFSIZE
  NPARTS
  PASIZE
  RSIZE
  SDSIZE
  SPSIZE
  ATL
  QUIESCE
  ```

  For details of the "signal shutdown" parameter QUIESCE, refer to the chapter "Initial Program Load" in the *z/VSE System Control Statements*.

  If your processor allows automatic timer-controlled IPL, set the BUFLD parameter to IGNORE if you use this function. Refer to the description of the SYS command in the manual *z/VSE System Control Statements* under "SYS" for further details.

  The automatic IPL function is closely related to the automatic timer-controlled power-on function of a processor. If these hardware functions are available, your processor provides panels for using them.
- **DPD**

  Used to modify page data set definitions. Be aware that:
  - Predefined environment A contains a page data set of 256 MB.
  - Predefined environment B contains a page data set of 512 MB.
  - Predefined environment C contains a page data set of 1792 MB.

  Note that:

– A DPD specification is not allowed if you define a system without a page data set. Refer to "Page Data Set Considerations" on page 17 for further details.
– You must ensure that no overlap occurs with other files when enlarging or relocating the page data set extents.

Refer to "Page Data Set Considerations" on page 17 for further page data set details.

- **DLF**

  Used to modify the lock file (also known as cross-system communication file) definition. This file is required when sharing disk devices (DASD sharing) among VSE systems.
- **DEF**

  Used to modify the definition of the physical device for the system recorder file and the hardcopy file (SYSREC) and the VSE/VSAM master catalog (SYSCAT).
- **ZONE**

  Used to modify the ZONE specifications (time difference between local time and Greenwich Mean Time).

  For further details about ZONE specifications, refer to Chapter 16, "ZONE Specifications and Daylight Saving Time," on page 201.
- **APPC/VM**

  Used to modify VSE APPC/VM resource definitions. Such definitions are required if you run z/VSE under VM and want to enable DB2 Server for VSE & VM applications to share one or more DB2® data bases.
- **SVA**

  Used to modify parameters of the Shared Virtual Area (SVA):

```
SDL ENTRIES  (max. number is 32765)
Additional PSIZE (24-bit, 31-bit)
Additional GETVIS (24-bit, 31-bit)
```

How to Add an IPL Procedure

The following steps are required if you want to create and add a new IPL procedure:

1. In the panel that lists the IPL procedures for your system enter **1** next to an existing IPL procedure. This procedure is used as a model. The values of the model are used as defaults for the new procedure.
2. The next panel requests you to enter the **name** of the new procedure (the first character must be $).
3. The subsequent panel shows the parameters that can be modified. They are listed under "IPL Parameters You Can Modify" on page 15. Enter **1** to the left of the parameter(s) you want to change. Figure 2 on page 17 shows, as an example, the panel for changing SUPERVISOR parameters.
4. A panel will be displayed for each parameter you select. Press **PF5** to save your changes.
5. On the panel displayed next, press **PF5** to create a jobstream to catalog your new or updated IPL procedure.
6. The *Job Disposition* panel is displayed. With it, you can submit the job to batch, file it in your VSE/ICCF primary library, or both.

```
 TAS$SUP1        TAILOR IPL PROCEDURE: SUPERVISOR PARAMETERS

 Enter the required data and press PF5=PROCESS

IPL CONSOLE................. ____       Console address used during IPL.  For
                                        valid console addresses enter a "?".

PAGE DATA SET................ _          Do you want to use a page data set?
                                        1=yes, 2 =no.

VSIZE....................... _______    Virtual address space; in M or G.

VIO......................... _______    Virtual I/O work space; in K or M.

VPOOL....................... ______     V-pool size; in K or M.

IODEV....................... ____       Number of I/O devices

IPL LOG OPTION.............. _          Logging of IPL commands on the IPL
                                        console required? 1=yes, 2=no.
 PF1=HELP      2=REDISPLAY  3=END                    5=PROCESS
```

Figure 2. Panel for Modifying Supervisor Parameters

Page Data Set Considerations

The page data set option allows to define a system without a page data set which is possible if enough real (processor) storage is available or if, in a VM environment, the VM virtual storage size is large enough to accommodate all z/VSE storage requirements.

If you specify 2 (no) for PAGE DATA SET, you get a system without a page data set. This system status is also referred to as NOPDS. For a NOPDS-system, a specification of VSIZE and the DPD parameter in the IPL procedure are not allowed.

The manual *z/VSE Planning* provides further details about NOPDS.

You may move the page data set from the system reserved space to another location and use the reserved space for your own definitions. You can release the reserved space by using the FREE key (PF6) on the DPD panel of the *Tailor IPL Procedure* dialog.

Note that starting with VSE/ESA 2.6, z/VSE checks automatically during initial installation whether the processor storage size is sufficient for a NOPDS system. If YES, no page data set is created and a NOPDS system is established.

IODEV Considerations

The IODEV parameter specifies the number of Input/Output (I/O) devices the z/VSE supervisor will support. The IODEV parameter also determines the shared addressing area in which various I/O control blocks will be allocated by the z/VSE supervisor.

You can set IODEV to either:

- **1023**, which means:
  - The z/VSE supervisor supports 1023 I/O devices.
  - The z/VSE supervisor will allocate the I/O control blocks in the 24-bit shared area, under the *16 MB line*.
- **1024**, which means:
  - The z/VSE supervisor supports 1024 I/O devices.

– The z/VSE supervisor will allocate the I/O control blocks in the 31-bit shared area (instead of in the 24-bit shared area), under the *2 GB bar*.

Overview of Startup Processing

When the operator initiates system startup, first the IPL procedure, then the JCL startup procedures and jobs are processed. As shipped, z/VSE uses the default procedure names for startup.

The JCL startup procedure for the BG partition calls, right at the beginning, the startup program **DTRISTRT**. DTRISTRT uses as input the following:

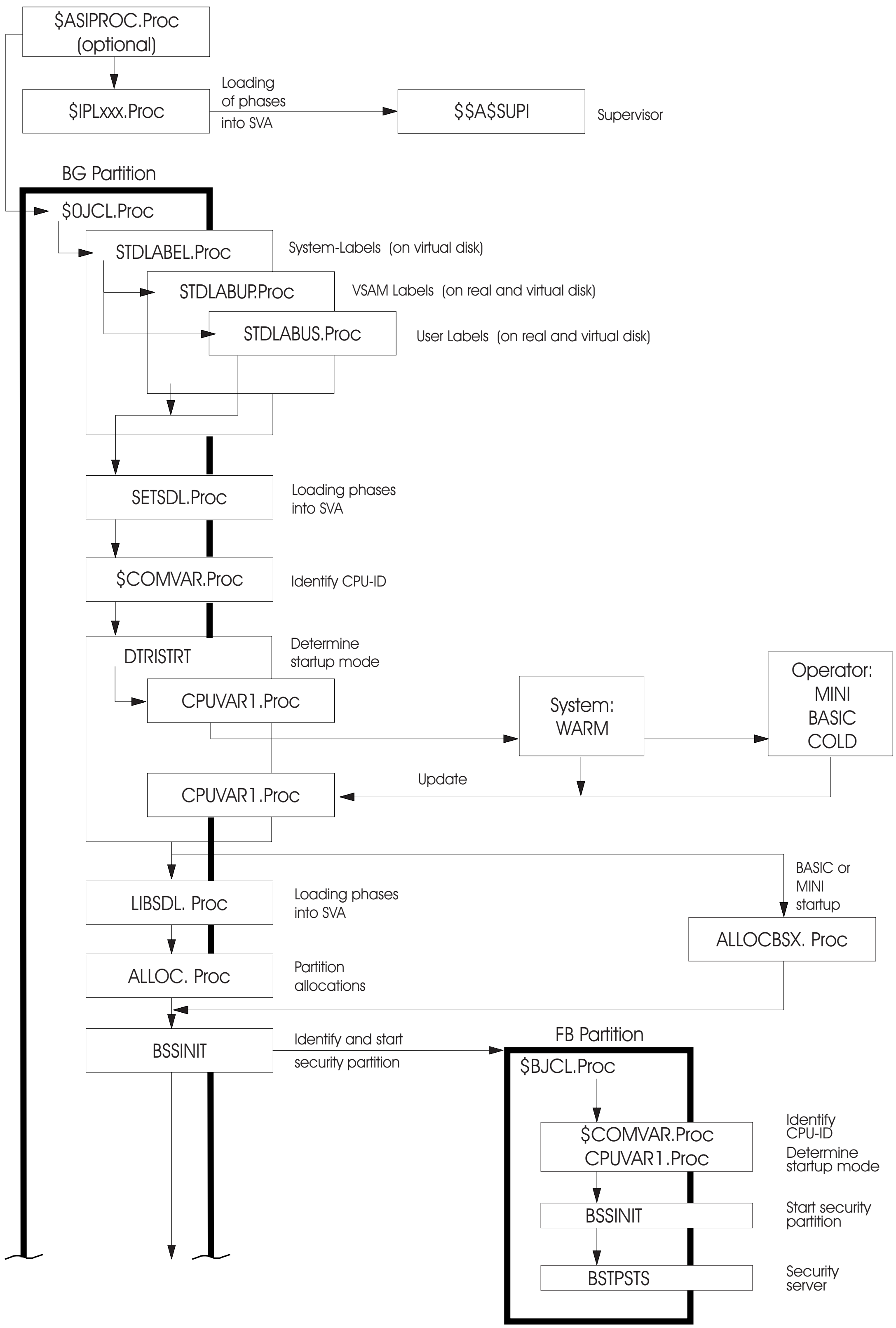
- System variables stored in procedure CPUVARn (CPUVAR1 procedure shipped with z/VSE and used in this description).
- The startup mode entered by the operator (BASIC, COLD, or MINI).

  Entering a startup mode is optional.

As output, DTRISTRT generates values for system variables and updates those variables in procedure CPUVAR1.

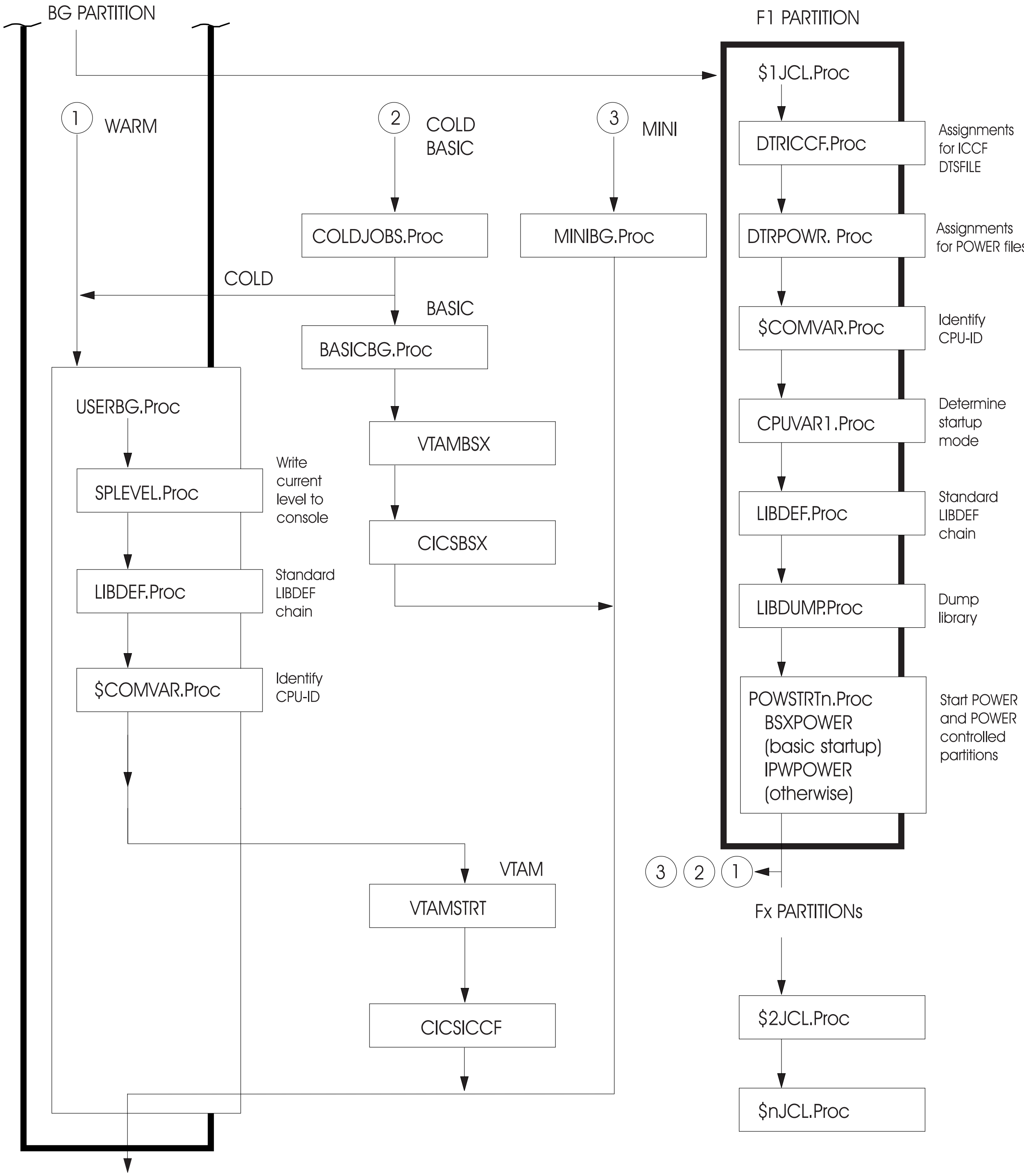
The JCL startup procedures retrieve the updated system variables from CPUVAR1. These variables control the subsequent startup process and determine the startup mode used. Refer to "JCL Startup Procedures and Jobs" on page 21 for further details.

Figure 3 on page 19 and Figure 4 on page 20 show the flow of main events during startup processing for predefined environments A, B, and C. Note that procedure POWSTRTn (where n identifies the predefined environment) also activates the dynamic partition support which is not shown in the figure.



Note: As shipped, z/VSE includes a $ASIPROC for initial installation only. If you want to use $ASIPROC, you must create your own. Refer also to "Using a $ASIPROC Procedure" on page 14.

Figure 3. z/VSE Startup Sequence for an Unmodified System (Part 1)



Note: The sequence of events shown in the F1 partition is for WARM startup and continues at entry point 1. A COLD or BASIC startup continues at entry point 2, a MINI startup at entry point 3, after the corresponding sequence in F1 has completed.

Figure 4. z/VSE Startup Sequence for an Unmodified System (Part 2)

JCL Startup Procedures and Jobs

After the successful processing of the IPL procedure, the JCL startup procedure for the BG partition **$0JCL** is called. The 0 in the second position indicates the BG partition. For details of the naming convention of JCL startup procedures, refer to the *z/VSE Planning*.

Following is a list of actions performed after $0JCL gets control. The list describes the flow of events (for WARM and RECOVERY startup) as shown in Figure 3 on page 19 and Figure 4 on page 20.

1. $0JCL calls procedure **STDLABEL** to write system file labels into the label information area on the virtual disk. STDLABEL calls label procedures **STDLABUP** (VSE/VSAM labels) and **STDLABUS** (user labels).

   The dialogs for VSE/VSAM objects automatically update STDLABUP. STDLABUS, however, must be updated for user files created in space not managed by VSE/VSAM by using skeleton STDLABUS.
2. Once standard labels are written, $0JCL calls procedure **SETSDL**. This procedure writes phase names into the system directory list (SDL) and optionally loads selected phases into the SVA.
3. After SETSDL completes, $0JCL calls procedure **$COMVAR** to identify the CPU where startup is to be performed and the SETPARM procedure to be used. For a single CPU system, the name of the SETPARM procedure shipped with z/VSE is **CPUVAR1**.
4. $0JCL then calls the startup program **DTRISTRT**. DTRISTRT decides which startup mode is to be used. It bases its decision on the information stored in CPUVAR1 and the startup mode requested by the operator (if any). DTRISTRT also decides about BASIC ($$JCLBSX) and MINI ($$JCLMIN) startup. You can modify the startup only, if DTRISTRT is included in $0JCL.
5. When DTRISTRT completes, $0JCL calls procedure **LIBSDL** to load phases for VTAM, CICS, REXX, High Level Assembler, LE/VSE, the VSE C Language Run-Time Support, as well as connector-related programs into the SVA.
6. Next, $0JCL calls procedure **ALLOC** for allocating static partitions.

   If startup mode is BASIC or MINI, procedure ALLOCBSX is called instead.
7. The security server partition is started by job SECSERV according to the return code of program BSSINIT. The related variable in CPUVARn is set. As shipped, the security server runs in the FB partition. The security server must run in a partition that is not under the control of VSE/POWER. BSSINIT is executed again to identify the security server partition. Based on the result, program BSTPSTS is executed, which is the actual security server program. The security server is stopped after VSE/POWER is terminated with the PEND command.
8. Once the partitions are allocated, partition F1 is started and $1JCL activates VSE/POWER.
9. $1JCL calls the following procedures:
   a. **DTRICCF** for assigning the VSE/ICCF DTSFILE.
   b. **DTRPOWER** for assigning VSE/POWER files.
   c. **$COMVAR** to identify the CPU where startup is to be performed and the SETPARM procedure to be used.
   d. **CPUVAR1** for retrieving environment information.
   e. **LIBDEF** for defining standard LIBDEF search chains for libraries and sublibraries.
   f. **LIBDUMP** for assigning the dump library for VSE/POWER.

g. **POWSTRTn** to start all VSE/POWER-controlled partitions allocated (except F1, which is already running). The following VSE/POWER startup procedures are provided:

   POWSTRTA (predefined environment A)

   POWSTRTB (predefined environment B)

   POWSTRTC (predefined environment C)

   When a partition has been started via POWSTRTn, the JCL ASI procedure for that partition is processed: $2JCL, $3JCL, and so on.

   Note the following:

   - If startup mode is COLD, the VSE/POWER queues are also formatted.
   - If startup mode is BASIC, partitions FB and F1 are started. The procedure $1JCLBSX:
     – formats temporary VSE/POWER queues on SYSWK1
     – brings up the F1 partition
     – starts partitions BG, F2, F3, F4, and F5.
   - If startup mode is MINI, $1JCLMIN initializes a two-partition system (BG and F1).

10. After partition startup completes, $0JCL resumes processing. The BG partition is now under control of VSE/POWER.

    Depending on the startup mode, the following procedures are called:

```
Mode:   Procedure(s) called:

WARM    USERBG
COLD    COLDJOBS, USERBG
BASIC   COLDJOBS, BASICBG
MINI    MINIBG
```

    These procedures determine the remaining startup activities.

USERBG

USERBG is processed if startup mode is WARM or COLD. It calls the procedures SPLEVEL, LIBDEF, $COMVAR, and CPUVAR1, and releases the startup jobs VTAMSTRT and CICSICCF.

COLDJOBS

COLDJOBS is processed if startup mode is COLD or BASIC. The procedure reloads jobs provided by z/VSE into the VSE/POWER reader queue.

BASICBG

BASICBG is processed if startup mode is BASIC. It loads the startup jobs VTAMBSX and CICSBSX into the VSE/POWER reader queue.

Together with COLDJOBS, BASICBG completes the building of a basic system without user modifications. A basic system helps you recover from a problem caused, for example, by your own modifications.

MINIBG

MINIBG is processed if startup mode is MINI. It completes the building of a two partition system, BG and F1 (VSE/POWER). You can use such a system for library maintenance, for example.

Procedures CPUVARn and $COMVAR

One or more CPUVARn procedures are the memory of the startup process. A CPUVARn procedure consists of JCL SETPARM statements which contain system variables. These variables describe each partition's status. Each CPU requires its own CPUVAR**n** procedure, where **n** identifies the CPU. For a single CPU system, z/VSE provides procedure CPUVAR1.

Procedure $COMVAR identifies the CPU number and the related CPUVARn procedure of the CPU on which the startup is to be performed.

Refer to "CPUVARn and Related Startup Processing" on page 25 for an example of a CPUVARn procedure.

Considerations for Tailoring System Startup

Plan for extensive testing if you modify the system startup. Before you implement your changes, ensure that they are error free and that system startup still works correctly.

Procedures and Jobs You Should Not Change

There are a number of startup procedures and jobs which you should not change. These procedures and jobs control the startup modes BASIC and MINI. Both startup modes enable you to bring up z/VSE even in case of problems. It is, therefore, essential that you can always perform a BASIC or MINI startup. The following procedures and jobs are used for a BASIC or MINI startup and should not be changed:

- $1JCLBSX
- $1JCLMIN
- $BJCLMIN
- BASICBG
- MINIBG
- ALLOCBSX
- CICSBSX
- VTAMBSX

Considerations for BASIC and MINI Startup

You should be aware of the following problems that may arise when you are using a MINI or a BASIC startup in a modified environment.

- BASIC startup

  A BASIC startup uses for partition allocations the procedure ALLOCBSX as shipped with z/VSE. If you change the virtual storage values in your IPL procedure or generate a supervisor which requires considerably more storage, the storage required by BASIC startup for partition allocation may not be available. You may then get a system with only a limited number of partitions active.

- MINI startup

  In an environment with VSE/POWER **shared spooling**, you have regenerated and modified the VSE/POWER phase IPWPOWER and probably given it a name of your own. If you select a MINI startup, the system uses the original

IPWPOWER phase in which shared spooling is defined as **SHARED=NO**. To avoid a possible damage of your VSE/POWER files, you must first shut down all the other involved VSE systems before you select a MINI startup.

Note: The VSE/POWER queue file resides in the partition GETVIS area. The partition allocation for VSE/POWER in case of a MINI startup is based on the **default** size of the VSE/POWER queue file. If you have increased the size of the queue file, it does no longer fit into the partition GETVIS area and the MINI startup does not work. If this is the case, perform a BASIC startup to get a system with the BG and F1 partition active.

Job Control Language Used

The startup procedures and jobs are written in the job control language (JCL). JCL functions such as conditional job control, symbolic parameters, and nested procedures can be used. When modifying system startup, you must adhere to the rules valid for these functions. For a detailed description of these functions, refer to the manual *z/VSE Guide to System Functions* under "Controlling Jobs".

A CPUVARn procedure consists of JCL SETPARM statements. The SETPARM statement is discussed in the manual *z/VSE System Control Statements* under "SETPARM".

Considerations for Naming Conventions

When tailoring system startup, you must plan for the names you are going to use for the changed startup procedures and jobs. The skeletons provided for startup tailoring reflect the original startup members and show the names used by z/VSE. These names reveal to you the logical structure and relationship of the startup procedures and jobs.

Using the Same Names as z/VSE

You can use the same names as z/VSE. That is, you do not change the names given in the skeletons. As a result, the changed member replaces the original member in libraries IJSYSRS.SYSLIB (and in PRD2.SAVE). This means that the original startup member is lost. Consequently, if your changes are incorrect, you will get startup problems. Although the manual *z/VSE Guide for Solving Problems* provides hints to overcome such a situation, try to avoid such a problem altogether. Use *one* of the following methods:

- Create backup copies of those startup members you are going to change. Create new ones as required but assign different names to them. After successful testing, rename them to the original z/VSE-supplied name.
- Change the skeletons (if necessary) so that your changed version is not stored in PRD2.SAVE, but only in IJSYSRS.SYSLIB. The original member is then still available in PRD2.SAVE. After successful testing, save the changed version in PRD2.SAVE as well.

Note: z/VSE uses PRD2.SAVE to save members. This is to ensure that the latest version of a member is not lost when performing an FSU (Fast Service Upgrade).

Using Your Own Naming Convention

If you want to use your own names, invent names that allow you easy identification of your own startup procedures and jobs.

For the JCL startup procedures **$nJCLnnn**, you must observe the following naming rules:

- The procedure name must always start with **$n** (where n is the partition number).
- The characters (maximum of six) following the first two ($n) characters must be the same for all JCL procedures.

So, if you choose your own eight character procedure names, you can change the last six characters. But these six characters must be the same for all JCL procedures required for a **single** startup.

The names chosen by you are not known to the system; you have two ways to define them to z/VSE:

- The operator enters them during IPL.
- You specify the names in a $ASIPROC master procedure. Refer to "Using a $ASIPROC Procedure" on page 14 for details.

CPUVARn and Related Startup Processing

SETPARM procedure CPUVARn controls startup processing together with startup program DTRISTRT and others. **CPUVAR1.PROC** is the default CPUVARn procedure shipped with z/VSE and shown in Figure 5.

Figure 5. Example of a CPUVAR1 Procedure

```
// SETPARM XSPINIT=FIRST          * SYSTEM VARIABLES *
// SETPARM XENVNR=A
// SETPARM DASD=''
// SETPARM XDOSRES=''
// SETPARM TPMODE=''
// SETPARM TAPECUU=''
// SETPARM TAPEMD1=''
// SETPARM TAPEMD2=''
// SETPARM CPUMODE=''
// SETPARM XS=''
// SETPARM XPWCNTL=ALL
// SETPARM XCUST=NOAUTO
// SETPARM XUSEBG=B0              * PROGRAM RUNNING IN PARTITION *
// SETPARM XUSEF1=PW
// SETPARM XUSEF2=CI
// SETPARM XUSEF3=VT
// SETPARM XUSEF4=B4
// SETPARM XUSEF5=B5
// SETPARM XUSEF6=NONE
// SETPARM XUSEF7=NONE
// SETPARM XUSEF8=NONE
// SETPARM XUSEF9=NONE
// SETPARM XUSEFA=NONE
// SETPARM XUSEFB=NONE
// SETPARM XSTATBG=INACTIVE       * STATUS OF PARTITION *
// SETPARM XSTATF1=INACTIVE
// SETPARM XSTATF2=INACTIVE
// SETPARM XSTATF3=INACTIVE
// SETPARM XSTATF4=INACTIVE
// SETPARM XSTATF5=INACTIVE
// SETPARM XSTATF6=''
// SETPARM XSTATF7=''
// SETPARM XSTATF8=''
// SETPARM XSTATF9=''
// SETPARM XSTATFA=''
// SETPARM XSTATFB=''
```

```
// SETPARM XPARTB0=BG             * PARTITION PROGRAM IS RUNNING IN *
// SETPARM XPARTPW=F1
// SETPARM XPARTCI=F2
// SETPARM XPARTVT=F3
// SETPARM XPARTB4=F4
// SETPARM XPARTB5=F5
// SETPARM XSECP=FB
// SETPARM XPWMODE=COLD           * STARTUP MODE FOR KEY PROGRAMS *
// SETPARM XSTRTPW=WARM
// SETPARM XSTRTCI=''
// SETPARM XSTRTVT=''
// SETPARM XBASIC=NONE
// SETPARM XCOLD=NONE
// SETPARM XMODEBG=COLD           * PARTITION STARTUP MODE *
// SETPARM XMODEF1=COLD
// SETPARM XMODEF2=COLD
// SETPARM XMODEF3=COLD
// SETPARM XMODEF4=COLD
// SETPARM XMODEF5=COLD
// SETPARM XMODEF6=''
// SETPARM XMODEF7=''
// SETPARM XMODEF8=COLD
// SETPARM XMODEF9=''
// SETPARM XMODEFA=''
// SETPARM XMODEFB=''
// SETPARM XAPPLF2=DBDCCICS       * CICS APPLICATION NAMES *
// SETPARM XAPPLF8=PRODCICS
// SETPARM SSLCAUT=NO             * SSL CLIENT AUTHENTICATION *
// SETPARM XFATD='N'              * FAT-3390 *
```

PARAMETERS:

CPUMODE
(reflects the supervisor mode; X stands for ESA)

DASD
(disk device type of DOSRES)

DISTRIB
(distribution medium)

TAPECUU
(tape address of installation tape)

TAPEMD1
(tape mode is streaming)

TAPEMD2
(tape mode is non-streaming)

TPMODE
(VTAM)

XDOSRES
(disk address of DOSRES)

XS
(subarea number if z/VSE is an unattended node; it is included for compatibility reasons only since unattended node environments are no longer supported)

XSECP
(security server partition)

VARIABLES:

Note: From z/VSE 5.1 onwards, CICS/VSE (Program Number 5686-026) is no longer shipped on the Extended Base Tape. Therefore, all references to CICS/VSE have been removed from this topic.

- **XAPPLyy** (where yy is the partition ID: F2 or F8)

Use: Identifies the application names of the CICS systems running per default in F2 and F8 (CICS TS). Must be changed if other partitions are used.

Value: DBDCCICS | PRODCICS

Set: For F2 during initial installation (DBDCCICS). For F8 (PRODCICS) by user when installing the corresponding CICS system.

- **XBASIC**

Use: Keeps request for a system BASIC startup from a z/VSE component program. This request will be processed at the next startup.

Value: BASIC | NONE

Initial:
Program DTRISTRT sets it to NONE after processing of first startup after initial installation.

Set: By z/VSE program or user application.

If a job or procedure requests the next startup to be a system BASIC start, the following JCL statement has to be used (assuming a CPU number of 2):

```
// EXEC DTRSETP,SIZE=AUTO,PARM='CPUVAR2;;SET XBASIC=BASIC'
```

- **XCOLD**

Use: Keeps request for a system COLD start. This request will be processed at the next startup.

Value: COLD | NONE

Initial:
Program DTRISTRT sets it to NONE after processing the request.

Set: By z/VSE program or user application. For example, when extending VSE/POWER queues or installing VSE/POWER PNET.

- **XCUST**

Use: Keeps request for automatic customization of an unattended node (it is included for compatibility reasons only since unattended nodes are no longer supported).

Value: AUTO | NOAUTO

Initial:
NOAUTO

Set: Via DTRSETP program during installation of the unattended node system at the service node.

- **XENVNR**

Use: Contains the character of the predefined environment selected during initial installation. This number determines the POWSTRTn procedure to be called by the JCL procedure of the VSE/POWER partition ($1JCL).

Value: A, B, or C.

Set: During initial installation.

- **XMODEyy** (where yy is a partition ID: BG, F1 to FB)

Use: Contains the startup mode to be performed for a particular partition.

Value: MINI | BASIC | COLD | RECOV | WARM

Set: Program DTRISTRT decides on the startup mode for each partition and sets all variables XMODEyy accordingly.

- **XPARTzz** (where zz is the symbolic partition ID as defined in XUSEyy)

  Use: Keeps partition ID of symbolic partition. Necessary for mapping to actual partitions. For example, statement START &XPARTPW in the BG ASI procedure will be changed (if this variable contains the value F1), into the statement START F1 to start the VSE/POWER partition.

  Value: BG | F1 through FB

  Set: By program DTRISTRT, but not for a BASIC startup. For a BASIC startup, the use of the partitions is predetermined.

- **XPWCNTL**

  Use: Keeps status of partition control during installation of an unattended node system (it is included for compatibility reasons only since unattended nodes are no longer supported).

  Value: ALL | NET

  Initial:
  ALL

  Set: By the system during initial installation of an unattended node system at the service node.

- **XPWMODE**

  Use: Contains the startup mode to be executed for the VSE/POWER partition.

  Value: Same as for XMODEyy above.

  Set: By program DTRISTRT. DTRISTRT decides on the startup mode for each partition and sets this variable accordingly.

- **XSECP**

  Use: Server partition as specified in the IPL SYS command.

  Value: FB | PARTITION ID

  Initial:
  FB

  Set: By the system during startup of BG partition.

- **XSPINIT**

  Use: Used by program DTRISTRT to keep track of how the installation of z/VSE is progressing.

  Value: FIRST | INSTALL | FINISHED

  Initial:
  FIRST

  Set: Is reset to INSTALL at initial installation time and set to FINISHED when initial installation is complete.

- **XSTATyy** (where yy is a partition ID: BG, F1 to FB)

  Use: Keeps status of each partition. This is used by the startup program DTRISTRT to decide between a WARM or RECOV (recovery) startup.

  Value: ACTIVE | INACTIVE | blank

  Set: By program DTRSETP: to ACTIVE when startup job begins, to INACTIVE when startup job ends.

- **XSTRTzz** (where zz is the symbolic partition ID as defined in XUSEyy)

  Use: Keeps request for a COLD startup from a z/VSE program or user application. This request will be processed at the next startup.

  Value: COLD | NONE

  Set: By z/VSE programs or user applications. Program DTRISTRT sets it to NONE after processing the request.

- **XUSEyy** (where yy is a partition ID: BG, F1 to FB)

  Use: Indicates use of partition. The value is used to construct the name of related variables. For example, VT to build the names XSTRTVT and XPARTVT.

  Value: NONE, or two alphanumeric characters that represent a symbolic partition ID. For example: B0, PW, CI, VT, C2, B4, and so on.

  Note: If you want to exclude a partition from startup processing, you must specify a value of NONE.

  The following values are reserved:

  B0 = Background partition

  PW = VSE/POWER

  CI = CICS with VSE/ICCF

  VT = VTAM

  Initial:
  Standard use of partitions:

  XUSEBG = B0

  XUSEF1 = PW

  XUSEF2 = CI

  XUSEF3 = VT

  XUSEF4 = B4

  XUSEF5 = B5

  Set: By **customer** when the use of a partition changes.

  Note that it is permitted to set several XUSEyy variables to the same value. For example, to B4 for a batch partition. The XUSEyy variables must be present for each partition generated in the supervisor.

- **SSLCAUT**

  YES Implement client authentication. For details, see Chapter 44, “Configuring for Client Authentication,” on page 505.

  NO Do not implement client authentication.

- **XFATD**

YES A FAT-3390 disk is used with VSE/VSAM. For details, see "Define Space" on page 215.

NO A FAT-3390 disk is not used with VSE/VSAM.

Startup Program DTRISTRT

DTRISTRT is activated by the JCL startup procedure for the BG partition: $0JCL. At that time, the system variables of CPUVARn still reflect the status of the latest shutdown or startup (if shutdown was not performed or not successful).

DTRISTRT analyzes the information stored in CPUVARn for each partition before making a decision about the startup mode. The JCL startup procedures retrieve the variables from CPUVARn to initiate startup with the proper startup mode.

In addition, DTRISTRT issues messages that allow the operator to intervene and request startup modes **MINI**, **BASIC**, or **COLD**, but not WARM or RECOV (recovery). Intervention is possible if it was requested using the:
- The IPL load parameter.
- Command **MSG BG** within a three-second time limit, after message IESI0211 was issued by z/VSE during the startup of partition BG.

Refer to the manual *z/VSE Guide to System Functions* under "Interrupt IPL Processing for Modifications" for details.

For the parameters and the syntax to be used when calling DTRISTRT, refer to "Tracing Startup Processing" on page 31. There, you also find the return codes issued by DTRISTRT.

Security Initialization During Startup

The z/VSE predefined environments use FB as security server partition. It is recommended not to switch to another partition for running the security server.

The name of the Basic Security Manager (BSM) server routine is BSTPSTS. If an External Security Manager (ESM) also requires a server partition like the BSM, the name BSTPSTS must be replaced by the name of the ESM server routine in the server partition startup procedure $BJCLxxx, which is the default.

BSSINIT is the common security initialization routine for the BSM or an ESM. The parameter setting in the IPL SYS command controls the initialization process. If neither an ESM initialization phase (SYS ESM=phase) nor recovery mode (SYS SEC=RECOVER) has been specified, BSSINIT will start BSM initialization. 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).

If the IPL SYS command includes SEC=YES, a minimum protection of libraries and files is active during startup. This protection is based on DTSECTAB definitions and is in effect until the BSM or an ESM is active and takes over protection. The BSM requires the security server.

Return Codes from BSSINIT

The return codes of the common security initialization routine BSSINIT provide status information for conditional processing during startup in the $xJCLxxx procedure. Note that these return codes are no error indicators.

```
0         Do nothing for security (security may be inactive)
1 to 11   Identifies the server partition (11, for example, is the FB partition)
99        Indicates that the current partition is the server partition
```

Other Startup Programs

Besides startup program DTRISTRT, the following programs are also involved in startup processing. For tailoring startup, you should know what they are used for:

DTRIBASE
Identifies system characteristics such as teleprocessing access method and environment number.

DTRIINIT
Loads jobs into the VSE/POWER reader queue.

This program is a general utility program. The manual *z/VSE System Utilities* describes the program in detail under “DTRIINIT Utility”.

DTRISCPU
Compares the CPU ID given as input parameter with the actual CPU ID. Called by $COMVAR if more than one CPU involved.

DTRSETP
Updates startup procedure CPUVARn.PROC.

IESWAITT
Waits an unlimited time until VTAM is active.

IESWAIT
Waits the specified number of seconds to allow the operator to request a startup mode. The number of seconds is passed as parameter with the PARM operand. See also parameter 6 under “Tracing Startup Processing.”

Tracing Startup Processing

z/VSE provides a parameter (TEST1) that allows you to trace the input processed and the output created by the startup program DTRISTRT. You can use this parameter to test your startup modifications. You call DTRISTRT with the following statement:

```
// EXEC DTRISTRT,SIZE=AUTO,PARM='CPUVAR2;;;;TEST1'
```

The parameters that can be specified are positional and must be enclosed in single quotes (')and separated by semicolons (;). The following parameters (PARM=) are supported:

Parameter 1:
Name of SETPARM procedure (CPUVAR2).

Parameter 2:
Used by z/VSE for BASIC startup.

Parameter 3:
Used by z/VSE for MINI startup.

Parameter 4:
Reserved.

Parameter 5:
Trace request (TEST1).

Parameter 6:
Number of seconds (two digits) for operator to request startup mode (IESWAIT). The default is 10 seconds.

The parameters of interest here are 1 (CPUVAR2) and 5 (TEST1). Specifying TEST1 for parameter 5 causes the trace to be activated. The trace information is shown on SYSLOG and printed on SYSLIST. Figure 6 shows a sample portion of a startup trace.

```
IESI0216I LOG DTRISTRT USING MEMBER CPUVAR2.PROC IN IJSYSRS.SYSLIB.
IESI0231I SYNTAX ERROR IN STATEMENT "SETPARM ...  ". STATEMENT WILL BE IGNORED.
IESI0211I ALL PARTITIONS WILL BE INITIALIZED IN xxxxxx START MODE. IF YOU
          WANT TO INTERRUPT ENTER "MSG BG".
IESI0217I LOG DTRISTRT INPUT FROM MEMBER : XUSEBG =B0.
IESI0217I LOG DTRISTRT INPUT FROM MEMBER : XBASIC =NONE.
IESI0217I LOG DTRISTRT INPUT FROM MEMBER : XCOLD =NONE.
IESI0218I LOG DTRISTRT PROCESSING BG DECIDES ON RECOV STARTUP MODE.
IESI0219I LOG DTRISTRT OUTPUT INTO MEMBER : XMODEBG=RECOV.
IESI0219I LOG DTRISTRT OUTPUT INTO MEMBER : XPARTB0=BG.
IESI0217I LOG DTRISTRT INPUT FROM MEMBER : XUSEFB =NONE.
IESI0219I LOG DTRISTRT OUTPUT INTO MEMBER : XMODEFB DELETED.
IESI0217I LOG DTRISTRT INPUT FROM MEMBER : XUSEFA =NONE.
IESI0219I LOG DTRISTRT OUTPUT INTO MEMBER : XMODEFA DELETED.
⋮
```

Figure 6. Portion of a Startup Trace

DTRISTRT issues the following return codes:

00 Successful processing (not MINI startup).

01 Successful processing (MINI startup).

08 Function partially executed (not MINI startup). Processing continues. Possible errors:
  Maximum number of variables exceeded.
  Syntax error in SETPARM statement.

09 Function partially executed (MINI startup). Processing continues. For possible errors see return code 08.

12 Error occurred. Processing is terminated. Possible errors:
  Parameter syntax incorrect.
  Library full on output.

16 Severe error occurred. Processing is terminated. Possible errors:
  Phase not found.
  GETVIS space exhausted.

Modifying Startup Processing Using CPUVARn Information

For example, you may want to process a user-written procedure in case of a specific partition startup condition. The following is assumed:

```
Partition: F8
CPU:       CPU1
Procedure: $8JCL.PROC
SETPARM procedure: CPUVAR1
Condition: COLD startup
```

To identify a COLD startup, you have to retrieve the corresponding system variable from CPUVAR1. Your statements in $8JCL may look as follows:

```
// EXEC PROC=CPUVAR1,XMODEF8
// IF &XMODEF8 = COLD THEN
// GO TO USRPROC
      .
      .
      .
/. USRPROC
// EXEC PROC=MYPROC
      .
      .
      .
```

Modifying Startup When Installing an Additional Program

Most likely you want to install additional programs on top of your z/VSE system. For example, z/VSE optional programs or your own application programs.

For startup, the following changes are required:

1. Update system variables XUSEyy and XSTATyy (set to INACTIVE) in CPUVARn.PROC for the partition used with the utility program DTRSETP. Refer to the manual *z/VSE System Utilities* under “DTRSETP Utility” for details about how to use this utility.
2. Catalog the startup job into a VSE sublibrary and load it into the VSE/POWER reader queue by using skeleton SKLOAD described under “Skeleton SKLOAD for Loading a Job” on page 63. Add the startup job also to the COLDJOBS load list for a COLD startup by using skeleton SKCOLD described under “Skeleton SKCOLD for Loading User Jobs During a COLD Startup” on page 61.
3. Update USERBG.PROC by including the name of the startup job by using skeleton SKUSERBG.
4. Update the LIBDEF chain if necessary.

Using Synchronization Points

Program DTRSETP provides a WAIT function that allows you to synchronize partitions at the JCL level. For example, partition F5 waits for partition F4 to reach a certain point in processing. To use this function, you must add a variable to the startup procedure CPUVARn. For example, USYNC01. Use program DTRSETP to add such a variable.

During system startup USYNC01 must be reset; to the value NO, for example. To do this, you must insert in the BG ASI procedure ($0JCL) an EXEC statement for program DTRSETP. Insert the statement after the JOB statement and before the EXEC DTRISTRT statement. Use skeleton SKJCL0 for that purpose.

The program running in F5 must use operation WAIT of program DTRSETP to initiate a wait loop for checking repeatedly the status of USYNC01. If the program running in F4 has reached its particular point of processing, it must use operation SET of program DTRSETP to set USYNC01 to an agreed value. This value will be recognized by the program running in F5 and synchronization can occur.

Refer to the manual *z/VSE System Utilities* under "DTRSETP Utility" for a detailed description of program DTRSETP.

Synchronizing Partition Startup Using IESWAITR Procedure

IESWAITR may be used to synchronize startup of:

- TCP/IP and CICS, for example, if the CICS TS Web Support (CWS) is to be used. If IESWAITR is coded in the CICS startup (see also skeletons SKCICS and SKCICS2), it ensures that TCP/IP is up and running when CICS is started.
- TCP/IP and the VSE Connector Server.
- Other applications and the DB2 Server.

IESWAIT is also used to check that DMF jobname `DMFSTART` has successfully initialized.

Changing Startup for DASD Sharing

For an environment with DASD sharing (sharing any disk devices among two or more CPUs, for example SCSI disks), you must create procedures CPUVAR2 through CPUVARn by tailoring procedure $COMVAR accordingly. Skeleton SKCOMVAR can be used to tailor procedure $COMVAR. Refer to "Skeleton SKCOMVAR for Tailoring $COMVAR Procedure" on page 63 for details.

For general guidelines for DASD sharing, refer to the manual *z/VSE Guide to System Functions* under "DASD Sharing with Multiple VSE Systems".

Changing Startup When Lock File Is Stored On SCSI DASD

Related Topic:
- "Using Shared SCSI Disks" on page 111

In an environment with DASD sharing where the lock file is stored on a SCSI DASD (sharing disk devices among two or more CPUs) *and the FCP adapter is not configured for NPIV mode*, you must use the dialog *Tailor IPL Procedure* (Fast Path **242**) to define a *physical* FCP adapter for the lock file.

To define a lock file on a SCSI disk, you must:

1. In the *Tailor IPL Procedure* dialog, enter '**2**' (= ALTER) next to the IPL procedure you wish to change, and press Enter.
2. Enter a '**1**' next to DLF ("Modify Lock File Definition") and press Enter. The *Lock File Definition* panel is then displayed:

```
 TAS$ICM4        TAILOR IPL PROCEDURE: LOCK FILE DEFINITION

 Enter the required data and press ENTER.

The  DLF  command  must be used if DASD sharing is specified in the supervisor.
If DASD sharing is not specified, any DLF command must be deleted.

VOLUME SERIAL................ WORK01     Volume Serial of lock file DASD
                                         (Enter blanks to delete command)
NUMBER OF CPUS.............. 9          Number of CPUs to be shared
START ADDRESS................ 200        Starting cylinder or block number
LOCK FILE SIZE............... 80         Number of cylinders or blocks (leave
                                         blank for VSE defaults)
SECURED DATA SET ?........... 1          2 = no, 1 = yes
FORMAT....................... 2          Formatting required (2 = no, 1
                                         = yes)
FCP.......................... ___        cuu of the FCP adapter,if lock file
                                         on SCSI
 WARNING: If the lock file was previously defined by  another  system,  specify
only the volume serial number and leave the other parameters blank.
 PF1=HELP      2=REDISPLAY  3=END
```

3. In the field “FCP” you enter the address of the FCP adapter you wish to use. If you are **not** using NPIV mode (see the note below), you specify the address of the FCP adapter used for connecting this SCSI DASD.

   Note:
   a. If you want to allocate a lock file on a SCSI disk, you have to have a unique FCP adapter installed for each VSE system sharing the lock file, and access the lock file via this unique FCP. The reason is, that the hardware does not reserve the SCSI disk (RESERVE command) per FCP cuu, but only *per FCP adapter*. This restriction does not apply if the FCP adapter has been configured in NPIV mode. *In this case you may leave the FCP parameter blank.*
   b. This operand will be ignored for non-SCSI disks (for example, ECKD™ or any other FBA device).
4. After you press Enter, the IPL procedure (for example, $IPLESA) will be updated with the above details. It is catalogued in the IJSYSRS library.

For further details about implementing SCSI support, see Chapter 8, “Configuring Your System to Use SCSI Disks,” on page 95.

Using Skeletons for Tailoring System Startup

z/VSE provides skeletons that help you tailor startup procedures and jobs. The skeletons are provided in VSE/ICCF library 59 and reflect the original startup members as provided by z/VSE for the predefined environments.

Note:
1. From z/VSE 5.1 onwards, CICS/VSE (Program Number 5686-026) is no longer shipped on the Extended Base Tape. Therefore, all references to CICS/VSE have been removed from this topic.
2. Before you change a skeleton, copy it to your primary VSE/ICCF library. Make your changes to that copy, not to the original. For details refer to “Copying Skeletons” on page 11.

On the following pages you find the statements of each startup skeleton listed. Those variables and names that have to be changed or that are likely candidates for changes are described and printed in bold. In general, you can change any statement of a skeleton (except for those mentioned under “Procedures and Jobs

You Should Not Change" on page 23). You must then find out, however, how such changes impact the statements of other skeletons and change them accordingly. Following is a list of skeletons available for tailoring system startup.

- Skeletons for static partition allocations:
  - SKALLOCA = Environment A (entry system)
  - SKALLOCB = Environment B (medium system)
  - SKALLOCC = Environment C (large system)
- Skeletons for starting up partitions:
  - SKJCL0 = BG partition
  - SKJCL1 = F1 partition (VSE/POWER)
  - SKJCL2-SKJCL9 = F2-F9 partitions
  - SKJCLA = FA partition
  - SKJCLB = FB partition
  - SKJCLDYN = Dynamic partitions
- Skeletons for called procedures and jobs:
  - SKUSERBG = Job release and LIBDEF processing
  - SKPWSTRT = VSE/POWER autostart
  - SKLIBCHN = Define library search chains (LIBDEFs)
  - SKCICS = Startup job CICS Transaction Server and VSE/ICCF (F2)
  - SKVTAM = Startup job VTAM (F3)
  - SKTCPSTR = Startup job TCP/IP (F7)
  - SKCOLD = Adding jobs for COLD startup
  - SKLOAD = Load job into VSE/POWER reader queue
  - SKCOMVAR = DASD sharing
  - SKVTASTJ = Startup of the Virtual Tape Server partition
  - SKVCSSTJ = Startup of the Connector Server partition
- Skeletons for ***environments B and C***
  - SKCICS2 = Startup job second CICS Transaction Server (without VSE/ICCF). For details, see Chapter 11, "Installing a Second Predefined CICS Transaction Server," on page 145.

For skeletons related to the z/VSE connectors support, refer to the *z/VSE e-business Connectors User's Guide*.

Skeleton for Cataloging Startup Changes (SKENVSEL)

Figure 7 on page 37 shows the statements of skeleton *SKENVSEL*. With it, you catalog changed startup procedures and jobs in library IJSYSRS.SYSLIB. In addition, copies of the cataloged startup members are saved in library PRD2.SAVE.

```
* $$ JOB JNM=ENVCAT,DISP=D,CLASS=0
// JOB ENVCAT
// EXEC LIBR,PARM='MSHP'
ACCESS S=IJSYSRS.SYSLIB
* $$ SLI ICCF=(SKALLOCA),LIB=(YY)
* $$ SLI ICCF=(SKALLOCB),LIB=(YY)
* $$ SLI ICCF=(SKALLOCC),LIB=(YY)
* $$ SLI ICCF=(SKJCL0),LIB=(YY)
* $$ SLI ICCF=(SKJCL1),LIB=(YY)
* $$ SLI ICCF=(SKJCL2),LIB=(YY)
* $$ SLI ICCF=(SKJCL3),LIB=(YY)
* $$ SLI ICCF=(SKJCL4),LIB=(YY)
* $$ SLI ICCF=(SKJCL5),LIB=(YY)
* $$ SLI ICCF=(SKJCL6),LIB=(YY)
* $$ SLI ICCF=(SKJCL7),LIB=(YY)
* $$ SLI ICCF=(SKJCL8),LIB=(YY)
* $$ SLI ICCF=(SKJCL9),LIB=(YY)
* $$ SLI ICCF=(SKJCLA),LIB=(YY)
* $$ SLI ICCF=(SKJCLB),LIB=(YY)
* $$ SLI ICCF=(SKBGSTRT),LIB=(YY)
* $$ SLI ICCF=(SKJCLDYN),LIB=(YY)
* $$ SLI ICCF=(SKUSERBG),LIB=(YY)
* $$ SLI ICCF=(SKPWSTRT),LIB=(YY)
* $$ SLI ICCF=(SKLIBCHN),LIB=(YY)
/*
/&
* $$ EOJ
```

Figure 7. Skeleton SKENVSEL for Cataloging Startup Changes

Skeleton SKALLOCC is used with the large Environment C, that was introduced with z/VSE 3.1.

Note: Several skeletons for startup are not included in SKENVSEL for the following reasons:
- They contain the necessary JCL and JECL statements for cataloging.
- Their output is cataloged in IJSYSRS.SYSLIB and loaded directly into the VSE/POWER reader queue (skeletons SKCICS, SKCICS2, and SKVTAM).

Each *** $$ SLI** statement includes a startup procedure created by a skeleton. Delete all SLI statements for skeletons that you did not modify. For example, you must delete 2 of the 3 statements for partition allocation procedures. The **YY** variable indicates your VSE/ICCF primary library that contains your modified copy of the skeleton.

After you have modified the skeleton, enter the following command from the editor's command line:

```
@DTRSEXIT
```

This command calls a macro that deletes specific comments from the file. You should do this *before* filing the skeleton.

Finally, submit job ENVCAT for processing.

Skeletons for Static Partition Allocations

Figure 8 on page 39, Figure 9 on page 40, Figure 10 on page 41 show the statements of the skeletons provided for static partition allocations. The comments included in the skeletons are not shown.

Note: ALLOC R and SETPFIX In releases prior to VSE/ESA 1.3, the skeletons for static partition allocations included ALLOC R definitions. Since VSE/ESA 1.3, the ALLOC R definition has been replaced by the JCL command SETPFIX. SETPFIX and ALLOC R are to be used as follows:

- SETPFIX command is to be used to set limits per partition for fixing pages (PFIX).
- An ALLOC R definition is to be used only to define storage per partition available for programs to be executed in **real mode**.

SETPFIX definitions are included in the startup procedures $1JCL (for VSE/POWER), VTAMSTRT (for VTAM), and CICSICCF (for CICS); the associated skeletons are SKJCL1, SKVTAM, and SKCICS. For TCP/IP the skeleton used is SKTCPSTR. For the second predefined CICS, the name of the startup procedure is CICS2, the name of the skeleton is SKCICS2.

Each skeleton's CATALOG statement shows the procedure name **ALLOC**. z/VSE uses this name for the allocation procedures provided, however, you may use your own name instead. To help you make a decision, read first “Considerations for Naming Conventions” on page 24. If you decide to use your own name, you must replace ALLOC (in skeletons SKALLOCx and SKJCL0) with the name you have chosen.

TCP/IP using OSAX-links require approximately 1 MB PFIX storage (above 16 MB) per link. For further details, refer to the skeleton SKTCPSTR.

After you have modified the skeleton, enter the following command from the editor's command line:

```
@DTRSEXIT
```

This command calls a macro that deletes specific comments from the skeleton. You should do this *before* filing the skeleton.

Use skeleton **SKENVSEL** for cataloging your changes. Refer to “Skeleton for Cataloging Startup Changes (SKENVSEL)” on page 36 for details.

Skeleton SKALLOCA

This skeleton applies to predefined environment A including 12 static partitions (one crossing the 16 MB line).

```
CATALOG ALLOC.PROC DATA=YES REPLACE=YES
ALLOC F1=6M
SIZE F1=1500K
ALLOC BG=6M
SIZE BG=1280K
ALLOC F2=50M
SIZE F2=2048K
ALLOC F3=15M
SIZE F3=600K
ALLOC F4=20M
SIZE F4=2M
ALLOC F5=1M
SIZE F5=768K
ALLOC F6=512K
SIZE F6=256K
ALLOC F7=20M
SIZE F7=1M
ALLOC F8=50M
SIZE F8=2M
ALLOC F9=512K
SIZE F9=256K
ALLOC FA=512K
SIZE FA=256K
ALLOC FB=1M
SIZE FB=512K
SYSDEF DSPACE,DSIZE=20M,COMMAX=20
NPGR BG=255,F2=255,F3=100,F4=200,F5=50,F6=50,F7=100,F8=200
NPGR F9=50,FA=50,FB=50
/+
CONN S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY ALLOC.PROC REPLACE=YES
```

Figure 8. Skeleton SKALLOCA (Static Partition Allocations)

Note: The SYSDEF command defines the VTAM® and VTAM application requirements for data spaces. This includes the VTAM requirements for VSE/POWER and CICS. This is also reflected in the // EXEC IPWPOWER,... statements of the VSE/POWER startup procedure (skeleton SKPWSTRT), the // EXEC DFHSIP,... statements of the CICS startup procedure (skeleton SKCICS), and the // EXEC ISTINCVT,... statement of the VTAM startup procedure (skeleton SKVTAM).

The value defined for DSIZE is a minimum value which must not be reduced. Add your own requirements for data spaces to this predefined value. The DSIZE value given here reflects the status at the time the manual was printed. The actual value shipped with z/VSE may be higher.

Skeleton SKALLOCB

This skeleton applies to predefined environment B prepared for a second CICS TS. It includes 12 static partitions (four crossing the 16 MB line).

```
CATALOG ALLOC.PROC   DATA=YES       REPLACE=YES
ALLOC BG=10M
SIZE BG=1280K
ALLOC F1=30M
SIZE F1=1500K
ALLOC F2=50M
SIZE F2=2M
ALLOC F3=15M
SIZE F3=600K
ALLOC F4=20M
SIZE F4=2M
ALLOC F5=5M
SIZE F5=768K
ALLOC F6=50M
SIZE F6=1M
ALLOC F7=20M
SIZE F7=1M
ALLOC F8=150M
SIZE F8=2M
ALLOC F9=5M
SIZE F9=1M
ALLOC FA=5M
SIZE FA=1M
ALLOC FB=1M
SIZE FB=512K
SYSDEF DSPACE,DSIZE=40M,COMMAX=20
NPGR BG=255,F2=255,F3=100,F4=200,F5=100,F6=100,F7=100,F8=200
NPGR F9=100,FA=100,FB=50
/+
CONN S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY ALLOC.PROC         REPLACE=YES
```

Figure 9. Skeleton SKALLOCB (Static Partition Allocations)

The SYSDEF command defines the VTAM and VTAM application requirements for data spaces. This includes the requirements for VSE/POWER and CICS. This is also reflected in the:

- // EXEC IPWPOWER,... statements of the VSE/POWER startup procedure (skeleton SKPWSTRT).
- // EXEC DFHSIP,... statements of the CICS startup procedures (skeletons SKCICS and SKCICS2).
- // EXEC ISTINCVT,... statement of the VTAM startup procedure (skeleton SKVTAM).

The value defined for DSIZE is an average size covering most application needs. It leaves about 4 MB for non-VTAM applications like TCP/IP for VSE/ESA or CICS shared data tables. The actual value shipped with z/VSE may be higher.

Skeleton SKALLOCC

This skeleton applies to predefined environment C prepared for a large environment. It includes 12 static partitions (eleven crossing the 16 MB line).

```
CATALOG ALLOC.PROC   DATA=YES       REPLACE=YES
ALLOC F1=32M
SIZE F1=1500K
ALLOC BG=32M
SIZE BG=1280K
ALLOC F2=256M
SIZE F2=2M
ALLOC F3=15M
SIZE F3=600K
ALLOC F4=32M
SIZE F4=2M
ALLOC F5=32M
SIZE F5=1M
ALLOC F6=32M
SIZE F6=1M
ALLOC F7=32M
SIZE F7=1M
ALLOC F8=512M
SIZE F8=2M
ALLOC F9=32M
SIZE F9=1M
ALLOC FA=32M
SIZE FA=1M
ALLOC FB=2M
SIZE FB=512K
SYSDEF DSPACE,DSIZE=256M,COMMAX=20
NPGR BG=255,F2=255,F3=100,F4=200,F5=100,F6=100,F7=100,F8=200
NPGR F9=100,FA=100,FB=50
/+
CONN S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY ALLOC.PROC         REPLACE=YES
```

Figure 10. Skeleton SKALLOCC (Static Partition Allocations)

The SYSDEF command defines the VTAM and VTAM application requirements for data spaces. This includes the requirements for VSE/POWER and CICS. This is also reflected in the:

- // EXEC IPWPOWER,... statements of the VSE/POWER startup procedure (skeleton SKPWSTRT).
- // EXEC DFHSIP,... statements of the CICS startup procedures (skeletons SKCICS and SKCICS2).
- // EXEC ISTINCVT,... statement of the VTAM startup procedure (skeleton SKVTAM).

The value defined for DSIZE is an average size covering most application needs. It leaves about 240 MB for non-VTAM applications like TCP/IP for VSE/ESA or CICS shared data tables. The actual value shipped with z/VSE may be higher.

Skeletons for Starting Up BG Partition

You can use the skeletons *SKJCL0* and *SKUSERBG* to tailor startup for the BG partition. The comments included in the skeletons are not shown in the following figures. To understand the functions included in the startup for the BG partition, refer to “Overview of Startup Processing” on page 18.

Both skeletons are shipped in VSE/ICCF library 59. “Skeleton SKJCL0 (Startup Procedure for BG Partition)” on page 42 shows the contents of skeleton SKJCL0 and “Skeleton SKUSERBG (Startup Procedure for BG Partition)” on page 46 the contents of skeletons SKUSERBG. Certain AR commands can be included or are recommended for inclusion in the $0JCL procedure. They are listed under “Including AR Commands” on page 45.

Skeleton SKJCL0 (Startup Procedure for BG Partition)

$0JCL is the procedure name used by z/VSE but you may use your own name instead. To help you make a decision, read first “Considerations for Naming Conventions” on page 24.

Note: The name for this procedure must always start with **$0** and the following six characters must be equal for all related JCL startup procedures.

If you decide to use your own name, you must update $ASIPROC (if used) or the operator must interrupt IPL processing to enter the correct name.

```
CATALOG $0JCL.PROC   DATA=YES   REPLACE=YES
STDOPT ACANCEL=NO,DECK=NO,DUMP=PART,SYSDUMP=YES,SXREF=YES
SYSDEF DSPACE,DSIZE=15M
SYSDEF SYSTEM,NTASKS=255,TASKS=OLD
// VDISK UNIT=FDF,BLKS=2880,VOLID=VDIDLA,USAGE=DLA
*  VDISK UNIT=CUU,BLKS=81920,VOLID=VDIWRK
// EXEC PROC=STDLABEL                    CALLS ALSO STDLABUP AND STDLABUS LOAD VDISK
// EXEC PROC=SETSDL                      SET SDL
PRTY BG,FA,F9,F8,F6,F5,F4,F2,F7,FB,F3,F1
ASSGN SYSLST,IGN
// JOB BGINIT
// LIBDEF DUMP,CATALOG=SYSDUMP.BG,PERM
// SETPARM XNCPU=' '
// EXEC PROC=$COMVAR,XNCPU
// EXEC DTRISTRT,SIZE=AUTO,PARM='CPUVAR&XNCPU;$$JCLBSX;$$JCLMIN'
/*
// SETPARM RETCODE=$RC
// SETPARM XSPINIT='FINISHED '
// SETPARM XMODEBG='MINI'
// SETPARM XPARTPW='F1'
// SETPARM XPWMODE='WARM'
// IF RETCODE=1 OR RETCODE=9 THEN
// GOTO ALLOCBSX
// EXEC PROC=CPUVAR&XNCPU,XMODEBG,XPARTPW,XPWMODE,XSPINIT
// IF XSPINIT ¬= FINISHED THEN
// GOTO NOSDL
```

Workfiles on Virtual Disk:

If you want to place workfiles on a virtual disk, you must activate the following statements in the skeleton:

```
*  VDISK UNIT=CUU,BLKS=81920,VOLID=VDIWRK
*  EXEC PROC=IESWORK      DEFINE WORK FILES ON VIRTUAL DISK   (on next page)
// EXEC PROC=LIBSDL                  PROVIDE CORRECT LIBDEF FOR SET SDL
SET SDL
LIST=$SVAVTAM
LIST=$SVACICS
LIST=$SVAREXX
LIST=$SVAASMA
LIST=$SVACONN
LIST=$SVACEE
LIST=$SVAEDCM
/*
* ------------------------------------------------------------------
* DEPENDING ON THE SVA SIZE AND ON THE LANGUAGES USED IN LE/VSE
* YOU MAY ADDITIONALLY LOAD RUNTIME SUPPORT FOR PL/I AND/OR COBOL.
* MOVE THE APPROPRIATE LIST STATEMENT ABOVE.
* ADDITIONAL AMOUNT OF STORAGE AS OF GA TIME:
*
*                                R-MODE 24   RMODE=ANY
* LIST=$SVAIGZM    LE COBOL          0KB       160KB
* LIST=$SVAIBMM    LE PL/I          36KB       208KB
* * ------------------------------------------------------------------
// LIBDROP PHASE
/. NOSDL
EXPLAIN ON
```

```
// IF XMODEBG=BASIC THEN
// GOTO ALLOCBSX
// EXEC PROC=ALLOC     CHANGE TO PROCNAME DEFINED IN SKALLOCx  x=A,B,C
*  EXEC PROC=IESWORK      DEFINE WORK FILES ON VIRTUAL DISK
// GOTO SECSTRT
/. ALLOCBSX
// EXEC PROC=ALLOCBSX     ALLOCs FOR BASIC START
// SETPARM XPARTPW=F1
*         IF YOU CHANGED THE PRIORITY IN  YOUR NORMAL SYSTEM
*         YOU HAVE TO RESET IT FOR BASIC START TO:
*         PRTY BG,FA,F9,F8,F6,F5,F4,F2,F7,FB,F3,F1
EXPLAIN ON
// PAUSE
/. SECSTRT
// ON $RC=99 GOTO ERROR
// EXEC BSSINIT
/*
// SETPARM RETCODE=$RC
// IF RETCODE¬=0 THEN
// GOTO PARTFB
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=RECOVERY
/*
// GOTO PWRSTRT
/. PARTFB
// IF RETCODE¬=11 THEN
// GOTO PARTFA
// IF XPARTPW = FB THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=FB
/*
// GOTO PWRSTRT
/. PARTFA
// IF RETCODE¬=10 THEN
// GOTO PARTF9
// IF XPARTPW = FA THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=FA
/*
// GOTO PWRSTRT
/. PARTF9
// IF RETCODE¬=9 THEN
// GOTO PARTF8
// IF XPARTPW = F9 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F9
/*
// GOTO PWRSTRT
/. PARTF8
// IF RETCODE¬=8 THEN
// GOTO PARTF7
// IF XPARTPW = F8 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F8
...
/*
// GOTO PWRSTRT
/. PARTF7
// IF RETCODE¬=7 THEN
// GOTO PARTF6
// IF XPARTPW = F7 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F7
/*
// GOTO PWRSTRT
/. PARTF6
```

```
// IF RETCODE¬=6 THEN
// GOTO PARTF5
// IF XPARTPW = F6 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F6
/*
// GOTO PWRSTRT
/. PARTF5
// IF RETCODE¬=5 THEN
// GOTO PARTF4
// IF XPARTPW = F5 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F5
/*
// GOTO PWRSTRT
/. PARTF4
// IF RETCODE¬=4 THEN
// GOTO PARTF3
// IF XPARTPW = F4 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F4
/*
// GOTO PWRSTRT
/. PARTF3
// IF RETCODE¬=3 THEN
// GOTO PARTF2
// IF XPARTPW = F3 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F3
/*
// GOTO PWRSTRT
/. PARTF2
// IF RETCODE¬=2 THEN
// GOTO PARTF1
// IF XPARTPW = F2 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F2
/*
/. PARTF1
// IF RETCODE¬=1 THEN
// GOTO ERROR
// IF XPARTPW = F1 THEN
// GOTO ERROR
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XSECP=F1
/*
/. PWRSTRT
SET MRCZERO
START &XPARTPW
STOP
ASSGN SYSIN,FEC,PERM
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
// IF XSPINIT ¬= FIRST THEN
// GOTO SKIP
* ==================================== *
*                                      *
*           INSTALLATION OF            *
// EXEC PROC=SPLEVEL
*                                      *
* ==================================== *
```

```
// EXEC PROC=LOADINST
// EXEC DTRSETP,PARM='CPUVAR1;;'
   SET XSPINIT=INSTALL
/*
// PWR PRELEASE RDR,INSTALL
// GOTO EXIT
/. SKIP
// IF XPWMODE=COLD OR XPWMODE=BASIC THEN
// GOTO COLDPART
// GOTO ENDCOLD
/. COLDPART
// ID USER=FORSEC                 !! NO PWD REQUIRED !!
// EXEC PROC=COLDJOBS
/. ENDCOLD
// IF XMODEBG ¬= MINI THEN
// GOTO NOTMINI
// EXEC PROC=MINIBG
// GOTO EXIT
/. NOTMINI
// IF XMODEBG ¬= BASIC THEN
// GOTO USER
// ID USER=FORSEC                !! NO PWD REQUIRED !!
// EXEC PROC=BASICBG
// GOTO EXIT
/. USER
// ID USER=FORSEC                !! NO PWD REQUIRED !!
// EXEC PROC=USERBG     CHANGE TO YOUR PROCNAME AS USED IN SKUSERBG
// GOTO EXIT
/. ERROR
* ERROR IN THE PARTITION SETUP FOR SECURITY SERVER. SERVER PARTITION
* MAY NOT BE &XPARTPW
/. EXIT
/&
/+
CONN S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY $0JCL.PROC      REPLACE=YES
```

In a system with security (access control) active, the // ID statements for user FORSEC ensure that startup procedures COLDJOBS and BASICBG have the appropriate access rights. In an ASI procedure no password is required for the ID statement.

Refer to chapter Chapter 33, “Access Rights/Checking in DTSECTAB,” on page 401 for details about the z/VSE access control support.

If you modified procedure USERBG and specified a different procedure name, you must change the procedure name here as well. Use the name entered in the CATALOG statement of skeleton SKUSERBG. Use skeleton **SKENVSEL** for cataloging your changes. Refer to “Skeleton for Cataloging Startup Changes (SKENVSEL)” on page 36 for details.

Including AR Commands

AR (attention routine) commands such as the following can be included in the $0JCL startup procedure for the BG partition:

```
BANDID
CACHE
DEBUG
EXPLAIN
FREE
LFCB
LUCB
OFFLINE
ONLINE
OPERATE
```

```
PRTYIO
RESERV
SETDF
SYSECHO
```

You find a detailed description of these commands in the manual *z/VSE System Control Statements*.

Skeleton SKUSERBG (Startup Procedure for BG Partition)

```
CATALOG USERBG.PROC DATA=YES REPLACE=YES
* START MODE FOR BG-PARTITION IS NORMAL
* *******************************************
*             YOUR SYSTEM IS                *
// EXEC PROC=SPLEVEL
* *******************************************
STDOPT DATE=MDY              CHANGE STANDARD OPTIONS IF WANTED
// EXEC PROC=LIBDEF               CHANGE TO YOUR OWN LIBDEF PROC
// EXEC ARXLINK                   INITIALIZE REXX/VSE
// LIBDEF DUMP,CATALOG=SYSDUMP.BG,PERM
*  TO ENABLE DB2, ENTER KEY AND CUSTOMER INFO, REMOVE ASTERISKS TO
*  ACTIVATE, CONTINUATION LINE START COLUMN 16.
*  BELOW KEY IS THE TRIAL KEY VALID FOR THE TRIAL PERIOD.
*  EXEC IVALPKEY,PARM='PRODUCT=DB2 KEY=0000-1111-2222-3333-4444 CUSTINF*
*              O=C111-111-1111'
*
* TO START CAPACITY MEASUREMENT, REMOVE ASTERISK BELOW.
* // EXEC PROC=CMTSTART
// SETPARM XNCPU=''
// EXEC PROC=$COMVAR,XNCPU              GET CPU NUMBER
// SETPARM XENVNR=''
// SETPARM SSLCAUT=''
// EXEC PROC=CPUVAR&XNCPU,XENVNR,SSLCAUT
// PWR PRELEASE RDR,VTAMSTRT            OR YOUR VTAM (SKVTAM)
// EXEC IESWAIT,PARM='03'
/*
// IF SSLCAUT NE YES THEN
// GOTO NOCAUT
// EXEC BSSDCERT,PARM='ACT'
/*
/. NOCAUT
// PWR PRELEASE RDR,CICSICCF              OR YOUR CICS (SKCICS)
// PWR PRELEASE RDR,CEEWARC               LE - AR INTERFACE
* // PWR PRELEASE RDR,CICS2               OR YOUR CICS2 (SKCICS2)
* // PWR PRELEASE RDR,TCPIP00             OR YOUR TCP/IP STARTUP
* // PWR PRELEASE RDR,STARTVCS            OR YOUR CONNECTOR SERVER
/+
CONN S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY USERBG.PROC      REPLACE=YES
```

USERBG is the procedure name used by z/VSE. If you modify the procedure, you may want to change the procedure name as well. To help you make a decision, read first "Considerations for Naming Conventions" on page 24.

If you decide to use your own name, you must also update procedure $0JCL. Use skeleton SKJCL0 and modify the corresponding EXEC PROC statement.

Procedure USERBG calls procedure LIBDEF and releases the startup jobs VTAMSTRT and CICSICCF (and CICS2 if installed). If you modified any of these and specified your own names, replace these names accordingly:
- LIBDEF
- VTAMSTRT
- CICSICCF
- CEEWARC

- CICS2
- STARTVCS

Use the name you entered in the CATALOG statement of the procedure or job changed. The names of the skeletons are:
- SKLIBCHN
- SKVTAM
- SKCICS
- SKCICS2
- SKTCPSTR
- SKVCSSTJ

Use skeleton **SKENVSEL** for cataloging your changes. Refer to “Skeleton for Cataloging Startup Changes (SKENVSEL)” on page 36 for details.

Enabling the DB2 Server for VSE

To enable the DB2 Server for VSE you must activate the EXEC IVALPKEY statement. Refer also to “System Provided Application Profiles” on page 133.

Skeletons for Starting Up VSE/POWER

You can use skeletons *SKJCL1* and *SKPWSTRT* to modify the startup of VSE/POWER in partition F1.

Both skeletons are shipped in VSE/ICCF library 59. Figure 11 on page 48 and “Skeleton SKPWSTRT (VSE/POWER Warm and Cold Starts)” on page 49 show the contents of the skeletons. Comments included in the skeletons are not shown.

Skeleton SKJCL1

```
CATALOG $1JCL.PROC DATA=YES REPL=YES
// JOB POWSTART
// OPTION SADUMP=5
// EXEC PROC=DTRICCF                  ASSIGNMENTS FOR DTSFILE
// EXEC PROC=DTRPOWR                  ASSIGNMENTS FOR VSE/POWER
// SETPARM DASD=''
// SETPARM XNCPU=''
// SETPARM XENVNR=''
// SETPARM XPWMODE=''
// SETPARM XPARTPW=''
// SETPARM XSECP=''
// SETPARM XSPINIT=''
// EXEC PROC=$COMVAR,XNCPU
// EXEC PROC=CPUVAR&XNCPU,DASD,XENVNR,XPWMODE,XPARTPW,XSECP,XSPINIT
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTAT&XPARTPW=ACTIVE'
/*
// EXEC PROC=LIBDEF
// EXEC PROC=LIBDUMP
// SETPFIX LIMIT=300K
// ID USER=FORSEC                     !! NO PWD REQUIRED !!
// EXEC PROC=POWSTRT&XENVNR,XPWMODE,XSECP
// ID USER=DUMMY                      !! NO PWD REQUIRED !!
/*
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTAT&XPARTPW=INACTIVE'
/*
// IF XSPINIT = FIRST OR XSPINIT = INSTALL THEN
// GOTO NOSEC
* ------------------------------------------------------------
* SECURITY SERVER PARTITION WILL BE STOPPED
* YOU MAY ENTER A PAUSE STATEMENT HERE IN CASE YOU DON'T WANT
* TO ALWAYS STOP. YOU WOULD HAVE TO ENTER // GOTO NOSEC IN
* CASE YOU DON'T WANT TO STOP.
* ------------------------------------------------------------
// EXEC DTRIATTN,PARM='MSG &XSECP,DATA=STOPNOREP'
/*
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;SET XSECP=RECOVER'
/*
/. NOSEC
/&
/+
CONN S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY $1JCL.PROC         REPLACE=YES
```

Figure 11. Skeleton SKJCL1 (Startup Procedure for VSE/POWER Partition)

If you modify this procedure, you may also change the procedure name $1JCL. To help you make a decision, read first "Considerations for Naming Conventions" on page 24. Note that the name for this procedure **must** always start with **$1** and the following six characters must be equal for all related JCL startup procedures.

In a system with security (access control) active, the // ID statement for user FORSEC ensures the appropriate access rights for procedure POWSTRTn. The // ID statement for user DUMMY turns off these access rights. In an ASI procedure no password is required for the ID statement.

Refer to Chapter 33, "Access Rights/Checking in DTSECTAB," on page 401 for details about the z/VSE access control support.

If you decide to use your own name, you must update $ASIPROC (if used) or the operator must interrupt IPL processing to enter the correct name.

For an explanation of the SETPFIX definition, refer to “Skeletons for Static Partition Allocations” on page 38. The EXEC PROC statement for LIBDEF calls the procedure that defines library search chains and assignments for the partitions which are controlled by VSE/POWER. If you modified the LIBDEF procedure (skeleton SKLIBCHN) and specified your own procedure name, change this statement accordingly. Use the name you entered in the CATALOG statement of SKLIBCHN.

The EXEC PROC statement for POWSTRT calls the procedure that defines warm and cold starts. &XENVNR identifies the predefined environment chosen (A, B, or C). If you modified this procedure (skeleton SKPWSTRT) and specified your own procedure name, change this statement accordingly. Use the name you entered in the CATALOG statement of SKPWSTRT.

After you have modified the skeleton, enter the following command from the editor's command line to delete specific comments from the skeleton:

```
@DTRSEXIT
```

Use skeleton *SKENVSEL* for cataloging your changes. Refer to “Skeleton for Cataloging Startup Changes (SKENVSEL)” on page 36 for details.

Skeleton SKPWSTRT (VSE/POWER Warm and Cold Starts)

This skeleton is in two parts; one for a COLD, the other for a WARM startup. The n in POWSTRTn.PROC reflects the predefined environment selected during initial installation: A, B, or C.

If you modify this procedure, you may also change the procedure name POWSTRTn (n identifies the environment selected during initial installation). Refer also to “Considerations for Naming Conventions” on page 24. If you decide to use your own name, you must update the corresponding EXEC PROC statement in procedure $1JCL (skeleton SKJCL1).

The statements

```
SET SYSID=Y
SET PNET=YYYYYYYY
```

for shared spooling and networking are optional. Delete them if you do not use these functions.

The statement

```
SET SECNODE=AAAA
```

is for a multiple-node environment with security (access control) active. It is required to distinguish the nodes for security processing.

The statement

```
SET SECAC=NO
```

indicates that the job and its output are not Spool Access Protected. NO is the default value.

The statement:

```
SET SJECL=YES
```

supports the cataloging of VSE/POWER jobs into a VSE library.

The statement:

```
SET WORKUNIT=PA
```

allows VSE/POWER to run on multiple CPUs in parallel.

A statement:

```
SET RBF=nnnnn
```

will cause VSE/POWER to cancel a job when the limit *nnnnn* for the number of output records (99999 in the skeleton) has been reached.

Do not change the following statement:

```
DEFINE L,CICSDATA,3F00,1,255,*
```

It specifies for the CICS Report Controller a Resource Security Level (RSL) value for reports generated by batch programs. Refer to "DEFINE: Specifying User-Defined Output Operands" in the manual *VSE/POWER Administration and Operation* for a detailed description of the DEFINE command.

The other DEFINE statements are for PSF/VSE (Print Services Facility/VSE), a z/VSE optional program which provides advanced printing support for certain types of IBM printers.

The VSE/POWER

```
PLOAD DYNC,ID=n,FORCE
```

command (shown on the following page) is optional and loads the dynamic class table DTR$DYNn (DTR$DYNC is the name of the table shipped with z/VSE) defining the parameters required for activating dynamic partitions. Dynamic partitions are supported for all predefined environments.

Note: This skeleton assumes that the Security Server runs in partition FB. If you have selected another partition for the Security Server, which is not recommended, you need to modify the skeleton accordingly. If there is a need to select another partition, you must select a static partition which is not controlled by VSE/POWER.

```
CATALOG POWSTRTn.PROC    REPL=YES  DATA=YES
// EXEC DTRWAITP
// IF XPWMODE ¬= COLD THEN
// GOTO WARM
// IF XSECP = FB THEN
// GOTO COLDFB
// ASSGN SYSLST,UA
// EXEC IPWPOWER,DSPACE=2M
SET SYSID=Y
SET PNET=YYYYYYYY
SET SECNODE=AAAA
SET SECAC=NO
SET SJECL=YES
SET WORKUNIT=PA
DEFINE L,CICSDATA,3F00,1,255,*
DEFINE L,CKPTPAGE,4,1,2,B,1,32767
DEFINE L,FORMDEF,1D,1,6,C
DEFINE L,PAGEDEF,1F,1,6,C
DEFINE L,PIMSG,21,2,3,C
DEFINE L,DATACK,2022,1,8,C
DEFINE L,FORMS,10,1,4,C
DEFINE L,PRMODE,18,1,8,C
DEFINE L,TRC,1A,1,3,C
FORMAT=D,A
PSTART BG,A0I
READER=FEC
PRINTERS=FEE
```

```
PUNCHES=FED
PSTART F2,L2
READER=FEC
PRINTERS=FEE,FEF
PUNCHES=FED
PSTART F3,K3
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F4,J4
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F5,H5
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F6,M6
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F7,N7
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F8,P8
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F9,Q9
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART FA,TV
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART FB,BU
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART DEV,YYYYYYYY,SYSPSFXX,P
PSTART TASKTR,ENAB,10
* PLOAD DYNC,ID=n,FORCE
/*
// GOTO EXIT
/. COLDFB            COLD START, SECURITY SERVER IS FB
// ASSGN SYSLST,UA
// EXEC IPWPOWER,DSPACE=2M
SET SYSID=Y
SET PNET=YYYYYYYY
SET SECNODE=AAAA
SET SECAC=NO
SET SJECL=YES
SET WORKUNIT=PA
DEFINE L,CICSDATA,3F00,1,255,*
DEFINE L,CKPTPAGE,4,1,2,B,1,32767
DEFINE L,FORMDEF,1D,1,6,C
DEFINE L,PAGEDEF,1F,1,6,C
DEFINE L,PIMSG,21,2,3,C
DEFINE L,DATACK,2022,1,8,C
DEFINE L,FORMS,10,1,4,C
DEFINE L,PRMODE,18,1,8,C
DEFINE L,TRC,1A,1,3,C
FORMAT=D,A
PSTART BG,A0I
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F2,L2
READER=FEC
PRINTERS=FEE,FEF
PUNCHES=FED
PSTART F3,K3
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F4,J4
READER=FEC
PRINTERS=FEE
```

```
PUNCHES=FED
PSTART F5,H5
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F6,M6
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F7,N7
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F8,P8
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F9,Q9
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART FA,TV
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART DEV,YYYYYYYY,SYSPSFXX,P
PSTART TASKTR,ENAB,10
* PLOAD DYNC,ID=n,FORCE
/*
// GOTO EXIT
/. WARM              POWER WARM START
// IF XSECP = FB THEN
// GOTO WARMFB
// EXEC IPWPOWER,DSPACE=2M
SET SYSID=Y
SET PNET=YYYYYYYY
SET SECNODE=AAAA
SET SECAC=NO
SET SJECL=YES
SET WORKUNIT=PA
DEFINE L,CICSDATA,3F00,1,255,*
DEFINE L,CKPTPAGE,4,1,2,B,1,32767
DEFINE L,FORMDEF,1D,1,6,C
DEFINE L,PAGEDEF,1F,1,6,C
DEFINE L,PIMSG,21,2,3,C
DEFINE L,DATACK,2022,1,8,C
DEFINE L,FORMS,10,1,4,C
DEFINE L,PRMODE,18,1,8,C
DEFINE L,TRC,1A,1,3,C
FORMAT=NO
PSTART BG,A0I
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F2,L2
READER=FEC
PRINTERS=FEE,FEF
PUNCHES=FED
PSTART F3,K3
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F4,J4
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F5,H5
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F6,M6
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F7,N7
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F8,P8
READER=FEC
```

```
PRINTERS=FEE
PUNCHES=FED
PSTART F9,Q9
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART FA,TV
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART FB,BU
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART DEV,YYYYYYYY,SYSPSFXX,P
PSTART TASKTR,ENAB,10
* PLOAD DYNC,ID=N,FORCE
/*
// GOTO EXIT
/. WARMFB            WARM START, SECURITY SERVER IS FB
// EXEC IPWPOWER,DSPACE=2M
SET SYSID=Y
SET PNET=YYYYYYYY
SET SECNODE=AAAA
SET SECAC=NO
SET SJECL=YES
SET WORKUNIT=PA
DEFINE L,CICSDATA,3F00,1,255,*
DEFINE L,CKPTPAGE,4,1,2,B,1,32767
DEFINE L,FORMDEF,1D,1,6,C
DEFINE L,PAGEDEF,1F,1,6,C
DEFINE L,PIMSG,21,2,3,C
DEFINE L,DATACK,2022,1,8,C
DEFINE L,FORMS,10,1,4,C
DEFINE L,PRMODE,18,1,8,C
DEFINE L,TRC,1A,1,3,C
FORMAT=NO
PSTART BG,A0I
READER=FEC
RINTERS=FEE
PUNCHES=FED
PSTART F2,L2
READER=FEC
PRINTERS=FEE,FEF
PUNCHES=FED
PSTART F3,K3
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F4,J4
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F5,H5
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F6,M6
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F7,N7
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F8,P8
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART F9,Q9
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART FA,TV
READER=FEC
PRINTERS=FEE
PUNCHES=FED
PSTART DEV,YYYYYYYY,SYSPSFXX,P
PSTART TASKTR,ENAB,10
* PLOAD DYNC,ID=N,FORCE
```

```
/*
/. EXIT
/+
CONNECT S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY POWSTRTn.PROC      REPLACE=YES
```

There are four commands for each of the twelve partitions named in the skeleton. For example:

```
PSTART BG,A0I
READER=FEC
PRINTERS=FEE
PUNCHES=FED
```

These four commands are required for each partition. They define some of the partition's operating characteristics and its device configuration for spooling. The manual *VSE/POWER Administration and Operation* describes these commands in detail.

z/VSE uses the first PRINTERS/PUNCHES as default device. For that reason, your real device should precede FEE or FED, respectively.

If more than one device is specified for PRINTERS or PUNCHES, it is recommended to code the POWER® * $$ LST and the * $$ PUN statements for device selection. Otherwise, the LST and PUN parameters are associated with the first device and the system will take VSE/POWER default values when spooling to a device other than the first device.

Be sure to delete the commands for the partition(s) that you do not want to have autostarted. Do this for both the cold and the warm start definitions.

You can use any device address. However, the device must be defined in the hardware configuration table. This is done either during initial installation by device sensing or by using the *Configure Hardware* dialog.

After you have modified the skeleton, enter the following command from the editor's command line:

```
@DTRSEXIT
```

This command calls a macro that deletes specific comments from the skeleton. You should do this *before* filing the skeleton.

Use skeleton **SKENVSEL** for cataloging your changes. Refer to "Skeleton for Cataloging Startup Changes (SKENVSEL)" on page 36 for details.

Skeleton SKLIBCHN for Defining Library Search Chains

You can use the skeleton *SKLIBCHN* to define search chains and assignments for the batch partitions controlled by VSE/POWER. The contents of the skeleton are shown below. Comments included in the skeleton are not shown. The name of the procedure created by this skeleton is **LIBDEF**.

```
CATALOG   LIBDEF.PROC  DATA=YES  REPLACE=YES
// LIBDEF  PHASE,SEARCH=(PRD2.CONFIG,
              YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
              YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
              PRD1.BASE,PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE,
              PRD2.COMM,PRD2.COMM2,PRD2.AFP,PRIMARY.$$C),
              CATALOG=YYYYYYY.YYYYYYYY,
              PERM
// LIBDEF    OBJ,SEARCH=(PRD2.CONFIG,
              YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
```

```
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               PRD1.BASE,PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE,
               PRD2.COMM,PRD2.COMM2,PRD2.AFP,PRIMARY.$$C),PERM
// LIBDEF SOURCE,SEARCH=(PRD2.CONFIG,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               PRD1.BASE,PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE,PRD2.COMM,
               PRD2.COMM2,PRD2.AFP,PRIMARY.$$C,PRD1.MACLIB),PERM
/*
/+
CATALOG   LIBDEFS.PROC  DATA=YES  REPLACE=YES
// LIBDEF  PHASE,SEARCH=(PRD2.CONFIG,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               PRD1.BASED,PRD1.BASE,PRD2.SCEEBASD,PRD2.SCEEBASE,
               PRD2.PROD,PRD2.DBASE,PRD2.COMM,PRD2.COMM2,PRD2.AFP,
               PRIMARY.$$C),
               CATALOG=YYYYYYY.YYYYYYYY,
               PERM
// LIBDEF    OBJ,SEARCH=(PRD2.CONFIG,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               PRD1.BASED,PRD1.BASE,PRD2.SCEEBASD,PRD2.SCEEBASE,
               PRD2.PROD,PRD2.DBASE,PRD2.COMM,PRD2.COMM2,PRD2.AFP,
               PRIMARY.$$C),PERM
// LIBDEF SOURCE,SEARCH=(PRD2.CONFIG,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               PRD1.BASED,PRD1.BASE,PRD2.SCEEBASD,PRD2.SCEEBASE,
               PRD2.PROD,PRD2.DBASE,PRD2.COMM,PRD2.COMM2,PRD2.AFP,
               PRIMARY.$$C,PRD1.MACLIBD,PRD1.MACLIB),PERM
/*
/+
CONNECT S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY LIBDEF.PROC   REPLACE=YES
COPY LIBDEFS.PROC  REPLACE=YES
```

Note: The second part of skeleton SKLIBCHN, called LIBDEFS, includes in addition libraries PRD1.BASED, PRD2.SCEEBASD, and PRD1.MACLIBD. These libraries are required only when applying service (PTFs) indirectly. Further details about these libraries are provided in the manual *z/VSE System Upgrade and Service* under "Job Sequence for PTF Application".

If you replace the procedure name LIBDEF by a name of your own, you must also change the name in skeletons such as SKJCL1, SKUSERBG, and SKINITNN.

Replace the **Y** strings in the three LIBDEF statements to define your permanent sublibrary search chains for phase, object, and source members. Note that you should also include the system sublibraries as defined in the skeleton. If you do not, the Interactive Interface may not function correctly. In the LIBDEF statement for member type PHASE, the **Y** string of the CATALOG operand defines the sublibrary that the linkage editor uses to catalog phases.

The skeleton reflects the search chains established after initial installation. If you do not install optional programs or if you install them in your own user libraries, some of the sublibraries remain empty and may be deleted. After initial installation the following sublibraries are empty:
- PRD2.COMM
- PRD2.COMM2
- PRD2.DBASE
- PRD2.PROD
- PRD2.AFP
- PRD2.DB2750
- PRD2.DB2750C

- PRD2.DFHDOC
- PRD2.DB2STP
- PRD2.OSASF
- PRIMARY.$$C
- PRIMARY.SUF

The sublibrary PRD2.AFP is for the z/VSE optional program PSF/VSE and the related z/VSE optional programs OGL/370 and PPFA/370.

After you have modified skeleton SKLIBCHN, enter the following command from the editor's command line:

@DTRSEXIT

This command calls a macro that deletes specific comments from the file. You should do this *before* filing the skeleton. Use skeleton *SKENVSEL* for cataloging your changes. Refer to "Skeleton for Cataloging Startup Changes (SKENVSEL)" on page 36 for details.

Skeleton SKCICS for Starting Up the CICS Transaction Server and VSE/ICCF

Skeleton *SKCICS* creates a startup job for the CICS Transaction Server and VSE/ICCF in default partition F2. If you want to create your own startup procedure for these components, use skeleton SKCICS. The job stream created with SKCICS catalogs the startup job CICSICCF into IJSYSRS.SYSLIB and loads it directly into the VSE/POWER reader queue.

If you have to perform a BASIC startup (because of CICS startup problems, for example) perform the following steps:

- Perform BASIC startup.
- Use skeleton SKCICS to create and catalog members CICSICCF.Z and LDCICS.PROC into IJSYSRS.SYSLIB.
- Perform a normal startup.
- Run procedure LDCICS.PROC to load the startup job CICSICCF into the VSE/POWER reader queue.
- Release startup job CICSICCF to continue normal startup.

```
* $$ JOB JNM=CATCICS,DISP=D,CLASS=0
// JOB CATCICS     CATALOG CICSICCF AND LDCICS, LOAD CICSICCF
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG  CICSICCF.Z        REPLACE=YES
$$$$ JOB JNM=CICSICCF,DISP=L,CLASS=2,EOJMSG=YES
$$$$ LST CLASS=A,DISP=D,RBS=100
// JOB CICSICCF         CICS/ICCF STARTUP
// OPTION SADUMP=5
// OPTION SYSDUMPC
// UPSI 11100000
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASED,PRD1.BASE,PRD2.PROD,        X
               PRD2.SCEEBASD,PRD2.SCEEBASE,PRD2.DBASE,PRD1.MACLIBD,    X
               PRD1.MACLIB,PRD2.DFHDOC)
// LIBDEF DUMP,CATALOG=SYSDUMP.F2
// SETPARM XNCPU=''
// SETPARM XMODEF2=AUTO
// SETPARM XAPPLF2=''
// SETPARM XSPINIT=''
// SETPARM XENVNR=''
// SETPARM XSECP=''
// EXEC PROC=$COMVAR,XNCPU
```

```
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATF2=ACTIVE'
$$/*
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XAPPLF2=DBDCCICS'
$$/*
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XPARTCI=F2'
$$/*
// EXEC PROC=CPUVAR&XNCPU,XMODEF2,XAPPLF2,XSPINIT,XENVNR,XSECP
// SETPFIX LIMIT=144K
// EXEC PROC=DTRCICST                  ASSGNS FOR CICS FILES
// EXEC PROC=DTRINFOA                  ASSGNS FOR INFO ANAL FILES
// EXEC PROC=DTRICCF                   ASSGN FOR DTSFILE
// OPTION SYSPARM='00'
// ASSGN SYS006,UA
// ASSGN SYS007,UA
// ASSGN SYS008,UA
// ASSGN SYS009,SYSLOG
LOG
// ID USER=DBDCCICS
NOLOG
// EXEC DTSANALS                       RECOVER IF DTSFILE DESTROYED
RECOVER OPT
$$/*
```

In a system with security (access control) active, the // ID statement for user DBDCCICS ensures that CICSICCF has the appropriate access rights to the control file. If you submit the job, your access rights are inherited by the CICSICCF startup job in the VSE/POWER reader queue. If your access rights as a user are adequate, it is recommended to delete the ID statement from the skeleton to avoid an exposure of the password. Chapter 33, "Access Rights/Checking in DTSECTAB," on page 401 describes the access control support in detail.

Instead of CICSICCF, you may use a name of your own for the startup job. To help you make a decision, read first "Considerations for Naming Conventions" on page 24. If you use your own name, you must also update procedures USERBG and COLDJOBS (skeletons SKUSERBG and SKCOLD) with the name you have chosen since they also call CICSICCF. With skeleton SKCOLD you can add your own jobs to the load list of procedure COLDJOBS. Note that these jobs must be cataloged in a VSE library.

For an explanation of the SETPFIX definition, refer to "Skeletons for Static Partition Allocations" on page 38.

For CLASS (in the $$$$ JOB statement) specify the identifier of the partition in which CICS is running. The skeleton assumes F2.

EOJMSG=YES is needed for an unattended node environment (it is included for compatibility reasons only since unattended nodes are no longer supported). At end-of-job (EOJ), VSE/POWER issues a message which initiates automated shutdown or the processing of other command lists.

Replace **F2** with the identifier of the partition if you run CICS with VSE/ICCF in another partition than F2.

```
*   WAITING FOR VTAM TO COME UP
// EXEC IESWAITT
$$/*
*   WAITING FOR TCP/IP TO COME UP
* // EXEC REXX=IESWAITR,PARM='TCPIP00'
$$/*
// OPTION SYSPARM='00'
// ASSGN SYS020,SYSLST
// SETPARM ELIM=14M
```

```
// IF XENVNR = B THEN
// SETPARM ELIM=25M
// IF XENVNR = C THEN
// SETPARM ELIM=200M
// IF XMODEF2 = COLD THEN
// GOTO COLDST
// SETPARM XMODEF2=AUTO
// GOTO STARTCIC
/. COLDST
// SETPARM XMODEF2=COLD
/. STARTCIC
// IF XSECP = RECOVERY THEN
// GOTO RECO
// EXEC DFHSIP,SIZE=DFHSIP,PARM='APPLID=&XAPPLF2.,START=&XMODEF2.,EDSAL*
               IM=&ELIM.,SI',DSPACE=2M,OS390
SIT=SP,STATRCD=OFF,NEWSIT=YES,
$$$$ SLI MEM=IESVAEXC.Z,S=IJSYSRS.SYSLIB
$$/*
// GOTO STAT
/. RECO
// EXEC DFHSIP,SIZE=DFHSIP,PARM='APPLID=&XAPPLF2.,START=&XMODEF2.,EDSAL*
               IM=&ELIM.,SI',DSPACE=2M,OS390
SEC=NO,SIT=SP,STATRCD=OFF,NEWSIT=YES,
$$$$ SLI MEM=IESVAEXC.Z,S=IJSYSRS.SYSLIB
$$/*
/. STAT
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATF2=INACTIVE'
$$/*
$$/&
$$$$ EOJ
/+
CATALOG  LDCICS.PROC     REPLACE=YES DATA=YES
// EXEC DTRIINIT
   LOAD CICSICCF.Z
/*
/+
/*
// EXEC PROC=LDCICS        TO LOAD CICSICCF INTO RDR QUEUE
/&
* $$ EOJ
```

The application IESWAITT (called with // EXEC IESWAITT) must be defined to z/VSE as a VTAM application. This definition exists if you are using z/VSE as shipped by IBM. If this definition is removed because of modifications, you must use the *Maintain VTAM Application Names* dialog and redefine IESWAITT as application name (APPLID) to VTAM.

If there is a need to wait until TCP/IP is up, activate statement

```
* // EXEC REXX=IESWAITR,PARM='TCPIP00'
```

by removing the preceding asterisk and blank character provided that the TCP/IP startup job is TCPIP00.

Note that the default setting for SVA in the DFHSIT is YES.

After you have modified the skeleton, enter the following command from the editor's command line:

```
@DTRSEXIT
```

This command calls a macro that deletes specific comments from the skeleton. You should do this *before* filing the skeleton.

The skeleton includes additional $$ characters. They are needed to mask off VSE/POWER JECL statements. Program DTRIINIT, described in the manual *z/VSE System Utilities* under "DTRIINIT Utility", replaces the $$ characters with VSE/POWER JECL statements for cataloging.

Skeleton SKVTAM for Starting Up VTAM

The SKVTAM skeleton creates a startup job for VTAM running in default partition F3. The job stream created with SKVTAM catalogs the startup job (VTAMSTRT) into IJSYSRS.SYSLIB and loads it directly into the VSE/POWER reader queue.

If you have to perform a BASIC startup (because of VTAM startup problems, for example), perform the following steps:

- Perform BASIC startup.
- Use skeleton SKVTAM to create and catalog members VTAMSTRT.Z and LDVTAM.PROC into IJSYSRS.SYSLIB.
- Perform a normal startup.
- Run procedure LDVTAM.PROC to load the startup job VTAMSTRT into the VSE/POWER reader queue.
- Release startup job VTAMSTRT to continue normal startup.

```
* $$ JOB JNM=CATVTAM,DISP=D,CLASS=0
// JOB CATVTAM                 CATALOG VTAMSTRT AND LDVTAM, LOAD VTAMSTRT
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG VTAMSTRT.Z   REPLACE=YES
$$$$ JOB JNM=VTAMSTRT,DISP=L,CLASS=3
// JOB VTAMSTRT                              START VTAM
// OPTION DUMP,SADUMP=5
// SETPARM XNCPU=''
// EXEC PROC=$COMVAR,XNCPU
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATF3=ACTIVE'
$$/*
// SETPFIX LIMIT=424K
* // SETPFIX LIMIT=(,300K)
// ASSGN SYS000,UA
// ASSGN SYS001,DISK,VOL=SYSWK1,SHR        TRACE FILE ASSIGNMENT
// ASSGN SYS004,DISK,VOL=SYSWK1,SHR        TRACE FILE ASSIGNMENT
// ASSGN SYS005,DISK,VOL=SYSWK1,SHR        NCP LOAD/DIAGNOSIS FILE ASSIGNMENT
```

VTAMSTRT is the name of the job used by z/VSE for VTAM startup in default partition F3. You may use your own name instead. To help you make a decision, read first "Considerations for Naming Conventions" on page 24. If you use your own name, you must also update procedures USERBG and COLDJOBS (skeletons SKUSERBG and SKCOLD) with the name you have chosen since they also call VTAMSTRT. With skeleton SKCOLD you can add your own jobs to the load list of procedure COLDJOBS. Note that these jobs must be cataloged in a VSE library.

For an explanation of the SETPFIX definition, refer to "Skeletons for Static Partition Allocations" on page 38.

For CLASS (in the $$$$ JOB statement), specify the identifier of the partition in which VTAM is running. The skeleton assumes F3.

For the // ASSGN statements, specify the VOLIDs of the disk devices where each file resides. SYS000 **must** be unassigned (UA) because VTAM uses it internally. The TRACE program addresses the TRACE file as SYS001. The TPRINT program addresses it as SYS004.

```
// LIBDEF  PHASE,SEARCH=(PRD2.COMM,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               PRD2.COMM2,PRD2.CONFIG,PRD1.BASED,PRD1.BASE,
               PRD2.SCEEBASD,PRD2.SCEEBASE,PRD2.PROD),PERM
// LIBDEF    OBJ,SEARCH=(PRD2.COMM,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               PRD2.COMM2,PRD2.CONFIG,PRD1.BASED,PRD1.BASE),PERM
// LIBDEF SOURCE,SEARCH=(PRD2.COMM,
               YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,YYYYYYY.YYYYYYYY,
               PRD2.COMM2,PRD2.CONFIG,PRD1.BASED,PRD1.BASE),PERM
// LIBDEF   DUMP,CATALOG=SYSDUMP.F3,PERM
// EXEC ISTINCVT,SIZE=ISTINCVT,PARM='CUSTNO=C555-555-5555,VTAMPW=5979-4*
               015-4627-6185-9388',DSPACE=2M
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATF3=INACTIVE'
$$/*
$$/&
$$$$ EOJ
/+
CATALOG LDVTAM.PROC REPLACE=YES DATA=YES
// EXEC DTRIINIT
   LOAD VTAMSTRT.Z
/*
/+
/*
// EXEC PROC=LDVTAM              TO LOAD VTAM STARTUP INTO RDR QUEUE
/&
* $$ EOJ
```

Replace the **Y** strings in each SEARCH chain. The LIBDEF statements define the permanent sublibrary search chains for phase, object, source, and dump library members.

Replace **F3** with the identifier of the partition if VTAM runs in another partition than F3.

After you make the changes, run the DTRSEXIT macro. This macro deletes specific comments from the file. You should do this before you file the skeleton. On the command line, enter:

```
@DTRSEXIT
```

The skeleton includes additional $$ characters. They are needed to mask off VSE/POWER JECL statements. Program DTRIINIT, described in the manual *z/VSE System Utilities* under "DTRIINIT Utility", replaces $$ characters with VSE/POWER JECL statements for cataloging.

Skeleton SKTCPSTR for Starting Up TCP/IP

The job shown below uses the "special task user-ID" VCSRV. It must be submitted from an administrator user-ID (for example, SYSA) when batch security is active (during IPL, SYS SEC=YES). For further details about "special task user-IDs", see "Access Control and CICS Region Prefix" on page 411.

The SKTCPSTR skeleton creates a startup job for TCP/IP running in default partition F7. The job stream created with SKTCPSTR catalogs the procedure LDTCPIP into IJSYSRS.SYSLIB, which loads the startup job (TCPIP00) into VSE/POWER.

```
* $$ JOB JNM=CATTCPIP,DISP=D,CLASS=0
// JOB CATTCPIP CATALOG TCPIP00 AND LDTCPIP PROCEDURE LOAD TCPIP00
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG  TCPIP00.Z        REPLACE=YES
$$$$ JOB JNM=TCPIP00,DISP=L,CLASS=7,EOJMSG=YES
$$$$ LST CLASS=A,DISP=D,RBS=100
```

```
// JOB TCPIP00          TCP/IP  STARTUP
// ID USER=VCSRV
// OPTION SADUMP=5
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.SCEEBASE)
// SETPFIX LIMIT=(400K)
// SETPFIX LIMIT=(,2100K)
// EXEC PROC=DTRICCF
// EXEC IPNET,SIZE=IPNET,PARM='ID=00,INIT=IPINIT00',DSPACE=4M
$$/*
$$/&
$$$$ EOJ
/+
CATALOG  LDTCPIP.PROC     REPLACE=YES DATA=YES
// EXEC DTRIINIT
   LOAD TCPIP00.Z
/*
/+
CONNECT S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY TCPIP00.Z REP=YES
/*
// EXEC PROC=LDTCPIP       TO LOAD TCP/IP  INTO RDR QUEUE
/&
* $$ EOJ
```

You should increase the SETPFIX value above 16 MB by 1 MB per OSAX link. Therefore, a default value of 2100 KB would allow for two OSAX links.

Replace **F7** with the identifier of the partition if TCP/IP runs in another partition than F7.

After you make the changes, run the DTRSEXIT macro. This macro deletes specific comments from the file. You should do this before you file the skeleton. On the command line, enter:

```
@DTRSEXIT
```

The skeleton includes additional $$ characters. They are needed to mask off VSE/POWER JECL statements. Program DTRIINIT, described in the manual *z/VSE System Utilities* under “DTRIINIT Utility”, replaces $$ characters with VSE/POWER JECL statements for cataloging.

You might also need to update the following procedures:
- USERBG (SKUSERBG)
- COLDJOBS (SKCOLD)
- STARTVCS (SKVCSSTJ)

Skeleton SKCOLD for Loading User Jobs During a COLD Startup

During a COLD startup z/VSE processes procedure COLDJOBS. This procedure activates program DTRIINIT and provides a list of jobs to be loaded into the VSE/POWER reader queue. The loading of the jobs is done by program DTRIINIT. Refer to the manual *z/VSE System Utilities* under “DTRIINIT Utility” for further details about program DTRIINIT. This manual also points out what must be considered when loading jobs in a system with security (access control) active.

With skeleton SKCOLD you can add your own jobs to the load list of procedure COLDJOBS. Note that these jobs must be cataloged in a VSE library.

```
* $$ JOB JNM=CATALOG,CLASS=0,DISP=D
// JOB CATALOG
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
```

```
CATALOG COLDJOBS.PROC   R=Y  DATA=YES
// EXEC DTRIINIT
  LOAD CICSICCF.Z
  LOAD STARTVCS.Z    LOAD CONNECTOR SERVER JOB
  LOAD TAPESRVR.Z    LOAD TAPE SERVER JOB
  LOAD TCPIP00.Z     LOAD TCP/IP STARTUP JOB
  LOAD CICS2.Z       LOAD CICS2; IF YOU DO NOT USE CICS2 DELETE IT
/*
// EXEC DTRIINIT
   ACCESS PRD2.SCEEBASE
   LOAD CEEWOPTJ.Z
   LOAD CEEWARC.Z
/*
// ID USER=DUMMY,PWD=DUMMY
// EXEC DTRIINIT
  LOAD VTAMSTRT.Z
  LOAD PAUSEBG.Z
  LOAD PAUSEFA.Z
  LOAD PAUSEFB.Z
  LOAD PAUSEF1.Z
  LOAD PAUSEF2.Z
  LOAD PAUSEF3.Z
  LOAD PAUSEF4.Z
  LOAD PAUSEF5.Z
  LOAD PAUSEF6.Z
  LOAD PAUSEF7.Z
  LOAD PAUSEF8.Z
  LOAD PAUSEF9.Z
  LOAD PRTDUMPA.Z
  LOAD PRTDUMPB.Z
  LOAD PRTDUC2A.Z    LOADED FOR CICS2; IF YOU DO NOT USE CICS2 DELETE IT
  LOAD PRTDUC2B.Z    LOADED FOR CICS2; IF YOU DO NOT USE CICS2 DELETE IT
/*
/+
/*
/&
* $$ EOJ
```

STARTVCS is the startup job for the Connector Server. TAPESRVR is the startup job for the Virtual Tape Data Handler.

In a system with security (access control) active, procedure COLDJOBS must be called with the appropriate access rights to load startup jobs CICSICCF and CICS2 into the VSE/POWER reader queue. The access rights are inherited by the jobs loaded. The // ID statement for user DUMMY turns off these access rights.

Refer to Chapter 33, “Access Rights/Checking in DTSECTAB,” on page 401 for a detailed description of the z/VSE access control support.

After you made 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

Skeleton SKLOAD for Loading a Job

Skeleton SKLOAD catalogs a job into IJSYSRS.SYSLIB and loads the job (via procedure LDPAUSEC) into the VSE/POWER reader queue. Job PAUSEC is used as an example.

```
* $$ JOB JNM=CATPAUSE,DISP=D,CLASS=0
// JOB CATPAUSE           CATALOG PAUSEC.Z AND LDPAUSEC, LOAD PAUSEC
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG  PAUSEC.Z         REPLACE=YES
$$$$ JOB JNM=PAUSEC,DISP=L,CLASS=C,EOJMSG=YES
$$$$ LST CLASS=A,DISP=D
// JOB PAUSEC
// PAUSE
$$/&
$$$$ EOJ
/+
CATALOG LDPAUSEC.PROC     REPLACE=YES DATA=YES
// EXEC DTRIINIT
   ACCESS IJSYSRS.SYSLIB
   LOAD PAUSEC.Z
/*
/+
/*
// EXEC PROC=LDPAUSEC     TO LOAD PAUSEC INTO RDR QUEUE
/&
* $$ EOJ
```

Before you file the skeleton, run the DTRSEXIT macro. This macro deletes specific comments from the skeleton. On the command line, enter:

```
@DTRSEXIT
```

The skeleton includes additional $$ characters. They are needed to mask off VSE/POWER JECL statements. Program DTRIINIT, described in the manual *z/VSE System Utilities* under “DTRIINIT Utility”, replaces the $$ characters with VSE/POWER JECL statements for cataloging.

Skeleton SKCOMVAR for Tailoring $COMVAR Procedure

You must complete this skeleton if your environment includes at least two CPUs which share disk devices (DASD sharing). Procedure $COMVAR serves to identify the currently active CPU. The default $COMVAR is set to a single CPU environment (XNCPU=1).

Skeleton SKCOMVAR provides statements for three CPUs. Change, add, or delete statements as required. You can add statements for up to 31 CPUs. Replace the

- **X** string with the 12 character CPU ID of your second CPU, and the
- **Y** string with the 12 character CPU ID of your third CPU.

$COMVAR.PROC is cataloged into library IJSYSRS.SYSLIB and PRD2.SAVE.

The manual *z/VSE Guide to System Functions* provides further details about DASD sharing under “DASD Sharing with Multiple VSE Systems”.

```
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG $COMVAR.PROC DATA=YES R=Y
SETPARM XNCPU=1
// EXEC DTRISCPU,SIZE=AUTO,PARM='XXXXXXXXXXXX'    (12 CHARS CPUID2)
IF $RC=0 THEN
SETPARM XNCPU=2
// EXEC DTRISCPU,SIZE=AUTO,PARM='YYYYYYYYYYYY'    (12 CHARS CPUID3)
IF $RC=0 THEN
SETPARM XNCPU=3
```

```
/*
/+
CONN S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY $COMVAR.PROC     REPL=YES
```

After you made 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

Skeleton SKVTASTJ for Starting Up the Virtual Tape Server

The SKVTASTJ skeleton creates a startup job for the Virtual Tape Server partition and catalogs it into libraries IJSYSRS.SYSLIB and PRD2.SAVE.

The procedure LDVTA then loads the startup job (TAPESRVR) into the VSE/POWER reader queue.

```
* $$ JOB JNM=CATSTVTA,DISP=D,CLASS=0
// JOB CATSTVTA       CATALOG TAPESRVR AND LDVTA, LOAD TAPESRVR
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG  TAPESRVR.Z        REPLACE=YES
$$$$ JOB JNM=TAPESRVR,DISP=L,CLASS=R,LOG=NO
$$$$ LST CLASS=A,DISP=D,PURGE=0004,RBS=500
// JOB TAPESRVR START UP VSE TAPE SERVER
// ID USER=VCSRV
// OPTION SYSPARM='00'
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.SCEEBASE)
// EXEC $VTMAIN,SIZE=$VTMAIN
$$/*
$$/&
$$$$ EOJ
/+
CATALOG  LDVTA.PROC     REPLACE=YES DATA=YES
// EXEC DTRIINIT
   LOAD TAPESRVR.Z
/*
/+
CONNECT S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY TAPESRVR.Z REP=YES
/*
// EXEC PROC=LDVTA        TO LOAD  TAPE SERVER  INTO RDR QUEUE
/&
* $$ EOJ
```

If you wish, you can replace **CLASS=R** with another class.

After you make the changes, run the DTRSEXIT macro. This macro deletes specific comments from the file. You should do this before you file the skeleton. On the command line, enter:

@DTRSEXIT

The skeleton includes additional $$ characters. They are needed to mask off VSE/POWER JECL statements. Program DTRIINIT, described in the manual *z/VSE System Utilities* under “DTRIINIT Utility”, replaces $$ characters with VSE/POWER JECL statements for cataloging.

Skeleton SKVCSSTJ for Starting Up VSE Connector Server

The SKVCSSTJ skeleton creates a startup job for the VSE Connector Server partition and catalogs it into libraries IJSYSRS.SYSLIB and PRD2.SAVE.

The procedure LDVCS then loads the startup job (STARTVCS) into the VSE/POWER reader queue.

```
* $$ JOB JNM=CATSTVCS,DISP=D,CLASS=0
// JOB CATSTVCS       CATALOG STARTVCS AND LDVCS, LOAD STARTVCS
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG  STARTVCS.Z        REPLACE=YES
$$$$ JOB JNM=STARTVCS,DISP=L,CLASS=R
$$$$ LST CLASS=A,DISP=D
// JOB STARTVCS START UP VSE CONNECTOR SERVER
// ID USER=VCSRV
*   WAITING FOR TCP/IP TO COME UP
// EXEC REXX=IESWAITR,PARM='TCPIP00'
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.SCEEBASE)
// OPTION SYSPARM='00'
// EXEC IESVCSRV,PARM='DD:PRD2.CONFIG(IESVCSRV.Z)'
$$/*
$$/&
$$$$ EOJ
/+
CATALOG  LDVCS.PROC     REPLACE=YES DATA=YES
// EXEC DTRIINIT
   LOAD STARTVCS.Z
/*
/+
CONNECT S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY STARTVCS.Z REP=YES
/*
// EXEC PROC=LDVCS        TO LOAD  VCS STARTUP  INTO RDR QUEUE
/&
* $$ EOJ
```

If you wish, you can replace **CLASS=R** with another class.

After you make the changes, run the DTRSEXIT macro. This macro deletes specific comments from the file. You should do this before you file the skeleton. On the command line, enter:

```
@DTRSEXIT
```

The skeleton includes additional $$ characters. They are needed to mask off VSE/POWER JECL statements. Program DTRIINIT, described in the manual *z/VSE System Utilities* under “DTRIINIT Utility”, replaces $$ characters with VSE/POWER JECL statements for cataloging.

Chapter 3. Modifying Predefined Environments

For a detailed description of the predefined environments provided by z/VSE, refer to the topic describing predefined system environments in the manual *z/VSE Planning*.

This chapter discusses:
- Modifying library search chains
- Changing use of static partitions
- Modifying static partition allocations
- Moving to another environment
- Modifying dynamic partition support

Modifying Library Search Chains

You can modify the library search chains for partitions controlled by VSE/POWER:
1. Copy skeleton SKLIBCHN to your VSE/ICCF primary library. Refer to "Skeleton SKLIBCHN for Defining Library Search Chains" on page 54 for details about the skeleton.
2. In the skeleton, modify the LIBDEF statements by adding your sublibraries in the correct sequence.
3. Copy skeleton SKENVSEL to your VSE/ICCF primary library. Modify it to catalog your changes in skeleton SKLIBCHN. Refer to "Skeleton for Cataloging Startup Changes (SKENVSEL)" on page 36 for details about skeleton SKENVSEL.

Changing Use of Static Partitions

To have a program run in another static partition, proceed as follows:
1. In CPUVARn.PROC, set the system variables XUSEyy and XSTATyy as required for the partitions involved via program DTRSETP. The manual *z/VSE System Utilities* describes program DTRSETP under "DTRSETP Utility" in detail.

   If you use a different partition for a CICS system, you must also change the variable XAPPLyy in CPUVARn which defines the application name for a CICS system. yy defines the partition used.
2. If necessary, modify the JCL startup procedures and the startup job(s) involved. If possible, use the skeletons provided. Also, if necessary, delete obsolete startup jobs in the VSE/POWER reader queue and update procedure COLDJOBS via skeleton SKCOLD.
3. Modify the partition allocations as needed. If possible, use the skeletons provided. Refer to "Modifying Static Partition Allocations" on page 68 for details.
4. Modify the VSE/POWER autostart statements in the POWSTRTn procedure. Use skeleton SKPWSTRT.
5. If required, put PARSTD labels into the STDLABUS procedure. Use skeleton STDLABUS. "Creating Standard Labels for Non-VSE/VSAM User Files" on page 234 describes skeleton STDLABUS.

6. If required, modify or add library search chain. Use skeleton SKLIBCHN which is described under "Skeleton SKLIBCHN for Defining Library Search Chains" on page 54.
7. You may want to change VSE/POWER job classes. They are defined in the POWSTRTn procedure and in the startup jobs in the VSE/POWER reader queue. Use the appropriate skeletons (SKPWSTRT, SKCICS, and SKVTAM).
8. If required, ask the operator to enter a PRTY command or change the JCL procedures accordingly. This may be necessary since the priorities for the programs running in the partitions are now different from those specified in $0JCL or your own procedure.

Modifying Static Partition Allocations

Depending on your environment selected during initial installation (A, B, or C), you must use one of the following skeletons:

SKALLOCA = 12 static partitions (small system)

SKALLOCB = 12 static partitions (medium system)

SKALLOCC = 12 static partitions (large system)

You can vary the partition sizes within the boundaries of the VSIZE specified for the predefined environment selected. Refer to the manual *z/VSE Planning* under "Selecting a Predefined Environment" for the VSIZE values of the predefined environments. You can increase the VSIZE via the *Tailor IPL Procedure* dialog which may also mean that a change in the DPD definitions of the IPL procedure is required.

To change partition allocations, perform the following steps:

1. Copy the corresponding SKALLOCx skeleton to your primary VSE/ICCF library and modify the copied skeleton as needed. Refer to "Skeletons for Static Partition Allocations" on page 38 for details about these skeletons.

   If you change the procedure name ALLOC in the CATALOG statement, you must use the new name as input for skeleton SKJCL0. In the skeleton, change the EXEC PROC statement for the allocation procedure accordingly. Refer to "Skeletons for Starting Up BG Partition" on page 41 for details on skeleton SKJCL0.
2. If you change the default procedure name $0JCL in the CATALOG statement of SKJCL0, you must specify the new name also in the $ASIPROC master procedure (if you created one) or the operator must enter it during IPL.
3. Copy skeleton SKENVSEL to your primary VSE/ICCF library and modify it to catalog your changes.

   Refer to "Skeleton for Cataloging Startup Changes (SKENVSEL)" on page 36 for details on skeleton SKENVSEL.
4. Submit the job stream created for processing.

Note:

1. You can change the partition size temporarily with the ALLOC command. Refer to the manual *z/VSE System Control Statements* under "ALLOC" for details about this command.
2. For the changes introduced with VSE/ESA 1.3 concerning ALLOC R, refer to "Skeletons for Static Partition Allocations" on page 38.

Moving to Another Environment

Moving to another environment involves a number of tailoring tasks. These vary according to your special system requirements. Consider the following as a guideline that helps you define the tailoring tasks to be performed for creating a specific environment.

Moving from Predefined Environment A to B/C, or from B to C

The predefined environments provided are described in detail in the manual *z/VSE Planning* under “Predefined System Environments”.

To move from predefined environment A to predefined environment B or C, proceed as follows:

- For an environment with a larger VSIZE, use the *Tailor IPL Procedure* dialog to also define new DPD extents. The dialog gives you hints for how much space you need. Use the *Tailor IPL Procedure* dialog (which is described under “Tailoring the IPL Procedure” on page 14) also if you want to change other IPL parameters.
- To change existing startup procedures, use the skeletons provided. Copy the skeletons first to your VSE/ICCF primary library.
- Set environment number XENVNR in CPUVAR1.PROC by running program DTRSETP in the BG partition:

  ```
  // EXEC DTRSETP,SIZE=AUTO,PARM='CPUVAR1;;SET XENVNR=n'
  ```

  where n is the environment number: A, B, or C.

  Refer to *z/VSE System Utilities* under “DTRSETP Utility” for details about how to use program DTRSETP.
- Change partition allocations as required by using skeleton SKALLOCn.
- Change PASIZE to at least the size of the largest partition.
- Change NPARTS, if you wish to define further dynamic partitions.

Moving to an Environment of Your Own Design

If you want to define an environment of your own design, proceed as follows:

1. Use the *Tailor IPL Procedure* dialog to modify IPL parameters such as virtual storage (VSIZE) and page data set extents (DPD). For details about the dialog, refer to “Tailoring the IPL Procedure” on page 14.
2. Depending on your requirements you may use one or more of the skeletons provided by z/VSE to tailor your startup. For partition allocations, consider using skeleton SKALLOCA as a sample. Refer to “Using Skeletons for Tailoring System Startup” on page 35 for details about the skeletons provided.
3. If needed, create or change startup jobs for the partitions. A startup job must be available as cataloged procedure in a VSE library. Load the startup jobs into the VSE/POWER reader queue. Use program DTRIINIT which is described in detail in the manual *z/VSE System Utilities* under “DTRIINIT Utility”.
4. Delete obsolete startup jobs in the VSE/POWER reader queue and update procedure COLDJOBS (skeleton SKCOLD) if necessary.

Modifying the Dynamic Partition Support

If you plan to use dynamic partitions, you should first consult the manual *z/VSE Planning* under "Planning for Dynamic Partition Support". It discusses planning aspects to be considered before using this support.

Note:

1. The maximum number of dynamic classes per table is **23**.
2. The maximum number of dynamic class tables is **36**.

Predefined environments A, B, and C provide dynamic partition support. If you want to modify this support or create a dynamic partition environment of your own, follow the steps outlined below. All steps can be performed while your system is running and are described in detail on the following pages.

1. Tailor the IPL Procedure via the *Tailor IPL Procedure* dialog.
2. Catalog the JCL startup procedure for a dynamic partition class via skeleton SKJCLDYN.
3. Tailor the VSE/POWER startup procedure via skeleton SKPWSTRT to have a specific dynamic class table be loaded automatically during startup.
4. Define one or more dynamic class tables via the *Maintain Dynamic Partitions* dialog according to your requirements.

You can activate a dynamic class table with the PLOAD command of VSE/POWER or have it activated automatically during startup (step 3).

Dynamic Partition Support - Tailoring the IPL Procedure

Use the *Tailor IPL Procedure* dialog (as described under "Tailoring the IPL Procedure" on page 14) if you want to change IPL parameters such as:

- NPARTS (IPL SYS command)

  Defines the total number of partitions (static and dynamic) you want to use. You can increase the number of dynamic partitions but not of static partitions (where the maximum is 12). The maximum number of dynamic partitions possible depends on the user environment. A theoretical value is 150 - 200.
- VSIZE (IPL supervisor parameters command)

  If you want to increase this value, you must also increase the values in the IPL DPD commands for the page data set extents. Consult the topic "z/VSE Disk Layouts" of the manual *z/VSE Planning* for the layout of DOSRES and SYSWK1 on the various disk device types.

  Note: You must ensure that no overlap occurs with other files when enlarging or relocating the page data set extents.

Depending on the number of the dynamic partitions you are going to use, there may also be a need to increase the GETVIS area. GETVIS is a parameter of the IPL SVA command. To activate your IPL changes, you must re-IPL the system.

Cataloging JCL Startup Procedures

For each dynamic class defined in the dynamic class table you must also define a startup procedure. This procedure is processed each time a dynamic partition is created. You may use the same procedure for more than one dynamic class. To catalog such a procedure, use skeleton **SKJCLDYN** which includes a sample profile. The layout of the skeleton SKJCLDYN is shown below.

```
CATALOG STDPROF.PROC  DATA=YES REPLACE=YES
// LIBDEF DUMP,CATALOG=SYSDUMP.DYN,PERM   ASSIGN SYSDUMP FOR DYN.PART.
// OPTION NODUMP
// EXEC PROC=LIBDEF      DEFINE THE PERMANENT LIBRARY SEARCH CHAIN
// SETPFIX LIMIT=48K
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
   .                     DEFINE YOUR OWN ASSIGNS IF NEEDED
   .                     ADD DLBL AND EXTENT STATEMENTS HERE IF NEEDED
/+
CONNECT S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY STDPROF.PROC  REPLACE=YES
```

Figure 12. Skeleton SKJCLDYN

You find the name of the procedure (STDPROF) under PROFILE in Figure 13 on page 72. For details about procedure LIBDEF refer to “Skeleton SKLIBCHN for Defining Library Search Chains” on page 54.

After you have modified skeleton SKJCLDYN, enter the following command from the editor's command line:

@DTRSEXIT

This command calls a macro which deletes specific comments from the file. You should do this before filing the skeleton. Use skeleton SKENVSEL (shown under “Skeleton for Cataloging Startup Changes (SKENVSEL)” on page 36) for cataloging your changes.

Tailoring VSE/POWER Startup Procedure

Activate or modify the PLOAD command in skeleton SKPWSTRT to have the required dynamic class table automatically loaded during startup. For details about skeleton SKPWSTRT refer to “Skeleton SKPWSTRT (VSE/POWER Warm and Cold Starts)” on page 49.

Activating a Dynamic Class Table

This step is **not** required if the operator will activate the dynamic class table at the console via the VSE/POWER command:

```
PLOAD DYNC,ID=n,FORCE
```

n identifies the table and FORCE ensures that even in case of wrong class definitions those classes which are correctly specified are activated.

Defining Dynamic Class Tables

z/VSE allows you to define up to 36 dynamic class tables. The name of the predefined table shipped is DTR$DYNC.

To define dynamic class tables, you use the *Maintain Dynamic Partitions* dialog. The dialog updates members DTR$DYNn.Z in system library IJSYSRS.SYSLIB when pressing PF5 on the panels shown later. For a detailed description of member DTR$DYNn.Z, refer to the manual *z/VSE Planning*.

To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

7 (Maintain Dynamic Partitions)

The dialog identifies invalid entries by displaying 2 or 5 in the OPT column of the panel shown in Figure 13. This may also happen when the dialog is reactivated and validity checking on a higher level is possible.

You get first panel TAS$DYNA listing the dynamic class tables defined for the system. With this panel you can add, alter or delete dynamic class tables. You can define up to **36** dynamic class tables with up to **23** dynamic classes per table. If you enter option 1 (to add a new table) next to an existing table, the specifications of the dynamic classes are used as default for the new table. First, you must define a name (DTR$DYNX in the example) for the new table in panel TAS$DYN3 before panel TAS$DYN1 is displayed.

```
TAS$DYN1                 MAINTAIN DYNAMIC PARTITIONS

Enter option(s) and press enter.

                                              Dynamic class table : DTR$DYNX
 OPTIONS:   1 = ADD            2 = ALTER             5 = DELETE
 OPT   DYNAMIC  ENABLED   MAX NO. OF    STORAGE    MAXIMUM   DYNAMIC   PROFILE
        CLASS   1=YES     PARTITIONS  ALLOCATION   PROGRAM   SPACE
                2=NO                               SIZE      GETVIS
  _       C       1          9             1M       500K    128K     STDPROF
  _       P       1         32             1M       512K    128K     PWSPROF
  _       R       1          3             8M      1024K    128K     STDPROF
  _       S       1          2            15M      1024K    128K     STDPROF
  _       Y       1          8             5M      1024K    128K     STDPROF
  _       Z       1          3             5M      1024K    128K     STDPROF




PF1=HELP       2=REDISPLAY  3=END                     5=PROCESS
```

Figure 13. Maintain Dynamic Partitions (TAS$DYN1)

If you enter option 1 (in panel TAS$DYN1) to add a new class definition you get the panel shown in Figure 14 on page 73. The system displays default values if you selected an empty line. If you selected an already defined class, as in the example, the dialog uses this class and its parameters as a model for the new class.

```
TAS$DYN2              MAINTAIN DYNAMIC PARTITIONS

Enter the required data and press ENTER.

DYNAMIC CLASS..........                Enter one of the classes C - E, G - Z
NUMBER OF PARTITIONS...  8             Enter a number between 1 and 32
STORAGE ALLOCATION.....  2M            Specify in M bytes
MAXIMUM PROGRAM SIZE...  1024K         Specify in K or M bytes
DYNAMIC SPACE GETVIS...  128K          Specify in K bytes
ENABLED................  1             1=YES, 2=NO
PROFILE................  STDPROF       Name of the JCL procedure
NUMBER OF LOGICAL UNITS  50            Enter a number between 10 - 255

READER DEVICE..........  FEC
LIST OF PRINTER DEVICES  FEE    ___    ___    ___    ___    ___    ___
                         ___    ___    ___    ___    ___    ___    ___

LIST OF PUNCH DEVICES..  FED    ___    ___    ___    ___    ___    ___
                         ___    ___    ___    ___    ___    ___    ___

PF1=HELP      2=REDISPLAY 3=END
```

Figure 14. Maintain Dynamic Partitions (TAS$DYN2)

Description of parameters:

DYNAMIC CLASS
Specifies the class to which a dynamic partition belongs. Except for the letters A, B, and F you can use all letters of the alphabet.

NUMBER OF PARTITIONS
Specifies the maximum number of partitions belonging to the same class.

STORAGE ALLOCATION
Specifies the virtual storage available for a dynamic partition. The allocation includes the dynamic space GETVIS area:

```
storage allocation = partition allocation + dynamic space GETVIS area
```

The maximum partition allocation calculates as follows:

```
2GB - size of shared areas - size of dynamic space GETVIS area
```

2GB is the maximum address space size possible. The shared areas include shared partitions, system GETVIS area, and the area in which the supervisor resides.

MAXIMUM PROGRAM SIZE
Specifies the amount of contiguous virtual storage in a dynamic partition reserved for program execution.

DYNAMIC SPACE GETVIS
Specifies the size of the dynamic space GETVIS area of a partition. This area can be considered as an extension of the system GETVIS area. In the chapter "Storage Management" of the manual *z/VSE Guide to System Functions* you find details about the layout of dynamic partitions including GETVIS areas.

ENABLED
Specifies whether the dynamic class is to be enabled when a PLOAD command for activating the dynamic class table is processed by VSE/POWER.

PROFILE
Specifies the name of the JCL startup procedure which is to be processed each time VSE/POWER activates (creates) a dynamic partition of this class.

NUMBER OF LOGICAL UNITS
Specifies the maximum number of programmer logical units which can be allocated to each dynamic partition of this class.

READER DEVICE
Specifies the address of the spooled reader device for the dynamic partitions of this class.

LIST OF PRINTER DEVICES
Specifies the address(es) of the spooled printer device(s) for the dynamic partitions of this class.

LIST OF PUNCH DEVICES
Specifies the address(es) of the spooled punch device(s) for the dynamic partitions of this class.

Chapter 4. Configuring an OSA Express Adapter

This chapter describes how you configure an OSA Express adapter for use within z/VSE. The OSA Express adapter is usually configured for *QDIO Mode*. However, you can also configure an OSA Express adapter in *non-QDIO mode* if (for example) you have your own applications that use *non-QDIO* processing.

This chapter contains these main topics:

Related Topics:

For details of how to ...	Refer to ...
configure the OSA Express hardware	*System z9 and zSeries Open Systems Adapter-Express Customer's Guide and Ref*.
configure a HiperSockets™ device	Chapter 5, "Configuring a HiperSockets Device," on page 81.
configure a Linux Fast Path	*z/VSE TCP/IP Support*.
configure a Virtual LAN (VLAN)	*z/VSE TCP/IP Support*.

Configuring an OSA Express Adapter in IOCP

OSA Express is identified in the I/O configuration by its channel path identifier (CHPID). The CHPID type for QDIO is **OSD**.

In this example, we define physical device addresses 1D00 and 1D03:

```
CHPID PATH=FC,TYPE=OSD
CNTLUNIT CUNUMBR=1D00,UNIT=OSA,PATH=FC
IODEVICE ADDRESS=(1D00,3),CUNUMBR=1D00,UNIT=OSA
IODEVICE ADDRESS=(1D03,3),CUNUMBR=1D00,UNIT=OSA
```

Figure 15. IOCP Statements Required for Configuring an OSA Express Adapter

Note:

- When defining devices for OSD CHPIDs, it is important to consider the maximum number of sub-channels 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).

- The candidate list should be specified *explicitly*, otherwise it defaults to all the LPARs defined on your RESOURCE statement in the IOCDS.

Defining an OSA Express Adapter in z/VSE

To access an OSA Express adapter in QDIO mode, you need *three* OSA Express devices:
- a read device
- a write device
- a datapath device.

You must specify these devices in the *IOCP generation macro* with device type **OSA** (as shown in Figure 15 on page 75).

For z/VSE, the corresponding *device type* is **OSAX**. It must be used for all the devices specified in the IOCP generation macro with CHPID type **OSD** (as shown in Figure 15 on page 75).

Here are some examples of the ADD statements that are used for OSAX devices:

```
ADD 1D00:1D02 AS D00:D02,OSAX
```

or

```
ADD 1D00 AS D00,OSAX
ADD 1D01 AS D01,OSAX
ADD 1D02 AS D02,OSAX
```

In these examples, the physical addresses `1D00`, `1D01`, and `1D02` are added with their corresponding VSE addresses `D00`, `D01`, and `D02`.

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 1D03:1D05 AS D03:D05,OSAX
```

The *Configure Hardware* dialog supports the definition of OSA Express devices:
1. Use Fast Path **241** to display the *Hardware Configuration: Unit Address List* panel.
2. Press **PF6** (ADD ADDR), and then enter the Address (the physical address, in this example `1D03`) and the VSE address (in this example `D03`) of the OSA Express device you wish to define.
3. Press Enter and the *Hardware Configuration: Device Group* panel is displayed.
4. Select *3 Com. Devices* and the panel *Selection List: Devices* is displayed, as shown in Figure 16 on page 77.

```
ADM$DEVL                   SELECTION LIST: DEVICES

Select one of the entries by entering 1.

                              VSE Address of the device to be defined: 400

 SEL      DEVICE      DESCRIPTION

 _        CTCA        Channel-to-Channel Adapter
 _        FCP         FCP Adapter
 _        OSAD        Open System Adapter Feature
 _        OSA         Open System Adapter Port
 _        OSAX        Open System Adapter Express
 _        2701        Data Adapter Unit
 _        2703        Transmission Control Unit
 _        3172        Interconnect Controller
 _        3745-130    Communications Controller
 _        3745-170    Communications Controller configurable with NCP
 _        3745-17A    Communications Controller configurable with NCP
 _        3745-210    Communications Controller
PF1=HELP       2=REDISPLAY  3=END
               8=FORWARD
```

Figure 16. Selection Panel for OSAX

5. Select OSAX (Open Systems Adapter Express) and press Enter. The *Hardware Configuration: Unit Address List* panel is then re-displayed. If the physical address, VSE address, and device type are correct, press **PF5** (Process) to enter these details in a job.
6. The *Hardware Configuration: Catalog Startup Members* panel is then displayed. Enter an 'X' next to “IPL Procedures” (remove any other 'X' entries) and Press Enter.
7. The *Job Disposition* panel is displayed. Enter a Job Name and other details, and press Enter to save the job.
8. Use Fast Path **51** to proceed to your *Primary Library*. You will see the job with the name you assigned. If the details are correct, submit this job to update your z/VSE system's IPL procedure (for example, $IPLESA.PROC).

Configuring an OSA Express Adapter in TCP/IP

The statements for using a HiperSockets connection vary depending upon the TCP/IP solution you have chosen.

Under TCP/IP for VSE/ESA, to use an OSA Express adapter in QDIO mode you must specify a 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=(8-byte-name),
       OSAPORT=(0|1)         (default: 0)
       FRAGMENT={NO|YES}     (default: NO)
                             (YES not supported by OSA Express adapter)
       ROUTER={NONE|PRIMARY|SECONDARY}     (default: NONE)
       ALTIP=(IP-Address1,IP-Address2,...,IP-Address9)
```

Figure 17. DEFINE LINK Statement for an OSA Express Adapter

Explanations:

1. cuu1,cuu2 are VSE addresses that correspond to physical addresses. These VSE addresses 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 no longer requires a PORTNAME.

   If the DEFINE LINK fails with message 0S39I .... REASON=0032, the PORTNAME specified does not match the name specified initially.
4. Specify OSAPORT=1 if you wish to use Port-1 of an OSA Express3 feature that supports two ports per CHPID.
5. 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.
6. If you wish to use:
   - your z/VSE system as a multi-homing host,
   - the TCP/IP stack as a gateway,

   you can specify up to nine additional IP addresses using the ALTIP parameter. In the two examples below, `DEV=(`**`D00,D01`**`)` and `DEV=`**`D04`** are VSE addresses that correspond to physical addresses.

Example 1:

```
DEFINE LINK,ID=...,TYPE=OSAX,
       DEV=(D00,D01),
       DATAPATH=D02,
       IPADDR=9.164.155.90,
       MTU=9000
```

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 one 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.

For detailed instructions on how to configure an OSA Express adapter in TCP/IP, refer to the *TCP/IP for VSE, Installation Guide* which you can find at this URL:

```
www.csi-international.com
```

Under IPv6/VSE, for details of how to specify the equivalent statements to those above, refer to:

```
http://www.ibm.com/systems/z/os/zvse/documentation/#tcpip
```

Configuring an OSA Express Adapter in Non-QDIO Mode

The *OSA-2 adapter* cannot be used with the current z/VSE Version/Release. Therefore, if you wish to use an OSA Express adapter so that it emulates an OSA-2 adapter (which operates in *non-QDIO* mode), you must:

1. Specify CHPID TYPE **OSE** in IOCP (shown in Figure 15 on page 75).
2. Specify these devices in the *Selection List: Devices* panel (shown in Figure 16 on page 77) with different addresses:
   - **OSAD**, which is the device type for the *OSA/SF for VSE/ESA* (OSA/SF) program. You can have only one OSAD definition (ADD statement) for an OSA-2 adapter. For details about how to set up and configure OSA/SF, see "Setting Up OSA/SF."
   - **OSA**, which is the device type for OSA-2 data transfer. You can have more than one OSA definition (ADD statement) for an OSA-2 adapter.
3. Specify device type **OSA2** in the TCP/IP DEFINE LINK statements (shown in Figure 17 on page 77).

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).

To use the jobs IOAMAIN and IOACMD, 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.

Chapter 5. Configuring a HiperSockets Device

This chapter describes how you configure a HiperSockets device for use within z/VSE.

It contains these main topics:

Related Topics:

For details of how to ...	Refer to ...
configure an OSA Express adapter	Chapter 4, “Configuring an OSA Express Adapter,” on page 75.
configure a Linux Fast Path	*z/VSE TCP/IP Support*.
configure a Virtual LAN (VLAN)	*z/VSE TCP/IP Support*.

An overview of the use of HiperSockets in z/VSE is provided in the chapter “TCP/IP and HiperSockets Support” in the *z/VSE Planning*.

Configuring a HiperSockets Device in IOCP

Each HiperSockets requires the definition of a channel path identifier (CHPID). The following rules and characteristics apply:
- You can define more than one HiperSockets per server and share them among LPARs (the number depends upon the IBM server model).
- 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.
- Only HiperSockets with the same CHPID PATH can communicate with each other.

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

Here is an example of how to configure a HiperSockets device in IOCP:

```
CHPID PATH=(FC),SHARED,PARTITION=(...),TYPE=IQD,OS=40
CHPID PATH=(FD),SHARED,PARTITION=(...),TYPE=IQD
CNTLUNIT CUNUMBR=1500,PATH=(FC),UNIT=IQD
CNTLUNIT CUNUMBR=1600,PATH=(FD),UNIT=IQD
IODEVICE ADDRESS=(1500,16),CUNUMBR=1500,UNIT=IQD
IODEVICE ADDRESS=(1600,3),CUNUMBR=1600,UNIT=IQD
```

Figure 18. IOCP Statements Required for Configuring a HiperSockets Device

Configuring HiperSockets Devices in z/VSE

In the IPL ADD command, CHPID type **IQD** devices have a corresponding z/VSE device type **OSAX**. For each HiperSockets link, you must define *three* such devices.

To distinguish CHPID type IQD devices from CHPID type OSD devices, a **mode of 1** must be specified as shown in the example below:

```
ADD 1500:1515 AS 500:515,OSAX,1
```

In the *Configure Hardware* dialog you must select **HiperSockets** together with MODE **01**:

```
COM$MODE               SELECTION LIST: DENSITY/MODE

Select one of the entries by entering 1.



 SEL      MODE        DESCRIPTION

 1        01          HiperSockets
 _        02          OSA Express for zBX
 _        EML         Emulated device
 _
 _
 _
 _
 _
 _
 _


PF1=HELP       2=REDISPLAY  3=END
```

Figure 19. Selecting the HiperSockets Mode

Configuring a HiperSockets Device in TCP/IP

The statements for using a HiperSockets connection vary depending upon the TCP/IP solution you have chosen.

Under TCP/IP for VSE/ESA, to define a Layer 3 IPv4 link you must 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)
```

cuu1,cuu2 are VSE addresses that correspond to physical addresses. These definitions are the same as those described for OSA Express (see “Configuring an OSA Express Adapter in TCP/IP” on page 77), **except** that:

1. HiperSockets do not require a PORTNAME.
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).

Under IPv6/VSE, for details of how to specify the equivalent statements to those above, refer to:

`http://www.ibm.com/systems/z/os/zvse/documentation/#tcpip`

TCP/IP Partition Resources Required for HiperSockets

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.

Chapter 6. Participating in an Intra-Ensemble Data Network

This chapter describes how you configure *OSA Express for zBX* devices for use with z/VSE to enable z/VSE to participate in an *Intra-Ensemble Data Network* (IEDN).

It contains these main topics:

Related Topics:

For details of how to ...	Refer to ...
configure an OSA Express adapter	Chapter 4, "Configuring an OSA Express Adapter," on page 75.
configure a Virtual LAN (VLAN)	*z/VSE TCP/IP Support*.

Overview of an IEDN

An IEDN provides connectivity between:

- A zEnterprise CEC (Central Electrical Complex) or later and System z Blade Center Extensions (zBXs).
- Two or more zEnterprise CECs or later.

You must configure an IEDN using the zEnterprise *Unified Resource Manager*.

z/VSE can participate in an IEDN within:

- A *z/VM environment* using dedicated *OSA Express for zBX* devices.
- An *LPAR environment* using *OSA Express for zBX* devices.
- A *z/VM environment* using z/VM VSWITCH support. If you use the VSWITCH *OSDSIM mode*, z/VSE can participate in an IEDN *transparently* using the functionality provided with *OSA Express* (CHPID type `OSD`) and TCP/IP.

Prerequisites for Participating in an IEDN

To allow your z/VSE system to participate in an IEDN, you must have:

- Configured three OSA Express for zBX devices in IOCP (described below).
- Defined OSA Express for zBX devices in z/VSE (described below).
- Configured the link information within TCP/IP (described below).
- Configured a VLAN (at least GLOBAL VLAN) for use with the TCP/IP LINK (described in the *z/VSE TCP/IP Support*).

  Note: If you are using z/VM VSWITCH support, the VLAN support is performed *by z/VM* (you are not required to perform any VLAN configuration within your z/VSE system).

Configuring OSA Express for zBX devices in IOCP

OSA Express for zBX devices are configured in IOCP using the CHPID TYPE=**OSX** (the channel path identifier) and UNIT=**OSA**.

You must define three devices as shown in Figure 20.

```
CHPID PATH=FC,TYPE=OSX
CNTLUNIT CUNUMBR=1D00,UNIT=OSA,PATH=FC
IODEVICE ADDRESS=(1D00,3),CUNUMBR=1D00,UNIT=OSA
```

Figure 20. IOCP Statements Required for Configuring an OSA Express for zBX Device

Defining OSA Express for zBX devices in z/VSE

In the IPL ADD command, CHPID type **OSX** devices have a corresponding z/VSE device type **OSAX**. To distinguish CHPID type OSX devices from CHPID type OSD devices, a **mode of 2** must be specified as shown in the following example:

```
ADD 1D00:1D02 AS D00:D02,OSAX,2
```

In the *Configure Hardware* dialog you must select **OSA Express for zBX** together with MODE **02**:

```
COM$MODE                SELECTION LIST: DENSITY/MODE

Select one of the entries by entering 1.



 SEL      MODE        DESCRIPTION

 _        01          HiperSockets
 1        02          OSA Express for zBX
 _        EML         Emulated device
 _
 _
 _
 _
 _
 _
 _


PF1=HELP       2=REDISPLAY  3=END
```

Figure 21. Selecting the IEDN Mode

Configuring TCP/IP to Use OSA Express for zBX devices

The TCP/IP `DEFINE LINK` statement for CHPID type OSX devices is identical to the TCP/IP LINK statement used for CHPID type OSD devices. This means, the `DEFINE LINK` statement has TYPE=**OSAX**.

For details, see "Configuring an OSA Express Adapter in IOCP" on page 75.

Chapter 7. Configuring Disk, Tape, and Printer Devices

This chapter describes how to use the *Configure Hardware* dialog to configure disk, tape, and printer devices. These devices are also referred to as *non-communication devices*.

This chapter contains these main topics:

- "Introduction to Configuring Disk, Tape, and Printer Devices"
- "Using the Configure Hardware Dialog" on page 88
- "Adding a Disk, Tape, or Printer Device" on page 89
- "Changing or Deleting a Disk, Tape, or Printer Device" on page 92

Related Topic:

For details of how to ...	Refer to ...
configure your z/VSE system to use SCSI disks	Chapter 8, "Configuring Your System to Use SCSI Disks," on page 95.

Introduction to Configuring Disk, Tape, and Printer Devices

During initial installation, you completed hardware configuration after signing on with the POST user-ID. This is described in detail in the manual *z/VSE Installation* in the topic "Installation Part 3 - Native and VM".

However, you can add, change, or delete devices on your system at any time. Use the *Configure Hardware* dialog to add or delete hardware addresses and to specify device characteristics.

By using the dialog as described in this topic, you can create a list that shows all the devices (and addresses) which are part of your system. Such a **configuration list** helps you control the hardware attached to your system. You should create a new list whenever a hardware change is implemented. To create a configuration list for your installation, perform the following steps:

1. Start with the Administrator *z/VSE Function Selection* panel and select Fast Path 241. You get the panel *Hardware Configuration: Unit Address List*.
2. Press PF9 (PRINT). On the panel displayed you can select the type of configuration list you want; SNA or non-SNA terminal list, for example. After selecting one or more lists, press ENTER.
3. The configuration list(s) created are stored as library member **CONFLIST** in your VSE/ICCF primary library.
4. You can print the library member by selecting option 3 (PRINT) in the FULIST display of your primary library. The output is placed in the VSE/POWER list queue for printing.

Using the Configure Hardware Dialog

Start with the Administrator *z/VSE Function Selection* panel and select Fast Path **241**. The *Hardware Configuration: Unit Address List* panel is displayed. It is shown in Figure 22.

```
ADM$HDWB          HARDWARE CONFIGURATION: UNIT ADDRESS LIST



OPTIONS: 2 = ALTER DEVICE TYPE CODE/MODE     3 = SELECT FOR FURTHER PROCESSING
         4 = LIST SIMILAR DEVICES            5 = DELETE A DEVICE

    OPT    VSE   PHYSICAL DEVICE     DTYPE     DEVICE  DEVICE  DEF
           ADDR  ADDR                CODE      MODE    DOWN    INCOMPL
     _      009   0009   3270CONS    3277
     _      00C   000C   2540-R      2540R
     _      A11   1A11   3390-X      ECKD
     _      A12   1A12   3390-X      ECKD
     _      D01   3D01   3592-E06    TPA11K    08
     _      D02   3D02   3592-E06    TPA11K    08
     _      D03   3D03   3592-E06    TPA11K    08
     _      D10   1D10   3390-X      ECKD
     _      D11   1D11   3390-X      ECKD
     _      D12   1D12   3390-X      ECKD
     _      D14   1D14   3390-X      ECKD
 POSITION NEAR ADDR == > ___
PF1=HELP       2=REDISPLAY  3=END                   5=PROCESS    6=ADD ADDR
               8=FORWARD    9=PRINT     10=SORT PHY
```

Figure 22. Unit Address List of Hardware Configuration Dialog

The unit address list consists of one or more panels. It shows all VSE and physical device addresses, and the related devices as defined for your z/VSE system.

- The VSE ADDR column contains a list of the VSE addresses (cuu). By default, the displayed information is based upon a sort of this VSE address list.
- The PHYSICAL ADDR column contains a list of the physical addresses (pcuu) that were defined in the IOCDS or z/VM configuration. If the address is FFF or less, a zero is automatically inserted at the start of the address.
- An 'X' in the column DEVICE DOWN indicates for a tape or disk device that this device is not available. By selecting 3 (SELECT FOR FURTHER PROCESSING), this status can be changed.
- An 'X' in the column DEF INCOMPL (definition incomplete) indicates that you should specify additional details for that particular device address.

Note: To change the VSE address of a device, you must first delete the device (using 5 = DELETE A DEVICE), and then re-add it with a new VSE address but with the *same* physical address (using PF6=ADD ADDR).

Various options and PF-key functions allow you to maintain your hardware configuration. They are listed below.

OPTIONS:

2 = ALTER DEVICE TYPE CODE/MODE
Select option 2 if you want to change the Device Type Code, or the Device Specification Mode.

3 = SELECT FOR FURTHER PROCESSING
Select option 3 if you want to change or add device characteristics other than Device Type Code or Device Specification Mode.

4 = LIST SIMILAR DEVICES
Select option 4 if you want only devices displayed that belong to a particular group. All disk devices or tapes for example.

5 = DELETE A DEVICE
Select option 5 if you want to delete a device (including both the physical address and VSE address) from the device address list.

POSITION NEAR ADDR:

This selection allows you to position FULIST close to an address. This address can be either a VSE address (when the list is sorted by the VSE address), or a physical address (when the list is sorted by the physical address). To skip to the top or to the bottom of the fulist, you can use either:
- 0 or FFF (where list is sorted by the VSE address).
- 0 or FFFF (where list is sorted by the physical address).

The input in this field is ignored when you press a PF key.

PF Keys:

5=PROCESS
Press PF5 if you have done all your changes using the options 2, 3, 5 or 6, or PF6.

6=ADD ADDR
With PF6 you add a new address (device) to your hardware configuration. Depending on the type of device, several panels are displayed. You have to select the device you want to add to your installation and enter all device specific information required.

9=PRINT
Use PF9 to get a printout of the device address list. If you then enter an 'X' next to an entry in the list, a library member **CONFLIST** is created in your VSE/ICCF primary library containing the appropriate listing.

10=SORT PHY or 10=SORT VSE
Use PF10 to display the information based upon a sort of the physical address list, or based upon the VSE address list.

Adding a Disk, Tape, or Printer Device

For most device parameters the system creates and uses defaults. But you must at least know the **device type** and **device address** before you can add a new device.

Let us assume that an IBM tape device of type **3592 Model E06** is to be added that has:
- “Physical” device addresses (pcuu) that are defined in the IOCDS, in the range X'3D01' to X'3D03'.
- Corresponding z/VSE addresses (cuu) in the range X'D01' to X'D03'.

The following steps are required:
1. Use Fast Path **241** from the Administrator *z/VSE Function Selection* panel. This gives you the first page of *Hardware Configuration: Unit Address List*. Figure 22 on page 88 shows this panel.
2. Press **PF6**. You get the panel *Hardware Configuration: Add a Device*. Now enter
   a. The (physical) starting address `3D01`.

b. The (physical) end address 3D03.

c. The VSE starting address D01.

d. The VSE end address will be automatically calculated by z/VSE. **Leave this field blank**.

e. The device name (3592-E06). Instead of entering the device name, you can enter a '?' to display the selection panel for the *Device Groups*. In this panel, enter:

   7 (Tape Units)

   Press **ENTER**.

3. You get the *Selection List: Devices* panel showing all tape devices supported by z/VSE. Select the correct device type:

   1 (3592-E06)

   Press **ENTER**.

   The system redisplays panel *Hardware Configuration: Unit Address List* with the newly added device.

4. Press **PF5** to process (catalog) the updated hardware configuration. You get the panel *Hardware Configuration: Catalog Startup Members* shown in Figure 23. In this panel, those startup members are marked by an 'X' which are affected by the change. In our example: IPL Procedures.

   Press **ENTER**.

5. You get the *Job Disposition* panel to submit the job to batch, file it in your VSE/ICCF primary library, or both.

```
 ADM$CRE1        HARDWARE CONFIGURATION: CATALOG STARTUP MEMBERS



Press ENTER to catalog the objects marked by an X. You may add or delete
an X as needed.


              X      IPL Procedures
              _      VTAM Book with Startup Options
              _      VTAM Books for Model Terminal Support
              _      VTAM Book for Local Non-SNA Terminals
              _      VTAM Book Local SNA Terminals
              _      VTAM Books for OSA or 3172 attached Terminals
              _      CICS CSD Group for terminals - VSETERM1
              _      CICS CSD Group for terminals - VSETERM2
              _      CICS CSD Group for terminals - VSETERM3



 PF1=HELP       2=REDISPLAY  3=END

 IPLPROC          SOURCE CREATED.
```

Figure 23. Panel for Cataloging Startup Members (Hardware Configuration)

For the example user here, the following statements will now be included in the IPL procedure:

```
ADD 3D01:3D03 AS D01:D03,TPA11K,08
```

Device Considerations

For most non-communication devices you only have to select the device type and enter the *cuu* address. The other device characteristics are known to the system. For disk devices and the IBM 3820 printer further panels prompt you for additional device characteristics. For tape devices, you can change and define additional characteristics with the SETMOD attention command.

You can also change the mode using option 2 (ALTER DEVICE TYPE CODE/MODE) on the *Hardware Configuration: Unit Address List* panel.

Disk Devices (Including FBA-SCSI Disks)

It is recommended that you initialize a disk device before you add it to your system. The topic "Initializing Disks and Placing the VTOC" in the manual *z/VSE Installation* contains details on how to initialize disk devices.

When you enter a disk device type, you get the panel *Hardware Configuration: Disk List* showing the disk devices specified and two columns with the options SHARED and DEVICE DOWN.

- SHARED

  Enter an **X** for those disk devices you want to share across systems. The following types of IBM disk devices can be used for (DASD) sharing:

  3380

  3390

  FBA

  FBA-SCSI
- DEVICE DOWN

  Enter an **X** for those disk devices that are not available (or should not be available) for operation.

  In a shared environment in which the volume labels are not unique, DEVICE DOWN may be required to prevent the system from accessing the wrong device when being addressed by volume label (VOLID).

Note: Configuring FBA-SCSI Disks The procedure for configuring FCP-attached SCSI disks (FBA-SCSI disks) is described in Chapter 8, "Configuring Your System to Use SCSI Disks," on page 95. This procedure is based upon a practical example that covers all aspects of configuring for SCSI use.

Tape Devices

For the IBM 3480 and IBM 3490 the *Configure Hardware* dialog offers two choices for device definition: one with data compaction, one without. For details about data compaction, refer to the ADD command in the manual *z/VSE System Control Statements* under "ADD".

For the IBM 3590 the *Configure Hardware* dialog offers three choices for device definition: with 128, 256, or 384 track capacity.

For the IBM 3592 the *Configure Hardware* dialog offers the:
- 512 track capacity for Model J1A.
- 896 track capacity for Model E05 (also referred to as the TS1120).
- 1152 track capacity for Model E06 (also referred to as the TS1130).
- 2176 track capacity for Model E07 (also referred to as the TS1140).

With option 2 (ALTER DEVICE TYPE CODE/MODE) the default mode setting can be changed. With option 3 (SELECT FOR FURTHER PROCESSING) the status of a tape device can be set to DEVICE DOWN (and reset).

Automated Tape Library Support

When using an IBM 3494 or 3584 tape library, it is necessary to request the automated tape library support in the IPL SYS command with the parameter ATL.

You can use the *Tailor IPL Procedure* dialog for setting ATL. You must also change the sample job TLSDEF to meet your own system requirements.

For details, see Chapter 21, “Implementing Tape Library Support,” on page 255.

IBM 3820 Printer

When you specify an IBM 3820 printer you have to define VTAM parameters associated with the printer. You define such parameters via the *Hardware Configuration: SNA Logical Unit List* panel.

You must provide the VTAM parameter **LOGAPPL**; for the parameters LOCAL ADDRESS, VTAM PARM TABLE, and LUNAME the system provides defaults.

Support of AFP Printers

You define printers for Advanced Function Printing (AFP) as any other printer via the *Configure Hardware* dialog. The IBM 3800-3, IBM 3825, and IBM 3827 are examples of such printers. To make use of their advanced functions, these printers require in addition the optional program PSF/VSE (Print Services Facility/VSE). z/VSE provides procedures and skeletons which support the installation and use of PSF/VSE. An example is the VSE/POWER startup procedure POWSTRTn which you can modify via skeleton SKPWSTRT. For details refer to “Skeleton SKPWSTRT (VSE/POWER Warm and Cold Starts)” on page 49.

Virtual Disk for Label Area

z/VSE provides a virtual disk with address FDF for holding the label area.

Considerations for Dummy Devices

You can change but not delete the following VSE/POWER and VSE/ICCF dummy devices: FED, FEE, FEF, FFD, FFE.

You cannot change or delete the following VSE/POWER and VSE/ICCF dummy devices: FEC, FFA, FFC.

You cannot change or delete the dummy device FFF which is a place holder for a dedicated **system console**.

Changing or Deleting a Disk, Tape, or Printer Device

Use panel *Hardware Configuration: Unit Address List*. You get to it by selecting Fast Path **241** from the Administrator *z/VSE Function Selection* panel.

Enter **2** (Alter Device Type Code/Mode) in the option column if you want to change the device type code or mode of a device. Enter **5** (Delete a Device) if you want to delete a device.

Enter **3** (Select for Further Processing) in the option column if you want to change other device characteristics. Note that the following non-communication devices have characteristics that can be changed with option **3**:

- Disk devices.
- Tape devices
- IBM 3820 printer.

After you have changed device characteristics or selected deletion, proceed as shown in steps 4 and 5 for “Adding a Disk, Tape, or Printer Device” on page 89.

Chapter 8. Configuring Your System to Use SCSI Disks

This chapter describes how you configure your z/VSE system to use Fibre-Channel-attached SCSI (Small Computer System Interface) disks.

It contains these main topics:

Related Topics:

For details of ...	Refer to ...
the disk layouts of DOSRES SCSI and SYSWK1 SCSI disks	"z/VSE Disk Layouts (DOSRES, SYSWK1)" in the *z/VSE Planning*.
• how to use SCSI disks under z/VM • how to IPL from a SCSI disk when running z/VSE under VM	"Running z/VSE Under VM" in the *z/VSE Planning*.
how to define a lock file on a SCSI disk.	"Changing Startup When Lock File Is Stored On SCSI DASD" on page 34.

Note: 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 from z/VSE 3.1 onwards, z/VSE 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.

These disk controllers are qualified to be used with z/VSE SCSI support:

- IBM System Storage® DS8000® and DS6000™.
- IBM TotalStorage Enterprise Storage Server® (ESS) Model F20, 800, and 800 Turbo.

The IBM System Storage *SAN Volume Controller* (SVC) is also qualified to be used with z/VSE's SCSI support.

VSE host programs running on IBM System z10™ (z10) and zEnterprise 196 (z196) platforms 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 System z server, it is possible to IPL z/VSE from a SCSI disk. Therefore, it is possible to build a "SCSI-only" z/VSE system.

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.

z/VSE FCP-attached SCSI disk support complements SCSI support in z/VM® Version 5 and *Linux on System z*. When operating as a guest under z/VM (using the VM-emulated FBA support), z/VM presents SCSI disks as 9336-20 FBA disks. In this situation, z/VSE "views" them as FBA, not SCSI disks.

z/VSE 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.

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 does **not** support FlashCopy® for SCSI-FCP disks.

Prerequisites for Using SCSI Disk Support

If you wish to attach SCSI disk devices to a System z server and access these disk devices from z/VSE, you require:
- An IBM System z FCP adapter (FICON Express adapter configured as CHPID type FCP).
- A disk controller. One of the following:
  - an IBM System Storage DS8000 disk controller.
  - an IBM System Storage DS6000 disk controller.
  - an Enterprise Storage Server (ESS) disk controller (such as the IBM 2105), or
  - a SAN Volume Controller (SVC).
    - IBM System Storage disk subsystems and OEM SCSI disk devices supported by the SVC.
  - an IBM Storwize V700 Midrange disk system.
  - an IBM XIV Storage System.
- A supported z/VSE release[1].
- z/VM 5.2 or later.

1. z/VSE SCSI support was first introduced with Version 3 Release 1

- If you are **not** using a point-to-point connection to attach your SCSI disks to the z/VSE host, you require an FCP switch (such as an IBM 2109).
- If you wish to perform an IPL from a SCSI disk CPU Feature Code 9904 must be installed.

Restrictions When Using SCSI Disk Support

There are some restrictions that apply to a SCSI-only system. For example:

- A stand-alone dump cannot be created on a SCSI disk. If you wish to create a stand-alone dump, you must create this dump using 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.
- z/VSE FlashCopy does not support the use of SCSI disks.

Special *migration* restrictions also apply:

- 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.

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.

Note: 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*.

Limitations When Defining SCSI Disks During IPL

You can define SCSI disks using either IPL `DEF SCSI` commands or AR/JCL `SYSDEF SCSI` statements:

- You must use IPL `DEF SCSI` commands to define SCSI system disks to be used *during IPL* (DOSRES, SYSWK1, and disks containing the PDS and lock file).
- You can define all other SCSI disks using either IPL `DEF SCSI` commands or AR/JCL `SYSDEF SCSI` statements.

The number of SCSI disks that can be defined using IPL `DEF SCSI` commands is limited to approximately **100 SCSI disks**. However, this number might be less depending upon your configuration.

IBM recommends that you define all SCSI disks that are *not* required during IPL using AR/JCL `SYSDEF SCSI` statements.

For further details on how to remove definitions of SCSI disks, refer to the *z/VSE System Upgrade and Service*.

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.

Space Requirements When SCSI Is Used As a System Disk

These are the main differences to the DOSRES and SYSWK1 disk layouts when SCSI is used as a system disk:

- 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 and SYSWK1, refer to the *z/VSE Planning*, SC34-2635.

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

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

Configuring FCP Adapters, SCSI Disks, and Connection Paths

This topic is based upon the example configurations Figure 24 on page 100 and Figure 25 on page 102.
- This topic begins with a brief description of LUNs, that are used for identifying SCSI disks.
- Next is an example that uses a *switch*, which enables you to create a configuration with a high degree of flexibility.
- The final example uses a *point-to-point connection*, which removes the requirement for a switch. This option is less expensive, but does not provide the flexibility of using a switch.

Use of Logical Unit Numbers (LUNs) With SCSI Disks

You must configure your SCSI disks in your disk controller. A SCSI disk in a disk controller is called a *Logical Unit Number (LUN)*.

z/VSE views SCSI disks as *FBA disks*. Therefore, for each LUN you must add a corresponding FBA device to your z/VSE IPL procedure, as described in "Adding a Disk, Tape, or Printer Device" on page 89.

Note: An FBA device must not exist in the IOCDS.

You must use a *connection path* to "link" each FBA device to a SCSI disk (LUN), as described in "Defining FCP Devices, SCSI Disks and Connection Paths to z/VSE" on page 105.

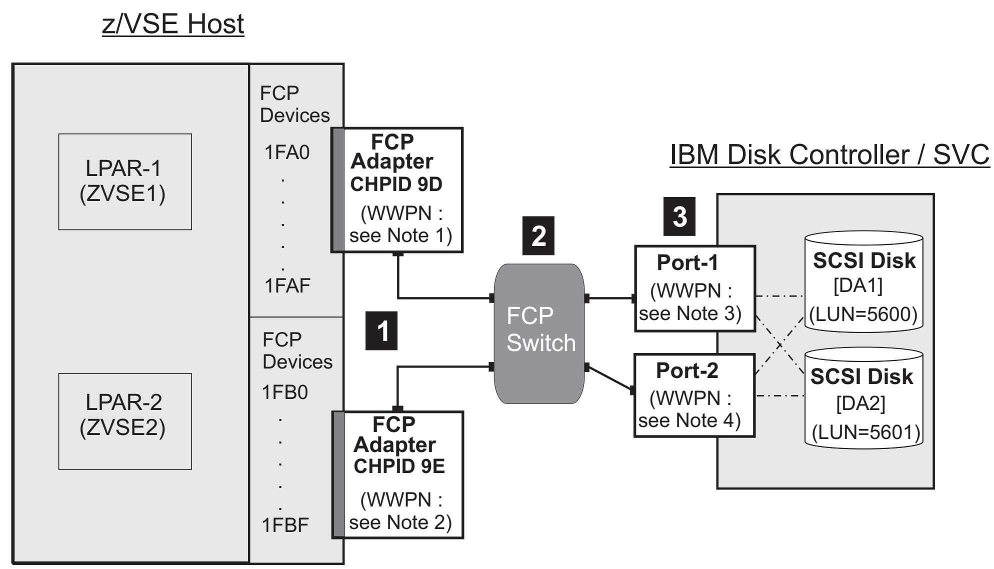
z/VSE's SCSI support can use LUNs that have been configured in *any* of the supported disk controllers.

Configuring LUNs in XIV, SVC, or Storwize V7000 disk systems:
- LUNs in XIV, SVC, or Storwize V7000 disk systems are represented by *decimal numbers*.
- LUNs used in z/VSE are represented by *hexadecimal numbers*.
- Each decimal LUN must therefore be translated into a hexadecimal number. For example: LUN number 12 would be translated in z/VSE to `LUN=000C`.

Example of a SCSI Environment That Uses a Switch

The example shown in Figure 24 and described in “Configuring FCP Adapters, SCSI Disks, and Connection Paths” on page 99, provide a practical example of how SCSI disks might be attached to a z/VSE host **via a switch**.

The SCSI disks can be configured either in an IBM disk controller or an IBM SAN Volume Controller (SVC). z/VSE does not distinguish between a SCSI disk that has been configured in the disk controller from a SCSI disk that has been configured in the SVC.



Notes: 1. The WWPN is 5005076300C295A5 (required during configuration only).
2 The WWPN is 5005076300C695A5 (required during configuration only).
3. The WWPN is 5005076300CA9A76.
4. The WWPN is 5005076300C29A76.

Figure 24. Example of a SCSI Environment Using a Switch

Note:

1. The configuration shown in Figure 24 includes the *physical addresses* of the FCP I/O devices. In “Defining FCP Devices, SCSI Disks and Connection Paths to z/VSE” on page 105, the VSE addresses (cuu) that correspond to these physical addresses (pcuu) are used by z/VSE.
2. The values used for WWPN and LUN in Figure 24 were taken from a *disk controller*. If an *SVC* is used instead of a disk controller, the format of these values will be different.

The example configuration shown in Figure 24 consists of:

1 Two physical System z FCP adapters with CHPIDs (channel path IDs) `9D` and `9E`. These physical FCP adapters (CHPIDs) are accessed as devices of type FCP that have been configured using the IOCP (Input/Output Configuration Program). Both physical FCP adapters (CHPIDs) are shared by the z/VSE systems running in LPAR-1 and LPAR-2. This is shown in the IOCP statements of Figure 26 on page 103. The physical FCP adapters can reside on the same FCP card or on different FCP cards (this is discussed in “Using Multipathing to Access SCSI Disks” on page 110).

The number of FCP devices that can be defined for each physical FCP adapter depends upon the hardware you are using. In Figure 24 on page 100:

- Sixteen FCP devices have been defined for physical FCP adapter (CHPID 9D) in the range `1FA0` to `1FAF`.
- Sixteen FCP devices have been defined for physical FCP adapter (CHPID 9E) in the range `1FB0` to `1FBF`.

Note: Within one z/VSE system, one FCP device per physical FCP adapter is sufficient for accessing *both* the FBA-SCSIs `DA1` and `DA2`.

2 Each physical System z FCP adapter is connected to the FCP switch (which might be an IBM 2109 switch) via a physical cable. The advantage of using a switch is that all ports on the disk controller can be physically accessed from FCP devices 1FA0 to 1FAF, and FCP devices 1FB0 to 1FBF.

3 The first port (Port-1) on the disk controller or SVC has the worldwide port number (WWPN) `5005076300CA9A76`. The second port (Port-2) on the disk controller or SVC has the WWPN `5005076300C29A76`. Both ports are connected to the FCP switch via physical cables. Ports are configured using the configuration software provided with the disk controller or SVC. An example for a disk controller is provided in "Configuring SCSI Disks in the Disk Controller" on page 103.

A WWPN is a unique 64-bit string (16 hexadecimal numbers) that represents a port.

The disk controller or SVC contains two SCSI disks with the logical unit numbers (LUN) `5600` and `5601`. A LUN represents a SCSI device. The LUNs have been configured so they can be accessed via both FCP adapters. LUNs are configured using the configuration software provided with the disk controller or SVC. The cuu addresses `DA1` and `DA2` are only used by z/VSE and are not part of the configuration process for the disk controller.

Example of a SCSI Environment That Uses Point-to-Point Connections

The example shown in Figure 25 on page 102 and described in "Configuring FCP Adapters, SCSI Disks, and Connection Paths" on page 99, provide a practical example of how SCSI disks might be attached to a z/VSE host **via point-to-point connections**.

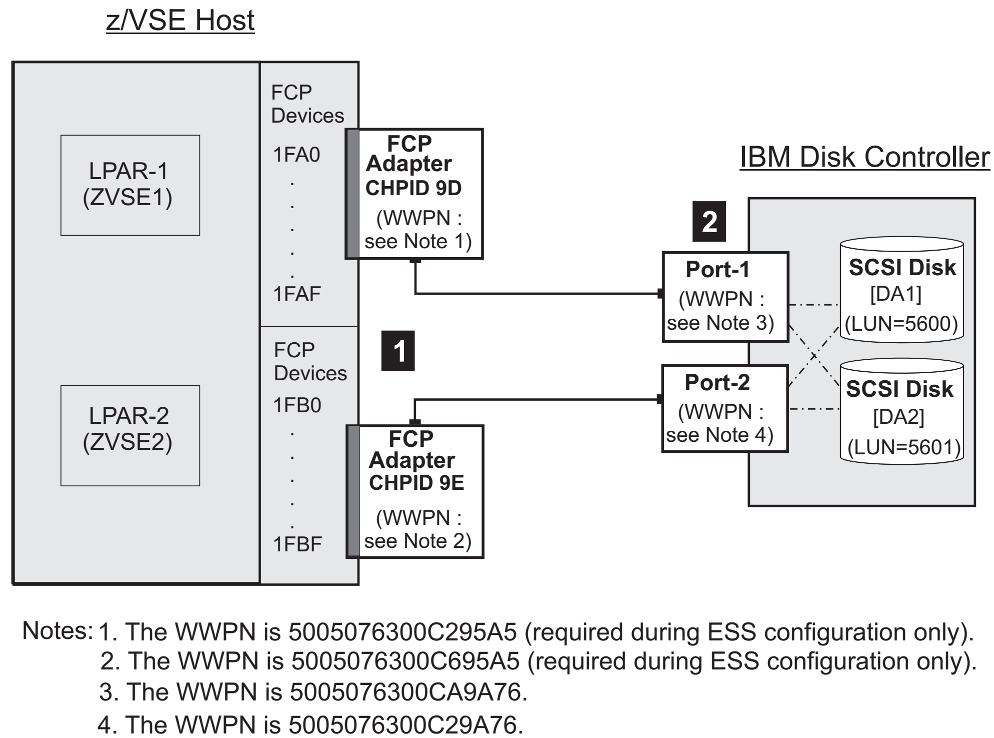


Figure 25. Example of a SCSI Environment Using Point-to-Point Connections

Note: The configuration shown in Figure 25 includes the *physical addresses* of the FCP I/O devices. In "Defining FCP Devices, SCSI Disks and Connection Paths to z/VSE" on page 105, these physical addresses are mapped to *VSE addresses* (cuu) that can be used by z/VSE.

The example configuration shown in Figure 25 consists of:

1 Two physical System z FCP adapters with CHPIDs (channel path IDs) `9D` and `9E`. These physical FCP adapters (CHPIDs) are accessed as devices of type FCP that have been configured using the IOCP (Input/Output Configuration Program). Both physical FCP adapters (CHPIDs) are shared by the z/VSE systems running in LPAR-1 and LPAR-2. This is shown in the IOCP statements of Figure 26 on page 103. The physical FCP adapters can reside on the same FCP card or on different FCP cards (this is discussed in "Using Multipathing to Access SCSI Disks" on page 110).

The number of FCP devices that can be defined for each physical FCP adapter depends upon the hardware you are using. In Figure 25:

- Sixteen FCP devices have been defined for physical FCP adapter (CHPID 9D) in the range `1FA0` to `1FAF`.
- Sixteen FCP devices have been defined for physical FCP adapter (CHPID 9E) in the range `1FB0` to `1FBF`.

Note: Within one z/VSE system, one FCP device per physical FCP adapter is sufficient for accessing *both* the FBA-SCSIs `DA1` and `DA2`.

2 The first port (Port-1) on the disk controller has the worldwide port number (WWPN) `5005076300CA9A76`. The second port (Port-2) on the disk controller has the WWPN `5005076300C29A76`. Each port is connected to **one** physical FCP adapter via a physical cable. Therefore:

- FCP devices 1FA0 to 1FAF can only access Port-1.

- FCP devices 1FB0 to 1FBF can only access Port-2.

Ports are configured using the configuration software provided with the disk controller, as described for the example provided in "Configuring SCSI Disks in the Disk Controller."

A WWPN is a unique 64-bit string (16 hexadecimal numbers) that represents a port.

The disk controller contains two SCSI disks with the logical unit numbers (LUN) 5600 and 5601. A LUN represents a SCSI device. The LUNs have been configured so they can be accessed via both FCP adapters. LUNs are configured using the configuration software provided with the disk controller, as described in "Configuring SCSI Disks in the Disk Controller." The cuu addresses DA1 and DA2 are only used by z/VSE and are not part of the configuration process for the disk controller.

Configuring FCP Adapters Using IOCP

Each FCP adapter that you plan to use with a System z server must be configured in the IOCP (Input/Output Configuration Program).

An FCP adapter is identified in the System z I/O configuration by its channel path identifier (CHPID). The channel type for an FCP adapter is FCP. For the examples shown in Figure 24 on page 100 and Figure 25 on page 102, you would use these type of statements:

```
⋮
CHPID PATH=(9D),SHARED,                                         *
      PARTITION=((ZVSE1,ZVSE2),(=)),TYPE=FCP
CHPID PATH=(9E),SHARED,                                         *
      PARTITION=((ZVSE1,ZVSE2),(=)),TYPE=FCP
⋮
CNTLUNIT CUNUMBR=1FA0,PATH=(9D),UNIT=FCP
CNTLUNIT CUNUMBR=1FB0,PATH=(9E),UNIT=FCP
⋮
IODEVICE ADDRESS=(1FA0,016),UNITADD=00,CUNUMBR=(1FA0),UNIT=FCP
IODEVICE ADDRESS=(1FB0,016),UNITADD=00,CUNUMBR=(1FB0),UNIT=FCP
```

Figure 26. IOCP Statements Used For Configuring FCP Adapters

Note: SCSI disks are *not* configured using the IOCP. Instead, the configuration programs supplied with the disk controller are used.

Configuring SCSI Disks in the Disk Controller

Note:

1. This topic provides you with an overview of the actions you must perform using the *ESS Specialist* (the configuration program of the *IBM TotalStorage ESS controller*). Other IBM disk controllers will have a different configuration process. For full details, refer to the documentation provided with your disk controller.
2. If you *are* using an IBM TotalStorage ESS controller, please be aware that the information in this topic may not exactly match the version of the ESS Specialist you might be currently using.
3. The configuration steps described below assume that you do **not** have *NPIV mode* enabled in your FCP adapter.

4. If you **do** have *NPIV mode* enabled, you must use the WWPN that is associated with an *FCP subchannel*. You should **not** use the WWPN associated with the *physical FCP adapter* (the CHPID).

1. **Define an FCP adapter**.For each of the two FCP adapters shown in Figure 24 on page 100 or Figure 25 on page 102, you must:
   a. From the main menu, select **Storage Location**, then **Open System Storage**, and then **Modify Host Systems**. Now you can enter a **Nickname**. For **Host-Type**, you must select RS/6000® or Linux. The Host-Type will determine the format of the LUN (logical unit number) to be generated.
   b. In the *Host Attachment* selection, select **Fibre Channel Attached**.
   c. In the field *Worldwide Port Name*, you must specify the WWPN of your FCP adapter (CHPID) (for the example, either `5005076300C295A5` or `5005076300C695A5`). A WWPN is automatically assigned to each physical FCP adapter (CHPID) when it is shipped from the factory. You can obtain the WWPN by displaying the CHPID information using the Service Element.
   d. In the *Fibre Channel Ports* selection, select the disk controller's ports that can be accessed by the FCP adapter. You can either select all installed ports, or you can select specific ports. In the example provided, you would select Port-1 and Port-2 of Figure 24 on page 100 or Figure 25 on page 102.
   e. Perform a configuration update.
2. **Configure the Disk Controller's Ports.** For each of the two ports (Port-1 and Port-2) shown in Figure 24 on page 100 or Figure 25 on page 102, you must:
   a. From the main menu, select **Storage Location**, **Open System Storage**, and then **Configure Host Adapter Ports**.
   b. Select the FCP Host Adapter port that you wish to configure.
   c. From the *FC Ports Attributes* selection, you select:
      1) **Fibre Channel Topology**, and then **Point to Point (Switched Fabric)**.
      2) **Fibre Channel Protocol**, and then **FCP (Open Systems)**.
   d. Perform a configuration update.
3. **Define the LUNs**.
   a. From the main menu, select **Storage Location**, **Open System Storage**, and then **Add Volumes**.
   b. Select **Host** and then the **Nickname** you specified for your physical FCP adapter (entered in Step 1 above).
   c. Select **Adapter**.
   d. Select a Disk Group that contains free space.
   e. Select a volume size, and then the number of volumes (SCSI disks) you wish to define for this size.
   f. Perform a configuration update.
4. **Specify Access to the LUNs**.
   a. From the main menu, select **Storage Location**, **Open System Storage**, and then **Modify Volume Assignments**.
   b. Select the volumes you wish to modify. In the example provided, this would be `5600` and `5601`.
   c. Click **Assign selected volumes to target hosts**, **Use same ID/LUN in source and target**, and then **Select Target Hosts**. Select the Nicknames of your physical FCP adapters (entered in Step 1 above). This will allow the FCP adapters to access the LUN.
   d. Perform a configuration update.

Defining FCP Devices, SCSI Disks and Connection Paths to z/VSE

The IUI *Hardware Configuration* dialog of z/VSE is used to define the FCP devices and SCSI disks to z/VSE.

In these examples (shown in Figure 24 on page 100 and Figure 25 on page 102), we define to z/VSE:

- FCP devices with VSE address (cuu) `FA0` that corresponds to physical address (pcuu) `1FA0`, and VSE address (cuu) `FB0` that corresponds to physical address (pcuu) `1FB0`.
- FBA-SCSI disk with VSE address (cuu) `DA1`.
- The connection path between:
  - FBA-SCSI disk with VSE address (cuu) `DA1`, and
  - the LUN 5600 via FCP device with VSE address (cuu) `FA0` that corresponds to physical address (pcuu) `1FA0`, and
  - Port-1 with WWPN `5005076300CA9A76`.

  Note that we cannot use any of the physical addresses that are defined in the IOCDS.
- FBA-SCSI disk with VSE address `DA2` (it has no physical address).
- The connection path between:
  - FBA-SCSI disk with VSE address (cuu) `DA2`, and
  - the LUN 5601 via FCP device with VSE address (cuu) `FA0` that corresponds to physical address (pcuu) `1FA0`, and
  - Port-1 with WWPN `5005076300CA9A76`.
- A second connection path between:
  - FBA-SCSI disk with VSE address (cuu) `DA1`, and
  - the LUN 5600 via FCP device with VSE address (cuu) `FB0` that corresponds to physical address (pcuu) `1FB0`, and
  - Port-2 with WWPN `5005076300C29A76`. This establishes a *multipathing* connection.

Finally, the definitions are processed to create the appropriate statements in the IPL procedure.

1. From the Administrator *z/VSE Function Selection* panel:
   a. Select *2 Resource Definition* and press **Enter**.
   b. The *Resource Definition* panel is displayed. Select *4 Hardware Configuration and IPL* and press **Enter**.
   c. Select *1 Configure Hardware* and press **Enter**.
   d. The *Hardware Configuration: Unit Address List* panel is displayed.

   Note: You can instead use the Fast Path **241** of the Interactive Interface to reach the *Hardware Configuration: Unit Address List* panel.
2. Press **PF6** (ADD ADDR) and you are prompted to enter the details for the first FCP device. Key the physical starting address `1FA0`, the corresponding VSE starting address `FA0`, and Device Name `FCP`. Press **Enter**.

```
ADM$ADD2            HARDWARE CONFIGURATION: ADD A DEVICE

 Enter the required data and press ENTER.

Specify the following physical addresses.
 STARTING ADDRESS........... 1FA0       The physical start address of an
                                        address range, or the only address to
                                        be added.
 END ADDRESS................ ____       The upper limit of the address range
                                        to be added.
Specify the following 3-digit VSE address,if needed.
 VSE STARTING ADDRESS........ FA0       The VSE address which is the mapping
                                        of the physical starting address.
 VSE END ADDRESS............ ___        The VSE address which is the mapping
                                        of the physical end address.
 DEVICE NAME................ FCP        The device you want to add or a "?"
                                        to get the group selection panel.



 PF1=HELP       2=REDISPLAY  3=END
```

The *Hardware Configuration: Unit Address List* panel is then displayed, showing the FCP device that has been added.

3. Repeat the Step 2 to add the physical Starting Address `1FB0` of the second FCP device, the corresponding VSE Starting Address `FB0`, and Device Name `FCP`.
4. Define a SCSI disk:
   a. Press **PF6** (ADD ADDR) to start the procedure of defining a SCSI disk.
   b. Key the VSE Starting Address `DA1` of the first SCSI disk, and Device Name `FBA-SCSI`. Press **Enter**. The *Hardware Configuration: Disk List* is displayed, showing the example SCSI disk with cuu address `DA1`.

```
ADM$DSK2             HARDWARE CONFIGURATION: DISK LIST



Options:  2 = Alter device type code/mode                   5 = Delete a disk
          3 = Specify Shared and/or Device Down by an 'X' in the appr. column
          8 = Specify DEF SCSI command
    OPT      VSE     DEVICE   DEVICE-TYPE  DEVICE SPEC   SHARED     DEVICE
             ADDR               CODE          MODE                   DOWN

    _        DA1     FBA-SCSI    FBA                        _          _
    _                                                       _          _
    _                                                       _          _
    _                                                       _          _
    _                                                       _          _
    _                                                       _          _
    _                                                       _          _
    _                                                       _          _
    _                                                       _          _
    _                                                       _          _
PF1=HELP       2=REDISPLAY  3=END                     5=PROCESS
```

Note: From this step onwards, the *VSE addresses (cuu) only* are required in this procedure!

5. Define a Connection Path.
   a. Key an '**8**' (Specify DEF SCSI Command) next to the FBA-SCSI disk `DA1` and press **Enter**. The *Hardware Configuration and IPL: DEF SCSI* panel is displayed.

b. In this example, you now define the connection path between the SCSI disk with cuu address `DA1` and the FCP device with cuu address `FA0`, via the Port-1 with WWPN `5005076300CA9A76`. The SCSI disk has the LUN `5600`.

```
TAS$ICME          HARDWARE CONFIGURATION AND IPL: DEF SCSI

Enter the required data and press ENTER.




FBA ..........      DA1                 cuu of the FBA-SCSI device

FCP ..........      FA0                 cuu of the FCP device

WWPN .........      5005076300CA9A76    World wide port name of the
                                        remote controller

LUN ..........      5600                Logical unit number of the SCSI



PF1=HELP       2=REDISPLAY  3=END
```

c. Press **Enter** and the *Hardware Configuration and IPL: DEF SCSI* panel is displayed, showing *all* the connection paths that have been created (the first entry below is from a previous entry, and is not part of this example).

```
TAS$ICMD          HARDWARE CONFIGURATION AND IPL: DEF SCSI

Enter the required data and press ENTER.

OPTIONS: 1 = ADD             2 = ALTER
         5 = DELETE

 OPT     FBA       FCP       WWPN              LUN
 _       233       C01       5005076300C693CB  5176
 _       DA1       FA0       5005076300CA9A76  5600
 _
 _
 _
 _
 _
 _
 _
 _


PF1=HELP       2=REDISPLAY  3=END                     5=PROCESS
```

d. **Press PF5 (Process) to process all the entries listed.** The *Hardware Configuration: Disk List* panel is displayed, showing the SCSI disk you have added. Again **press PF5 (Process) to process the entries listed.** The *Hardware Configuration: Unit Address List* panel is then displayed

6. Now repeat Steps 4 and 5 to:
   a. Add the second SCSI disk with VSE Starting Address `DA2`.
   b. Define the connection path between the SCSI disk with cuu address `DA2` and the FCP device with cuu address `FA0`, via Port-1 with the WWPN `5005076300CA9A76`. The SCSI disk has LUN `5601`.

When you have repeated Steps 4 and 5, the *Hardware Configuration: Unit Address List* panel now includes both SCSI disks `DA1` and `DA2`, and both FCP devices `FA0` and `FB0`.

7. Create a *multipathing* connection. Now we will create a connection path between the SCSI disk with cuu address DA1 and the FCP device with cuu address FB0, via Port-2 with the WWPN 5005076300C29A76. The SCSI disk has the LUN 5600.
   a. From the *Hardware Configuration: Unit Address List* panel, key a '**3**' (Select for Further Processing) next to the FBA-SCSI disk DA1, and press **Enter**. The *Hardware Configuration: Disk List* panel is displayed, showing the SCSI disk with cuu address DA1.
   b. Key an '**8**' (Specify DEF SCSI Command) next to this FBA-SCSI disk and press **Enter**. The *Hardware Configuration and IPL: DEF SCSI* panel is again displayed, showing the two connection paths you have already defined.
   c. Key a '**1**' next to the connection path entry for DA1 and press **Enter**. The *Hardware Configuration and IPL: DEF SCSI* panel is displayed with the fields for FBA and LUN already completed.

```
TAS$ICME          HARDWARE CONFIGURATION AND IPL: DEF SCSI

Enter the required data and press ENTER.




FBA ...........      DA1                cuu of the FBA-SCSI device

FCP ...........                         cuu of the FCP device

WWPN ..........                         World wide port name of the
                                        remote controller

LUN ...........      5600               Logical unit number of the SCSI



PF1=HELP       2=REDISPLAY  3=END
```

   d. Key the FCP cuu address FB0 and the WWPN (for Port-2) 5005076300C29A76, and press **Enter**. The *Hardware Configuration and IPL: DEF SCSI* panel is displayed, showing *all* the connection paths that have been created (the first entry below is not part of this example).

```
TAS$ICMD          HARDWARE CONFIGURATION AND IPL: DEF SCSI

Enter the required data and press ENTER.

OPTIONS: 1 = ADD             2 = ALTER
         5 = DELETE

 OPT     FBA       FCP       WWPN              LUN
 _       233       C01       5005076300C693CB  5176
 _       DA1       FA0       5005076300CA9A76  5600
 _       DA1       FB0       5005076300C29A76  5600
 _       DA2       FA0       5005076300CA9A76  5601
 _
 _
 _
 _
 _
 _

PF1=HELP       2=REDISPLAY  3=END                    5=PROCESS
```

e. **Press PF5 (Process) to process all the entries listed.** The *Hardware Configuration: Disk List* panel is displayed. **Again Press PF5 (Process).**

f. The *Hardware Configuration: Unit Address List* panel is then displayed. **Again press PF5 (Process) to process all the entries listed.** The *Hardware Configuration: Catalog Startup Members* panel is then displayed. Press **Enter**.

8. The *Job Disposition* panel is then displayed. You can select:
   - Option 1. Submit the job to the VSE/POWER queue.
   - Option 2. File the job in your primary ICCF library.
   - Option 3. Perform both the above actions.

   Press **Enter**. The following statements will now be included in the IPL procedure:

   ```
   ADD DA1:DA2,FBA
   ADD 1FA0 AS FA0,FCP
   ADD 1FB0 AS FB0,FCP
   DEF SCSI,FBA=DA1,FCP=FA0,WWPN=5005076300CA9A76,LUN=5600
   DEF SCSI,FBA=DA1,FCP=FB0,WWPN=5005076300C29A76,LUN=5600
   DEF SCSI,FBA=DA2,FCP=FA0,WWPN=5005076300CA9A76,LUN=5601
   ```

   You can list this procedure using the LIBR LIST member.

For details of the ADD and DEF statements, refer to the *z/VSE System Control Statements*, SC34-2637.

Using JCL Statements to Define or Delete Connection Paths

You can use JCL statements to define or delete connection paths between an FBA-SCSI disk and the associated LUN. If you define a connection path, the FBA-SCSI device and the FCP device must have been added during IPL.

Note: in the statements that follow, *cuu* and *cuu2* are *VSE addresses*.

To define a connection path, you can enter this statement in the attention routine (AR):

```
SYSDEF SCSI,FBA=cuu,FCP=cuu2,WWPN=portname1,LUN=lun
```

Alternatively, you can define a connection path in the BG partition using this statement:

```
// SYSDEF SCSI,FBA=cuu,FCP=cuu2,WWPN=portname1,LUN=lun
```

To delete a connection path, you must:

1. Set the FBA-SCSI offline using the attention routine (AR) OFFLINE command.
2. Use this statement in the attention routine (AR):

   ```
   SYSDEF SCSI,DELETE,FBA=cuu,FCP=cuu2,WWPN=portname1,LUN=lun
   ```

Note: Any connection paths you define using the SYSDEF statement will only exist until the next IPL of your system is made. If you wish to specify *permanent* connection paths, you must use the Interactive Interface.

For details of the SYSDEF SCSI command, refer to the *z/VSE System Control Statements*, SC34-2637.

Checking Which SCSI Devices Are Available

To obtain the configuration of all SCSI devices in the system, you can use the JCL QUERY SCSI command:

```
QUERY SCSI
```

To obtain the configuration of a single SCSI device in the system, you can also use the JCL QUERY SCSI command:

```
QUERY SCSI,cuu
```

Using Multipathing to Access SCSI Disks

Multipathing means that one or more alternate connection paths exist to a SCSI disk. It is used to increase the availability of a SCSI disk.

To implement multipathing, the FCP devices used to access a SCSI disk (LUN) must be on *different* physical FCP adapters (CHPIDs). If one connection path is no longer available due to an outage of an FCP adapter, the alternate connection path is used.

An FCP card can contain more than one physical FCP adapters (CHPIDs). Because maintenance activities might affect all physical FCP adapters (CHPIDs) contained on one FCP card, you might wish to use CHPIDs belonging to *different* FCP cards in your multipathing configuration. If you also wish to protect against the possible outage of a port, you can define an alternate connection path via a different port.

The examples of Figure 24 on page 100 and Figure 25 on page 102 shows a multipathing configuration for the z/VSE system in LPAR-1 and the SCSI disk device DA1. In this configuration, you are protected against outages of either the physical FCP adapter or the port.

```
DEF SCSI,FBA=DA1,FCP=FA0,WWPN=5005076300CA9A76,LUN=5600
DEF SCSI,FBA=DA1,FCP=FB0,WWPN=5005076300C29A76,LUN=5600
```

If you perform a QUERY SCSI command, the information displayed would look like this:

```
AR 0015 FBA-CUU   FCP-CUU   WORLDWIDE PORTNAME   LOGICAL UNIT NUMBER
AR 0015     DA1       FA0     5005076300CA9A76      5600000000000000
AR 0015     DA1MP     FB0     5005076300C29A76      5600000000000000
```

The first connection path that is displayed is used by z/VSE to access the LUN.

If you were to use the same WWPN, you would only be protected against an outage of the physical FCP adapter, as shown below.

```
DEF SCSI,FBA=DA1,FCP=FA0,WWPN=5005076300CA9A76,LUN=5600
DEF SCSI,FBA=DA1,FCP=FB0,WWPN=5005076300CA9A76,LUN=5600
```

Using Shared SCSI Disks

DASD sharing means that a SCSI disk can be shared between more than one z/VSE system.

DASD sharing requires that a *lock file* is defined on a shared disk. For details of how to specify a lock file for use with FCP-attached FBA-SCSI disks, see "Changing Startup When Lock File Is Stored On SCSI DASD" on page 34.

Important:
- If the lock file resides on a SCSI disk, you *must* ensure that each z/VSE system that shares this lock file has its own physical FCP adapter (CHPID). However:
  - This is not applicable if you have *NPIV mode* enabled[2] in your FCP adapter.
  - In *NPIV mode*, the FCP devices that are used to access the lock file can reside on the *same* physical FCP adapter (CHPID).
- In each DLF command, you must specify an FCP device that references a physical FCP adapter. The FCP device used in a DLF command of another z/VSE system must reference a *different* physical FCP adapter. However:
  - If you have *NPIV mode* enabled in your FCP adapter, the FCP parameter is **not** required.
- If the lock file resides on a SCSI disk, you are strongly recommended **not** to define other system files on the lock file device.
- If the lock file resides on a SCSI disk and the z/VSE system abends, you must ensure that you IPL your z/VSE system immediately. To do so, you must use the **same** IPL procedure that you used to IPL the z/VSE system. This is required in order to free the SCSI disk for use by other z/VSE systems.

In addition, the use of the lock file device has these restrictions:
- The lock file must not reside on the DOSRES or SYSWK1 SCSI disks.
- A multipath connection to the lock file device is not allowed (it will be rejected by z/VSE).

In Figure 24 on page 100 and Figure 25 on page 102, the lock file resides on SCSI disk `DA2`. This lock file is shared between ZVSE1 (the first z/VSE system) and ZVSE2 (the second z/VSE system). ZVSE1 has a connection path to the SCSI disk that uses the FCP device `FA0`. ZVSE2 has a connection path to the SCSI disk that uses, for example, the FCP device `FB1`.

The FCP device `FB0` is already used by ZVSE1 for multipathing.

Providing *NPIV mode* is **not** enabled in your FCP adapter, the statements for this shared DASD configuration would be:

```
DEF SCSI,FBA=DA2,FCP=FA0,WWPN=5005076300CA9A76,LUN=5601
DLF UNIT=DA2,BLK=...,NBLK=...,FCP=FA0  (configured in ZVSE1)

DEF SCSI,FBA=DA2,FCP=FB1,WWPN=5005076300C29A76,LUN=5601
DLF UNIT=DA2,BLK=...,NBLK=...,FCP=FB1  (configured in ZVSE2)
```

If *NPIV mode* **is** enabled in your FCP adapter (for example, CHPID 9D), the FCP device in the `DEF SCSI` statement for system ZVSE2 could reside on CHPID 9D. The statements would be:

2. NPIV is an abbreviation for *N_Port ID Virtualization*. NPIV is available, for example, with the FiconExpress2 card of an IBM System z9 server.

```
DEF SCSI,FBA=DA2,FCP=FA1,WWPN=....,LUN=5601
DLF UNIT=DA2,BLK=...,NBLK=...          (configured in ZVSE2)
```

The dialog *Tailor IPL Procedure* supports the use of the DLF command. For details, see "Changing Startup When Lock File Is Stored On SCSI DASD" on page 34.

For further details about:
- The DEF and DLF statements, refer to the manual *z/VSE System Control Statements*.
- DASD sharing, refer to the manual *z/VSE Guide to System Functions*.

Using the Attention Routine OFFLINE / ONLINE Commands

If you enter the command:

```
OFFLINE cuu
```

where *cuu* is the FCP device address, all connection paths containing this FCP device address will be terminated. All currently ongoing I/Os against the SCSI devices will be cancelled.

If you enter the command:

```
ONLINE cuu
```

where *cuu* is the FCP device address, all previously-defined connection paths containing this FCP device address will be reactivated.

For further details about the OFFLINE and ONLINE commands, refer to the manual *z/VSE System Control Statements*.

Performing an IPL of z/VSE From a SCSI Disk

Note: The information provided here is based upon the examples provided in Figure 24 on page 100 and Figure 25 on page 102. For detailed information about how to perform an IPL of z/VSE, refer to the manual *z/VSE Operation*.

When you perform an IPL from a *non-SCSI* disk, the IPL process uses channel-attached devices. From z/VSE 3.1 onwards, you can perform an IPL from an *FCP-attached SCSI disk*.

To perform an IPL of z/VSE from an FCP-attached SCSI disk, you use the *machine loader* (a platform-independent hardware tool). You can start the IPL from either:
- A VM guest (described in "Initiating an IPL of z/VSE From a VM Guest" on page 113).
- An LPAR (described in "Initiating an IPL of z/VSE From an LPAR" on page 114).

Prerequisites For Performing an IPL of z/VSE From a SCSI Disk

- To perform an IPL of z/VSE from a SCSI disk, the SCSI IPL hardware feature must already be installed and enabled on your System z platform.
- If your z/VSE system is running under z/VM, the z/VM system must also support an IPL from SCSI.

Initiating an IPL of z/VSE From a VM Guest

This topic provides an overview of the steps you must follow if z/VSE is to be IPL'd from a SCSI disk, where the IPL is initiated from a VM guest. The VM guest's virtual memory is loaded with the:
- machine loader (a platform-independent hardware tool).
- parameters required to access the SCSI disk, which you define using the `SET LOADDEV` command (described below).

1. **Use the SET LOADDEV Command to Supply the Required Parameters**. You use the SET LOADDEV command to provide the machine loader with the parameters this program needs in order to access a SCSI disk. These are the parameters you must provide to the machine loader:
   - WWPN used to access the SCSI disk.
   - LUN of the SCSI disk.

   Using the examples shown in Figure 24 on page 100 and Figure 25 on page 102, to IPL your z/VSE system from the second SCSI disk that is accessed via the WWPN `5005076300C29A76` and has the LUN `5601000000000000`, you would use the command:
   ```
   SET LOADDEV PORTNAME 50050763 00C29A76 LUN 56010000 00000000
   ```
   (You must pad the LUN with zeros until it reaches 16 characters).

   You can also use the QUERY LOADDEV command to display the parameters that have been set for the machine loader. In this example, if you enter Q LOADDEV, the displayed information would look like this:
   ```
   PORTNAME  50050763 00C29A76    LUN 56010000 00000000    BOOTPROG 0
   BR_LBA    00000000 00000000
   ```
   For details of the SET LOADDEV and QUERY LOADDEV commands, refer to the manual *z/VM CP Command and Utility Reference*, SC24-6008.
2. **IPL the FCP Device**. z/VM commands are always used together with the *physical addresses* (pcuu) and not VSE addresses (cuu). Therefore, the syntax of the z/VM command is:
   ```
   IPL fcp_device_number    (pcuu)
   ```
   Using the examples shown in Figure 24 on page 100 and Figure 25 on page 102, to IPL the FCP device with the physical address `1FA0`, you would enter:
   ```
   IPL 1FA0
   ```

Initiating an IPL of z/VSE From an LPAR

This topic provides an overview of the steps you must follow if z/VSE is to be IPL'd from a SCSI disk, where the IPL is initiated from an LPAR. You use the *Hardware Management Console* (HMC) to load the z/VSE operating system into an LPAR. For details of how to navigate to the HMC *Load* panel, you should refer to the operating procedure manual for the IBM server you are using.

In the *Load* panel, first click **SCSI** from the selection shown. Then you must enter these values:

Load Address
: This is the physical address of the FCP device. In the examples of Figure 24 on page 100 and Figure 25 on page 102, this is `1FA0`.

World Wide Port Name (WWPN)
: This is the WWPN of the port on the disk controller which is used to connect to the SCSI disk. In the examples of Figure 24 on page 100 and Figure 25 on page 102, this is `5005076300C29A76`.

Logical Unit Number
: The LUN number of the SCSI disk from which the z/VSE operating system is to be IPL'd. In the examples of Figure 24 on page 100 and Figure 25 on page 102, this is `5601000000000000`.

Note: You must not change any of these fields in the *Load* panel:
- Boot program selector
- Boot record logical block address
- OS specific load parameters

(The defaults are correct).

Understanding IPL Messages Relating to SCSI Disks

During an IPL, the informational message **0I04I** displays:
- The FBA-SCSI device address (IPLDEV=...).
- SCSI parameters you have specified either using LOADDEV (under z/VM) or using the *Load* panel: FCP=..., WWPN=..., and LUN=...

Here is an example message:

```
0I04I IPLDEV=X'600',VOLSER=DOSRES,CPUID=FF0198142064
      FCP=X'1D00',WWPN=5005076300CA9A76,LUN=5606000000000000
```

Note: The FCP address X'1D00' is the *physical address* (pcuu) of the FCP device.

The FBA-SCSI device address is always the one that was used during the *previous* IPL. If the previous device address cannot be determined, z/VSE generates its own device address to be used temporarily (X'FF0' in the example below).

```
0I04I IPLDEV=X'FF0',VOLSER=DOSRES,CPUID=FF0198142064
      FCP=X'1D00',WWPN=5005076300CA9A76,LUN=5606000000000000
```

z/VSE also expects a DEF SCSI command for the SYSRES SCSI disk in your IPL procedure. If the DEF SCSI commands for the SYSRES device do not specify the same parameters that were used for the IPL, they are considered to be additional connection paths. z/VSE will always first use the IPL'd path (that was defined using LOADDEV or the *Load* panel). In the example shown below, the DEF SCSI statement for the SYSRES device uses a different path than the one used for the IPL.

```
BG 0000 DEF SCSI,FBA=600,FCP=C01,WWPN=5005076300CA9A76,LUN=5606000000000000
BG 0000 DEF SCSI,FBA=601,FCP=D00,WWPN=5005076300C69A76,LUN=5607000000000000
```

Note: The VSE address (cuu) X'D00' is now assigned to the physical address (pcuu) X'1D00'.

If you perform a QUERY SCSI command, the information displayed would look like this:

```
AR 0015 FBA-CUU   FCP-CUU   WORLDWIDE PORTNAME  LOGICAL UNIT NUMBER
AR 0015     600       D00     5005076300CA9A76     5606000000000000
AR 0015     600MP     C01     5005076300CA9A76     5606000000000000
AR 0015     601       D00     5005076300C69A76     5607000000000000
```

Note: The IPL path is given by FCP-CUU D00, assigned from the physical address (pcuu) 1D00.

Errors That Might Occur During Configuration

This topic describes the errors that might occur when you configure your SCSI devices for use with z/VSE. Possible solutions are provided.

Error Message	Reason Code	Cause	Remedy
0S40I	0018	The FCP adapter is probably not authorized to access the port identified by *WWPN*.	Select the port (WWPN) in the *Fibre Channel Ports* selection. For details, see "Configuring SCSI Disks in the Disk Controller" on page 103.
0S40I	002F	The FCP devices used for establishing multipathing are defined for the same physical FCP adapter.	Use FCP devices defined on different physical FCP adapters. For details, see "Using Multipathing to Access SCSI Disks" on page 110.
0S40I	0023	The port identified by *WWPN* is not registered in the Nameserver of your FCP switch. Either: 1. The port does not exist (an incorrect WWPN was specified). 2. The port has been set "Offline" in the FCP switch. 3. The physical cable between FCP switch and disk controller is not properly connected.	Either: 1. Enter a valid WWPN. 2. Set the port to "Online" in the FCP switch. 3. Ensure that the physical cable between FCP switch and disk controller is properly connected.
0S40I	0102, 0018	Confirm that FSFCMD=00000005 and that FSFSTAT shows BADDEF. In this case, the port identified by WWPN is not an Open FCP port.	Correct your settings in the *FC Ports Attributes* selection. For details, see "Configuring SCSI Disks in the Disk Controller" on page 103.
0S41I	–	Your SCSI disk does not support ANSI SCSI Version 3.	You must use a SCSI disk that supports ANSI SCSI Version 3.
0S42I	–	The block size of your SCSI disk is not **512 bytes**.	Re-configure your disk controller.
0S43I	–	The size of your SCSI disk is less than 8 MB, which is the minimum supported size.	Increase your SCSI disk size.

SCSI Disks

Error Message	Reason Code	Cause	Remedy
0S44I	–	The size of your SCSI disk is greater than 24 MB, which is the maximum supported size. z/VSE will only use the first 24 MB of your SCSI disk.	This message is provided for information only. It informs you that you are wasting disk space.
0S46I	052500 0B2500	Either: 1. The LUN does not exist in the disk controller. 2. The physical FCP adapter is not authorized to access the LUN.	Either: 1. Enter a valid LUN. 2. Select the Nickname of your physical FCP adapter in the *Assign selected volumes to target hosts* selection. For details, see "Configuring SCSI Disks in the Disk Controller" on page 103.

Chapter 9. Configuring Your System to Use PAV

This chapter describes how you configure your z/VSE system to use *Parallel Access Volume* (PAV) support that is available with the DS6000 and DS8000 Series.

It contains these main topics:

Related Topic:

For details of how to ...	Refer to ...
add an ECKD volume using the Interactive Interface	"Using the Configure Hardware Dialog" on page 88
add an ECKD volume using the ADD statement	*z/VSE System Control Statements*
start and stop PAV support using the `SYSDEF SYSTEM,PAV=xxxxx` command	

Overview of PAV Support

PAV reduces storage management costs that are associated with maintaining a large numbers of volumes that would otherwise be required.

The *PAV-alias* volumes simulate "alternate paths" to a PAV-base volume. For example, a 3390 with one PAV-alias represents one "physical" volume, but two paths can be used for I/O operations.

For S/390® and System z platforms, PAV-support:
- Enables *simultaneous* data-transfer operations, to/from the same volume.
- Allows multiple users and jobs to *simultaneously* access a volume.
- Allows read and write operations to be performed *simultaneously* to/from different domains (the domain of an I/O operation is the specified extent to which the I/O operation applies).

A PAV volume has one volume serial number and multiple device addresses. The device addresses are "represented" by subchannels (these terms are sometimes used interchangeably). When accessing a PAV volume *in parallel*, the I/O operations will be executed *simultaneously* (except for I/O operations that address the same extent).

Using an internal algorithm and load-balancing methods, z/VSE distributes the I/O operations between the available PAV-base and PAV-alias volumes.

Prerequisites for Using PAV Support

Before you can use PAV support, you must have:

- Obtained a PAV license. When you order a PAV license, you specify the feature code that represents the physical capacity allowed for the function. Therefore, only a subset of the physical capacity installed may be assigned to PAV-usage.
- Activated the appropriate PAV feature code.
- Purchased the FICON/ESCON attachment feature (for TURBO models).
- Configured IOCP so that your configuration reflects and matches the DS6000/DS8000 configuration.
- Configured both your storage unit and operating system to use PAVs. You can use the logical configuration definition to define PAV-bases, PAV-aliases, and their relationship in the storage unit hardware. This unit address relationship creates a single logical volume, allowing concurrent I/O operations. The topic "Modifying zSeries® Volumes" in the IBM *System Storage DS® Storage Manager* also describes how to configure and maintain base and alias volumes.
- Created a configuration on the ESS that matches the configuration you created in the IOCP. As each alias volume also has a uniquely subchannel, the IOCP must also contain an IODEVICE statement for each alias volume.
- Used an `ADD` statement to add the PAV-base in the z/VSE IPL procedure, as described in "Defining PAV Volumes to z/VSE" on page 120.
- Activated z/VSE PAV support via the PAV operand of the `SYSDEF SYSTEM` statement. This can be issued as:
  - an attention routine (AR) command from the system console or master console, or
  - a JCL command included in the startup procedure ($0JCL) of the BG partition.

Restrictions/Considerations When Using PAV Support

- z/VSE only supports up to 7 PAV-alias devices per PAV-base volume.
- Device addresses that are configured as PAV-alias volumes cannot be added or used in the z/VSE system.
- If a PAV-alias device is specified as IPL device, the system will try to automatically identify the SYSRES base device and switch to it as IPL device. If the PAV-base device *cannot* be identified, the system will enter a hardwait.
- PAV-alias device addresses are also subject to the z/VSE 3-digit cuu restriction.
- During PAV activation, z/VSE will consider all available PAV-alias devices for a PAV-base to be usable. z/VSE does not allow you to exclude or only use some of the available PAV-aliases.
- z/VSE does not allow you to *dynamically* add a *PAV-alias device*. z/VSE ignores any such device that becomes ACTIVE. However, if a *PAV-base device* becomes READY when PAV is support is ACTIVE, all the corresponding PAV-aliases will also be activated.
- z/VSE will remove from processing any alias devices that become non-operational. To notify you that an alias device has been removed from processing, the PAV-alias `cuu` will be surrounded by parentheses on the output from the `VOLUME cuu,DETAIL` command.
- Although a PAV-alias might be considered to be a "virtual" device, it *does* use copy blocks, channel queue entries, and other I/O-related resources (in the same way a "real" device does). Therefore when you allocate such resources, you *must* consider alias devices.

- For transparency with earlier releases, the z/VSE system services and messages will only consider the PAV-base device.
- PAV processing will *not* improve the performance of a “single application” PAV-base. However if you have many applications running in parallel that issue I/Os against a single device, PAV processing *will* improve performance.
- The Subsystem Monitoring Facility (SMF) is enabled to collect information from the *PAV-base device only*.
- If you are running under z/VM and have the required licenses and hardware, z/VM allows (per default) a specified control unit to operate with *HyperParallel Access Volume* (HyperPAV) devices. HyperPAV devices that are attached to z/VSE *cannot be used with z/VSE's PAV support!* For details of how to configure your PAV devices for use with a z/VSE system that runs under z/VM, refer to the description of the QUERY CU and SET CU commands in the *z/VM CP Commands and Utilities Reference* (SC24-6081).

Configuring PAV Volumes Using IOCP

The configuration that you define on the DS8000 or DS6000 must match your IOCP (Input/Output Configuration Program) volume-device configuration. Two examples are provided below of how an IOCP configuration might be defined.

The first example shows a basic IOCP configuration for a PAV-base volume and PAV-alias volumes.

```
⋮
**********************************************************************
* DEFINE 2105-E20 LOGICAL CONTROL UNIT 0                             *
**********************************************************************
      CNTLUNIT CUNUMBR=0700,PATH=(70,71,72,73),UNITADD=((00,256)),   *
             LINK=(24,2D,34,3D),CUADD=0,UNIT=3990

**********************************************************************
* DEFINE 3390-9 BASE AND ALIASES ADDRESS ON LOGICAL CONTROL UNIT 0   *
*       16 BASE ADDRESS, 3 ALIASES PER BASE                          *
**********************************************************************
      IODEVICE ADDRESS=(900,016),CUNUMBR=(0700),STADET=Y,UNIT=3390
      IODEVICE ADDRESS=(301,048),CUNUMBR=(0700),STADET=Y,UNIT=3390
**********************************************************************
⋮
```

Figure 27. Basic IOCP Configuration for PAV Volumes

The second example shows how you can use IOCP to maintain an overview of which volumes have been defined as the PAV-base, and which have been defined as PAV-aliases. Although the IOCP configuration is the same as in the previous example, the use of the UNIT parameter provides additional clarity.

```
⋮
*********************************************************************
* DEFINE 2105-E20 LOGICAL CONTROL UNIT 1                            *
*********************************************************************
      CNTLUNIT CUNUMBR=0701,PATH=(70,71,72,73),UNITADD=((00,128)),   *
             LINK=(24,2D,34,3D),CUADD=1,UNIT=2105

*********************************************************************
* DEFINE 3390-9 BASE AND ALIASES ADDRESS ON LOGICAL CONTROL UNIT 1   *
*       16 BASE ADDRESS, 3 ALIASES PER BASE                         *
*********************************************************************
      IODEVICE ADDRESS=(A00,016),CUNUMBR=(0701),STADET=Y,UNIT=3390B
      IODEVICE ADDRESS=(B00,048),CUNUMBR=(0701),STADET=Y,UNIT=3390A
*********************************************************************
⋮
```

Figure 28. IOCP Configuration for PAV Volumes With Additional Information

Defining PAV Volumes to z/VSE

The IUI *Hardware Configuration* dialog of z/VSE is used to define the *PAV-base* to z/VSE.

Note:

1. You must *not* define your *PAV-alias* volumes to z/VSE!
2. An ADD statement for an alias device will be ignored.

Note:

You define the PAV-base volume to z/VSE in the same way as for any other ECKD volume (see “Using the Configure Hardware Dialog” on page 88). For details of how to use an ADD command to define a volume to z/VSE, refer to *z/VSE System Control Statements*, SC34-2637.

Activating PAV Using AR Commands or JCL Statements

To activate PAV support, you can enter this command in the attention routine (AR):

```
SYSDEF SYSTEM,PAV=START
```

During PAV-activation, *all* PAV-aliases of the PAV-base volume will be included in PAV processing.

To quiesce (stop) PAV support, you can enter this command in the attention routine (AR):

```
SYSDEF SYSTEM,PAV=STOP
```

There might be a delay before the STOP processing finishes. This is because the I/O traffic must first complete for all PAV-aliases that are being used.

Any PAV activation will only exist until the next IPL of your system is made. If you wish to specify *permanent* PAV activation, you must include an appropriate *JCL statement* in your startup procedure for the BG partition ($0JCL). To do so, use skeleton SKJCL0 in ICCF Library 59.

For details of the `SYSDEF SYSTEM` command, refer to *z/VSE System Control Statements*, SC34-2637.

Checking Which PAV Volumes Are Available

- You can use the QUERY SYSTEM command to display the current PAV setting (highlighted below):

```
QUERY SYSTEM
AR 0015   NUMBER OF TASKS TOTAL LIMIT: 255
AR 0015   OLD SUBTASKS LIMIT:         163   IN USE:   3   MAX. EVER USED:   4
AR 0015   NEW SUBTASKS LIMIT:           0   IN USE:   0   MAX. EVER USED:   0
AR 0015   PARALLEL ACCESS VOLUME (PAV):  ACTIVE
```

- You can use the VOLUME command to display PAV information about one or more volumes (the *B extension is highlighted below in the CODE column):

```
VOLUME 777
AR 0015 CUU  CODE  DEV.-TYP   VOLID   USAGE     SHARED   STATUS   CAPACITY
AR 0015 777  6E*B  2105-000   DOSRE1   USED                           1200 CYL
```

- You can use the VOLUME command together with the DETAIL parameter to display the relationship between base and alias volumes:

```
VOLUME 777,DETAIL
AR 0015 CUU  CODE  DEV.-TYP  VOLID   USAGE   SHARED   STATUS  CAPACITY
AR 0015 777  6E*B  2105-000  DOSRE1   USED                        1200 CYL
AR 0015      BASE  TO 778,xxx,...    (up to 7 alias cuu's)
```

Note: If a cuu is enclosed in parentheses, this indicates that the PAV-alias device is *not operational*. For example:

```
AR 0015      BASE  TO 778,77E,779,(77A),77B
```

For further details of the QUERY SYSTEM and VOLUME commands, refer to *z/VSE System Control Statements*, SC34-2637.

Chapter 10. Tailoring the Interactive Interface

z/VSE provides dialogs for tailoring the Interactive Interface according to the needs of your installation. You can:

- Define user profile information.
- Change the selections offered by the Interactive Interface panels.
- Include your own CICS applications so that you can access them from the Interactive Interface.

z/VSE provides four dialogs to help you tailor the Interactive Interface according to your needs:

Maintain User Profiles

Maintain Selection Panels

Maintain Application Profiles

Maintain Synonyms

- The *Maintain User Profiles* is described in Chapter 25, "Maintaining User Profiles via BSM Dialogs," on page 289.
- The remaining three dialogs are described in detail in this chapter.

In addition, the panel *User Interface Tailoring* offers two more dialogs which allow the system administrator to:

- *Maintain PRIMARY Sublibraries*

  This function is for creating, maintaining, and deleting PRIMARY sublibraries. A VSE (not VSE/ICCF) PRIMARY sublibrary is created for any user who is authorized in the user profile.
- *Customize z/VSE Workstation Platform*

  This dialog supports workstation integration and allows the specification of up to 3 classes for file transfer between a workstation and VSE sublibraries. Refer to the manual *VSE/ESA Programming and Workstation Guide* under "The Librarian Transaction Server" for further details about this dialog.

Related Topic:

For details of how to ...	**Refer to the ...**
use the Interactive Interface to define *user entries*	Chapter 25, "Maintaining User Profiles via BSM Dialogs," on page 289.

Planning Considerations for Using the Interactive Interface

Before you change the Interactive Interface provided by z/VSE, you should carefully plan how you want your system to look like. Review the information in the following topics about the dialogs provided.

You should be aware that:

- For each user signing on, about 2 KB of virtual storage are needed in the associated CICS partition.
- VSE/ICCF is available for type 1 and 2 users with a 4-character user-ID.

  VSE/ICCF is **not** available for type 3 users and type 1 and 2 users with a 5 to 8-character user-ID.

VSE CONTROL FILE

User interface tailoring is done by maintaining records in the z/VSE control file. The file is used by each CICS TS and by the BSM (Basic Security Manager). To close the file in case of an update or the alteration of VSE/VSAM options, use the following command:

```
CEMT SET FILE=(IESCNTL) CLOSE
```

Issue the command for each CICS TS. To close the control file for the BSM, issue:

```
MSG FB,DATA=CLOSECNTL
```

The control file is a VSE/VSAM KSDS file. It contains the following types of records:

1. User profile record

   A user profile record exists for each system user. The records are defined and maintained by using the *Maintain User Profiles* dialog or the batch utility IESUPDCF (described in Chapter 26, “Maintaining User Profiles via Batch Program IESUPDCF,” on page 311).

   User profile records are used by the BSM for security checking.
2. Selection panel record

   Selection panel records are used to build selection panels. z/VSE ships records for all the selection panels in the Interactive Interface. You can create and maintain other records using the *Maintain Selection Panels* dialog.
3. Application profile record

   An application profile record contains execution information about a CICS application. z/VSE ships records for the Interactive Interface dialogs.

   z/VSE also provides additional application profiles which you can include in the Interactive Interface. Under “Dialogs of the Interactive Interface”, the manual *z/VSE Planning* lists the dialogs and additional applications.

   You can create and maintain other application profile records using the *Maintain Application Profiles* dialog. This lets you incorporate your own CICS applications into the Interactive Interface.
4. Synonym record

   A synonym record exists for each system user who has defined synonyms for accessing dialogs. Each user can create and maintain own synonym records using the *Maintain Synonyms* dialog.
5. News record

   A news record contains a *news item* which is a message that the system displays when a user signs on. You define these records using the *Enter News* dialog.

   Note: Items 6, 7, 8, and 9 are defined and maintained by the *Customize z/VSE Workstation Platform* dialog.
6. System ID record

   This record exists for each z/VSE system.
7. Class for job submission

   Defines in which class the job is to run.
8. Class for librarian transaction server

   Defines the class(es) for the execution of the transaction server.
9. Library record

   This record is for VSE libraries and sublibraries that are visible to programmable workstation users.

Maintaining Selection Panels

z/VSE lets you change the structure of the Interactive Interface. You can create your own selection panels and corresponding HELP panels. In this way, you can have many interactive panel hierarchies for different users of your system.

Maintaining Selection Panels without VSE/ICCF

You can also update selection panels in an environment without VSE/ICCF (in case of a second CICS, for example), or if:

- VSE/ICCF has been terminated.
- The VSE/ICCF DTSFILE has been disconnected.
- The system administrator is a non-VSE/ICCF user.

Introduction to Maintaining Selection Panels

The *Maintain Selection Panels* dialog helps you create, change, or delete selection panels. You define the selections you want on the panel and specify what is invoked for each selection. Each selection can invoke:

1. Another selection panel. It can be a panel shipped by z/VSE or one that you create.
2. An Interactive Interface dialog.
3. An additional z/VSE application.

   z/VSE provides a number of applications that are not included in the default panel hierarchies of the Interactive Interface. If you wish, you can invoke one or more of these applications from selection panels that you create.

   In the topic “Additional z/VSE Applications”, the manual *z/VSE Planning* has an appendix that gives an overview of those applications not included in the default panel hierarchies.
4. One of your own CICS applications.

   You must define your application to the system using the *Maintain Application Profiles* dialog.

Each selection panel is defined by a selection panel record. The system stores the records in the z/VSE control file.

You can also write your own HELP information for the selection panels you create. You use the *Maintain Selection Panels* dialog to process the HELP text. The system stores HELP text in the system's text repository file IESTRFL. You can use the dialog to add, update, or delete HELP information in the text file.

z/VSE automatically manages the display of HELP panels that you create. It displays your HELP text whenever you press PF1 from a selection panel that you have created. z/VSE automatically handles backward and forward scrolling. “Creating HELP Panels” on page 130 describes how you create HELP panels.

Before you create selection panels, refer to *z/VSE Planning* under “Planning for Tailoring the Interactive Interface” for additional information about user interface tailoring. To access the dialog, start with the *z/VSE Function Selection* panel and select:

2 (Resource Definition)

1 (User Interface Tailoring)

2 (Maintain Selection Panels)

Maintaining Selection Panels

An entry panel of the *Maintain Selection Panels* dialog appears where you can press ENTER to list all available panel names, or enter the name or the first characters of the panel you want to be listed.

```
IESADMUIFS            MAINTAIN SELECTION PANELS

Specify the prefix of the Selection Panels you want to be listed and
press the ENTER key.



  SELECTION.......... ________   1 - 8 prefix characters, e.g.
                                 'AB' for all Selection Panels
                                 starting with AB.
                                 Press ENTER to list all Selection
                                 Panels.







PF1=HELP                  3=END        4=RETURN
```

A FULIST displays the selection panels defined for the system. If you want to locate a particular entry, enter the selection panel name in the LOCATE NAME field.

```
IESADMSPL                        MAINTAIN SELECTION PANELS     Page   1 of  10

CONTROL FILE
OPTIONS:   1 = ADD            2 = CHANGE             5 = DELETE
           6 = UPDATE HELP     7 = DELETE HELP

 OPT     PANEL NAME  HELP  SELECTION   SELECTABLE PANELS OR APPLICATIONS

  _       IESEADM     *       1-4     IESEINST IESEDEF  IESEOPS  IESEPROB
                              5-8     IESEGDEV IESNICCF IESECICA
                                9
  _       IESEASAV    *       1-4     IESEVSAM IESELIBR IESS$BAC IESS$RHS
                              5-8     IESEBKDT IESERSTD IESECPDD
                                9
  _       IESEBKCA    *       1-4     IESC$BMT IESC$BMD IESC$BUT IESC$BUD
                              5-8
                                9
  _       IESEBKDT    *       1-4     IESU$DMV IESU$DMF
                              5-8
                                9
PF1=HELP       2=REFRESH    3=END        4=RETURN
               8=FORWARD    9=PRINT

LOCATE NAME ==> ________
```

The options you can choose are shown at the top of the FULIST. Enter an option number in the OPT column to the left of the panel you want to process.

The dialog processes HELP text whenever you select options:

- 1 (ADD)
- 6 (UPDATE HELP)
- 7 (DELETE HELP)

When you select one of these options, the dialog searches the following libraries for the VSE/ICCF library member *which has the same name as your selection panel*:

- Primary library

- Connected library
- Common library

It then either copies the member to the text repository file or deletes it from the text file, depending on the option you selected.

Selection panel names that begin with the following characters are reserved:
- IES
- INW

You **cannot** change or delete them. You can use them as models to define your own panels.

To create a status report of selection profiles/panels that are stored in the VSE Control File, press **PF9**. A status report is then created using the reporting tool IESXSPR, and stored in the VSE/POWER List Queue. This List Queue entry has the job name IESXSTX.

The skeleton IESXSTX is provided in VSE/ICCF Library 59. This skeleton contains the source code of the report format. To create your own report layouts, you can modify this source code.

If you change the skeleton IESXSTX, you must activate the related phase using the CEMT SET PROG(IESXSTX) NEWCOPY command.

To create a status report of user profiles that are stored in the VSE Control File, press **PF9**. A status report is then created using the reporting tool IESXSPR, and stored in the VSE/POWER List Queue. The job name of this List Queue entry is IESXSUSP.

Add or Change a Panel

If you add a new panel, enter option number **1** next to the panel you want to use as a model. The model provides default values.

If you change a panel, enter option number **2** next to the panel you want to change.

After you make your selection, the dialog displays an additional panel. You need the following information:

SELECTION PANEL NAME
Specify a unique name (when adding only) for the selection panel. The name cannot begin with the characters IES or INW. These prefixes are reserved for z/VSE.

SEQ The sequence numbers of the selections on the panel. You can specify the numbers **1 - 9**, for up to nine options on the panel. The dialog automatically sorts the sequence numbers and the corresponding selection text in ascending order.

NAME
Enter a **1 - 8** character name indicating what is invoked when this selection is chosen. It can be:

- An application profile name.

  It can be a z/VSE dialog or application or your own CICS application which you have added using the *Maintain Application Profiles* dialog.

- The name of another selection panel.

  It can be a panel shipped with the system or one that you create.

The Appendix "Additional z/VSE Applications" in the manual *z/VSE Planning* lists the dialogs and additional application profiles which z/VSE provides.

TYPE This indicates whether you entered an application profile or a selection panel name in the NAME field. Enter:

1 - Application profile

2 - Selection panel

SELECTION TEXT
This is the explanation text that is shown to the right of the sequence number on the selection panel.

After you type in your information, press **ENTER**. The dialog formats the information, checks for editing errors, and redisplays the panel. Check your entries and make any changes.

When you are done, press **PF5** to update the z/VSE control file and store the selection panel record. If you are adding a new panel, the dialog also searches for corresponding HELP text. If it locates the VSE/ICCF library member, it formats the HELP text and adds it to the text repository file. If you are changing a selection panel, the dialog does **not** process HELP text.

The dialog continues and redisplays the FULIST.

Delete a Panel

Option 5 (DELETE) deletes an existing selection panel record from the z/VSE control file. If you have HELP text for the panel, the dialog also deletes it from the text repository file. However, it does **not** delete the library member which contains the HELP text from the VSE/ICCF library.

Use option 7 to delete your HELP text in both the system's text repository file and the VSE/ICCF library member.

Update HELP

Option 6 (UPDATE HELP) replaces the selection panel HELP text in the system text file with HELP text from the VSE/ICCF library member.

Delete HELP

Option 7 (DELETE HELP) deletes the selection panel HELP text from both the system text file and the VSE/ICCF library member that contains the HELP information.

The dialog does not check whether the VSE/ICCF member is found. If the correct library is not accessed, the member may not be deleted.

Rebuild Default Selection Panels

z/VSE ships selection panels for three default hierarchies:

- System administrator
- Programmer
- Operator

"z/VSE Profiles" on page 5 describes the default hierarchies.

If the default selection panel records are damaged, you can rebuild them. This can only be done using the default administrator user-ID SYSA.

When user-ID SYSA accesses the dialog, the FULIST displays PF6=SYSTEM. PF6 is only displayed for user-ID SYSA. It rebuilds the shipped selection panel records for the three default hierarchies.

Migrating Selection Panel Definitions to a Second z/VSE System

The easiest method of migrating your selection panel records is to use the IESBLDUP utility. For details, refer to the chapter "Migrating From Earlier Releases" in the *z/VSE Planning*, SC34-2635.

Alternatively, you can use the migration facility described below which enables you to:

- Generate statements containing your selection panel definitions for a z/VSE system.
- Run REXX/VSE procedure UPCNTLSP, which uses these statements to add your selection panel definitions to the VSE Control File (IESCNTL) of a second z/VSE system.

To generate your selection panel definitions, you can use tool IESXSPR to generate a job that calls REXX procedure UPCNTLSP. To do so, you must:

1. Define a transaction (for example, XSPR) to reference program IESXSPR.
2. Run transaction XSPR with parameter IESXSSPU (provided in ICCF library 59) to generate the REXX/VSE job to be run on the second z/VSE system.
3. The generated job is written to the punch queue. The name of the generated job in the punch queue will be the same as the job name of your CICS startup job (for example, CICSICCF).

Here is an example of the generated output:

```
* $$ JOB JNM=UPCNTLSP,CLASS=0,DISP=D
* $$ PUN DISP=I,CLASS=0,PRI=9
// JOB UPCNTLSP
* VSE CONTROL FILE DATA AS OF 04/06/06  18:09:44
// EXEC REXX=UPCNTLSP
TX AMADADM  64 E
60 P    IESEINST Installation
60 P    IESEDEF  Resource Definition
60 P    IESEOPS  Operations
60 P    IESEPROB Problem Handling
60 P    IESEGDEV Program Development
60 A    IESNICCF Command Mode
60 P    IESECICA CICS-Supplied Transactions
60 A    IESDITTO Ditto
60
TX BQUL     64 E
```

```
60 A    IESBQU   BQ
60
60
60
60
60
60
60
60
:
60
/*
* TOTAL OF 51      SELECTION PANELS
/&
* $$ EOJ
```

Creating HELP Panels

You can create your own HELP panels for the selection panels you create. You simply create a VSE/ICCF library member with the same name as the name of your selection panel. For example, if you create a selection panel named USERSEL, create a VSE/ICCF library member named USERSEL for your HELP text.

You can use the *Program Development Library* dialog to create VSE/ICCF library members. The topic "Handling VSE/ICCF Library Members" in the manual *VSE/ESA Programming and Workstation Guide* describes the dialog in detail.

After you create your library member, edit the member and type in your HELP text.

Do not enter lines longer than 68 characters. Lines which are longer are truncated. You can have blank lines, but trailing blanks are suppressed.

The system formats the HELP text in a way that one panel (page) of HELP text consists of sixteen lines, including blank lines. The text can have a maximum of 4000 characters, not including trailing blanks. This is approximately 6 - 8 panels of text. Note that you do not need to define how the system should manage the panel display of HELP text or forward and backward paging. The system does this automatically for you.

After creating the HELP text, you can incorporate it into the system using the *Maintain Selection Panels* dialog.

Additional Considerations When Maintaining Selection Panels

1. z/VSE ensures that only one user can access the *Maintain Selection Panels* dialog at one time.
2. Do not use the following prefixes for the name of your selection panels:
   - IES
   - INW

   These prefixes are reserved by z/VSE.
3. If you create HELP text before you create your selection panel, the dialog automatically adds the HELP to the system text file when you add (option 1) the new selection panel.

   If you create the HELP text after you create your selection panel, you can add the HELP information to the text file using option 6 (UPDATE HELP).

4. When you use the following options, the correct VSE/ICCF libraries must be accessed for correct HELP text processing:
   - 1 (ADD)
   - 6 (UPDATE HELP)
   - 7 (DELETE HELP)

   When the dialog searches for the VSE/ICCF library member with the same name as the selection panel, it searches in the following order:
   - Primary library
   - Connected library
   - Common library
5. If you specify that one of your own CICS applications is invoked from a selection panel, the application must be defined by an application profile record. You can define your own applications using the *Maintain Application Profiles* dialog.

Maintaining Application Profiles

You can include your own CICS applications in the z/VSE system and access them from the Interactive Interface. The application can be accessed from a selection panel or invoked directly when a user signs on.

Maintaining Application Profiles without VSE/ICCF

You can also update application profiles in an environment without VSE/ICCF (in case of a second CICS, for example), or if:
- VSE/ICCF has been terminated.
- The VSE/ICCF DTSFILE has been disconnected.
- The system administrator is a non-VSE/ICCF user.

The *Maintain Application Profiles* dialog helps you include your own applications in the Interactive Interface. Each application is defined by an *application profile record*. The record defines the name and characteristics of the application. The system stores application profile records in the z/VSE control file.

Before you include an application, review *z/VSE Planning* under “Planning for Tailoring the Interactive Interface” for information on user interface tailoring. There are many things to consider in terms of user profiles and selection panels before you change your system.

To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)
1 (User Interface Tailoring)
3 (Maintain Application Profiles)

An entry panel of the *Maintain Application Profiles* dialog appears where you can press ENTER to list all available application profiles, or enter the name or the first characters of the profile you want to be listed.

```
IESADMUIFA           MAINTAIN APPLICATION PROFILES

Specify the prefix of the Application Profiles you want to be listed
and press the ENTER key.



  APPLICATION........ ________   1 - 8 prefix characters, e.g.
                                 'AB' for all Application Profiles
                                 starting with AB.
                                 Press ENTER to list all Application
                                 Profiles.








PF1=HELP                    3=END        4=RETURN
```

A FULIST displays the applications defined for the system. If you want to locate a particular entry, enter the application name in the LOCATE NAME field.

```
IESADMAPL                      MAINTAIN APPLICATION PROFILES    Page   1 of  14

CONTROL FILE
OPTIONS:   1 = ADD             2 = CHANGE             5 = DELETE

                                    EXECUTION   UPPER     SHOW    SYSTEM USE
 OPT      NAME       ACTIVATE         CODE      CASE      INPUT      ONLY

  _     IESA$FST     DTRDDMGR          6         1         1         1
  _     IESA$HDW     DTRDDMGR          6         1         1         1
  _     IESA$LB      DTRDDMGR          6         1         1         1
  _     IESA$NLS     DTRDDMGR          6         1         1         1
  _     IESAPM       IETF              1         1         1         0
  _     IESAPM2      IESA              1         1         1         0
  _     IESBQU       IESQ              1         1         1         0
  _     IESC$ACT     DTRDDMGR          6         1         1         1
  _     IESC$APP     DTRDDMGR          6         1         1         1
  _     IESC$BMD     DTRDDMGR          6         1         1         1
  _     IESC$BMT     DTRDDMGR          6         1         1         1

PF1=HELP      2=REFRESH    3=END        4=RETURN
              8=FORWARD    9=PRINT

LOCATE NAME ==> ________
```

The options you can choose are shown at the top of the FULIST. Enter the appropriate option number in the OPT column to the left of the application you want to process.

To create a status report of application profiles that are stored in the VSE Control File, press **PF9**. A status report is then created using the reporting tool IESXSPR, and stored in the VSE/POWER List Queue. The List Queue entry is called IESXSAP.

The skeleton IESXSAP is provided in VSE/ICCF Library 59. This skeleton contains the source code of the report format. To create your own report layouts, you can modify this source code.

If you change the skeleton IESXSAP, you must activate the related phase using the CEMT SET PROG(IESXSAP) NEWCOPY command.

Here is an example of the use of IESXSAP to print application profiles to the VSE/POWER List Queue.

```
VSE APPLICATION PROFILE LIST
VSE CONTROL FILE
        PROFILE         EXECUTION
NUMBER  NAME     ACTIVATE CODE   PARAMETER (30 CHAR)
    1  IESA$FST DTRDDMGR  6     $$$$EASY ADM$FST

    2  IESA$HDW DTRDDMGR  6     EZSIZE=400.$$$$EASY ADM$HDW

    3  IESA$LB  DTRDDMGR  6     $$$$EASY ADM$LB

    4  IESA$NLS DTRDDMGR  6     $$$$EASY TES$MSG  27 482

    5  IESAPM   IETF      1     APM

    6  ........
```

System Provided Application Profiles

z/VSE provides a number of application profiles. The names of these profiles begin with the prefix IES and are reserved by the system. You **cannot** change or delete them. You can use them as models when you add your own application. Useful applications are, for example:

```
IESDITTO    Access DITTO/ESA for VSE
IESISQL     Access DB2 Server for VSE
```

Add or Change an Application Profile

If you add a new application profile, enter option number **1** next to the profile you want to use as a model. The model provides default values. If you change a profile, enter option number **2** next to the profile you want to change.

After you make your selection, the dialog displays an additional panel. You need the following information:

NAME
Required when adding an application profile. Specify a unique application name of **1 - 8** characters. It identifies the application to the system.

CODE Specify how the application is initiated.

- **1** = Initiate transaction via CICS START.
- **2** = LINK to a program. A CICS LINK is performed to a CICS program using the current TCA.
- **3** = ATTACH a non-conversational transaction. The transaction begins as if a transaction code had been entered from the terminal. For non-conversational transactions, END-OF-TASK does not necessarily mean “end of application”. You have two choices:
  1. You can add a line of code to your last transaction program so that it transfers control back to the Interactive Interface. (See Figure 29 on page 137.)
  2. The user can press **PF3** to return to the Interactive Interface.
- **4** = ATTACH a conversational transaction. The transaction begins as if a transaction code had been entered from the terminal. For conversational transactions, z/VSE assumes that END-OF-TASK means “end of the application”. The user is automatically returned to the selection panel.

It is recommended that you use the CICS START technique (CODE=1) for not directly connected applications.

You can only use one of these four codes. Some z/VSE application profiles use different codes besides the four listed above. If you select a z/VSE profile as a model and it is defined with another code, the dialog sets the CODE field to an underscore (_) when it displays the panel. Enter a code (**1 - 4**) for your own applications.

Refer also to “Example of Application Coding for the Interactive Interface” on page 136.

ACTIVATE
Specify the name to activate the application.

If you enter 2 for CODE, this is a **1 - 8** character program name. If you enter 1, 3, or 4 for CODE, this is a **1 - 4** character transaction ID.

CASE How terminal input is passed to the application:
- **1** = Uppercase (CICS performs Upper Case Translation – UCTRAN).
- **2** = Upper- and lowercase (CICS does not perform UCTRAN).

DATA Up to 136 characters of data which is passed to the transaction or program.

If CODE=1, data is passed as interval control data. For the other three codes, data is passed in the TIOA. Note that “Example of Application Coding for the Interactive Interface” on page 136 has coding examples for retrieving data.

SHOW
Used only if you specify input data (DATA). Specify whether the data which is passed should be displayed on the user's terminal:
- **1** = data displayed on user's terminal.
- **2** = data not displayed on user's terminal.

After you type in your information, press **ENTER**. The dialog formats the information and redisplays the panel. Check your entries and make any changes.

When you are done, press **PF5** to update the z/VSE control file and store the information. The dialog continues and redisplays the FULIST.

Delete an Application Profile

Option 5 deletes an existing application profile record from the z/VSE control file. You cannot delete applications with the prefix IES, INW, and INF.

Rebuild Default Application Profiles

z/VSE ships application profile records for each Interactive Interface dialog. If any application profiles for these dialogs are damaged, you can rebuild them. This can only be done using the default administrator user-ID SYSA. When user-ID SYSA accesses the dialog, the FULIST displays PF6=SYSTEM. PF6 is only displayed for user-ID SYSA. It rebuilds the application profile records which z/VSE provides for the Interactive Interface.

Migrating Your Application Profile Definitions to a Second z/VSE System

The easiest method of migrating your application profile definitions is to use the IESBLDUP utility. For details, refer to the chapter “Migrating From Earlier Releases” in the *z/VSE Planning*, SC34-2635.

Alternatively, you can use the migration facility described below which enables you to:

- Generate statements containing your application profile definitions for a z/VSE system.
- Run REXX/VSE procedure UPCNTLAP, which uses these statements to add your application profile records to the VSE Control File (IESCNTL) of a second z/VSE system.

To generate your application profile definitions, you can use tool IESXSPR to generate a job that calls REXX procedure UPCNTLAP. To do so, you must:

1. Define a transaction (for example, XSPR) to reference program IESXSPR.
2. Run transaction XSPR with parameter IESXSAPU (provided in ICCF library 59) to generate the REXX/VSE job to be run on the second z/VSE system.
3. The generated job is written to the punch queue. The name of the generated job in the punch queue will be the same as the job name of your CICS startup job (for example, CICSICCF).

Here is an example of the generated output:

```
* $$ JOB JNM=UPCNTLAP,CLASS=0,DISP=D
* $$ PUN DISP=I,CLASS=0,PRI=9
// JOB UPCNTLAP
* VSE CONTROL FILE DATA AS OF 03/31/06  08:18:23
// EXEC REXX=UPCNTLAP
IESA$FST 64 E 6       B2     DTRDDMGR      17
$$$$EASY ADM$FST
IESA$HDW 64 E 6       B2     DTRDDMGR      28
EZSIZE=400.$$$$EASY ADM$HDW
IESA$LB  64 E 6       B2     DTRDDMGR      17
$$$$EASY ADM$LB
:
IESU$RSF 64 E 6       B2     DTRDDMGR      43
&EZYZ.IAPCLS=B&EZYZ.$$$$EASY UTL$RST RSTFIL
IESU$RSV 64 E 6       B2     DTRDDMGR      43
&EZYZ.IAPCLS=B&EZYZ.$$$$EASY UTL$RST RSTVOL
/*
* TOTAL OF 186     APPLICATION PROFILES
/&
* $$ EOJ
```

Additional Considerations When Maintaining Application Profiles

1. Do not use the prefixes IES and INW for the names of your application profiles. These prefixes are reserved for z/VSE.
2. If you add your own CICS applications, they can be invoked either from a selection panel or directly after signing on as specified in the user's profile. You can maintain selection panels and user profiles using the following dialogs:
   - *Maintain User Profiles*
   - *Maintain Selection Panels*

3. After integrating an application, you must define the relevant programs to CICS.

Function Selection Within an Application

Many applications present their own menus (selection panels). Some require a keyword along with the initial transaction code to access a subfunction. You can simplify how end users work with these applications by creating a selection panel that replaces the application menu. The panel's selections can be the different subfunctions of the application. In this way, you have:

- Uniform types of selection panels throughout your system.
- The ability to create your own HELPs for an application.
- A way for users to access new subfunctions. Additional selections on the panel can be used for new subfunctions.

For applications in which a single transaction code is entered with a key word for the desired subfunction, you can point to different application profile records in a selection panel record. Each application profile names the same transaction code, but passes the former key word as data. The correct application subfunction is thus presented to the user.

Example of Application Coding for the Interactive Interface

The **entry** and **exit** of an application in the Interactive Interface must be prepared differently, depending on the CODE value in the application profile record. Refer to Figure 38 on page 142 for an example of the *Maintain Application Profiles* dialog, where the CODE value is defined.

Entry and exit handling is as follows:

Code 1 or 3

- Entry – Do a normal RECEIVE of TIOA data.
- Exit – Do an XCTL to program IESFPEP.

Codes 2 or 4

- Entry – Do a normal RECEIVE of TIOA data.
- Exit – Do a normal RETURN to CICS.

Figure 29 on page 137 is an example of command level coding for Code 1.

```
When you enter the application, check how it was started:
       .
       EXEC CICS HANDLE CONDITION,    Set up in case it was not
            ENDDATA(NOTVSE),          started by the Interactive
            NOTFND(NOTVSE)            Interface.

       EXEC CICS RETRIEVE,            Retrieve data passed by the
            SET(SOMEREG),             Interactive Interface.  Give
            LENGTH(HALFWORD)          register for data address and
                                      how long it is.
*
* If START with data is possible by other means, check for character
* string that could only have come from the Interactive Interface.
*
       CLC     =C'MY CHAR STRING',0(SOMEREG)
       BNE     NOTVSE
*
* Next instruction assumes program started by the Interactive Interface.
*
NOTVSE DS      0H     Come here if not started by Interactive Interface.
        .
        .
END    DS      0H     Return to Interactive Interface only if we
        .             came from there.
        .
       EXEC CICS XCTL PROGRAM('IESFPEP')
```

Figure 29. CICS Command Level Coding Example for Code 1 (Start)

Creating a User-Defined Selection Panel

This topic describes the steps necessary for creating a user-defined selection panel. The example used shows how to create a panel with the following selections:

1 User Application A

2 Personal Computer Move Utilities

3 Program Development

4 File Management

5 Retrieve Message

6 Display Active Users/Send Message

7 Maintain Synonyms

Selection 1 gives access to a user-provided application. Selections 2 through 7 are standard z/VSE dialogs.

Note: The *VSE/ESA Programming and Workstation Guide* describes selections 2 through 7 in detail.

The following steps are required for creating and using the selection panel outlined above:

1. Create a user profile with the *Maintain User Profiles* dialog.
2. Create the selection panel with the *Maintain Selection Panels* dialog.
3. Create an application profile for *User Application A* with the *Maintain Application Profiles* dialog.

For the example, the following main parameters are used:

user-ID:
ENDU

User Type:
2 (Programmer)

Password:
PNV48

Initial Name:
ENDSEL (Name of selection panel)

Catalog Name:
IJSYSCT

Primary Library:
16

Name: USAPPL (Name of user application)

Activate:
APIS (Program or transaction to be activated)

The 4-character user-ID (ENDU) allows this programmer to access and use VSE/ICCF.

Creating the User Profile

The *Maintain User Profiles* dialog is used to enter the information shown below. There are four panels for entering user profile information and one or more panels to define the user's VSE/ICCF primary library and VSE/ICCF parameters.

The values you can enter are described in detail under "Adding/Changing a User-ID and Profile Definitions" on page 290.

First Panel

```
IESADMUPBA                ADD OR CHANGE USER PROFILE
Base     II       CICS     ResClass ICCF

To CHANGE, alter any of the entries except the userid.

  USERID............ ENDU      4 - 8 characters (4 characters for ICCF users)

  INITIAL PASSWORD... ________  3 - 8 characters

  DAYS...............  80      0-365 Number of days before password expires
  REVOKE DATE........ 12/01/11  Date when Userid will be revoked (mm/dd/yy)

  USER TYPE.......... 1         1=Administrator, 2=Programmer, 3=General
  INITIAL NAME....... ENDSEL    Initial function performed at signon
  NAME TYPE.......... 2         1=Application, 2=Selection Panel
  SYNONYM MODEL...... ________  Userid to be used as model for synonyms
  PROGRAMMER NAME....                      Supplementary user name



PF1=HELP                  3=END                    5=UPDATE
               8=FORWARD
```

Figure 30. First Panel of Defining a User Profile

After entering the information you must press **PF8** (FORWARD) for the next panel.

Second Panel

```
IESADMUPII                    USER AUTHORIZATION
Base     II       CICS     ResClass ICCF
Answer yes or no to the following questions for userid ENDU
Enter 1 for yes, 2 for no
  NEWS...................... 1   Should user receive news items?
  ESCAPE.................... 1   Can user escape to CICS?
  CONFIRM DELETE............ 2   Does user want a confirmation message?
  VSE PRIMARY SUBLIBRARY..... 1   Does user want a PRIMARY sublibrary?
  SUBMIT TO BATCH........... 1   Can user submit to Batch?
  VSAM FILES................ 1   Can user define VSAM files?
  VSAM CATALOGS............. 1   Can user manage VSAM catalogs?
  OLPD...................... 1   Can user delete OLPD incidents?
  CONSOLE COMMANDS.......... 1   Can user enter all commands?
  CONSOLE OUTPUT............ 1   Can user see all messages?
  BATCH QUEUES.............. 1   Can user manage all POWER jobs?
  APPLICATION PROFILES...... 1   Can user maintain application profiles?
  SELECTION PANELS.......... 1   Can user maintain selection panels?
  USER PROFILES............. 1   Can user maintain user profiles?
  DEFAULT USER VSAM CATALOG.. IJSYSCT

PF1=HELP                    3=END                    5=UPDATE
PF7=BACKWARD   8=FORWARD
```

Figure 31. Second Panel of Defining a User Profile for a Type 2 User

After entering the information you must press **PF8** (FORWARD) for the next panel.

Third Panel

Select one or more operator class values from 1 to 24, which identify this user to the CICS Transaction Server system. Class 1 is set per default if you do not specify other operator classes. For a detailed description of these characteristics consult the CICS Transaction Server documentation.

```
IESADMUPCI                ADD OR CHANGE CICS SEGMENT
Base     II       CICS     ResClass ICCF

   OPERATOR ID............ SYA   Enter 3 character id for user ENDU
   OPERATOR PRIORITY....... 000   Operator priority between 0-255
   XRF SIGNOFF............ 2     Sign off after XRF takeover (1=yes,2=no)
   TIMEOUT................ 00    Minutes until sign off between 0-60

   PRIMARY LANGUAGE........       National language for CICS messages


     Place an 'X' next to the operator classes for this user

   01 X    02 _     03 _     04 _     05 _     06 _     07 _     08 _
   09 _    10 _     11 _     12 _     13 _     14 _     15 _     16 _
   17 _    18 _     19 _     20 _     21 _     22 _     23 _     24 _




PF1=HELP                    3=END                    5=UPDATE
PF7=BACKWARD   8=FORWARD
```

Figure 32. Third Panel of Defining a User Profile for a Type 2 User

After entering the information you must press **PF8** (FORWARD) for the next panel.

Fourth Panel

Note that the transaction security keys and the access rights chosen are examples. When planning security for your installation consult the CICS Transaction Server documentation.

```
IESADMUPR1          ADD OR CHANGE RESOURCE ACCESS RIGHTS
Base     II       CICS     ResClass ICCF

     Place an 'X' next to the transaction security keys for user ENDU
   01 X   02 X   03 X   04 X   05 X   06 X   07 X   08 X   09 X   10 X   11 X
   12 X   13 X   14 X   15 X   16 X   17 X   18 X   19 X   20 X   21 X   22 X
   23 X   24 X   25 X   26 X   27 X   28 X   29 X   30 X   31 X   32 X   33 X
   34 X   35 X   36 X   37 X   38 X   39 X   40 X   41 X   42 X   43 X   44 X
   45 X   46 X   47 X   48 X   49 X   50 X   51 X   52 X   53 X   54 X   55 X
   56 X   57 X   58 X   59 X   60 X   61 X   62 X   63 X   64 X

     Specify the access rights for 1-32 DTSECTAB access control classes
     ( _=No access, 1=Connect, 2=Read, 3=Update, 4=Alter )
   01 _   02 _   03 _   04 _   05 _   06 _   07 _   08 _   09 _   10 _   11 _
   12 _   13 _   14 _   15 _   16 _   17 _   18 _   19 _   20 _   21 _   22 _
   23 _   24 _   25 _   26 _   27 _   28 _   29 _   30 _   31 _   32 _

   READ DIRECTORY..... 1   User can read directory with Connect (1=yes, 2=no)
   B-TRANSIENTS....... 1   User can manipulate B-Transients     (1=yes, 2=no)

PF1=HELP                    3=END                    5=UPDATE
PF7=BACKWARD   8=FORWARD
```

Figure 33. Fourth Panel of Defining a User Profile

After entering the information you must press **PF5** (UPDATE) for processing the data you entered in the last four panels.

Panel for Specifying VSE/ICCF Primary Library

```
                                               User Id = ENDU


LIBRARY.................... 16     The primary library for this user.



DEFAULTS.................. 1      Do you accept the remaining defaults?
                                   Enter 2 = no, 1 = yes.
                                   (Do not change defaults, without care-
                                    ful consideration)
```

Figure 34. Panel for Defining Primary Library in User Profile

After entering the information you must press **ENTER** for processing.

Creating the Selection Panel

The *Maintain Selection Panels* dialog is used to enter the information shown below. In the first panel, IESEADM is chosen as a model for the selection panel. In the second panel, the required selections are defined.

Note: For the *name* to be specified for an application refer also to the manual *z/VSE Planning*. In this manual, the appendixes "Dialogs of the Interactive Interface" and "Additional z/VSE Applications" show the names of the application profiles provided and which can be specified for selection panels.

```
OPT    PANEL NAME   HELP   SELECTION   SELECTABLE PANELS OR APPLICATIONS

 1      IESEADM      *        1-4    IESEINST IESEDEF  IESEOPS  IESEPROB
                              5-8    IESEGDEV IESNICCF IESECICA
                                9
 _      IESEASAV     *        1-4    IESEVSAM IESELIBR IESS$BAC IESS$RHS
                              5-8    IESEBKDT IESERSTD IESECPDD
                                9
          .
          .
          .
```

Figure 35. First Panel of Defining a Selection Panel

After entering the information you must press **ENTER**.

```
SELECTION PANEL NAME.... ENDSEL

SEQ  NAME    TYPE             SELECTION TEXT

 1  USAPPL     1  User Application A
 2  IESEIWS    2  Personal Computer Move Utilities
 3  IESEGDEV   2  Program Development
 4  IESVSAM    1  File Management
 5  IESIMSG    1  Retrieve Message
 6  IESUSER    1  Display Active User/Send Message
 7  IESSYN     1  Maintain Synonym
 8  IESISQL    1  Access DB2 Server for VSE
 9             1
```

Figure 36. Second Panel of Defining a Selection Panel

After entering the information you must press **PF5** (UPDATE) for processing.

Creating the Application Profile

The *Maintain Application Profiles* dialog is used to enter the information shown below. In the first panel, IESAPM is chosen as a model for the application profile. In the second panel, the application related information is entered.

```
        APPLICATION               EXECUTION  UPPER    SHOW   SYSTEM USE
 OPT       NAME       ACTIVATE      CODE     CASE     INPUT     ONLY

  _       IESA$FST     DTRDDMGR       6        1        1        1
  _       IESA$HDW     DTRDDMGR       6        1        1        1
  _       IESA$LB      DTRDDMGR       6        1        1        1
  1       IESAPM       IESA           1        1        1        0
            .
            .
            .
```

Figure 37. First Panel of Defining an Application Profile

After entering the information you must press **ENTER**.

```
NAME......... USAPPL   Unique application name, 1-8 characters.

CODE......... 1        1=START trans ID, 2=LINK to program, 3=ATTACH NON-
                       CONVERSATIONAL trans ID with data, 4=ATTACH
                       CONVERSATIONAL trans ID with data.

ACTIVATE..... APIS     Name to activate, a 1-8 character program name or
                       a 1-4 character transaction ID.

CASE......... 1        Terminal input passed to application in uppercase
                       only(CASE=1) or upper/lowercase(CASE=2).

DATA......... USERDATA

                                                                     <==

                       Optional input data to pass to application.

SHOW......... 2        Show input data(SHOW=1) or do not show it(SHOW=2).
```

Figure 38. Second Panel of Defining an Application Profile

After entering the information you must press **PF5** (UPDATE) for processing.

Accessing the Newly Created Selection Panel

After completing the above steps, user ENDU can log on with password PNV48. The *z/VSE Function Selection Panel* displayed shows the selections defined for user ENDU. User ENDU can work with the selections displayed except for selection 1 (*User Application A*). To access *User Application A*, the application must first be installed (define programs, maps, transactions, and so on to CICS). For an overview on how to install a user-written application, refer to the manual *z/VSE Planning*.

Maintaining Synonyms

With this function you can define private synonyms for accessing panels. To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

1 (User Interface Tailoring)

4 (Maintain Synonyms)

When selecting this dialog, you get a list of synonyms active for your user-ID. You can locate a particular entry by using the LOCATE field.

Adding, Changing, or Deleting a Synonym

To **add** a synonym, press **PF6**. You get the *Add New Synonyms* panel. On this panel you can enter up to 13 new synonyms and paths. You can view the updated list of synonyms by pressing PF6 again. In addition, you can **change** a synonym (option 2) and **delete** a synonym (option 5).

A synonym must consist of 1-8 alphameric characters, including the characters $, #, and @. The first character cannot be a number.

Additional Considerations When Maintaining Synonyms

Users can have their own private synonyms or use a synonym model defined in their user profile by the system administrator. z/VSE provides synonyms for users SYSA, PROG, and OPER. They can be used as models for other user-IDs. Chapter 54, “Fast Paths and Synonyms for Dialogs,” on page 649 shows the synonyms for these users.

You assign a synonym model by specifying the user-ID of the synonym model owner in the SYNONYM MODEL parameter of a user's profile. Specifying PROG, for example, allows a user to use the synonyms defined for user-ID PROG. You define the SYNONYM MODEL parameter with the *Maintain User Profiles* dialog. Refer to Chapter 25, “Maintaining User Profiles via BSM Dialogs,” on page 289 for a description of the dialog and the SYNONYM MODEL parameter. To change synonyms, the following applies:

- Private synonyms can be changed by each individual user.
- The synonyms of a synonym model can only be changed by the owner (user-ID) of the model.
- If a user (but not the owner) wants to change the synonyms of the synonym model specified in the provided user profile, z/VSE does not allow it. However, z/VSE creates a copy of the model and allows the user to change the copied synonyms. The user can then use the private synonyms but no longer those of the synonym model. Only if all private synonyms are deleted, the synonyms of the synonym model become accessible again.

Assigning synonym models is a useful method if you have many users that access the same panels and functions. By allowing only the administrator to maintain such models, you can keep control of the synonyms used at your system.

Password Expiration

user-IDs should be defined with an expiration date for the password. If a password expires in seven days or less and a user signs-on, the system displays the following message after sign-on:

```
YOUR PASSWORD EXPIRES IN x DAYS
```

In the message, x specifies the number of days before the password expires.

It is recommended that the user changes the password during the next sign on.

In addition, the person who is responsible for maintaining user profiles, usually the system administrator, can change a password using the *Maintain User Profiles* dialog. The dialog is described under Chapter 25, “Maintaining User Profiles via BSM Dialogs,” on page 289.

How the Password History Is Stored

By default, z/VSE retains the last **twelve** passwords for each user and prevents these passwords from being reused. If a user then reenters a password that was last used “thirteen passwords previously”, z/VSE will permit this previously-used password.

If the user changes his/her password during the sign-on procedure, z/VSE will check the password history. z/VSE does *not* check the password history:

- If the user changes his/her password using the:

  – Batch facility IESUPDCF.
  – *User Profile Maintenance* dialog.
- If password-history checking has been turned off using the `BSTADMIN PERFORM PASSWORD` command.

Please note that you cannot change the rules for the number of passwords that are stored in the password history.

Resetting a Revoked User-ID

The z/VSE Basic Security Manager (BSM) revokes a user-ID if the number of invalid sign-on attempts exceeds a specified limit. This limit for sign-on attempts is specified in the skeleton IESELOGO. Refer to "Setting a Limit for Invalid Sign-On Attempts" on page 188 for detailed information.

Once a user-ID is revoked, only a user with system administrator authority can reset it. To reset a revoked user-ID, use the dialog *Maintain User Profiles*. See Figure 78 on page 291 for details. In the field REVOKE DATE specify the appropriate date, or set it to zero if the user should never be revoked.

To reset a revoked user-ID, you can also use the batch utility program IESUPDCF. You achieve this with the ALTER command and Revoke 0 for the user to be reset. An example now given:

```
* $$ JOB JNM=UPDATECF,CLASS=0,DISP=D
* $$ PUN DISP=I
// JOB UPDATECF
// PAUSE DISCONNECT DTSFILE, IESCNTL
// EXEC PROC=DTRICCF
// DLBL IESCNTL,'VSE.CONTROL.FILE',,VSAM,CAT=VSESPUC
// EXEC IESUPDCF,SIZE=64K
ICCF=YES
             ALT SYSA,PWD=ABCDEF,REVOKE=0
/*
/&
* $$ EOJ
```

Chapter 11. Installing a Second Predefined CICS Transaction Server

z/VSE provides support for installing a second, predefined CICS Transaction Server. The following text uses also the short form CICS for CICS Transaction Server.

Note: From z/VSE 5.1 onwards, CICS/VSE (Program Number 5686-026) is no longer shipped on the Extended Base Tape. Therefore, all references to CICS/VSE have been removed from this chapter.

The following skeletons are provided:

SKCICS2
(for defining startup)

SKPREPC2
(for defining resources)

The second CICS Transaction Server under z/VSE does not include VSE/ICCF and can have any of three relationships to the primary CICS:

1. No communication with the primary CICS.
2. Communication to the primary CICS via *Multiregion Operation* (MRO). The two CICS Transaction Servers run in the same server.
3. The two CICS Transaction Servers can also communicate via *Intersystem Communication* (ISC), running on the same or different servers. With ISC any type of CICS subsystem can communicate with each other.

z/VSE's support addresses only the first two cases, and so does this chapter. For information on ISC, consult the IBM manual *CICS Intercommunication Guide*.

Before you begin with the installation process, for planning information you should consult the manual *z/VSE Planning*, topic “Planning for the Second CICS Transaction Server”.

The *z/VSE Planning* manual also provides detailed information about the “CICS Transaction Server Monitoring and Statistics Support”.

Installation Tasks for a Second CICS Transaction Server

The tasks to be performed are described below. Follow the given sequence.

Task 1: Modify the CICS Predefined Environment

The following is assumed:

- Your second CICS is to run in partition F8.

  This means that you can use skeleton SKALLOCB or SKALLOCC to change the partition values as shown below.

  – For environment A, the allocation reserves the following space in F8 for the second CICS:

    ```
    ALLOC F8=50M
    SIZE F8=2M
    ```

– For environment B, the allocation reserves the following space in F8 for the second CICS:

```
ALLOC F8=150M
SIZE F8=2M
```

– For environment C, the allocation reserves the following space in F8 for the second CICS:

```
ALLOC F8=512M
SIZE F8=2M
```

If you decide not to use F8, you must select a partition of the appropriate size and modify the startup job accordingly.

Note that these are recommended average values. Depending on your applications they may not be sufficient.

For the second, predefined CICS the name of the startup job is CICS2, the name of the corresponding skeleton is SKCICS2.

Before you edit the skeletons, copy them first from VSE/ICCF library 59 to your primary library.

Increasing partition sizes might also mean that to meet your total system requirements you have to increase the VSIZE.

The predefined page data set size (VSIZE) for:
- Predefined environment A is 256 MB.
- Predefined environment B is 512 MB.
- Predefined environment C is 2 GB.

However, you can change the VIO and VSIZE values using the *Tailor IPL Procedure* dialog. Refer to "Tailoring the IPL Procedure" on page 14 for details.

In addition to running a CICS Transaction Server in a static partition, you may also run it in a **dynamic partition** of sufficient size:
- Predefined environment A allows for 50 MB partitions.
- Predefined environment B allows for 150 MB partitions.
- Predefined environment C allows for 512 MB partitions.

However, you should adjust the value of EDSALIM in the DFHSIT table or the startup override accordingly:
- Predefined environment A has a predefined EDSALIM size of 25 MB.
- Predefined environment B has a predefined EDSALIM size of 120 MB.
- Predefined environment C has a predefined EDSALIM size of 450 MB.

Note also that the partition parameters and values for dynamic partitions are defined in the active dynamic class table. You can modify this table with the *Maintain Dynamic Partitions* dialog. For details refer to "Defining Dynamic Class Tables" on page 72.

Running your second CICS in another static partition than F8 (or in a dynamic partition) requires changes in skeletons SKCICS2 and SKPREPC2. Affected are mainly statements which include partition-related information. Further details are provided with the skeleton descriptions.

Do not run the CICS Transaction Server in partition F4 because of possible storage key problems if storage protection is set in DFHSIT.

Note: Before you submit skeletons SKCICS2 and SKPREPC2 later, ensure that you did run at least once skeleton SKCOLD. It updates procedure COLDJOBS which loads jobs into the VSE/POWER reader queue that are important for a COLD startup.

Task 2: Modify the Skeletons Provided by z/VSE

Copy a skeleton first from VSE/ICCF library 59 to your primary library and modify the skeleton here. This ensures that you have a backup version of the original skeleton available.

Skeleton SKUSERBG

1. Locate the statement
```
* // PWR PRELEASE RDR,CICS2
```
2. Delete the asterisk and the blank in the first two columns. Refer also to “Skeleton SKUSERBG (Startup Procedure for BG Partition)” on page 46.

   When the modified procedure is processed during system startup, the statement causes the second CICS subsystem to be started.

Skeletons SKCICS2 and SKPREPC2

Refer to “Skeleton SKCICS2” on page 152 and to “Skeleton SKPREPC2” on page 154 where the skeletons are shown in detail. Comments point out what might or should be modified and why a modification should be considered.

Modify the skeletons **but do not submit** them now.

Task 3: Modify CICS Control Tables

The control tables that may need to be modified for the second CICS subsystem are the following (their source can be obtained from VSE/ICCF library 59):

System Initialization Table (DFHSITC2)

Destination Control Table (DFHDCTC2)

File Control Table (DFHFCTC2)

The FCT will be migrated to the CSD file and further modifications should be done there.

Differences of significance are indicated in subsequent paragraphs that discuss the various tables. These paragraphs point out possible modifications. Some changes have to be made to the tables to include or exclude CICS-to-CICS communication support via MRO. You may also add entries to meet local requirements.

If your modified skeleton SKPREPC2 includes DLBL and EXTENT statements for CICS **Journal Files**, you must provide the specifications for the corresponding Journal Control Tables (DFHJCTs). You can use skeleton DFHJCTSP, stored in VSE/ICCF library 59, for that purpose. The skeleton reflects the journal control tables for the primary CICS. Your modification consists of changing the suffix 'SP' into 'C2' wherever it occurs. For a description of the DFHJCT macro, refer to the CICS Transaction Server manual *CICS Resource Definition Guide*.

Modify CICS Control Tables - System Initialization Table

This table, shipped as member **DFHSITC2**, **includes** significant differences compared to DFHSITSP as follows:

– Except for PLTPI and PLTSD, the table suffix is C2. Note that FCT is NO. Affected operands are:

```
DCT=C2      PLTPI=P2
FCT=NO      PLTSD=S2
```

– The application name of the CICS subsystem:

```
APPLID=PRODCICS
```

PRODCICS is also the user-ID defined in the VSE.CONTROL.FILE and used in the ID statement as user-ID for startup.

– The table activates the spool support of CICS with the specification

```
SPOOL=(YES,B,A)
```

– The internal trace function is set to off (the second CICS Transaction Server is assumed to support applications in production):

```
TRTABSZ=256
TRTRANSZ=128
TRTRANTY=TRAN
```

– As shipped, table DFHSITC2 does not activate CICS-to-CICS communication via MRO. The related specifications are set as follows:

```
GRPLIST=VSELST2
IRCSTRT=NO
ISC=YES
SYSIDNT=CIC2
```

SKPREPC2 defines list VSELST2.

If **MRO communication is to be used**, IRCSTRT=NO has to be changed into IRCSTRT=YES (in table DFHSITC2). In table DFHSITSP for the primary CICS system, the settings for MRO must also be changed from NO to YES:

```
IRCSTRT=YES
ISC=YES
```

Modify CICS Control Tables - Destination Control Table

This table is shipped as member **DFHDCTC2**; it includes no significant differences. Any TYPE=SDSCI entries that you need in addition are to be added immediately behind the box labeled

```
LOCAL ENTRIES FOR TYPE=SDSCI SHOULD BE PLACED BELOW THIS BOX
```

Modify CICS Control Tables - File Control Table

This table is shipped as member **DFHFCTC2**; it includes no significant differences. Any entries that you need in addition are to be added immediately behind the box labeled

```
LOCAL ENTRIES SHOULD BE PLACED BELOW THIS BOX
```

The default FCT (DFHFCTC2) is migrated into the CSD (CICS System Definition) file (as group FCTC2) and FCT=NO is set. Thus modifying DFHFCTC2 does not change the system unless the FCT is migrated to the CSD after changing it. Migration is done using utility DFHCSDUP:

```
// EXEC DFHCSDUP,SIZE=600K
   DELETE GROUP(FCTC2)
   MIGRATE TABLE(DFHFCTC2)
/*
```

The initial setup using SKPREPC2 will migrate the DFHFCTC2.

You may also use RDO (CEDA command) for defining FCT entries.

Task 4: Submit the Modified Skeletons

After having modified the skeletons as described under "Skeleton SKCICS2" on page 152 and "Skeleton SKPREPC2" on page 154, submit the skeletons from the FULIST of your primary VSE/ICCF library.

Ensure to submit them in the sequence shown:

1. SKCICS2 – use option **7**.

   Before you submit the next skeleton, close the Message Routing file. Use the CICS command:

   ```
   CEMT SET FILE(IESROUT) CLOSE
   ```

   If you define the Workstation File Transfer Support for your second CICS subsystem, close also the Host Transfer file. Use the CICS command:

   ```
   CEMT SET FILE(INWFILE) CLOSE
   ```
2. SKPREPC2 – use option **7**.
3. The CICS control tables that you modified or coded – use option **7**.
4. When processing of these skeletons is complete, reopen the file(s) that you have closed. Use the CICS command(s):

   ```
   CEMT SET FILE(IESROUT) OPEN ENA
   CEMT SET FILE(INWFILE) OPEN ENA
   ```

There is no need for you to define any terminals to the second CICS. Also, the name of the second CICS (PRODCICS) is already defined to VTAM.

If **MRO communication is to be used**, however, you should assign unique CICS terminal IDs to the terminals of the second CICS. Refer also to "Tailoring Autoinstall Terminals" on page 151.

Task 5: Definitions for MRO

For this task use the RDO (Resource Definition Online) function described in the CICS Transaction Server manual *CICS Resource Definition Guide*. The function is a convenient means for setting up a communication path to the primary CICS subsystem and for defining terminals.

1. **Define CICS-to-CICS MRO communication**

   This requires the definition of a connection and an associated sessions definition for each of the two CICS subsystems.

   For a **connection** definition, enter the RDO command

   ```
   CEDA DEFINE CONNECTION
   ```

   and provide specifications as listed below. Accept the defaults for the data-entry fields not listed here.

Panel Line	Specifications DBDCCICS Side	Specifications PRODCICS Side	Comment
Connection:	CIC2	CIC1	SYSID in DFHSITC2
Group:	VSEIRC1	VSEIRC2	
Netname:	PRODCICS	DBDCCICS	
ACcessmethod:	IRc	IRc	
Protocol:			Must be blank
AUToconnect:	Yes	Yes	

For the associated **sessions** definition, enter the RDO command

```
CEDA DEFINE SESSIONS
```

Provide the specifications listed below to have 10 send and receive sessions. Accept the defaults for the data-entry lines not listed here.

Panel Line	Specifications DBDCCICS Side	Specifications PRODCICS Side	Comment
Sessions:	CICSS2	CICSS1	arbitrary
Group:	VSEIRC1	VSEIRC2	
Connection:	CIC2	CIC1	SYSID in DFHSITC2
Protocol:	LU61	LU61	
RECEIVEPfx:	TR	PR	
RECEIVECount:	010	010	
SENDPfx:	TS	PS	
SENDCount:	010	010	
SENDSize:	4096	4096	See Note a below
RECEIVESize:	4096	4096	See Note a below
OPERRsl:	0	0	See Note b below
OPERSecurity:	1	1	See Note b below
AUToconnect:	Yes	Yes	
INservice:	Yes	Yes	
RELreq:	Yes	Yes	
Discreq:	Yes	Yes	

Note:

a. A general recommendation: certain CICS applications may require specific values to be specified. Check the applicable manuals.

b. This is the default: gives the terminal operator access to unprotected resources only.

2. **Define terminals**

Use the autoinstall function of CICS to define the terminals that are to be supported by your second CICS.

Enter RDO commands as follows:

```
CEDA ADD GROUP(VSEIRC2) LIST(VSELST2)
```

The statements cause all of list VSELIST to be copied into the new CSD list, and the new group VSEIRC2 (which you had defined via CEDA DEFINE SESSIONS), to be added to the list.

```
CEDA ADD GROUP(VSEIRC1) LIST(VSELIST)
CEDA INSTALL GROUP(VSEIRC1)
```

The above statements add and install the new CSD group, `VSEIRC1`, which you had defined via CEDA DEFINE SESSIONS for your primary CICS.

After successful completion of the above procedure, the required definitions for your second CICS are complete:

- At the next startup of your z/VSE system, your second CICS will be available.
- If you do *not* want to restart the second CICS Transaction Server, you should run a **CEDA INSTALL GROUP(VSEIRC2)** on a terminal that is signed-on to the second CICS Transaction Server.

Tailoring Autoinstall Terminals

For using unique CICS terminal IDs, you have to perform the following steps:

- In the *Hardware Configuration* dialog select option 3 for logical unit (further processing) for autoinstall terminals.
- If the entry for CICS TERM ID is displayed, use option 6 and delete the entry for TERM ID (CICS).

The TERM IDs (such as A001, A002 and so on) are used for the first and a second CICS.

To prevent the use of duplicated CICS TERM IDs for the second CICS, do the following:

- Access and copy the autoinstall exit member IESZATDX from VSE/ICCF library 59 to your own VSE/ICCF library.
- Locate the field `PREFIX   DC    C'ABCDEFGHIJ*'`.
- To ensure that different terminal prefixes are used for the CICS TERM ID, change the prefix vector to a vector containing different characters (for example `C'KLMNOPQRST*'`).
- Submit the changed member IESZATDX.
- Ensure that the phase is being cataloged into a sublibrary unique to the second CICS Transaction Server.

Using Traces for Problem Solving

As shipped, the startup job stream (skeleton SKCICS2) defines and allocates an AUXTRACE file (trace file A). However, tracing the flow of transactions through the system is **not** activated automatically. For how to use AUXTRACE, refer to the CICS Transaction Server manual *CICS Problem Determination Guide*.

Skeleton DFHAUXPR, which is stored in VSE/ICCF library 59, provides a job stream for analyzing AUXTRACE data. You must adapt the label information in the job stream for the second CICS (PRODCICS).

If you require a *second* auxiliary trace file, you can use the skeleton DFHAUXB (which is also stored in VSE/ICCF library 59) to define a second trace file.

Skeletons for Second CICS Transaction Server

This appendix lists and describes skeletons:

SKCICS2

SKPREPC2

These skeletons are shipped as members of VSE/ICCF library 59. The skeletons as listed in this topic include comments that you may find helpful when modifying them. Also note that names of resources (such as volumes) that you might want or have to change are highlighted in bold.

Skeleton SKCICS2

This is the startup procedure for the second CICS Transaction Server. You can submit this skeleton unchanged if your second CICS Transaction Server is to run in partition F8. Else, change the highlighted specifications accordingly and ensure that XAPPLF8 in CPUVAR1 is modified accordingly.

The first loading of CICS2 is done via skeleton SKPREPC2. If there is a need to load CICS2 again, you have to remove the asterisk (*) in front of the command `EXEC PROC=LDCICS2` to activate the loading into the VSE/POWER reader queue.

After you have modified skeleton SKCICS2, enter the following command from the editor's command line **before** you file the skeleton **@DTRSEXIT**. This command calls a macro that deletes specific comments from the skeleton.

```
* $$ JOB JNM=CATCICS2,DISP=D,CLASS=0
// JOB CATCICS2                 CATALOG CICS2 AND LDCICS2
// EXEC LIBR,PARM='MSHP'
   ACCESS S=IJSYSRS.SYSLIB
   CATALOG CICS2.Z REPLACE=YES
$$$$ JOB JNM=CICS2,DISP=L,CLASS=8,EOJMSG=YES
$$$$ LST CLASS=A,DISP=D,RBS=100
// JOB CICS2         STARTUP OF SECOND CICS WITHOUT ICCF
// OPTION SADUMP=5
// OPTION SYSDUMPC
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASED,PRD1.BASE,PRD2.PROD,        X
               PRD2.SCEEBASD,PRD2.SCEEBASE,PRD2.DBASE),PERM
// LIBDEF DUMP,CATALOG=SYSDUMP.F8
// SETPARM XNCPU=''
// SETPARM XMODEF8=AUTO
// SETPARM XAPPLF8=''
// EXEC PROC=$COMVAR,XNCPU
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATF8=ACTIVE' **F8 ASSUMED
$$/*
// EXEC PROC=CPUVAR&XNCPU,XMODEF8,XAPPLF8        **F8 ASSUMED
// SETPFIX LIMIT=256K
LOG
// ID USER=PRODCICS
NOLOG
// EXEC PROC=DTRCICS2                  LABELS FOR CICS FILES
*   WAITING FOR VTAM TO COME UP
// EXEC IESWAITT
$$/*
*   WAITING FOR TCP/IP TO COME UP
* // EXEC REXX=IESWAITR,PARM='TCPIP00'
$$/*
// ASSGN SYS020,SYSLST
* // ASSGN SYS023,DISK,VOL=DOSRES,SHR   IF JOURNALING
// IF XENVNR = A THEN
// SETPARM ELIM=25M
// IF XENVNR = B THEN
// SETPARM ELIM=120M
// IF XENVNR = C THEN
// SETPARM ELIM=450M
```

Figure 39. Skeleton SKCICS2 Part 1 (Starting Up Second CICS in Partition F8)

Note:
1. In a system with security (access control) active the // ID statement (for user PRODCICS) ensures that CICS2 has the appropriate access rights to the control file. When you submit the job your access rights are inherited by the CICS2 startup job in the VSE/POWER reader queue, provided that the // ID statement is for PRODCICS or is of the same model type like DBDCCICS or PRODCICS. In this case, no password is required. To get inheritence the job must be submitted when security is active.
2. Chapter 33, "Access Rights/Checking in DTSECTAB," on page 401 provides details about the z/VSE access control support.
3. If you use another name than CICS2, you must also update procedures such as USERBG and COLDJOBS (via skeletons SKUSERBG and SKCOLD).
4. ELIM is the value of EDSAMIM. For environment C, the specified value requires a partition of at least 480 MB.

```
// IF XMODEF8 = COLD THEN
// GOTO COLDST
// SETPARM XMODEF8=AUTO
// GOTO STARTCIC
/. COLDST
// SETPARM XMODEF8=COLD
/. STARTCIC
// EXEC DFHSIP,SIZE=DFHSIP,PARM='APPLID=&XAPPLF8.,START=&XMODEF8.,SI', *
               DSPACE=2M,OS390
SIT=C2,STATRCD=OFF,NEWSIT=YES,
$$$$ SLI MEM=IESVAEXC.Z,S=IJSYSRS.SYSLIB
$$/*
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATF8=INACTIVE'
$$/*
$$/&
$$$$ EOJ
/+
CATALOG  LDCICS2.PROC     REPLACE=YES DATA=YES
// EXEC DTRIINIT
   LOAD CICS2.Z
/*
/+
/*
* // EXEC PROC=LDCICS2       LOAD CICS2 INTO RDR QUEUE
/*
/&
* $$ EOJ
```

Figure 40. Skeleton SKCICS2 Part 2 (Starting Up Second CICS in Partition F8)

The skeleton includes additional $$ characters. They are needed to mask off VSE/POWER JECL statements. Program DTRIINIT, described in the manual *z/VSE System Utilities* under "DTRIINIT Utility", replaces the $$ characters with VSE/POWER JECL statements for cataloging.

Skeleton SKPREPC2

Note: Before you submit SKPREPC2 for processing, you may have to change skeleton SKUSERBG in VSE/ICCF library 59. See also the "Comment" in Part 6 of this skeleton.

The skeleton defines the resources for a second CICS Transaction Server and catalogs the required label information.

For space requirements, consult the manual *z/VSE Planning*.

```
* $$ JOB JNM=VSAMDEF2,DISP=D,CLASS=0
// JOB VSAMDEF2 - DEFINE VSAM CLUSTERS FOR SECOND CICS
* ********************************************
*  DEFINE AND INITIALIZE VSAM FILES FOR CICS2
```

```
* ******************************************
// EXEC IDCAMS,SIZE=AUTO
 /*                                                                    */
 /*  DELETE VSAM FILES                                                 */
 /*                                                                    */
```

Note:

1. The delete job below ensures that no catalog entries with identical file IDs exist.
2. You might have files with an ID identical to the ones specified in the job and also under control of the specified user catalog. If you need those files further on (to operate with three CICS Transaction Servers, for example), rename the IDs in the skeleton. Suggested approach: change `CICS2` to `CICSB`.
3. If you do not use the default user catalog, do a global change for the catalog name, **VSESPUC**, and the catalog ID, **VSESP.USER.CATALOG**. The occurrences of these specifications in the skeleton are highlighted.

```
   DELETE (CICS2.GCD) CL NOERASE PURGE -
   CATALOG(VSESP.USER.CATALOG)
   DELETE (CICS2.LCD) CL NOERASE PURGE -
   CATALOG(VSESP.USER.CATALOG)
   DELETE (CICS2.ONLINE.PROB.DET.FILE) CL NOERASE PURGE  -
   CATALOG(VSESP.USER.CATALOG)
   DELETE (CICS2.DFHTEMP) CL NOERASE PURGE  -
   CATALOG(VSESP.USER.CATALOG)
   DELETE (CICS2.TD.INTRA) CL NOERASE PURGE  -
   CATALOG(VSESP.USER.CATALOG)
   DELETE (CICS2.RSD) CL NOERASE PURGE  -
   CATALOG(VSESP.USER.CATALOG)
   SET MAXCC = 0
 /*                                                                */
 /*  DEFINE VSAM FILES                                             */
 /*                                                                */
```

Note: The required VSE/VSAM files are defined to reside on the SYSWK1 volume with DOSRES specified as the secondary volume for allocations. If you plan to have the files allocated in VSE/VSAM space on different volumes, change the volume names accordingly.

```
DEFINE CLUSTER(NAME(CICS2.GCD)            -
    RECORDSIZE (4089 4089)                -
    RECORDS (2000 200)                    -
    KEYS   (28 0)                         -
    REUSE                                 -
    INDEXED                               -
    FREESPACE (10 10)                     -
    SHR(2)                                -
    CISZ(8192)                            -
    VOL(SYSWK1 DOSRES))                   -
    DATA(NAME(CICS2.GCD.@D@))             -
    INDEX (NAME (CICS2.GCD.@I@))          -
    CATALOG(VSESP.USER.CATALOG)
    DEFINE CLUSTER(NAME(CICS2.LCD)        -
    INDEXED                               -
    RECORDSIZE (45 124)                   -
    RECORDS (3000 200)                    -
    KEYS   (28 0)                         -
    REUSE                                 -
    FREESPACE (10 10)                     -
    SHR(2)                                -
    CISZ(2048)                            -
    VOL(SYSWK1 DOSRES))                   -
    DATA(NAME(CICS2.LCD.@D@))             -
    INDEX (NAME (CICS2.LCD.@I@))          -
    CATALOG(VSESP.USER.CATALOG)
    DEF CLUSTER(NAME(CICS2.ONLINE.PROB.DET.FILE) -
     FILE(IESPRB)                          -
     VOL(SYSWK1 DOSRES)                    -
```

```
     RECORDS (300 100)                -
     RECORDSIZE (4000 4089)           -
     INDEXED                          -
     KEYS(2 0)                        -
     SHR(2))                          -
     DATA (NAME (CICS2.ONLINE.PROB.DET.FILE.@D@) CISZ(4096)) -
     INDEX (NAME (CICS2.ONLINE.PROB.DET.FILE.@I@) CISZ(512)) -
     CATALOG(VSESP.USER.CATALOG)
 /*                                                          */
 DEF CLUSTER(NAME(CICS2.DFHTEMP)      -
     VOL(SYSWK1 DOSRES)               -
     RECORDS (100)                    -
     RECORDSIZE (16377 16377)         -
     CISZ   (16384)                   -
     NONINDEXED                       -
     SHR(2))                          -
     DATA(NAME(CICS2.DFHTEMP.ESDS))   -
     CATALOG(VSESP.USER.CATALOG)
 /*                                                          */
 DEF CLUSTER(NAME(CICS2.TD.INTRA)     -
     VOL(SYSWK1 DOSRES)               -
     RECORDS (100)                    -
     RECORDSIZE (4089 4089)           -
     CISZ   (4096)                    -
     NONINDEXED                       -
     SHR(2))                          -
     DATA(NAME(CICS2.TD.INTRA.ESDS))  -
     CATALOG(VSESP.USER.CATALOG)
 /*                                                          */
 DEF CLUSTER(NAME(CICS2.RSD)          -
     INDEXED                          -
     RECORDSIZE (2000 2000)           -
     RECORDS (250 100)                -
     KEYS   (22 0)                    -
     FREESPACE (20 20)                -
     SHR(2)                           -
     VOL(SYSWK1 DOSRES))              -
     DATA(NAME(CICS2.RSD.@D@))        -
     INDEX (NAME (CICS2.RSD.@I@))     -
     CATALOG(VSESP.USER.CATALOG)
 /*                                                           */
```

Note: The following files are shared with CICSICCF, the primary CICS Transaction Server and need not be defined:

VSE.TEXT.REPOSITORY.FILE

VSE.MESSAGE.ROUTING.FILE

CICS.CSD

VSE.CONTROL.FILE

```
/*
// IF $RC > 0 THEN
CANCEL
*
*    INITIALIZE THE CICS2 RESTART DATA SET
*
// DLBL DFHRSD,'CICS2.RSD',0,VSAM,CAT=VSESPUC
// DLBL DFHGCD,'CICS2.GCD',0,VSAM,                                     X
               CAT=VSESPUC
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE   -
        (SYSIPT                 -
         ENVIRONMENT            -
          (RECORDFORMAT (FIXUNB) -
           BLOCKSIZE(80)        -
           RECORDSIZE (80)))    -
        OUTFILE (DFHRSD)
ACTL 0001
/*
```

```
// EXEC IDCAMS,SIZE=AUTO          INIT GCD FILE
   REPRO INFILE                                     -
           (SYSIPT                                  -
           ENVIRONMENT                              -
             (RECORDFORMAT(FIXUNB)                  -
              BLOCKSIZE(80)                         -
              RECORDSIZE(80)))                      -
         OUTFILE(DFHGCD)
/*
// DLBL DFHLCD,'CICS2.LCD',0,VSAM,                                    X
                CAT=VSESPUC
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD2.SCEEBASE,PRD1.BASE)
// EXEC DFHCCUTL,SIZE=300K        INITIALIZE CICS CATALOG
/*
/&
* $$ EOJ
* $$ JOB JNM=DTRCICS2,DISP=D,CLASS=0
// JOB DTRCICS2 - DEFINE LABELS FOR SECOND CICS
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG DTRCICS2.PROC  D=YES  R=YES  EOD=/+
// ASSGN SYS018,DISK,VOL=SYSWK1,SHR
// DLBL DFHDMPA,'CICS2.DUMPA',0,VSAM,
               CAT=VSESPUC,RECSIZE=7200,
               DISP=(NEW,KEEP),RECORDS=(300,0)
// DLBL DFHDMPB,'CICS2.DUMPB',0,VSAM,
               CAT=VSESPUC,RECSIZE=7200,
               DISP=(NEW,KEEP),RECORDS=(100,0)
// DLBL DFHAUXT,'CICS2.AUXTRACE',0,VSAM,
               CAT=VSESPUC,RECSIZE=4096,
               DISP=(NEW,KEEP),RECORDS=(400,0)
// DLBL DFHTEMP,'CICS2.DFHTEMP',0,VSAM,
               CAT=VSESPUC
// DLBL DFHNTRA,'CICS2.TD.INTRA',0,VSAM,
               CAT=VSESPUC
// DLBL DFHRSD,'CICS2.RSD',0,VSAM,
               CAT=VSESPUC
 // DLBL DFHLCD,'CICS2.LCD',0,VSAM,
               CAT=VSESPUC
// DLBL DFHGCD,'CICS2.GCD',0,VSAM,
               CAT=VSESPUC
// DLBL IESPRB,'CICS2.ONLINE.PROB.DET.FILE',,VSAM,
               CAT=VSESPUC
* ******************************************************************** C
* REMOVE COMMENTS AND TRAILING 'C' IN COLUMN 71 IN CASE JOURNALLING   C
* IS USED. ADJUST LABELS AS SPECIFIED IN SKJOUR2 - DEPENDING ON YOUR  C
* DISK TYPE.                                                          C
*                                                                     C
*  DLBL DFHJ01A,'CICS2.SYSTEM.LOG.A',0,SD                             C
*  EXTENT SYS023,DOSRES,1,0,XXXX,60                                   C
*  DLBL DFHJ01B,'CICS2.SYSTEM.LOG.B',0,SD                             C
*  EXTENT SYS023,DOSRES,1,0,XXXX,60                                   C
*  DLBL DFHJ02A,'CICS2.USER.JOURNAL.A',0,SD                           C
*  EXTENT SYS023,DOSRES,1,0,XXXX,60                                   C
*  DLBL DFHJ02B,'CICS2.USER.JOURNAL.B',0,SD                           C
*  EXTENT SYS023,DOSRES,1,0,XXXX,60                                   C
* ******************************************************************** C
/+
  CONNECT S=IJSYSRS.SYSLIB:PRD2.SAVE
  COPY DTRCICS2.PROC  REPLACE=YES
/*
/&
* $$ EOJ
```

Note:

1. If you plan to define and format one or more CICS journal files, insert the required DLBL and EXTENT statements at this point. Member SKJOURN of the VSE/ICCF library 59 includes suitable sample statements for all types of the

supported IBM disk devices. For further details about defining journal files, refer to “Task 3: Modify CICS Control Tables” on page 147.

2. The SETPARM values specified in the job below assume that your second CICS will run in partition F8. If this is your intent, no partition-related specifications need be changed. However, if your second CICS Transaction Server is to run in another partition, F5 for example, change the SETPARM values as shown:

Present Specification	Change To
XPARTC2='F8'	XPARTC2='F5'
XUSEF8='CI'	XUSEF5='CI'
XAPPLF8='PRODCICS'	XAPPLF5='PRODCICS'

3. Do not use the partition F4 because of possible storage key problems if storage protection is set in DFHSIT. Also, be aware of the size requirements of the partition.
4. Modify the job if necessary. If journaling is used, the journal files are located on DOSRES. However, for environment C you should *locate the journal files to a disk other than DOSRES*. Otherwise, the page data set might be overwritten!

```
* $$ JOB JNM=GLOBVAR,DISP=D,CLASS=0
// JOB GLOBVAR - DEFINE GLOBAL VARIABLES
// SETPARM XNCPU='
// EXEC PROC=$COMVAR,XNCPU
// EXEC DTRSETP,PARM='CPUVAR&XNCPU'
   SET XPARTC2='F8'
   SET XUSEF8='CI'
   SET XAPPLF8='PRODCICS'
/*
/&
* $$ EOJ
```

Note: The following job replaces the currently used startup procedure USERBG for the BG partition. In the SLI statement, replace the library number 59 by the number of your primary library (assuming that you had applied the changes to SKUSERBG in your primary library). Jobs PRTDUC2A and PRTDUC2B are for printing the dump data sets of CICS.

```
* $$ JOB JNM=COPYUBG,DISP=D,CLASS=0
// JOB COPYUBG - COPY SKUSERBG FROM ICCF LIBRARY
// EXEC LIBR,PARM='MSHP'
   ACCESS S=IJSYSRS.SYSLIB
* $$ SLI ICCF=(SKUSERBG),LIB=(59)
/*
/&
* $$ EOJ
```

```
* $$ JOB JNM=ADDJOBS,CLASS=0,DISP=D
// JOB ADDJOBS ADD JOBS TO POWER READER QUEUE
// EXEC DTRIINIT
   LOAD CICS2.Z
   LOAD PRTDUC2A.Z
   LOAD PRTDUC2B.Z
/*
/&
* $$ EOJ
```

Note: The job below changes the share option definition from `SHAREOPTIONS(2)` to `SHAREOPTIONS(4)` for the Message Routing and the Host Transfer File. Replace the ID of the VSE/VSAM user catalog if you do not use the default user catalog of z/VSE.

```
* $$ JOB JNM=SHARE4,CLASS=0,DISP=D
// JOB SHARE4 CHANGE SHAREOPTIONS
* PLEASE CLOSE FILES IESROUT AND INWFILE ON DBDCCICS
* AND ALSO ON ALL OTHER CICS PARTITIONS USING THE FILES.
*        CEMT SET FI(XXXXXXX) CLOSE
* A RETURN CODE OF 4 IS OK. IF THE INWFILE DOES NOT EXIST, RETURN
```

```
* CODE WILL BE 12.
* IF OTHER FILES SHOULD ALSO BE SHARED AMONG SYSTEMS CHANGE
* THE SHAREOPTIONS ACCORDINGLY.
// PAUSE
// EXEC IDCAMS
   ALTER VSE.MESSAGE.ROUTING.FILE.@I@ -
   SHAREOPTIONS(4)                    -
   CATALOG(VSESP.USER.CATALOG)
   /**/
```

Note:

1. Delete the two ALTER statements for the Host Transfer File if the available Workstation File Transfer Support is to be used only from your primary CICS Transaction Server or not at all.
2. If the Workstation File Transfer Support support is to be used from both CICS Transaction Servers, the share option of the Host Transfer File must be changed. This may slightly impact file transfer speed.

```
   ALTER VSE.MESSAGE.ROUTING.FILE.@D@ -
   SHAREOPTIONS(4)                    -
   CATALOG(VSESP.USER.CATALOG)
   /**/
   ALTER PC.HOST.TRANSFER.FILE.INDEX  -
   SHAREOPTIONS(4)                    -
   CATALOG(VSESP.USER.CATALOG)
   /**/
   ALTER PC.HOST.TRANSFER.FILE.DATA   -
   SHAREOPTIONS(4)                    -
   CATALOG(VSESP.USER.CATALOG)
/*
* OPEN FILES AGAIN
*        CEMT SET FI(XXXXXXX) OPEN
// PAUSE
/&
* $$ EOJ
* PLEASE CLOSE DFHCSD IN DBDCCICS

// PAUSE
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.SCEEBASE)
// EXEC DFHCSDUP,SIZE=600K        INIT AND LOAD CICS
  DELETE GROUP(FCTC2)
  DELETE LIST(VSELST2)
  ADD GROUP(VSETYPE) LIST(VSELST2)
  ADD GROUP(VSETERM) LIST(VSELST2)
  ADD GROUP(VSETERM1) LIST(VSELST2)
  APPEND LIST(DFHLIST) TO(VSELST2)
  ADD GROUP(DFHRCF) LIST(VSELST2)
  ADD GROUP(DFHCLNT) LIST(VSELST2)
  ADD GROUP(CICREXX) LIST(VSELST2)
  ADD GROUP(TCPIP) LIST(VSELST2)
  ADD GROUP(VSEAI62) LIST(VSELST2)
  ADD GROUP(EZA) LIST(VSELST2)
  ADD GROUP(DFH$WBSN) LIST(VSELST2)
* $$ SLI MEM=IESZFCT2.Z
  ADD GROUP(VSESPG) LIST(VSELST2)
  ADD GROUP(FCTC2) LIST(VSELST2)
  ADD GROUP(CEE) LIST(VSELST2)
  LIST ALL
/*
* PLEASE OPEN DFHCSD AGAIN
// PAUSE
/&
* $$ EOJ
```

Note: When you have completed both skeletons (SKCICS2 and SKPREPC2), continue with “Task 3: Modify CICS Control Tables” on page 147.

Chapter 12. Maintaining VTAM Application Names and Startup Options

Maintaining VTAM Application Names

The dialog *Maintain VTAM Application Names* helps you maintain VTAM application names: To access the dialog, start with the *z/VSE Function Selection* panel and select:

2 (Resource Definition)

5 (Maintain VTAM Application Names)

You get the panel *VTAM APPLID Maintenance: APPLID List* as shown in Figure 41.

```
COM$APPA         VTAM APPLID MAINTENANCE: APPLID LIST



OPTIONS:     2 = ALTER AN APPLID            5 = DELETE AN APPLID


   OPT       APPLID   APPLICATION  APPLICATION   DEFAULT
                         TYPE       PROPERTY     LOGAPPL
    _       DBDCCICS     CICS       LOGAPPL        X
    _       PRODCICS     CICS       LOGAPPL        _
    _       POWER        RJE                       _
    _       PNET         PNET                      _
    _       ________                               _
    _       ________                               _
    _       ________                               _
    _       ________                               _
    _       ________                               _
    _       ________                               _
    _       ________                               _

PF1=HELP       2=REDISPLAY  3=END                    5=PROCESS    6=ADD APPL
```

Figure 41. Panel for VTAM APPLID Maintenance

The panel lists the application names (APPLIDs) of the VTAM applications installed on your system. The APPLIDs listed are defined as minor nodes of the VTAM application major node VTMAPPL. The applications shown are of type:

- CICS

  for primary CICS Transaction Server (DBDCCICS) or additional CICS systems (PRODCICS, for example).
- RJE

  for the standard VSE/POWER RJE (Remote Job Entry) definition.
- PNET

  for the VSE/POWER networking support program.

Other *possible* application types are:

- TCP/IP

  if you are using the z/VSE base program TCP/IP for VSE/ESA.
- PSF

if you are using the z/VSE optional program PSF/VSE (Print Services Facility/VSE).

- SELF-DEFINED

  for user-defined applications.

If you define your own applications (SELF-DEFINED), you have to include the application macro definition in a special library member in VSE/ICCF library 2. This member is named E$$VTMAP and included in the VTAM major node VTMAPPL during generation. If you have that member in your private VSE/ICCF library, it is retrieved from there and not from library 2. “Application Major Node” in the manual *z/VSE SNA Networking Support* provides further details about this facility.

Application property LOGAPPL, as shown for application type CICS, indicates that you can set up a direct sign on to that application. An 'X' in the last column indicates that this APPLID is taken as default value for the LOGAPPL parameter for terminal configuration. The dialog offers the following functions:

- Add (**PF6**)
- Alter (Option **2**)
- Delete (Option **5**)

After entering your changes, press **PF5**. You get the *Job Disposition* panel to submit the job created to batch, or file it in your VSE/ICCF primary library, or both.

Maintaining VTAM Startup Options

With the dialog *Maintain VTAM Startup Options* you maintain VTAM parameters for startup. To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

6 (Maintain VTAM Startup Options)

You get the panel *VTAM Start Options Maintenance* that allows you to maintain the following parameters stored in the VTAM startup member ATCSTR00:

HOSTSA

This is the SNA subarea number of this host. The value (in hexadecimal) you can specify depends on the naming convention selected during initial installation. The 2-digit subarea naming convention allows a value from 1 to 255; the 4-digit subarea naming convention from 1 to 65535. Note, however, that for a value greater 511 each subarea range of 256 requires an additional 7KB of VTAM buffer.

z/VSE supports per default 511 subareas. This is the default value in the start option MXSUBNUM.

If you change the HOSTSA value, the system renames **on request** all VTAM resource names. The suffix of the resource names is changed to the HOSTSA value.

PROMPT

Enter 1 if you want the operator to be prompted for entering the startup options during system startup.

Enter 2 if you want the system to take the values as defined on the panel. This results in an automatic startup of VTAM. No operator intervention is required. z/VSE creates VTAM startup member ATCSTR00 and stores it in

VSE/ICCF library 51. You can apply permanent changes to the VTAM startup book(s) (if they were created via dialog) by using member E$$VTMST in VSE/ICCF library 2. Member E$$VTMST is automatically included in startup member ATCSTR00. “Values Entered through a Dialog” in the manual *z/VSE SNA Networking Support* provides further details.

NETID
This is the 1 to 8 character name of the network this VTAM is part of. The name should be unique within interconnected networks.

After typing in your changes, press **ENTER**. You get the *Job Disposition* panel to submit the job created to batch, or file it in your VSE/ICCF primary library, or both.

Chapter 13. Maintaining and Cataloging Printer Information

Maintaining Printer FCB

You use the *Maintain Printer FCB* dialog to define and maintain printer forms control buffers (FCBs). For the FCBs provided by z/VSE, refer to "Creating Print Buffers for a System Printer" in the manual *z/VSE Installation*.

To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)
4 (Hardware Configuration and IPL)
3 (Maintain Printer FCB)

The panel you get displays the FCBs currently defined. You can scroll through the entries using **PF7** and **PF8**. The options you can select are at the top of the display. Enter the option number in the OPT column to the left of the FCB name you want to process.

When you are finished, press **PF5**. The dialog processes the information and redisplays the selection panel.

The options you can select for maintaining FCBs are:

1 (Add)
2 (Alter)
5 (Delete)
7 (Catalog)

The dialog creates a job for cataloging the new or changed FCB in the IJSYSRS.SYSLIB library. After pressing **PF5** you get the *Job Disposition* panel. With it, you can submit the job to batch, file it in your default VSE/ICCF primary library, or both.

Add or Change an FCB

If you add (ADD) or change (ALTER) an FCB, you need the following information on the printer's characteristics:

FCB NAME
If you add an FCB, enter the new FCB name. If you alter an FCB, you can change the name.

The name identifies the FCB to the Interactive Interface. It is **not** the phase name. You can use the same name that you use for the phase name. The FCB name must be unique.

DEVICE TYPE
Enter **3203, 3211, 3262, 3289, 4245, 4248**, or **6262**.

PHASE NAME
The library phase name. You use this name either in the LFCB command or in a VSE/POWER LST control statement to load the FCB.

The system automatically loads one FCB at IPL. The phase name is fixed for each printer type. Unless an FCB has the standard name for the printer type, it must be loaded by the operator or by VSE/POWER JECL.

LINES PER INCH
Enter either **6** or **8** for the number of lines per inch. If you want to change the lines-per-inch setting anywhere on the form (when using an IBM 4248 or IBM 3262 printer for example), leave this field blank.

FORM LENGTH
Enter the page length, in inches. For example; 11, 12, and 8.5 are acceptable.

3211 INDEXING
Specify the following information for IBM 3211 printers with the indexing feature:

- SHIFT DIRECTION

  0 - No indexing

  1 - Right

  2 - Left

- SHIFT NUMBER

  The number of positions to be shifted (**1 - 31**). For no indexing, enter **0**.

CHANNEL POSITIONS
Enter the print line position for channels 1 - 12. If the channel is not used, enter **0**.

Channel 1 **cannot** be 0.

VERIFICATION MESSAGE
Enter a message that is printed on SYSLST when the FCB is loaded. The message is printed after the following header (where xxxxxxxx is the phase name):

```
xxxxxxxx  LOADED
```

The following parameters are only required for IBM 4248 printers.

STACKER-LEVEL CONTROL
Specify **1**, **2**, **3**, or **4**. The meaning is as follows:

```
1 = Automatic stacker level control.
2 = 25 mm (1 in) below automatic stacker level control.
3 = 51 mm (2 in) below automatic stacker level control.
4 = 76 mm (3 in) below automatic stacker level control.
```

HORIZONTAL COPY
Specify **1** (for YES) or **2** (for NO).

PRINT SPEED
Specify **1**, **2**, **3**, or **4**. The meaning is as follows:

```
1 = No change of the print-speed setting.
2 = Low speed    (2.200 lines per minute).
3 = Medium speed (3.000 lines per minute).
4 = High speed   (3.600 lines per minute).
```

TRAY DROP RATE

Specify **1**, **2**, **3**, or **4**. The meaning is as follows:

1 = 7 forms cause a drop.
Form thickness: 0.5 mm (0.02 in).
2 = 13 forms cause a drop.
Form thickness: 0.2 mm (0.007 in).
3 = 19 forms cause a drop.
Form thickness: 0.1 mm (0.005 in).
4 = 25 forms cause a drop.
Form thickness: < 0.1 mm (0.003 in).

OFFSET COUNT BYTE

If *horizontal copy* is on, the offset count byte specifies where the duplicate print line begins.

The dialog creates a job for cataloging the new or changed FCB in the IJSYSRS.SYSLIB library. After pressing **PF5** you get the *Job Disposition* panel. With it, you can submit the job to batch, file it in your default VSE/ICCF primary library, or both.

Additional Considerations When Maintaining Printer FCB

If you want to use an FCB that you create, you must load it into the printer. No re-IPL is necessary. The operator (or VSE/POWER JECL) can load it by using the LFCB command.

Cataloging Printer UCB

You use the *Catalog Printer UCB* dialog to catalog a universal character set buffer (UCB). For the UCBs provided by z/VSE, refer to the manual *z/VSE Installation* under "Creating Print Buffers for a System Printer".

A UCB converts bit patterns sent to the printer into specific locations on the print train. By using a UCB, you can take advantage of options such as different print trains, and upper- and lowercase printing.

The *Catalog Printer UCB* dialog creates a job to catalog a standard UCB or to assemble and catalog a non-standard UCB. To access this dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

4 (Hardware Configuration and IPL)

4 (Catalog Printer UCB)

You get the *Catalog Printer UCB: Select Function* panel. From this panel you can select three different catalog dialogs:

- **1** (Catalog IPL loaded standard train)

  Select this option to define a new UCB that is automatically loaded at IPL. The new UCB replaces the current UCB that is loaded at IPL.
- **2** (Catalog user loaded standard train)

  Select this option if you want to define a new UCB, but you do **not** want to replace the current UCB loaded at IPL. The new UCB is assigned a name that can be used when a program is run.
- **3** (Catalog user loaded non-standard train)

Select this option if you want to define your own load phase. The source for the phase is in a VSE/ICCF library member. The dialog helps you assemble and catalog the non-standard UCB. The current UCB is still loaded at IPL.

After you make your selection, the dialog displays various panels. The input you need depends on the type of UCB you catalog (standard or non-standard). Review the descriptions below.

Standard UCB

If you select option 1 or 2 (standard train), you need the following information:

PRINTER TYPE
Select the printer type you are using.

PRINT TRAIN
Select the train type for the printer you are using.

BUFFER NAME
This is the phase name that is assigned to the UCB in the VSE library. The name **cannot** be:

- ALL
- S
- ROOT

You are asked for the buffer name, only if you are cataloging a user loaded standard train (option 2 on the selection panel).

After entering the information required, the *Job Disposition* panel is displayed. With it, you can submit the job to batch, file it in your VSE/ICCF primary library, or both.

Non-Standard UCB

If you select option 3 (non-standard train), you need the following information:

MEMBER NAME
The name of the VSE/ICCF library member that contains the source for the UCB you create.

PASSWORD
The password for the VSE/ICCF member. This is needed, if the member is password-protected.

LIBRARY NUMBER
The number of the library that contains the VSE/ICCF member.

BUFFER NAME
The phase name that is assigned to the UCB in the VSE library. The name **cannot** be:

- ALL
- S
- ROOT

After entering the information required, the *Job Disposition* panel is displayed. With it, you can submit the job to batch, file it in your VSE/ICCF primary library, or both.

Additional Considerations When Cataloging Printer UCB

1. If you select option 1 (Catalog IPL Loaded Standard Train), the dialog catalogs the UCB in the system library with a standard name. The system automatically loads it into the printer at IPL.
2. If you select option 2 (Catalog User Loaded Standard Train), you define the UCB with a phase name. The operator has to load it into the printer (with the LUCB command) before you can use it.
3. If you select option 3 (Catalog User Loaded Non-Standard Train), you must create a VSE/ICCF library member that contains the source for the UCB phase. The operator has to load the UCB into the printer (with the LUCB command).

Cataloging Your Own Print Control Buffer Phases

This step applies if the print trains (or belts) used on your location's printers do not match the default FCB and UCB images that are loaded automatically during IPL. You may catalog your own FCB (forms control buffer) or UCB (universal character set buffer) image phases, if there is a need. For a list of default FCB and UCB image phases, refer to the manual *z/VSE System Control Statements* under “Buffer Load Phases”; this list includes the names of the UCB-image object modules shipped with z/VSE for possible link-editing. The manual describes, in addition, how to create FCB or UCB image phases and how to catalog them. Your control statements for cataloging buffer-image phases (assuming that an IBM supplied UCB-image object module can be used) should be:

```
// JOB    CATALOG BUFFER IMAGE
// OPTION CATAL
LIBDEF *,SEARCH=IJSYSRS.SYSLIB,CATALOG=IJSYSRS.SYSLIB
  PHASE $$BUCBxx,*
  INCLUDE IJBxxxxx
   ...
   ...       PHASE and INCLUDE statements for additional
   ...       buffer-image phases.
   ...
// EXEC LNKEDT,PARM='MSHP'
/*
/&
```

On completion of this job step, you can load any of the newly cataloged buffer-image phases by entering an LFCB or LUCB command in the format:

```
LFCB cuu,phasename
LUCB cuu,phasename,NOCHK
```

Chapter 14. Extending and Tailoring System Files

Extending the VSE/ICCF DTSFILE

If necessary, you can allocate more space to the VSE/ICCF DTSFILE. You can do this by defining a larger extent on SYSWK1 or by defining extents on several volumes.

Estimating Used Space

Before you extend the amount of space reserved for the DTSFILE, you should make sure that you actually need to do so. Space allocated to the DTSFILE cannot be used for any other purpose.

The topic “z/VSE Disk Layouts” in the manual *z/VSE Planning* documents the initial space allocations for the DTSFILE (ICCF.LIBRARY) on SYSWK1. You can estimate how much of that space is currently in use in two ways:

1. During system startup, look for the following message:

   ```
   K088I HI FILE RECORDS=number (nn%)
   ```

   This message is issued for the F2 (CICS-VSE/ICCF) partition. If **nn%** is near 0%, it may mean that the reserved space is almost used up.
2. Access the **DTSUTIL** utility of VSE/ICCF. From the Administrator *z/VSE Function Selection* panel select the *Command Mode*. Enter first *$DTSUTIL* to invoke the utility and then

   ```
   DISPLAY LIBRARY
   ```

   to display library information. The resulting display shows the total amount of DTSFILE space and the amount of free space left. The following example shows that at least 372,333 records of the space reserved for the libraries are still available:

   ```
   RECORDS IN FILE     432,040
   HI FILE RECORDS     372,333
   ```

It is also recommended that you back up and restore the VSE/ICCF libraries before you allocate more space to the DTSFILE. This may free up some space that currently cannot be used. *z/VSE Operation* has information about backing up and restoring the VSE/ICCF libraries under “Backing Up VSE/ICCF Libraries” and under “Restoring VSE/ICCF Libraries”.

Using Skeleton SKDTSEXT

With the skeleton SKDTSEXT, you create a job that extends the VSE/ICCF DTSFILE to a multivolume, multiextent file. The skeleton is shipped in VSE/ICCF library 59. Figure 42 on page 173 shows the skeletons and the variables you must specify.

To extend the DTSFILE, you must perform the following steps:

1. Backup the DTSFILE on tape. Use the *Backup the ICCF Library on Tape* dialog.
2. Create a restore job for the DTSFILE using the *Restore the ICCF Library from Tape* dialog. Submit the job. Defer its execution by using disposition L.

Note: The restore job **must** be in the VSE/POWER reader queue, before you run the extend job. In this way, it can be released while VSE/ICCF is down. If the job is not there, you might later have to reinstall your system in order to use VSE/ICCF again.

3. Prepare the extend job by using skeleton SKDTSEXT. Refer to Figure 42 on page 173 for details of the skeleton and what needs to be observed when using it. Comments included in the skeleton are not shown.

   Submit the extend job with disposition L. As a result, the job is stored in the VSE/POWER reader queue but not released for processing.

4. Update the **label information** in the STDLABEL procedure. The reason is that the job you created (via SKDTSEXT) can only update the temporary label area on disk but not the STDLABEL procedure itself. To update the label information permanently, perform the following steps:

   a. Copy the STDLABEL procedure from IJSYSRS.SYSLIB to your primary VSE/ICCF library. For copying, use the VSE/ICCF LIBRP command. In command mode, enter:

      ```
      LIBRP IJSYSRS.SYSLIB STDLABEL.PROC STDLABEL
      ```

   b. Update the DTSFILE label information.

   c. Ensure that all necessary JCL statements are present and submit procedure STDLABEL for processing (cataloging).

5. Shutdown the system and perform a MINI startup. Release the jobs in the reader queue. Release first the extend job and after its successful completion the restore job.

6. To activate the changed characteristics of the DTSFILE, shut down the system and perform a normal startup.

```
* $$ JOB JNM=ICCFEXT,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
      .
      .
// JOB ICCFFORM  FORMAT YOUR  z/VSE ICCF DTSFILE ON NEW EXTENTS
// DLBL DTSFILE,'ICCF.LIBRARY',99/366,DA
// EXTENT SYS010,SYSXXX,1,0,NNNNN,MMMMM
// EXTENT SYS011,SYSYYY,1,1,NNNNN,MMMMM
// ASSGN SYS010,DISK,VOL=SYSXXX,SHR
// ASSGN SYS011,DISK,VOL=SYSYYY,SHR
// PAUSE     BE SURE ICCF IS NOT OPERATIONAL
// EXEC DTSUTIL
FORMAT LIB(199) USERS(199)
/*
/&
// JOB UPDATE    UPDATE STDLABEL AREA  AND DTRICCF.PROC
// OPTION STDLABEL=DELETE
DTSFILE
/*
// OPTION STDLABEL=ADD
// DLBL DTSFILE,'ICCF.LIBRARY',99/366,DA
// EXTENT SYS010,SYSXXX,1,0,NNNNN,MMMMM
// EXTENT SYS011,SYSYYY,1,1,NNNNN,MMMMM
/*
      .
      .
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG DTRICCF.PROC DATA=YES REPL=YES
// ASSGN SYS010,DISK,VOL=SYSXXX,SHR
// ASSGN SYS011,DISK,VOL=SYSYYY,SHR
/+
CONNECT S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY DTRICCF.PROC REPLACE=YES
/*
/&
* $$ EOJ
```

Figure 42. Extending the VSE/ICCF DTSFILE (Skeleton SKDTSEXT)

Observe the following when using skeleton SKDTSEXT:

- First copy skeleton SKDTSEXT to your primary VSE/ICCF library.
- Assign a logical unit to every extent that you define. SYS010 must be the logical unit assigned to the first extent, SYS011 to the second extent, and so on. However, if you define the second extent on the same volume as the first extent, you must use SYS010 also for the second extent. Skeleton SKDTSEXT uses two extents on different volumes.

  Ensure that no overlap occurs with the extents of other files.
- SYS**XXX** and SYS**YYY** define the disk volume(s). SYSWK1, for example.
- **NNNNN** defines the beginning of an extent.
- **MMMMM** defines the total amount of space to be reserved for the extent.
- The disk volume used for the extent must have the SHR (share) option.
- In the skeleton, you also have to complete statements that update the system's standard labels and the procedure DTRICCF. DTRICCF contains the assignments for the DTSFILE. It is processed during startup of VSE/POWER and CICS.

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
```

Reformatting the VSE/ICCF DTSFILE

The DTSFILE generated by z/VSE defines 199 libraries and 199 VSE/ICCF user-ID records. You can *reformat* the DTSFILE to create additional libraries and user-ID records.

Note:

1. Before changing the original definitions for libraries and user-IDs, refer to “Formatting the Library File or Changing its Size” in the manual *VSE/ICCF Administration and Operation*. This manual has detailed information about formatting the DTSFILE.
2. For z/VSE, you *must* define VSE/ICCF libraries with the DATE option.

Skeleton SKICCFMT has the values that z/VSE specifies for the file. If you use the skeleton, first copy it from VSE/ICCF library 59 to your primary VSE/ICCF library. Then edit the copied file.

To reformat the DTSFILE, perform the following steps:

- Backup the DTSFILE on tape. Use the *Backup the ICCF Library on Tape* dialog.
- Create a restore job for the DTSFILE using the *Restore the ICCF Library from Tape* dialog. Replace line "RESTORE ALL" by "RESTORE LIBRARIES(n) USERS(u) ALL", and insert before the changed restore command a format command as follows:

  ```
  FORMAT LIBRARIES(n) USERS(u),
  ```

  where "n" is the number of desired libraries and "u" the number of desired VSE/ICCF users.
- Prepare the job to add new libraries according to skeleton SKICFFMT. Be aware that no format command is needed and only those libraries must be mentioned, that have not already been added when the DTSFILE backup was done. Insert a PAUSE statement before calling DTSUTIL to be able to disconnect the DTSFILE when running the job. Submit the job.

Figure 43 on page 175 shows the skeleton. Comments included in the skeleton are not shown.

The sample has only one variable, **-V001-**. You can also change, add, or delete other statements or parameters.

```
* $$ JOB JNM=SKICFFMT,DISP=D,CLASS=0
// JOB SKICFFMT
// ASSGN SYS010,DISK,VOL=-V001-,SHR
// ASSGN SYS011,DISK,VOL=-V002-,SHR
// EXEC DTSUTIL
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

               .
               .        (Additional ADD LIBRARY statements)
               .

ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
```

Figure 43. Skeleton SKICFFMT, Part 1 of 2 (Formatting the VSE/ICCF DTSFILE)

In the ASSGN statement, replace the variable **-V001-**. Specify the volume number of the disk where the DTSFILE resides.

The skeleton adds the following libraries:

1 = For VSE/ICCF administrator
2 = Common library
3-6 = Public libraries

Libraries 7 - 49 are private libraries that can be assigned to users.

Note: Libraries 8, 9, and 10 are used as primary libraries by the predefined z/VSE users OPER, PROG, and SYSA.

```
*  ADD LIBRARIES 50 THRU 68  . . .

ADD LIBRARY DATE NOCOMMON PUBLIC
ADD LIBRARY DATE NOCOMMON PUBLIC

            .
            .           (Additional ADD LIBRARY statements)
            .

ADD LIBRARY DATE NOCOMMON PUBLIC
ADD LIBRARY DATE NOCOMMON PUBLIC
*  ADD LIBRARIES 69 THROUGH 99

ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE

            .
            .           (Additional ADD LIBRARY statements)
            .

ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
*  ADD LIBRARIES 100 THROUGH 199.

ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE

            .
            .           (Additional ADD LIBRARY statements)
            .
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
ADD LIBRARY MAXDIR(200) FREESPACE(25) DATE
DSERV ALL COMMON SORTED
END
/*
/&
* $$ EOJ
```

Figure 44. Skeleton SKICFFMT, Part 2 of 2 (Formatting the VSE/ICCF DTSFILE)

The skeleton adds libraries 50 - 68 which are reserved for z/VSE and are used by the Interactive Interface. The skeleton also adds libraries 69 - 99 which are private and can be assigned to users.

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.

Extending VSE/POWER Files

If there is a need to extend the VSE/POWER files, do it carefully. Incorrect specifications are likely to cause startup problems. Two sample job streams (skeletons) are provided, the first one for extending both the data file (IJDFILE) and the queue file (IJQFILE), the second one for extending the data file (IJDFILE) only.

Extending the Queue File and Data File by a VSE/POWER Cold Start

The VSE/POWER IJQFILE is designed for about 500 to 1000 queue file records. Extending it or placing the queue and/or the data file at a different disk location can only be done by a VSE/POWER cold start.

Note: If it is necessary to increase the queue file, you must also increase the size of the partition GETVIS area (skeleton SKALLOCx). For the values to be specified, refer to the topic “Planning for VSE/POWER” in the manual *VSE/POWER Administration and Operation*.

The following steps are required to extend/move the space for the VSE/POWER data file and queue file:

1. Use the POFFLOAD BACKUP command to save the queue entries on tape.
2. Update the file extent information in the label procedure STDLABEL.PROC. Proceed as follows:
   a. Copy the label procedure from IJSYSRS.SYSLIB to your primary VSE/ICCF library. For copying, use the VSE/ICCF command LIBRP. In Command Mode, enter:
   ```
   LIBRP IJSYSRS.SYSLIB STDLABEL.PROC STDLABEL
   ```
   b. Update the file extent information NNNNN (beginning of extent) and MMMMM (amount of space). Sample statements are shown below:
   ```
   // DLBL IJQFILE,'VSE.POWER.QUEUE.FILE',99/366,DA
   // EXTENT SYS001,DOSRES,1,0,NNNNN,MMMMM
   // DLBL IJDFILE,'VSE.POWER.DATA.FILE',99/366,DA
   // EXTENT SYS002,SYSWK1,1,0,NNNNN,MMMMM
   ```
   For a description of the DLBL and EXTENT statements, see the manual *z/VSE System Control Statements* under “DLBL” and under “EXTENT”.

   To extend the queue file or the data file, ensure that enough space is available on the disk volume(s) that you use. The queue file can have only one extent. The data file can have up to 32 extents, and all of its extents must reside on disk volumes of the same device type.

   Note:
   1) If you extend the VSE/POWER data file (IJDFILE) over more than one volume, you have to use consecutive SYSnnn numbers starting with SYS002 and update the label information in STDLABEL.PROC accordingly.
   2) If you define extents on multiple volumes or if you move to a different volume than SYSWK1, you must update procedure DTRPOWR in IJSYSRS.SYSLIB. DTRPOWR includes the ASSGN statements for the VSE/POWER account, queue, and data files. Refer also to the skeleton SKPWREXT shown in Figure 45 on page 178.
   3) If you move the VSE/POWER queue file (IJQFILE) to a different volume than DOSRES, you must update procedure DTRPOWR in IJSYSRS.SYSLIB.

   c. Add to your edited member the control statements listed below. In front of your member add:
   ```
   * $$ JOB JNM=RECAT,CLASS=0,PRI=9
   // JOB RECAT
   // LIBDEF *,CATALOG=IJSYSRS.SYSLIB
   // EXEC LIBR,PARM='MSHP'
   ACC S=IJSYSRS.SYSLIB
   ```

At the end of your member add:

```
/&
* $$ EOJ
```

3. Submit the updated label procedure by selecting option 7 or by entering: SUBMIT STDLABEL. This causes the procedure to be written back into IJSYSRS.SYSLIB, replacing the original label-information statements.
4. Shut down your system; follow the procedure described in the *VSE/ESA Operation* manual under "Shutting Down the System".
5. Perform a COLD startup of your sytem.
6. Reload the queue entries saved on tape. Use the POFFLOAD LOAD command. The manual *z/VSE Operation* describes in detail how to use the POFFLOAD command under “Offloading and Loading VSE/POWER Queues”. For a detailed description of the LIBRP command, refer to the manual *VSE/ICCF User's Guide* under “LIBRP Macro”.

You may replace steps 2 and 3 by using skeleton SKPWREXT provided in VSE/ICCF library 59 and follow its instructions.

Figure 45. Skeleton SKPWREXT

```
* $$ JOB JNM=POWEREXT,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB POWEREXT
* -------------------------------------------------------------------
*  STEP 1
* -------------------------------------------------------------------
*  CHANGE THE LABEL PROCEDURE
*      STDLABEL.PROC IN IJSYSRS.SYSLIB AS FOLLOWS:
*        1. COPY THE PROCEDURE INTO YOUR PRIMARY LIBRARY USING LIBRP.
*        2. MODIFY THE LABELS FOR POWER DATA AND/OR  ACCOUNT
*           FILE AND SAVE THE MODIFIED FILE.
*           IF CHANGING THE QUEUE FILE IT MIGHT BE NECESSARY TO
*           ADJUST THE PARTITION SIZE, REFER ALSO TO THE
*           ADMINISTRATION GUIDE.
*        3. INSERT THE NAME OF THIS ICCF MEMBER IN THE SUBSEQUENT
*           INCLUDE STATEMENT - VARIABLE --V001--
*       OR USE DITTO AND CHANGE THE PROCEDURE DIRECTLY, DON'T
*       FORGET TO CHANGE ALSO IN PRD2.SAVE AND REMOVE FOLLOWING
*       STEP.
// EXEC LIBR,PARM='MSHP'
AC S=IJSYSRS.SYSLIB
/INCLUDE --V001--
// EXEC LIBR,PARM='MSHP'
CON S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY STDLABEL.PROC R=Y
/*
* -------------------------------------------------------------------
*  STEP 2
* -------------------------------------------------------------------
*  CHANGE PROCEDURE DTRPOWR
*      IF ANY OF THE POWER FILES WAS MOVED TO A DIFFERENT VOLUME
*      THE ASSIGNMENT MUST ALSO BE CHANGED.
*      CHANGE THE PROCEDURE AS YOU CHANGED THE LABEL PROCEDURE,
*      THE NAME OF THE PROCEDURE HAS TO BE CHANGED IN THE SUBSEQUENT
*      INCLUDE STATEMENT - VARIABLE --V002--
*      OR USE DITTO AND CHANGE THE PROCEDURE DIRECTLY, DON'T
*      FORGET TO CHANGE ALSO IN PRD2.SAVE AND REMOVE FOLLOWING
*      STEP.
// EXEC LIBR,PARM='MSHP'
AC S=IJSYSRS.SYSLIB
/INCLUDE --V002--
// EXEC LIBR,PARM='MSHP'
CON S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY DTRPOWR.PROC R=Y
/*
* -------------------------------------------------------------------
*  STEP 3
* -------------------------------------------------------------------
*  MAKE SURE A POWER COLD START IS PERFORMED
```

```
* ---------------------------------------------------------------
// SETPARM XNCPU=' '
// EXEC PROC=$COMVAR,XNCPU
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;'
   SET XPWMODE=COLD
/*
* ---------------------------------------------------------------
*  STEP 4
* ---------------------------------------------------------------
*  FOR THE FOLLOWING COLD START, BACKUP THE DATA FILE AS FOLLOWS:
*             1. REPLY "(END/ENTER)" TO FINISH THIS JOB
*             2. SHUTDOWN ALL PARTITIONS EXCEPT POWER
*             3. POFFLOAD YOUR POWER QUEUES
*                  POFFLOAD BACKUP,ALL,CUU
*                                      CUU OF THE TAPE DRIVE
*             4. IPL FROM DOSRES, SYSTEM WILL ISSUE A COLD START
*             5. WHEN VSE/POWER IS UP AFTER IPLING, LOAD THE DATA
*                BACK INTO THE SYSTEM:
*                  POFFLOAD LOAD,ALL,CUU
*                              CUU OF THE TAPE DRIVE
* ---------------------------------------------------------------
// PAUSE
/&
* $$ EOJ
```

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.

Extending the Data File during a VSE/POWER Warm Start

If there is only a need for more VSE/POWER data file space, for example to be able to store more LST output, you may increase the data file during a VSE/POWER Warm start. This is possible without reformatting the existing queue file and data file extents as required in case of a Cold start. Therefore, the Data File extension will **not** affect the already spooled data and cause no long system-down-time.

To trigger Data File extension during Warm start, append **one** extent with ascending sequence number to the existing IJDFILE DLBL/EXTENT statements. The new extent must be added as the last extent, because VSE/POWER accesses the extents as a contiguous stream of DBLKs, starting with DBLK #0 and ending with DBLK #n. DBLKs on existing extents are already referred to by their number which cannot be changed. VSE/POWER will detect the appended extent during the next Warm start and will ask the operator to confirm Data File extension by:

```
1QD2D DATA FILE EXTENT NO. mm FOUND - TO FORMAT REPLY 'YES' ELSE 'NO'
(// EXTENT SYSxxx,volid,1,nnn,start,length)
```

When the operator replies YES, VSE/POWER verifies the specified location of the appended extent. If the new extent is accepted, formatting of the new extent takes place after Warm start has been completed. While the additional extent is formatted, spooling is already enabled. More details about this process can be found in the manual *VSE/POWER Administration and Operation*.

Note: As long as the maximum number of Data File extents is not yet reached, "Data File extension during Warm start" can be repeated during a subsequent VSE/POWER Warm start. Therefore, it is recommended to define a Queue File larger than needed during a VSE/POWER Cold start to avoid any further VSE/POWER Queue File extensions. The following steps are required:

1. Update the file extent information in the label procedure STDLABEL.PROC as follows:
   a. Copy the label procedure from IJSYSRS.SYSLIB to your primary VSE/ICCF library. For copying, use the VSE/ICCF command LIBRP. In *Command Mode*, enter:
      ```
      LIBRP IJSYSRS.SYSLIB STDLABEL.PROC STDLABEL
      ```
   b. Append another EXTENT statement for the data file (IJDFILE). Sample statements show DLBL, EXTENT, and ASSGN statements before and after appending a new EXTENT. The old DLBL/EXTENT and ASSGN statements have the following values:
      ```
      // DLBL IJDFILE,'VSE.POWER.DATA.FILE',99/366,DA
      // EXTENT SYS002,SYSWK1,1,0,start1,length1
      // EXTENT SYS002,SYSWK1,1,1,start2,length2
      // EXTENT SYS003,SYSWK4,1,2,start3,length3

      // ASSGN SYS002,DISK,VOL=SYSWK1,SHR      POWER DATA FILE 1 + 2
      // ASSGN SYS003,DISK,VOL=SYSWK4,SHR      POWER DATA FILE 3
      ```
      The new DLBL/EXTENT and ASSGN statements have the following values:
      ```
      // DLBL IJDFILE,'VSE.POWER.DATA.FILE',99/366,DA
      // EXTENT SYS002,SYSWK1,1,0,start1,length1
      // EXTENT SYS002,SYSWK1,1,1,start2,length2
      // EXTENT SYS003,SYSWK4,1,2,start3,length3
      // EXTENT SYS004,SYSWK2,1,3,start4,length4

      // ASSGN SYS002,DISK,VOL=SYSWK1,SHR      POWER DATA FILE 1 + 2
      // ASSGN SYS003,DISK,VOL=SYSWK4,SHR      POWER DATA FILE 3
      // ASSGN SYS004,DISK,VOL=SYSWK2,SHR      POWER DATA FILE 4
      ```
      For a description of the DLBL and EXTENT statements, refer to the topics “DLBL” and “EXTENT” in the manual *z/VSE System Control Statements*.

      To extend the Data File, ensure that enough space is available on the disk volume(s) that you use. The Data File can have up to 32 extents, and all these extents must reside on disk volumes of the same device type.

      Note:
      1) If you extend the VSE/POWER data file (IJDFILE) over more than one volume, you have to use consecutive SYSnnn numbers starting with SYS002 and update the label information in STDLABEL.PROC accordingly.
      2) If you define extents on multiple volumes or if you move to a different volume than SYSWK1, you must update procedure DTRPOWR in IJSYSRS.SYSLIB. DTRPOWR includes the ASSGN statements for the VSE/POWER account, queue, and data files. Refer also to skeleton SKPWRDAT shown in Figure 46 on page 182.
   c. Add to your edited member the control statements listed below.

      In front of your member:
      ```
      * $$ JOB JNM=RECAT,CLASS=0,PRI=9
      // JOB RECAT
      // LIBDEF *,CATALOG=IJSYSRS.SYSLIB
      // EXEC LIBR,PARM='MSHP'
      ACC S=IJSYSRS.SYSLIB
      ```
      At the end of your member:
      ```
      /&
      * $$ EOJ
      ```
2. Submit the updated label procedure by selecting option 7 or by entering:
   ```
   SUBMIT STDLABEL
   ```

This causes the procedure to be written back into IJSYSRS.SYSLIB, replacing the original label-information statements.

3. Shut down your system; follow the procedure described in the *z/VSE Operation* manual under “Shutting Down the System”.
4. Re-IPL your system which will prompt you with message 1QD2D.

For a detailed description of the LIBRP command, refer to the topic “LIBRP Macro” in the manual *VSE/ICCF User’s Guide*

You may replace steps 1 and 2 by using skeleton SKPWRDAT provided in VSE/ICCF library 59 and follow its instructions.

```
* $$ JOB JNM=POWERDAT,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB POWERDAT
* ----------------------------------------------------------------------
*  STEP 1
* ----------------------------------------------------------------------
*  CHANGE THE LABEL PROCEDURE
*      STDLABEL.PROC IN IJSYSRS.SYSLIB AS FOLLOWS:
*        1. COPY THE PROCEDURE INTO YOUR PRIMARY LIBRARY USING LIBRP.
*        2. APPEND ONE EXTENT FOR POWER DATA FILE IJDFILE AND
*           SAVE THE MODIFIED FILE.
*     NOTE: THE ADDITIONAL EXTENT MUST EITHER RESIDE ON THE SAME
*           DISK AS THE LAST EXTENT AND USE THE SAME LOGICAL UNIT
*           NUMBER (SYSNNN) OR MUST RESIDE ON A DISK
*           CONTAINING NO DATA FILE EXTENTS SO FAR AND THE LOGICAL
*           UNIT NUMBER SYSNNN MUST BE INCREMENTED BY ONE.
*
*        3. INSERT THE NAME OF THIS ICCF MEMBER IN THE SUBSEQUENT
*           INCLUDE STATEMENT - VARIABLE --V001--
*       OR USE DITTO AND CHANGE THE PROCEDURE DIRECTLY, DON'T
*       FORGET TO CHANGE ALSO IN PRD2.SAVE AND REMOVE FOLLOWING
*       STEP.
// EXEC LIBR,PARM='MSHP'
AC S=IJSYSRS.SYSLIB
/INCLUDE --V001--
// EXEC LIBR,PARM='MSHP'
CON S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY STDLABEL.PROC R=Y
/*
* ----------------------------------------------------------------------
*  STEP 2
* ----------------------------------------------------------------------
*  CHANGE PROCEDURE DTRPOWR
*      IF THE APPENDED EXTENT OF THE POWER DATA FILE RESIDES ON
*      A NOT YET ASSIGNED VOLUME, YOU MUST ADD THE ASSIGNMENT.
*      CHANGE THE PROCEDURE AS YOU CHANGED THE LABEL PROCEDURE,
*      THE NAME OF THE PROCEDURE HAS TO BE CHANGED IN THE SUBSEQUENT
*      INCLUDE STATEMENT - VARIABLE --V002--
*      OR USE DITTO AND CHANGE THE PROCEDURE DIRECTLY, DON'T
*      FORGET TO CHANGE ALSO IN PRD2.SAVE AND REMOVE FOLLOWING
*      STEP.
// EXEC LIBR,PARM='MSHP'
AC S=IJSYSRS.SYSLIB
/INCLUDE --V002--
// EXEC LIBR,PARM='MSHP'
CON S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY DTRPOWR.PROC R=Y
/*
* ----------------------------------------------------------------
*  STEP 3
* ----------------------------------------------------------------
*  DURING THE FOLLOWING WARM START, VSE/POWER WILL EXTEND THE DATA FILE
*             1. REPLY "(END/ENTER)" TO FINISH THIS JOB
*             2. IPL FROM DOSRES, SYSTEM WILL ISSUE A WARM START
*             3. WHEN VSE/POWER REQUESTS CONFIRMATION FOR
*                DATA FILE EXTENSION BY MESSAGE 1QD2D, REPLY 'YES'
* ----------------------------------------------------------------
// PAUSE
/*
/&
* $$ EOJ
```

Figure 46. Skeleton SKPWRDAT

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.

Chapter 15. Tailoring Terminal Functions and Console Definitions

This chapter describes several tasks for tailoring terminal functions. For the task of changing z/VSE console definitions, refer directly to "Tailoring Console Definitions" on page 193.

Using Skeleton IESxLOGO

The **IESxLOGO** skeleton allows you to modify the *z/VSE Online* panel and related functions. With this panel you sign on to z/VSE. The '**x**' in the logo name refers to the language being used:

E English

J Japanese

An example of this panel is shown in Figure 1 on page 10.

Using IESxLOGO (where 'x' refers to the language you are using: E or J) you can:

1. Change the **logo** that is displayed on the panel. The default logo is z/VSE. You can implement your own logo design for the panel display.
2. Set a **limit** for invalid **sign-on** attempts. The MAXNUMSO parameter is retained for compatibility reasons only. To modify the number of invalid sign-on attempts, you must use the BSTADMIN command PERFORM PASSWORD (for details, see "Overview of BSM BSTADMIN Commands and Their Syntax" on page 358).
3. Allow **every** CICS user to **escape** to CICS from the panel without signing on to the Interactive Interface.

   To implement the escape function, you can either:

   - Specify that PF6 and PF9 are displayed on the panel. These PF keys are used for the escape facility.
   - Specify a 1 - 4 character string for the escape facility. A user can then enter this character string from the *z/VSE Online* panel to escape to CICS.

   Note: The security functions of the Interactive Interface (user-ID, password) are bypassed when allowing to "escape" to CICS and use it in native mode.
4. Specify the offset or **cuu** for non-SNA terminals in the **netname** to use the PF3 function key when running under VM.
5. Configure the "logon here" function.

IESxLOGO modifications become effective for all the terminals defined to a CICS subsystem.

The skeleton is shipped in VSE/ICCF library 59. First copy it to your VSE/ICCF primary library and edit the copied skeleton. Refer to "Copying Skeletons" on page 11 for information on copying skeletons.

Figure 47 through Figure 49 on page 188 shows the skeleton. A description of the statements and changes follows each part of the skeleton.

```
* $$ JOB JNM=IESELOGO,CLASS=A,DISP=D,NTFY=YES
* $$ LST CLASS=Q,DISP=H
// JOB IESELOGO ASSEMBLE
// LIBDEF *,CATALOG=PRD2.CONFIG
* IN CASE GENERATION FEATURE IS INSTALLED ACTIVATE THE FIRST LIBDEF
* // LIBDEF SOURCE,SEARCH=(PRD2.GEN1,PRD1.BASE,PRD1.MACLIB)
// LIBDEF SOURCE,SEARCH=(PRD1.BASE,PRD1.MACLIB)
// OPTION CATAL,LIST
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAX*
               -200K,ABOVE)'
           .
           .
         GBLB  &PUNCH               SHALL WE PUNCH A CATALOG STATEMENT?
&PUNCH   SETB  1                    THIS TIME THE ANSWER IS         YES
         AIF   (NOT &PUNCH).BYPUN   IF NO CATALOG STATEMENT REQUIRED
         PUNCH ' CATALOG IESELOGO.OBJ REP=YES'
         PUNCH ' PHASE  IESELOGO,S'
.BYPUN   ANOP                       NO CATALOG STATEMENT REQUIRED
LOGO     TITLE 'z/VSE --  USER CHANGEABLE LOGO PHASE'
IESELOGO CSECT
         DC    CL8'IESLOGO'       MODULE IDENTIFIER
         DC    X'64'              VSE/ESA 2.4.0 AND HIGHER
         DC    AL1(LOGOLINS)      NUMBER OF LINES OF LOGO TEXT
         DC    H'0'               ...  RESERVED  ...
         DC    A(LOGOBA)          ADDRESS OF THE LOGO TEXT
         DC    A(ESCAPESW)        ADDRESS OF THE ESCAPE SWITCH
         DC    A(MAXNUMSO)        ADDRESS MAX. NUMBER SIGNON ATTEMPTS
         DC    A(0)               *
         DC    A(UCESCSTR)        ADDRESS OF THE UPPER CASE ESCAPE
*                                    CHARACTER STRING
         DC    A(MCESCSTR)        ADDRESS OF THE MIXED CASE ESCAPE
*                                    CHARACTER STRING
         DC    A(CUUOFFS)         ADDRESS OF CUU OFFSET IN NETNAME
         DC    A(SIGNONH)         ADDRESS OF SIGNON-HERE SWITCH
*-------------------------------------------------------------------*
*          THE LINES ABOVE THIS BOX MUST NOT BE CHANGED             *
*-------------------------------------------------------------------*
         SPACE 2
           .
         SPACE 2
           .
         EQU   70                           FIXED LENGTH OF EACH LINE
           .
         SPACE 2
           .
         SPACE 2
```

Figure 47. IESELOGO Skeleton, Part 1 of 3

Note:

1. The statement

   ```
   // EXEC ASMA90....
   ```

   calls the High Level Assembler. Refer to the manual *z/VSE Guide to System Functions* under "High Level Assembler Considerations" for further details.
2. Do **not** change the statements in this part of the skeleton, except for the case described below: You may have a system with multiple CICS partitions and for each of those you want individual logos displayed. The logo created with this skeleton is cataloged in library PRD2.CONFIG. If you want to create a second logo, you must change the library definition (PRD2.CONFIG) in the LIBDEF

statement. Otherwise, the second logo simply replaces the logo you created first. Choose an appropriate sublibrary of your installation. Change the library search chain of the related CICS so that the sublibrary with the logo is early or first in the search chain.

3. To define the maximum number of sign-on attempts, you must use the BSTADMIN command PERFORM PASSWORD. For details, see “Overview of BSM BSTADMIN Commands and Their Syntax” on page 358.

Changing the LOGO Design

You use this section of the skeleton to change the logo design.

```
LOGOBA   EQU   *                 THIS LABEL MUST PRECEDE YOUR LOGO TEXT
*---------------------------------------------------------------------*
LOGOSKEL EQU   *                 THE SKELETON LOGO BEGINS HERE
*DC CL(L)'?????????????????????????????????????????????????????????????*
               ?????????'
*DC CL(L)'??????????????? YOU CAN REPLACE THE AREA ????????????????????*
               ?????????'
*DC CL(L)'?????????????? FILLED WITH QUESTION MARKS ???????????????????*
               ?????????'
*DC CL(L)'??????????????? WITH YOUR OWN LOGO TEXT ?????????????????????*
               ?????????'
*DC CL(L)'?????????????????????????????????????????????????????????????*
               ?????????'
*DC CL(L)'?????????????????????????????????????????????????????????????*
               ?????????'
*DC CL(L)'?????????????????????????????????????????????????????????????*
               ?????????'
*DC CL(L)'?????????????????????????????????????????????????????????????*
               ?????????'
*DC CL(L)'?????????????????????????????????????????????????????????????*
               ?????????'
*DC CL(L)'?????????????????????????????????????????????????????????????*
               ?????????'
*---------------------------------------------------------------------*
LOGOEND  EQU   *                  THIS LABEL MUST FOLLOW YOUR LOGO TEXT
```

Figure 48. IESELOGO Skeleton, Part 2 of 3

The LOGOBA label **must** be before the logo text and should **not** be changed. In the *DC CL(L) statements:

- Replace the question marks (**?**) with your own logo design.
- Replace the asterisks (*) in column 1 with blanks for each DC statement that you use.

You can replace the question marks with text, block letters, or blank lines. Do **not** change the format of the skeleton; that is, the beginning and ending columns of the lines. The format follows the rules of Assembler language coding. If you change the format, there may be assembly errors or the sign-on program itself may not work correctly.

The LOGOEND label **must** follow the logo text. To implement the change, proceed as follows:

1. Submit skeleton IESxLOGO (where 'x' refers to the language you are using: E or J) for processing.
2. Restart CICS.

Setting a Limit for Invalid Sign-On Attempts

A user without a valid password can try to gain access to the system again and again. To limit the chances of gaining unauthorized access through this trial-and-error method, you can restrict the number of unsuccessful sign-on attempts. If the limit is reached, z/VSE revokes the user-ID to prevent further use. System administrator authority is required to use the dialog for resetting a revoked user-ID. Refer to "Resetting a Revoked User-ID" on page 144 for further details.

You must use the BSTADMIN `PERFORM PASSWORD` command to set a limit for invalid sign-on attempts. For details, see "PERFORM | PF Command" on page 367.

Note: In Figure 49, the variable MAXNUMSO is retained for compatibility reasons only. Any modification of this variable will have no effect!

```
LOGOLINS EQU   (LOGOEND-LOGOBA)/L  NUMBER OF LOGO TEXT LINES
         SPACE 3
           .
MAXNUMSO DC    H'5'               MAX. NUMBER INVALID SIGNON ATTEMPTS
         SPACE 1
           .
SIGNONH DC     C'Y'               SIGNON-HERE CAPABILITY
         SPACE 1
           .
         SPACE 1
           .
ESCAPESW DC    C'N'               ESCAPE SWITCH
         SPACE 2
           .
UCESCSTR DC    CL4'    '           THIS IS THE CHARACTER STRING THE
*              TERMINAL OPERATOR SHOULD KEY INTO  . . .
*
*
           .
MCESCSTR DC    CL4'    '           THIS IS THE CHARACTER STRING THE
*              TERMINAL OPERATOR SHOULD KEY INTO  . . .
           .
         SPACE 2
           .
CUUOFFS  DC    H'1'               CUU OFFSET (0-5) IN NETNAME
         SPACE 2
*----------------------------------------------------------------------*
         END   ,                   NOTE --> NO LABEL ON END CARD
/*
// EXEC LNKEDT
/*
/&
* $$ EOJ
```

Figure 49. IESELOGO Skeleton, Part 3 of 3

To implement the change, proceed as follows:

1. Submit skeleton IESELOGO for processing.
2. Restart CICS.

Controlling the Escape Facility

With line ESCAPESW in skeleton IESELOGO (shown in Figure 49 on page 188), you can control whether PF6 and PF9 are displayed on the sign-on panel. These two PF keys are used for the CICS escape facility. A user can press PF6 or PF9 to escape to native CICS without signing on to the Interactive Interface. In this case, security values for z/VSE and CICS are **not** established.

If you want to have the escape facility with the PF keys, change the N value in line ESCAPESW to **Y**.

If you have terminals which do not have PF6 and PF9, you can specify a 1 - 4 character string for the escape facility. You would do the following:

1. Change the N value in line ESCAPESW to **Y**.
2. In the following statements, insert a 1 - 4 character string between the single quotes (' '):
   - UCESCSTR DC CL4' '
   - MCESCSTR DC CL4' '

   UCESCSTR is for escape with uppercase (equivalent to PF6). MCESCSTR is for escape with mixed case (equivalent to PF9). Transaction IDs are translated in uppercase (UCTRANID).

   You can use special characters, but you cannot specify lowercase letters. If your character string is shorter than four characters, it must be padded with blanks on the right.

After filing the skeleton, perform the following steps to implement your changes:

1. Submit the completed skeleton for processing. The skeleton assembles and catalogs the logo module.
2. Check for any errors in the assembly. Correct any errors before proceeding.
3. Restart CICS.

Specifying cuu in Netname

According to the naming convention for non-SNA terminals, the generated VTAM netname contains the **cuu** (channel and unit number) in position 2-4:

x**cuu**xxxx

This is needed by z/VSE when running under VM to offer the PF3 function key (RETURN TO VM) on the *z/VSE Online* panel. If you use your own naming convention, you can specify the position of cuu using the IESxLOGO (where 'x' refers to the language you are using: E, G, S, or J) skeleton.

Perform the following steps:

1. Locate line CUUOFFS in skeleton IESxLOGO (shown in Figure 49 on page 188). In this line, change the default offset 1 (H'1') to the offset at which *cuu* starts in *netname*.
2. Submit skeleton IESxLOGO for processing.
3. Restart CICS.

Configure 'Logon Here'

If a user wants to sign-on to a z/VSE system but is already signed-on at another terminal, the message

```
USER ID 'xxxxx    ' IS ALREADY IN USE AT TERMINAL 'nnnn'
```

is displayed and the entry panel provides a PF key (PF12) with the function LOGON HERE.

LOGON HERE specifies that if this user-ID is already logged on, it should be disconnected from its current terminal and reconnected at the terminal where this logon is requested.

If you want to disable this function, change the statement SIGNONH in skeleton IESxLOGO (where 'x' refers to the language you are using: E or J) to **N** (No):

```
:
SIGNONH DC     C'N'              SIGNON-HERE CAPABILITY
:
```

```
IESADMSO1                         z/VSE ONLINE
    5609-ZV5 and Other Materials (C) Copyright IBM Corp. 2011 and other dates



                                ++
                               ++    VV    VV    SSSSS    EEEEEEE
                zzzzzz        ++     VV    VV   SSSSSSS   EEEEEEE
                zzzzz        ++      VV    VV  SS         EE
                  zz        ++       VV    VV   SSSSSS    EEEEEE
                 zz       ++         VV    VV    SSSSSS   EEEEEE
                zzzzzz   ++           VV  VV         SS  EE
               zzzzzzz  ++             VVVV     SSSSSSS   EEEEEEE
                                        VV       SSSSS    EEEEEEE

        Your terminal is A001 and its name in the network is D3010001
        Today is 10/22/2011   To sign on to DBDCCICS  --  enter your:

       USER-ID........ ________   The name by which the system knows you.
       PASSWORD.......            Your personal access code.

PF1=HELP      2=TUTORIAL   3=TO VM      4=REMOTE APPLICATIONS     6=ESCAPE(U)
                           9=Escape(m) 10=NEW PASSWORD
```

Figure 50. Logon Here Panel

Recovering Terminal Connections

When a terminal is switched off without signing off or loses its VTAM connection to the CPU, the Interactive Interface is unaware of it. The user-ID and the related control blocks are not freed. This prevents a signing on with the same user-ID from another terminal.

To help avoid such situations, z/VSE provides the program IESCLEAN. This program frees all the resources related to specific user-ID and performs a sign off from the Interactive Interface. A user can then sign on again with the freed user-ID from a different terminal but without the reconnect facility available.

Implementing Program IESCLEAN

z/VSE uses the CICS node error program DFHZNEP to provide a link to program IESCLEAN. Refer to the manual *CICS Customization Guide* for details on node error processing.

In VSE/ICCF library 59, z/VSE provides three sample programs to modify program IESCLEAN:

IESZNEP

IESZNEPS

IESZNEPX

Program IESCLEAN is invoked if one of the following CICS error codes occur:

10 Node not activated

49 Node session terminated

57 Terminal released by master terminal operator

61 (with Sense=0831) POWER OFF at SNA terminals.

A7 Bracket error

D1 Node unrecoverable

You can add or remove error codes as needed by your installation.

You can use the sample programs provided as follows:

- **IESZNEP** includes a complete node error program (NEP) DFHZNEP which is active by default. By changing and submitting skeleton IESZNEP, you can replace NEP DFHZNEP.
- If you currently operate with a user supplied node error program, you can either use sample IESZNEPS or sample IESZNEPX:
  - Sample **IESZNEPS**:

    This sample assumes that you use the CICS sample node error program (DFHNEPS macro).

    IESZNEPS contains an error processor to be included into an existing sample NEP with the statement: COPY IESZNEPS. When being submitted, the sample creates member IESZNEPS.A and stores it in library PRD2.CONFIG. The group number assigned to the error processor must be unique.
  - Sample **IESZNEPX**:

    This sample assumes that you use your own user written node error program.

    IESZNEPX contains an error processor to be included into an existing user written NEP with the statement: COPY IESZNEPX. When being submitted, the sample creates member IESZNEPX.A and stores it in library PRD2.CONFIG.

To modify program IESCLEAN, the following steps are required:

1. Select one of the sample programs according to the needs of your installation.
2. Submit the sample program for processing.
3. Prepare your own node error program (if necessary).
4. Assemble your node error program (if necessary).
5. Restart CICS.

Signing On to Different CICS System

The sign-on exit **IESEXIT** (skeleton SKEXIT1) allows a user to sign on to different CICS subsystems in the same z/VSE system with the same user-ID and password and receive different initial selection or application panels.

The initial selection panel or application name is defined for a given user in the user profile. In a z/VSE system with several CICS subsystems, the users are defined in the same control file meaning that for all CICS systems the same initial selection panel or application name is selected for a user at sign on. By means of this exit, you are able to specify an initial selection panel or application name depending on different CICS subsystems which use the same control file.

The program needs to be defined in the CICS CSD file (CEDA). Control must be given back via CICS RETURN.

The following parameters are provided through the CICS communication area (COMMAREA):

```
* PARAMETER LIST FOR SIGNON EXIT
PARMX    DS    0H                 PARAMETER LIST FOR EXIT1
APPLIDX  DS    CL8                APPLICATION ID (padded with X'40's)
USERIDX  DS    CL8                USERID (padded with X'00's)
CURRSELX DS    CL8                CURRENT SELECTION/APPLICATION
NEWSELX  DS    CL8                NEW SELECTION/APPLICATION
CURRTYPX DS    CL1                CURRENT TYPE SELECTION/APPLICATION
NEWTYPX  DS    CL1                NEW TYPE SELECTION/APPLICATION
PARMLENX EQU   *-PARMX            LENGTH OF PARAMETER AREA
```

Following is an example of how to retrieve a parameter in IESEXIT1:

```
         OC    EIBCALEN,EIBCALEN   COMMUNICATION AREA PROVIDED ?
         BZ    RETURN              NO, RETURN
         L     R1,DFHEICAP         GET POINTER TO COMMUNICATION AREA
         USING PARMX,R1
```

The parameters have the following meaning:

APPLIDX:
ID of CICS as DBDCCICS, PRODCICS or any other.

USERIDX:
User-ID as defined for z/VSE.

CURRSELX:
Name of initial selection panel or application as defined in the user profile.

NEWSELX:
Name of initial selection panel or application to be used for this sign on. These names must be defined to the system.

CURRTYPX:
Current type which can be selection (S) or application (A).

NEWTYPX:
New type which can be selection (S) or application (A).

You should catalog program IESEXIT into sublibrary PRD2.CONFIG.

Tailoring Console Definitions

z/VSE uses predefined *console definitions* which you may modify if required. Note that these definitions are used for **all** active consoles at your z/VSE installation. Individual console tailoring is not possible.

Consult also the manual *z/VSE Planning* under "Console Support" before you start tailoring console definitions. The manual has a chapter which introduces and provides an overview of the console support provided.

Console definitions define:
- Panel data (fixed text displayed on the panel)
- PF key settings
- Local messages (related to the *Console* dialog)

The console definitions are shipped in source format as well as in phase format. The source of the console definitions is available as member IJBxDEF.Z, the corresponding phase is $IJBxDEF.PHASE. x defines one of the following languages:

E = English

J = Japanese

The basic version of the system contains:

IJBEDEF.Z and **$IJBEDEF.PHASE** (English)

The following multicultural support version (source and phase) is also available:

IJBJDEF.Z and $IJBJDEF.PHASE (Japanese)

$IJBEDEF.PHASE is always part of the system independent of the language ordered. Member IJBxDEF.Z contains definitions for:
- Panel data (either in English or in a national language).
- PF keys (either in English or in a national language).
- Local messages consisting of two entries each (the first entry specifies the local message text in English and the second one in a national language).

In order to modify console definitions, you must proceed as follows:
1. Edit the source in corresponding member IJBxDEF.Z.
2. Copy the member first from IJSYSRS.SYSLIB to your primary VSE/ICCF library. Use the VSE/ICCF command LIBRP.
3. Assemble the edited member and catalog the resulting object module as phase $IJBxDEF.PHASE into the system library IJSYSRS.SYSLIB.

The input required to create phase $IJBxDEF consists of multiple invocations of the macro IJBDEF as shown in "Member IJBEDEF.Z" on page 196.

Using Macro IJBDEF

The IJBDEF macro can be used to create entries for:

- Panel data
- PF key settings
- Local messages.

The macro does extensive validity and syntax checking for each macro definition. Every invocation of macro IJBDEF creates one entry.

Note: When reading this macro description refer to "Member IJBEDEF.Z" on page 196 for easier understanding. The figure includes coded examples of the different types of entries.

The IJBDEF macro has the following general format:

```
label    IJBDEF  parameters
```

The parameters are positional. Code a comma for any parameter that you omit (except for the last one).

The first parameter determines the type of definition:

PANEL
Defines a panel data table entry

PFKEY
Defines a PF key table entry

MSG Defines a local message text table entry

GEN Starts the table generation.

Note: The IJBDEF macro has no default for the first parameter.

Defining Panel Data

For defining panel data, the macro has the following format:

```
IJBDEF  PANEL,type,,,'edata','ndata'
```

One invocation defines an entry in the panel data table which consists of 30 entries allocated in IJBxDEF. The meaning of each parameter is as follows:

type Specifies the type of the particular panel data table entry. The constants to be specified for type are predefined and **cannot** be changed.

'edata' A string of data in English, enclosed in apostrophes, to be displayed at a defined position on the panel.

'ndata' A string of data in a national language, enclosed in apostrophes, to be displayed at a defined position on the panel.

Note: In case 'ndata' is identical with 'edata', the specification of '=' is sufficient for 'ndata'.

The maximum data length that can be displayed varies with the panel type. The following list shows the maximum number of characters allowed for each panel type:

```
TPANL     10
TTITL     20
TSYST      8
TAPPL      8
```

```
TTIME       6
TPAGE       8
TTIMX       8
TACMD       3
TCMD      126
TDMSG      78
TFILT       8
TFILTR      8
TACTM       8
TFILX      17
THOLD       8
TNOHLD      8
TDIRC       6
TNUM1       8
TPAUSE      6
TNUM2       4
TIMSG       8
TMESG       8
TSUSP       8
TMODE       6
TMODXC     14
TMODXCM    14
TMODXCD    14
TMODXR     14
TMODXE     14
TMODXH     14
```

Defining PF Key Settings

For defining PF key settings, the macro has the following format:

- **`IJBDEF  PFKEY,n,m,'etext','ntext','command'`**

The meaning of each parameter is as follows:

n
PF key number in the range 1-12, or ENTER, or CLEAR

m
One of the modes under which the PF key is valid:

C console mode

R redisplay mode

E explain mode

H help mode

'etext'
The descriptive text in English to be displayed on the panel, with a maximum length of 8 characters, enclosed in apostrophes.

'ntext'
The descriptive text in a national language to be displayed on the panel, with a maximum length of 8 characters, enclosed in apostrophes.

Note:

1. The total length of the descriptive text for the PF keys (1-12) per mode ('etext' mode, 'ntext' mode) must not exceed 80 characters. For each single specification one extra space must be counted.
2. In case 'ntext' is identical with 'etext', the specification of '=' is sufficient for 'ntext'. See also “Member IJBEDEF.Z” on page 196.

'command'
A string of data being a local command, a z/VSE command or any other data such as a reply. It must be enclosed in apostrophes and may consist of up to 10 substrings separated from each other by 2 apostrophes.

Defining a Local Message Text

This function can be used when writing your own console applications. For defining the text of a local message, the macro has the following format:

- **Mn   IJBDEF MSG,'etext','ntext'**

The meaning of each parameter is as follows:

Mn
n is a number between 1 and 256, each of which specifies a particular message table entry. The number **cannot** be changed for system-supplied local messages. These are the numbers M1 through M80 which are reserved for use by z/VSE.

'etext'
Text of local message in English enclosed in apostrophes.

'ntext'
Text of local message in a national language enclosed in apostrophes.

Note: In case 'ntext' is identical with 'etext', the specification of **'='** is sufficient for 'ntext'. See also "Member IJBEDEF.Z."

Starting Table Generation

For starting the generation of console definition tables, the macro has the following format:

```
IJBDEF GEN,phase
```

IJBDEF GEN must be the last specification in a sequence of IJBDEF specifications. The variable "phase" specifies the name of the phase to be generated:

$IJBEDEF
(English, which is the default)

$IJBJDEF
(Japanese)

Member IJBEDEF.Z

The example below shows member IJBEDEF.Z stored in system library IJSYSRS.SYSLIB. Note that the contents of IJBEDEF.Z as shown reflects the status at the time the manual was printed. The actual contents shipped may differ from the one shown in the example below.

The following job stream example can be used to reassemble member IJBEDEF.Z if tailoring is required:

```
* $$ JOB JNM=IJBEDEF,DISP=D,CLASS=Z,LDEST=(*,user)
* $$ LST CLASS=X,DISP=D
// JOB CATAL IJBEDEF CONSOLE DEFINITIONS
// OPTION NOSYSDUMP
// ID USER=FORSEC,PWD=FORSEC
// LIBDEF *,CATALOG=IJSYSRS.SYSLIB
// OPTION ERRS,SXREF,SYM,NODECK,CATAL
// EXEC ASMA90,SIZE=(ASMA90,64K)
          .
          .
    member IJBEDEF.Z
          .
          .
/*
// EXEC LNKEDT,PARM='MSHP'
/&
* $$ EOJ
```

```
***********************************************************************
*        PANEL DATA DEFINITIONS                                       *
***********************************************************************
```

```
         IJBDEF PANEL,TSYST,,,'SYSTEM:','='
         IJBDEF PANEL,TSYSX,,,' ','='
         IJBDEF PANEL,TTITL,,,'      z/VSE 4.3     ','='
         IJBDEF PANEL,TUSER,,,'USER:','='
         IJBDEF PANEL,TUSEX,,,' ','='
         IJBDEF PANEL,TTIME,,,'TIME:','='
         IJBDEF PANEL,TTIMX,,,' ','='
         IJBDEF PANEL,TDISP,,,'TURBO','='
         IJBDEF PANEL,TDISPX,,,' ','='
         IJBDEF PANEL,TACMD,,,'==>','='
         IJBDEF PANEL,TCMD,,,' ','='
         IJBDEF PANEL,TDMSG,,,' ','='
         IJBDEF PANEL,TFILT,,,' ','='
         IJBDEF PANEL,TFILTR,,,'FILTER:','='
         IJBDEF PANEL,TACTM,,,'ACT_MSG:','='
         IJBDEF PANEL,TFILX,,,' ','='
         IJBDEF PANEL,THOLD,,,'HOLD','='
         IJBDEF PANEL,THRUN,,,'HOLDRUN','='
         IJBDEF PANEL,TNOHLD,,,'NOHOLD','='
         IJBDEF PANEL,TDIRC,,,' ','='
         IJBDEF PANEL,TPAUS,,,'PAUSE:','='
         IJBDEF PANEL,TPAUSX,,,' ','='
         IJBDEF PANEL,TSCRL,,,'SCROLL:','='
         IJBDEF PANEL,TSCRLX,,,' ','='
         IJBDEF PANEL,TIMSG,,,' ','='
         IJBDEF PANEL,TMESG,,,'MESSAGE','='
         IJBDEF PANEL,TSUSP,,,'SUSPEND','='
         IJBDEF PANEL,TMODE,,,'MODE:','='
         IJBDEF PANEL,TMODXC,,,'CONSOLE','='
         IJBDEF PANEL,TMODXCM,,,'CONSOLE ..MORE','='
         IJBDEF PANEL,TMODXCD,,,'CONSOLE ..HOLD','='
         IJBDEF PANEL,TMODXR,,,'REDISPLAY','='
         IJBDEF PANEL,TMODXE,,,'EXPLANATION','='
         IJBDEF PANEL,TMODXH,,,'HELP','='
*********************************************************************
*        PF KEY DEFINITIONS                                         *
*********************************************************************
         IJBDEF PFKEY,1,C,'1=HLP','=','%HELP'
         IJBDEF PFKEY,2,C,'2=CPY','=','%COPY ''?CL'
         IJBDEF PFKEY,3,C,'3=END','=','%END'
         IJBDEF PFKEY,4,C,'4=RTN','=','%RETURN'
         IJBDEF PFKEY,5,C,'5=DEL','=','%DELETE ''?CL'',''?IN'
         IJBDEF PFKEY,6,C,'6=DELS','=','%DELETE ''?CL'',''SYSTEM'
         IJBDEF PFKEY,7,C,'7=RED','=','%REDISPLAY ''?IN'
         IJBDEF PFKEY,8,C,'8=CONT','=','%CONTINUE'
         IJBDEF PFKEY,9,C,'9=EXPL','=','%EXPLAIN ''?TK'
         IJBDEF PFKEY,10,C,'10=HLD','=','%CHANGE ''HOLD'
         IJBDEF PFKEY,11,C,'11=PCUU','=','%EXCUU ''?CL'
         IJBDEF PFKEY,12,C,'12=RTRV','=','%RETRIEVE'
         IJBDEF PFKEY,ENTER,C,'INPUT','=','?IN'
         IJBDEF PFKEY,CLEAR,C,'CLEAR','=','%CLEAR'
*         IJBDEF PFKEY,1,R,'1=HLP','=','%HELP'
         IJBDEF PFKEY,2,R,'2=CPY','=','%COPY ''?CL'
         IJBDEF PFKEY,3,R,'3=END','=','%REDISPLAY E'
         IJBDEF PFKEY,4,R,'      ','=',''
         IJBDEF PFKEY,5,R,'     ','=',''
         IJBDEF PFKEY,6,R,'6=CNCL','=','%REDI C'
         IJBDEF PFKEY,7,R,'7=BWD','=','%REDI ''?CL'';''B,''?IN'
         IJBDEF PFKEY,8,R,'8=FWD','=','%REDI ''?CL'';''F,''?IN'
         IJBDEF PFKEY,9,R,'9=EXPL','=','%EXPLAIN ''?TK'
         IJBDEF PFKEY,10,R,'10=INP','=','?IN'
         IJBDEF PFKEY,11,R,'11=PCUU','=','%EXCUU ''?CL'
         IJBDEF PFKEY,12,R,'12=INFO','=','%CHANGE INFO'
         IJBDEF PFKEY,ENTER,R,'REDISPLY','=','%REDI ''?CL'';''?IN'
         IJBDEF PFKEY,CLEAR,R,'CLEAR','=','%CLEAR'
*
         IJBDEF PFKEY,1,E,'1=HLP','=','%HELP'
         IJBDEF PFKEY,2,E,'2=CPY','=','%COPY ''?CL'
         IJBDEF PFKEY,3,E,'3=END','=','%END'
         IJBDEF PFKEY,4,E,'      ','=',''
         IJBDEF PFKEY,5,E,'     ','=',''
```

```
         IJBDEF PFKEY,6,E,'      ','=',''
         IJBDEF PFKEY,7,E,'7=BWD','=','%BACKWARD'
         IJBDEF PFKEY,8,E,'8=FWD','=','%FORWARD'
         IJBDEF PFKEY,9,E,'9=EXPL','=','%EXPLAIN ''?TK'
         IJBDEF PFKEY,10,E,'10=INP','=','?IN'
         IJBDEF PFKEY,ENTER,E,'EXPLAIN','=','%EXPLAIN ''?TK'
         IJBDEF PFKEY,CLEAR,E,'CLEAR','=','%CLEAR'
*
         IJBDEF PFKEY,1,H,'1=HLP','=','%HELP'
         IJBDEF PFKEY,2,H,'     ','=',''
         IJBDEF PFKEY,3,H,'3=END','=','%END'
         IJBDEF PFKEY,4,H,'      ','=',''
         IJBDEF PFKEY,5,H,'     ','=',''
         IJBDEF PFKEY,6,H,'      ','=',''
         IJBDEF PFKEY,7,H,'7=BWD','=','%BACKWARD'
         IJBDEF PFKEY,8,H,'8=FWD','=','%FORWARD'
         IJBDEF PFKEY,9,H,'      ','=',''
         IJBDEF PFKEY,10,H,'10=INP','=','?IN'
         IJBDEF PFKEY,ENTER,H,'HELP','=','%HELP'
         IJBDEF PFKEY,CLEAR,H,'CLEAR','=','%CLEAR'
*
***********************************************************************
*        CONSOLE ROUTER LOCAL MESSAGE DEFINITIONS, RANGE M1 - M20    *
***********************************************************************
M1       IJBDEF MSG,'0D18I INVALID INPUT','='
M2       IJBDEF MSG,'0D14I COMMAND IGNORED','='
M3       IJBDEF MSG,'0D11I INVALID REPLY-ID','='
M4       IJBDEF MSG,'0D10I COMMAND/REPLY NOT AUTHORIZED','='
M5       IJBDEF MSG,'0D19I ATTENTION ROUTINE NOT ACTIVE','='
M6       IJBDEF MSG,'0D24I REDISPLAY PROCESSOR NOT ACTIVE','='
M7       IJBDEF MSG,'0D21I INPUT REJECTED BY EXTERNAL EXIT','='
M8       IJBDEF MSG,'0D91I INPUT NOT ACCEPTED DUE TO REMOTE OPERATING M-
               ODE','='
M9       IJBDEF MSG,'0D92I REDISPLAY MODE ALREADY ACTIVE FOR OTHER USER-
               ','='
M10      IJBDEF MSG,'0D93I COMMAND NOT ACCEPTED','='
***********************************************************************
*        HARD COPY FILE LOCAL MESSAGE DEFINITIONS, RANGE M21 - M40   *
***********************************************************************
M21      IJBDEF MSG,'0D26E I/O ERROR ON HARD COPY FILE','='
M22      IJBDEF MSG,'0D29E INCORRECT LENGTH DURING I/O FOR HARD COPY FI-
               LE','='
M23      IJBDEF MSG,'0D51I EXTENT FAILED','='
M24      IJBDEF MSG,'0D52I GETVIS FAILED','='
M25      IJBDEF MSG,'0D56E INCONSISTENT STATE DURING HARD COPY FILE PRO-
               CESSING','='
M26      IJBDEF MSG,'0D80I INVALID REDISPLAY COMMAND','='
M27      IJBDEF MSG,'0D81I A TRAILING COMMA IS NOT VALID','='
M28      IJBDEF MSG,'0D82I FUNCTION HOLD AND A SUBFILTER ARE NOT COMPAT-
               IBLE','='
M29      IJBDEF MSG,'0D83I REDISPLAY COMMAND IS CANCELLED','='
M30      IJBDEF MSG,'0D84I REDISPLAY MODE IS TERMINATED','='
M31      IJBDEF MSG,'0D85I ACTION CANCEL DOES NOT ALLOW OTHER OPERANDS'-
               ,'='
M32      IJBDEF MSG,'0D86I NO REDISPLAY COMMAND/MODE IS ACTIVE, COMMAND-
                IGNORED','='
M33      IJBDEF MSG,'0D22I INSUFFICIENT GETVIS FOR REQUESTED FUNCTION',-
               '='
*
***********************************************************************
*        CONSOLE APPLICATION LOCAL MESSAGES, RANGE M41 - M80         *
***********************************************************************
M41      IJBDEF MSG,'0D61I PRESS CONTINUE TO RESUME','='
M42      IJBDEF MSG,'0D62I SCREEN IS FULL WITH HOLD MESSAGES (SET ACT_M-
                SG TO NOHOLD)','='
M43      IJBDEF MSG,'0D63I PF/PA KEY NOT DEFINED','='
M44      IJBDEF MSG,'0D64I COMMAND NOT ALLOWED IN THIS MODE','='
M45      IJBDEF MSG,'0D65I COMMAND NOT ALLOWED FROM THE INPUT LINE','='
M46      IJBDEF MSG,'0D66I INVALID CURSOR POSITION/LINE NUMBER FOR THIS-
                COMMAND','='
M47      IJBDEF MSG,'0D67I COMMAND INVALID','='
```

```
M48      IJBDEF MSG,'0D68I OPERAND INVALID','='
M49      IJBDEF MSG,'0D69I PRESS END TO RESUME','='
M50      IJBDEF MSG,'0D70I NO MORE EXPLAIN/HELP DATA AVAILABLE','='
M51      IJBDEF MSG,'0D71I NO EXPLAIN/HELP DATA FOUND','='
M52      IJBDEF MSG,'0D72I TRY AGAIN LATER','='
M53      IJBDEF MSG,'0D73I CONSOLE DEACTIVATED, HIT ENTER TO RESUME','=-
               '
M54      IJBDEF MSG,'0D74I EXPLAIN FILE ACCESS FAILURE','='
M55      IJBDEF MSG,'0D75I EXPLAIN SUPPORT NOT ACTIVE','='
M56      IJBDEF MSG,'0D76I EXPANSION FAILURE','='
M57      IJBDEF MSG,'0D77I DICTIONARY COULD NOT BE LOADED','='
*
 *********************************************************************
*        CONSOLE PARAMETER SETTINGS                                  *
*********************************************************************
*
         IJBDEF DEFAULT,HOLD,RUN     (YES/RUN/NO)  DEFAULT=RUN  PN78356
         IJBDEF DEFAULT,ALARM,YES    (YES/NO)             DEFAULT=YES
         IJBDEF DEFAULT,INFO,NONE    (NONE/TSTAMP/USERID) DEFAULT=NONE
         IJBDEF DEFAULT,PAUSE,1      (00 GE NN LE 99)     DEFAULT=1
         IJBDEF DEFAULT,SCROLL,1     (0 GE N LE 9)        DEFAULT=1
*
*********************************************************************
*        GENERATE THE TABLES                                         *
*********************************************************************
         IJBDEF GEN,$IJBEDEF
```

Chapter 16. ZONE Specifications and Daylight Saving Time

With the *Tailor IPL Procedure* dialog, you can add or modify zone specifications to define time zones, and to be able to switch between standard and daylight saving time (summertime) without changing the IPL startup procedure each time.

To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

4 (Hardware Configuration and IPL)

2 (Tailor IPL Procedure)

You get a list of the IPL parameters that can be modified. Select the ZONE parameter by entering 1 next to it.

```
TAS$MAS2                  TAILOR IPL PROCEDURE

Enter the required data and press ENTER.

                                               IPL procedure = $

To modify one or more of the following IPL parameters, place a  1 next to it.

     _     Supervisor  Modify console, supervisor- and storage option
     _     SYS         Modify SYS command parameters
     _     DLA         Delete label area definition
     _     DPD         Modify page data set definition
     _     DLF         Modify lock file definition
     _     DEF         Modify recorder file and catalog assignment
     1     ZONE        Modify ZONE specifications
     _     APPC/VM     Modify VSE APPC/VM specification
     _     SVA         Modify shared virtual area definition



PF1=HELP      2=REDISPLAY  3=END                  5=PROCESS
```

Figure 51. Tailor IPL Procedure Dialog

On the *ZONE SPECIFICATION* panel, you can define the zone direction and the zone hours, or you can specify a zone id. These two possibilities define static values in your IPL procedure.

```
TAS$ICM6          TAILOR IPL PROCEDURE: ZONE SPECIFICATION

 Enter the required data and press ENTER.



ZONE DIRECTION............... _           Direction to you from Greenwich
                                          England (1 = East   2 = West)

ZONE HOURS................... __          Hours to you from Greenwich, England
                                          (Two digits between 00 and 23)

ZONE ID...................... CES         Time Zone Definition.






 PF1=HELP      2=REDISPLAY  3=END                     5=PROCESS
               8=FORWARD
```

Figure 52. Panel for Modifying Zone Specifications

Note: If you want to use the possibility to switch between times, make sure that you have no static ZONE definition specified on the panel shown above (TAS$ICM6). The switching will be ignored when the IPL procedure contains a static SET ZONE definition.

To add or change the Time Zone Definition (ZONE ID), delete the values shown on the ZONE DIRECTION and ZONE HOURS lines and press PF8 to define the characteristics of the ZONDEF specifications (see Figure 53 on page 203). If your definitions are already completed on this panel, press PF5 to process the entered values.

Note that if you do not want to have a SET ZONE command in your IPL procedure, (for example, if you are running under VM), you need to delete all the values entered on panel TAS$ICM6.

ZONEDEF Specification

Use the *Zonedef Specification* dialog to define system time zones according to their difference from Greenwich Mean Time (GMT).

```
TAS$ICMB          TAILOR IPL PROCEDURE: ZONEDEF SPECIFICATION

Enter the required data and press ENTER.

OPTIONS: 1 = ADD   2 = ALTER    5 = DELETE

 OPT       ZONE ID       ZONE DIRECTION    ZONE HOURS

  _         CES           1                  00
  _         CET           1                  01
  _         EST           2                  05
  _         CST           2                  06
  _         EDT           2                  04
  _         CDT           2                  05
  _         ___           _                  __
  _         ___           _                  __
  _         ___           _                  __
  _         ___           _                  __


PF1=HELP       2=REDISPLAY  3=END                    5=PROCESS
PF7=BACKWARD   8=FORWARD
```

Figure 53. ZONEDEF Specification Panel

The following values can be specified:

ZONE ID
Enter a three character name for this ZONE definition. You can select any name you want. It refers to a specific zone value. You can define up to 10 new ZONE IDs. The examples shown, are the names of the official time zones division. For example:

- CES for Central European Summertime
- CET for Central European Standard Time
- EST for Eastern Standard Time
- EDT for Eastern Daylight Saving Time
- CST for Central Standard Time
- CDT for Central Daylight Saving Time

ZONE DIRECTION
Enter the direction to you from Greenwich, England. 1 = East, 2 = West. For example, define 2 for USA which is to the west of Greenwich.

ZONE HOURS
Enter the hours to you from Greenwich, England. Enter two digits between 00 and 23. Only zone hours, no zone minutes are supported.

Press PF8 to get the *ZONEBDY Specification* panel, or press PF5 to process the entered values.

ZONEBDY Specification

```
TAS$ICMC         TAILOR IPL PROCEDURE: ZONEBDY SPECIFICATION

Enter the required data and press ENTER.

OPTIONS: 1 = ADD               5 = DELETE

 OPT      BEGIN DATE        BEGIN TIME    ZONE ID
          mmddyyyy          hhmmss

  _       04012004          000001          CES
  _       10012004          000001          CET
  _       04012005          000001          CES
  _       10012005          000001          CET
  _       ________          ______          ___
  _       ________          ______          ___
  _       ________          ______          ___
  _       ________          ______          ___
  _       ________          ______          ___
  _       ________          ______          ___

PF1=HELP       2=REDISPLAY  3=END                     5=PROCESS
PF7=BACKWARD
```

Figure 54. ZONEBDY Specification Panel

With the *ZONEBDY Specification* panel you can define the date and time when z/VSE should begin to use a given time zone. Usually, you use these definitions to switch between standard and daylight saving local times.

BEGIN DATE
: Enter the date, in the format mmddyyyy, on which z/VSE should begin using a given time zone. You can define up to 20 dates.

BEGIN TIME
: Enter the local time in the format hhmmss, on which z/VSE should begin using a given time zone.

ZONE ID
: Enter a three character time zone definition you specified before on the *ZONEDEF Specification* panel.

Press PF5 to process the data, or press PF7 to re-display the entered values.

Note that you have to IPL the system in order to switch to the new time zone.

The statements created by the dialog are documented, together with examples, in the manual *z/VSE System Control Statements* under “SET ZONEDEF” and “SET ZONEBDY”.

Part 2. FILES AND TAPES

Chapter 17. Managing VSE/VSAM Files and Catalogs

Related Topic:

For details of how to ...	Refer to ...
backup VSE/VSAM files and datasets	"Backing Up and Restoring Data" in the *z/VSE Operation*

Overview of File and Catalog Management Dialogs

For the model system administrator (SYSA), the Interactive Interface offers the *File and Catalog Management* dialog. The user profile of SYSA allows the definition, deletion, and processing of VSE/VSAM files and user catalogs.

For the model programmer (PROG), the Interactive Interface offers the *File Management* dialog. The user profile of PROG allows the definition, deletion, and processing of VSE/VSAM files, **but not** of user catalogs.

Some dialogs process the information immediately. Others create a job. You can submit the job for processing or store it as a VSE/ICCF library member in your default VSE/ICCF primary library. Column positions of some parameters in these jobs are essential. Be careful not to change any parameter positions when looking at a job stored as a VSE/ICCF library member.

To access the dialog, start with the Administrator *z/VSE Function Selection* panel. As administrator (SYSA), select:

2 (Resource Definition)
2 (File and Catalog Management)

The programmer (PROG) must choose selection **6** of the *z/VSE Function Selection* panel. The default synonym is the same as for the administrator. Below, the selections for the administrator are shown. For the default programmer (PROG), only the first four selections are displayed.

The panel displayed for the administrator offers six selections:

1 (Display or Process a File)
2 (Define a New File)
3 (Define a Library)
4 (Define an Alternate Index or Name)
5 (Display or Process a Catalog, Space)
6 (Define a New User Catalog)

Note: For selections 1 through 4, a default catalog name is displayed as defined in the user profile. This name can be changed on the panel.

The dialog authorization for the administrator and programmer is based on a general authorization concept. You can have the authority to:

- Define/delete files.
- Process catalogs.

This authorization is part of the user profile. When you define a user profile, you specify whether the user has the authority to define/delete files and the authority to process catalogs. This does not depend on whether the user is an administrator or programmer.

Table 3 illustrates the selections which the panel displays and which you can access based on the authorization you have. If the panel displays a selection which you cannot access and you enter that selection number, the dialog displays an error message.

Table 3. Relationship Between VSE/VSAM Authorization in User Profile and Dialog Selections

Define/Delete Files	Process Catalogs	Selections Displayed and Accessible
YES	YES	All selections displayed. All selections can be accessed.
YES	NO	Selections 1 - 4 displayed. Selections 1 - 4 can be accessed.
NO	YES	Selections 1 - 6 displayed. Only selections 1, 5, and 6 can be accessed.
NO	NO	Selections 1 - 4 displayed. Only selection 1 can be accessed.

Displaying or Processing a VSE/VSAM File

The dialog *Display or Process a File* provides a FULIST that shows the file IDs and file names of all files in the specified catalog. Use **PF7** and **PF8** to scroll through the list. Use **PF9** to list a subset of the files by entering a prefix. With **PF2** you can refresh the panel display. To locate a particular file, enter the file ID in the LOCATE FILE ID field. The FILE TYPE field contains either the letter *A* or *B* to show the type of file.

```
A  -  Alternate index
B  -  Base file
```

The options you can choose are at the top of the FULIST. Enter an option number in the OPT column to the left of the file ID you want to process. The options available are listed below. They are described in detail in the manual *VSE/ESA Programming and Workstation Guide* in the topic “Display or Process a File”.

1 (Show)
Displays details about the characteristics of a VSE/VSAM file or alternate index.

2 (Sort)
Sorts a VSE/VSAM file. You must have:

- The z/VSE optional program DFSORT/VSE or a compatible program installed.
- Both input and output files already defined in the catalog.

3 (Print)
Prints one or more records of a VSE/VSAM file on the system printer.

4 (Copy)
Copies all or part of a file to another file. You can also copy a VSE/VSAM file to and from tape.

5 (Delete)
Deletes a VSE/VSAM file or an alternate index and name. You **cannot** delete system files.

6 (Verify)
Compares the end-of-file information in the catalog with the end-of-file indicator(s) in the file. If the information does not agree, the catalog information is corrected. You **cannot** verify an alternate index.

7 (Load)
Loads data from a VSE/ICCF library member into a base file, or loads an alternate index from a base file.

Defining a New VSE/VSAM File

With the dialog *Define a New File* you can create a new VSE/VSAM file in the catalog specified. To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

2 (File and Catalog Management)

2 (Define a New File)

If you want to specify a catalog and not use the default catalog, select Fast Path **22**.

The dialog displays several panels. You need to enter different file characteristics, depending on the type of file you are defining.

The dialog defines the new file and adds a label to the system standard label area with the file name, file ID, and catalog name of the new file. It also adds label information to the VSE/VSAM label procedure STDLABUP in IJSYSRS.SYSLIB.

You need the following information:

FILE ID
Enter up to five segments for the file ID. You cannot enter more than 38 characters, including dots.

FILE NAME
Enter **1 - 7** alphameric characters. The first character must be alphabetic.

FILE ORGANIZATION

1 - Non-keyed (ESDS)

2 - Keyed (KSDS)

3 - Numbered (RRDS)

4 - Numbered (VRDS)

5 - Sequential (SAM ESDS)

FILE ADDRESSABILITY

1 - Default addressing (32-bit RBAs).

2 - XXL addressing. Define a VSE/VSAM KSDS file larger than 4 GB.

FILE ACCESS
For the VSE/VSAM Share option, specify:

1 - Multiple Read OR Single Write

2 - Multiple Read AND Single Write

3 - Multiple Read AND Write (no integrity)

4 - Multiple Read AND Write (with integrity)

FILE USAGE

1 - Data file (NOREUSE)

2 - Work file (REUSE)

If the catalog owns space on more than one volume, at this point a list is displayed which shows these volumes and their device type. You can then select the volume you want the primary space allocated on and the volume(s) you want the secondary space allocated on.

EXPIRATION DATE
Enter four digits for the year, and three digits for the day of the year (**YYYYDDD**).

ALLOCATION UNIT
Required only for **CKD disk devices**. Specify one of the following:

1 - Cylinder

2 - Track

For FBA devices, the allocation unit “Block” is used automatically.

PRIMARY and SECONDARY ALLOCATION
The number of allocation units for the initial (primary) and subsequent (secondary) allocations.

CONTROL INTERVAL SIZE
Specify the Control Interval size of the data component for all file types. For file types that have indexes, the value of the index component is calculated by VSAM.

AVERAGE and MAXIMUM RECORD SIZE
The average and maximum length of the data record, in bytes. For RRDS files, the average and maximum record sizes are the same. If you are defining a **sequential** file, you do **not** need this information.

DATA COMPRESSION
Data compression is available for VSE/VSAM files of type ESDS, KSDS, and VRDS. By specifying **1**, data compression will be enabled.

The manual *z/VSE Planning*, topic “Data Compression Support”, provides introductory information about the data compression support.

Additional information required for specific file types:

Note: Keyed (KSDS) Files For keyed (KSDS) files, specify KEY LENGTH and POSITION. Enter the key length from **1 - 255**. The key position is the offset of the key from the beginning of the record.

Note: Sequential Files If you define a sequential file, specify the following file characteristics:

RECORD FORMAT

1 - Fixed, unblocked

2 - Fixed, blocked

3 - Variable, unblocked

4 - Variable, blocked

5 - Undefined

6 - No control interval format

RECORD SIZE
Fixed record formats only (RECORD FORMAT options 1 and 2). Enter the record length.

BLOCK SIZE
Fixed, blocked format only (RECORD FORMAT option 2). Enter the block length.

AVERAGE RECORD SIZE
Variable length and undefined formats only (RECORD FORMAT options 3, 4, or 5). Enter the average length of the record.

MAXIMUM RECORD SIZE
Variable length and undefined formats only (RECORD FORMAT options 3, 4, or 5). Enter the maximum length of the record.

On the *Job Execution* panel, select:

1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed

Option **1** stores the job control statements as a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Defining a VSE/VSAM Library

With the dialog *Define a Library* you can create a VSE library in VSE/VSAM managed space. To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

2 (File and Catalog Management)

3 (Define a Library)

If you want to specify a catalog and not use the default catalog, select Fast Path 22.

You need the following information:

LIBRARY NAME
Specify the library file name. 'VSE.file name.LIBRARY' is the default for the file ID created by the dialog.

PRIMARY ALLOCATION
Enter the number of 1K library blocks.

SECONDARY ALLOCATION
Enter the number of 1K library blocks.

EXTENTS
Enter either **1** (for a maximum of 16 extents) or **2** (for a maximum of 32

extents). You can specify a maximum of 32 extents when the library is a multi-volume file and is defined in space managed by VSE/VSAM. If you specify a maximum of 32 extents (MAX32), you must also select SECONDARY ALLOCATION on the next *Select Space* panel.

If the catalog owns space on more than one volume, a list is displayed that shows these volumes and their device type code. You can then select the volume you want the primary space allocated and the volume(s) you want the secondary space allocated.

On the *Job Execution* panel, select:

1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed

Option **1** stores the job control statements in a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Note: To access a library you need at least one sublibrary. Use the librarian (LIBR) program to define sublibraries. The topic "Using VSE Libraries" in the manual *z/VSE Guide to System Functions* describes program LIBR in detail.

Defining a VSE/VSAM Alternate Index or Name

With the dialog *Define an Alternate Index or Name* you can create either an alternate index or an alternate name for an existing VSE/VSAM file. A FULIST displays the file IDs and names of the files in the catalog. To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

2 (File and Catalog Management)

4 (Define an Alternate Index or Name)

If you want to specify a catalog and not use the default catalog, select Fast Path 22. Use **PF7** and **PF8** to scroll through the list. Use **PF9** to list a subset of the files by entering a prefix. With **PF2** you can refresh the display. To locate a particular file, enter the file ID in the LOCATE FILE ID field.

Alternate Index

This task defines an alternate index over an existing base file. When you define an alternate index, two things are defined:

1. The alternate index cluster.
2. The path.

The name and ID you specify become the *path* name and ID. The system generates the name of the alternate index cluster internally. You need the following information:

ALTERNATE INDEX ID and NAME
Specify the ID and the name of the alternate index.

KEY POSITION and LENGTH
Specify the position and length of the alternate key within the base record. The key length can be **1 - 255**.

KEYS Specify the maximum number of non-unique keys in the alternate index. The dialog uses this value to calculate the maximum record length of the alternate index file.

The dialog adds a label to the system standard label area with the file name, file ID, and catalog name of the alternate index. It also adds label information to the VSE/VSAM label procedure STDLABUP.

On the *Job Execution* panel, select:

```
1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed
```

Option **1** stores the job control statements as a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Alternate Name

This task defines an alternate name for the file. It also adds label information to the VSE/VSAM label procedure STDLABUP.

If a file does not have a file name (it has no label in the system standard label area), you can use this task to define the file name. You **should not** define alternate names for libraries.

You only need to specify the alternate file name.

On the *Job Execution* panel, select:

```
1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed
```

Option **1** stores the job control statements in a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Displaying or Processing a VSE/VSAM Catalog or Space

The dialog *Display or Process a Catalog, Space* provides a FULIST that shows the catalog IDs and names in the system. To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

2 (File and Catalog Management)

5 (Display or Process a Catalog, Space)

Use **PF7** and **PF8** to scroll through the list. With **PF2** you can refresh the display.

You can select the following options:

1 (Show space)

2 (Define alternate name)

3 (Print catalog contents)

4 (Define space)

5 (Delete catalog)

6 (Delete space)

Show Space

This task displays details about the space owned by the catalog selected. A panel lists the volumes owned by the catalog. It displays the allocated, used, and free space on each volume.

Define Alternate Name

This task defines an alternate name for the catalog. The dialog defines the name and adds a label to the system standard label area with the alternate name. It also adds label information to the VSE/VSAM label procedure STDLABUP.

You should use alternate names for catalogs carefully. On the *File and Catalog Management* panel, if you:

- Select options 1 or 4

  AND
- Specify an alternate catalog name in the CATALOG NAME field

The FULIST only displays file names for files which are defined with the alternate catalog name. The FULIST displays *NONE* as the file name for files defined in the same catalog with a different catalog name.

You only need to specify the alternate catalog name.

On the *Job Execution* panel, select:

1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed

Option **1** stores the job control statements in a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Print Catalog Contents

This task creates a LISTCAT of the selected catalog. You do not have to specify any information. On the *Job Execution* panel, select:

1 - Delayed, Submission is handled by user
2 - Immediate, Job is submitted

Option **1** stores the job control statements as a VSE/ICCF library member in your default primary library.

Option **2** submits the job automatically. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Define Space

This task defines VSE/VSAM space that is to be used for the allocation of files. It is recommended that you define space owned by a catalog on the same volume on which the catalog resides. You need the following information:

VOLUME NAME
Enter the six character VOLID of the disk where the space should be defined.

ALL FREE SPACE
Specify whether you want all available space on the volume for VSE/VSAM:

1 - YES

2 - NO

If you specify 1 (YES), all free space (up to 16 extents) on the volume is dedicated to VSE/VSAM.

If you specify 2 (NO), a panel displays the free extents on the volume. Select one extent. Enter the beginning allocation and the amount of space to be allocated.

In some circumstances, VSE/VSAM rounds the specified values to a higher number. If the rounded extent exceeds the original one, the space definition fails. To avoid this, choose values which result in a smaller extent than the one shown.

SPACE AVAILABLE TO CURRENT FILES
Specify whether the files currently owned by the catalog can access the new space for secondary allocation:

1 - YES

2 - NO

If you specify 1 (YES), the dialog alters the catalog entries of the current files so new space can be used for secondary allocation. The dialog changes the catalog entries, if:

- The secondary allocation for the file is greater than 0.
- You define new space on disk devices with the same device type code as the primary allocation for the file(s).

If a current file already accesses space on a volume, it keeps that access when you define new space on the same volume.

If you specify 2 (NO), current files **cannot** access the new space.

FAT-3390 DISK
Specify extended space on a 3390 disk.

1 - YES

2 - NO

If you specify 1 (YES), the dialog defines up to 65520 cylinders of space (3390 ECKD only).

If you specify 2 (NO), the limit of 10017 cylinders is valid. You should specify NO for all other disks.

On the *Job Execution* panel, select:

1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed

Option **1** stores the job control statements in a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Note:

1. For CKD disk devices, the units of allocation are *cylinders* and **not** tracks.
2. If you are using emulated FBA disks or virtual FBA disks, VSE/VSAM can use as VSE/VSAM space the first 4194240 blocks (2 GB) in units of 960 blocks.
3. If you are using FBA-SCSI disks, VSE/VSAM can use as VSE/VSAM space the first 33546240 blocks (16 GB) in units of 30720 blocks. For further details, refer to the manual *z/VSE Planning*, SC34-2635.

Delete Catalog

This task deletes a VSE/VSAM catalog or alternate catalog name. The dialog removes the user catalog entry from the VSE/VSAM master catalog. Before you delete a catalog, you should do the following:

- Delete all files which the catalog owns.
- If the catalog owns space on more than one volume, first delete the space on the volumes other than the catalog volume.

The dialog deletes the catalog and removes the label from the system standard label area and the VSE/VSAM label procedure STDLABUP. If the catalog has alternate names, specify what you want to delete:

1 - Delete catalog name only
2 - Delete actual catalog, including alternate names

Verify that the catalog is the one you want to delete.

On the *Job Execution* panel, select:

1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed

Option **1** stores the job control statements in a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Delete Space

This task deletes VSE/VSAM data spaces. A panel displays a list of the volumes owned by the catalog. It shows the allocated, used, and free space on each volume.

Enter **5** in the OPT column next to the volume you want to select. On the *Job Execution* panel, select:

1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed

Option **1** stores the job control statements in a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Defining a New VSE/VSAM User Catalog

With the dialog *Define a New User Catalog* you can create a new user catalog and, optionally, space for file allocation.

Data Compression Support: The dialog automatically creates for each user catalog a CCDS (Compression Control Data Set) which is named VSAM.COMPRESS.CONTROL. This cluster is required to enable data compression for ESDS, KSDS, and VRDS files defined in the catalog. The topic "Data Compression Support" in the manual *z/VSE Planning* provides an overview about the data compression support available.

To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

2 (File and Catalog Management)

6 (Define a New User Catalog)

The dialog adds a label to the system standard label area with the catalog name. It also adds label information to the VSE/VSAM label procedure STDLABUP.

Space which belongs to a catalog should be on the same volume as the catalog itself. When you define the catalog, you can use only part of a volume and then use more of the volume later. However, it is better if you take as much space on the volume as you will need. Additional space may not be available later. You need the following information:

USER CATALOG ID and NAME

For the new catalog, enter up to five segments for the file ID. You cannot enter more than 38 characters, including dots. For the name, enter **1 - 7** alphameric characters.

Note: The catalog name IJSYSUC is reserved for system use.

VOLUME NAME

Enter the six character volume ID of the disk where the catalog will be defined.

ALL FREE SPACE

Specify whether you want all available space on the volume for VSE/VSAM:

1 - YES

2 - NO

If you specify 1 (YES), the dialog uses all free space (up to 16 extents) on the volume for both the catalog and files. VSE/VSAM determines the size of the space reserved for the catalog and for the files.

If you specify 2 (NO), you are defining space for the catalog, and space for file allocation. A panel displays the free extents on the volume. Select one extent. Enter the beginning allocation and the amount of space to be allocated.

This is how space will be allocated:

- For *FBA devices* (FBA SCSI, or FBA devices under z/VM):
  - 3072 blocks will be allocated for the catalog space
  - the remaining blocks will be allocated for the data space

- For *ECKD devices*:
  - 75 tracks will be allocated for the catalog space
  - the remaining tracks will be allocated for the data space

In some circumstances, VSE/VSAM rounds the specified values to a higher number. If the rounded extent exceeds the original one, the space definition fails. To avoid this, choose values which result in a smaller extent than the one shown.

FAT-3390 DISK
Specify extended space on a 3390 disk.

1 - YES

2 - NO

If you specify 1 (YES), the dialog defines up to 65520 cylinders of space (3390 ECKD only).

If you specify 2 (NO), the limit of 10017 cylinders is valid. You should specify NO for all other disks.

On the *Job Execution* panel, select:

1 - Delayed, Submission is handled by user
2 - Immediate, Job is executed

Option **1** stores the job control statements in a VSE/ICCF library member in your default primary library. The default member name is **F$xxxx**, where xxxx is your user-ID. You can change the name on the panel.

Option **2** automatically runs the job online. The terminal is locked until the job finishes. It also stores the VSE/ICCF member **F$xxxx.P** (xxxx is your user-ID) in your default primary library. If there are no errors, the member contains one record with an asterisk (*). If errors occurred, the control statements and VSE/VSAM (Access Method Services) error messages are stored in this member. You are notified if you should review the contents of the library member.

Chapter 18. Performing a FlashCopy

This chapter describes how you can perform a *FlashCopy* using the **AR** (Attention Routine) **IXFP SNAP** command. In addition, FlashCopy provides the IXFP **DDSR** and **STATUS** functions.

z/VSE provides support for:

- FlashCopy Version 1 (volume-based FlashCopy), including the NOCOPY option which 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.
- FlashCopy Version 2 ("FlashCopy 2"), which includes the features of FlashCopy Version 1 *plus* extensions designed to improve capacity management and disk utilization.

Note: Except for FILE SNAPping (DSN=data-set-name), VSE will **not**:
1. perform VTOC checking on the specified target device.
2. provide warning messages of any kind, be it overlapping extents, secured- or unexpired files, or anything else.

Cylinder or volume copying will be done *unconditionally* within the specified or assumed boundaries.

This chapter contains these main topics:

- "Installing FlashCopy" on page 222
- "Issuing IXFP Commands From a Batch Job" on page 222
- "Using IXFP SNAP Function With VM Minidisks" on page 222
- "Using the FlashCopy Space Efficient (SE) Feature" on page 223
- "Using the FlashCopy Consistency Group Support" on page 225
- "VSE/Fast Copy (FCOPY) Exploitation of FlashCopy" on page 226

Related Topics:

For details of ...	Refer to ...
the IXFP SNAP, IXFP DDSR, and IXFP STATUS commands (their syntax and parameters)	*z/VSE System Control Statements*.
how to backup VSE/VSAM files and datasets	"Backing Up and Restoring Data" in the *z/VSE Operation*.

For a *general description* of FlashCopy, refer to "Hardware Support" in the *z/VSE Planning*.

Installing FlashCopy

Hardware Prerequisite

FlashCopy support has the following hardware prerequisite:

- IBM TotalStorage Enterprise Storage Server (ESS) or DS8000 series with the FlashCopy Version 1 or Version 2 feature installed.

Shipment and Installation

The FlashCopy support is part of z/VSE as shipped, and is provided via the IXFP command.

Issuing IXFP Commands From a Batch Job

A small REXX/VSE procedure can be used to issue IXFP commands from a batch job. The following example shows such a procedure:

```
* $$ JOB JNM=IXFPREXX,CLASS=0,DISP=D
// JOB IXFPREXX
// EXEC LIBR
ACC S=PRD2.CONFIG
CAT IXFPREXX.PROC  R=Y
/*    rexx/vse procedure     */
/* to issue console commands */
trace off
rc = SENDCMD('your-console-cmd-1')   /* enter your 1st IXFP cmd here */
call sleep 5                         /* wait for 5 seconds           */
rc = SENDCMD('your-console-cmd-2')   /* enter your 2nd IXFP cmd here */
exit rc
/+
/*
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE)
// EXEC REXX=IXFPREXX
/&
* $$ EOJ
```

More information on the REXX/VSE Console Automation Capability can be found in the *REXX/VSE Reference* manual, SC33-6642.

Using IXFP SNAP Function With VM Minidisks

If used on a z/VSE system running under z/VM, the IXFP SNAP function:

- does **not** allow volume or cylinder relocation for **partial minidisks**, and therefore
- **only works** (and will only be accepted as a valid command by z/VM) for **full-pack minidisks** or **dedicated devices**.

Please be aware that for minidisks which use MDC (Mini Disk Caching), the MDC buffer must be flushed *before* performing a SNAP or DDSR function. Otherwise, data can be incomplete.

The above MDC problem is solved by VM APAR VM61486, which is included in z/VM Version 4 onwards.

Note: Other host caching products (for example, Cache Magic) will have the same requirements.

Using the FlashCopy Space Efficient (SE) Feature

Overview of the FlashCopy SE Feature

FlashCopy SE is optimized for use in situations where a small percentage of the source volume is updated during the life of the relationship.

- If much more than 20 per cent of the source is expected to change, there may be tradeoffs in terms of performance versus space efficiency. In this case, “standard” FlashCopy might be considered a better alternative.
- Since background copy would update the *entire target volume*, it would not make much sense and is therefore not permitted with FlashCopy SE.
- If *performance* on the source or target volumes is of primary importance, standard FlashCopy is recommended.

The FlashCopy SE feature of the DS8000 series allocates storage space on an “as needed” basis by only using space on a target volume when it actually copies tracks from the source volume to the target volume. Using FlashCopy SE, disk space will only be consumed on the target volume when:
- a write to the source volume needs to be hardened on disk, or
- a write is directed to the target volume.

FlashCopy SE is designed for *temporary copies*. The duration of the copy should generally not be longer than 24 hours, unless the source and target volumes have little write activity.

Since space is allocated “as needed” on the target volumes, data location on the target volumes is *independent* of data location on the source volumes:
- Data is not physically ordered in the same sequence on the target volume as it is on the source volume.
- A mapping structure is created to keep track of where the data on the target volume is physically located.

Note: FlashCopy SE relations might fail if sufficient space is not available in the storage that is allocated for the target volumes. However, this relation-failure does prevent I/O failures on the source volumes. If the target volume becomes unusable, standard FlashCopy relations require your intervention to remove the relation.

For details of how you implement FlashCopy SE, refer to the documentation that accompanies the DS8000 Series.

Dealing With an Out-of-Space Condition

If the space-efficient (SE) volume encounters an out-of-space condition, the FlashCopy relationship will fail. It will *not* perform a write-inhibit to the source volume or extent.

Further Host writes *will* be allowed to the source volume/extent, but Host write and reads will be *rejected* on the target volume/extent. As a result:
- The volume will appear as if it is offline.
- An “Equipment Check Sense” will be displayed.

To clear this condition and remove the relation, you must issue an `IXFP DDSR` command. For details, refer to “Job Control and Attention Routine” in the *z/VSE System Control Statements*.

Recognizing a Space-Efficient Volume

If you enter the AR (Attention Routine) command `VOLUME cuu`, if the volume is a Space Efficient volume then **SE** (Space Efficient) is included in the display. For example, if a volume has a cuu of `e59` and is a Space Efficient volume, the display would look like this:

```
volume e59

AR 0015 CUU  CODE DEV.-TYP   VOLID  USAGE   SHARED    STATUS    CAPACITY
AR 0015 E59  6E   2107-900  *NONE*  UNUSED            DOWN  SE  65520 CYL
```

For further details, refer to "Job Control and Attention Routine" in the *z/VSE System Control Statements*, SC34-2637.

Verifying the Status of a Space-Efficient Target Volume

If you enter the AR (Attention Routine) command `IXFP STATUS` when the target volume is a Space Efficient (SE) volume, the following additional message is displayed:

```
IXFP74I FL/SE QUERY CUU=nnn POOLID=nnn ALLOCATED SPACE='nnnn' CYL POOL SIZE='nnnn' CYL
```

The cuu=*nnn* provided in this message is the currently-defined FlashCopy Space-Efficient TARGET repository. This additional information might also be displayed:
- Extent Pool ID.
- Space that is currently allocated in the Extent Pool's Repository.
- Size of the Pool's Extent Repository.

For further details of the `IXFP STATUS` command, refer to "Job Control and Attention Routine" in the manual *z/VSE System Control Statements*.

Additional Messages That Might Occur When Running With DEBUG ON

These are the messages you might receive if you are running your z/VSE system using the Attention Routine (AR) `DEBUG ON` option:
- If you are running z/VSE as guest under *z/VM 5.4 or earlier*, these messages *will not appear*. This is because FlashCopy SE is *not* supported by z/VM 5.4 or earlier.
- If you are running z/VSE as guest under *z/VM 6.1 or later* (or if you are running in LPAR mode), these messages *will appear*.

```
AR 0024 ATTENTION-MSG read for DEVICE=0E11  Path=80
AR 0024 00100601  10000001  000E0000  00000000
AR 0024 AOMOS01I POOL=001, SPACE EFFICIENT TARGET REPOSITORY HAS REACHED A WARNING WATERMARK

AR 0024 ATTENTION-MSG read for DEVICE=0E11  Path=80
AR 0024 00100602  00000001  000E0000  00000000
AR 0024 AOMOS02I POOL=001, SPACE EFFICIENT TARGET REPOSITORY HAS BEEN EXHAUSTED
```

Using the FlashCopy Consistency Group Support

Using the FlashCopy *Consistency Group* support, you can take a FlashCopy across *multiple volumes* on an ESS, DS6000, or DS8000 storage system. This is useful when applications have their data spread over multiple volumes.

If you run z/VSE under z/VM, to use Consistency Group support you must have:
- Installed APAR VM64693.
- Specified `STDEVOPT DASDSYS DATAMOVER` in the CP directory.

You can use *consistency groups* to help create a consistent Point-in-Time copy across multiple volumes, and even across multiple ESS, DS6000, or DS8000 storage systems, thus managing the consistency of *dependent writes*:
- If the start of one write operation is dependent upon the completion of a previous write, the writes are *dependent*. Application examples for dependent writes are databases with their associated logging files. In addition, updates to catalogs, VTOCs as well as VSAM indexes and VSAM data components rely on dependent writes. For example, the database logging file will be updated after a new entry has been successfully written to a table space.
- The chronological order of dependent writes to the FlashCopy source volumes is the basis for providing consistent data at the FlashCopy target volumes.

To ensure that order of dependent writes to the FlashCopy source volume is consistent with the data on the target volume, you can use the FlashCopy FREEZE parameter. When you invoke an IXFP SNAP with parameter FREEZE, the storage system will stop the I/O activity to the source volume for a period-of-time by putting the source volume in an *extended long busy* state. Therefore, a *time slot* can be created during which the dependent write updates will not occur. FlashCopy uses the time slot to establish a consistent point-in-time copy of the related volumes.

I/O activity resumes when either of the following occurs:
- You invoke an IXFP DDSR with parameter THAW for the source volume (after all FlashCopies have been established). THAW indicates that all FlashCopy volumes in this logical subsystem (specified by one volume in the unit parameter) will be released.
- The extended “long busy” time slot has expired (the default is two minutes).

You can use Consistency Groups to minimize the application impact on your production data. This is because prior to a Consistency Group FlashCopy, you have to:
1. quiesce the application,
2. establish their FlashCopy relationships, and
3. restart the application,

which is a *disruptive* process. It might cause application outages or data unavailability for an unacceptable period of time.

Note:
1. The FREEZE parameter always affects the *complete volume*. Therefore, even if the IXFP SNAP has been used to specify data sets (rather than complete volumes), the complete volume will nevertheless be “long busy” to host operations. The timer is set to a default of two minutes, called the “Consistency

Group timer” on the WEB panel that displays server properties. You can change the timer value to suit your needs.
2. The FlashCopy command and FlashCopy FREEZE are on a *volume* basis, but the THAW command is at the *LSS (logical subsystem)* level. This means, if there are more than one set of volumes using consistency, the THAW command will affect *all* of the Consistency Groups.
3. An I/O request to a source device that is in a FREEZE state will be in a “long busy” state until either:
   - The extended long busy time slot (120 seconds) has expired.
   - A THAW command is issued.

   This might cause:
   - Delays to I/O requests for system files residing on a disk that is in a FREEZE state.
   - Delays to AR commands.
   - Unpredictable results for application programs that use their own error processing routines.
   - A THAW command to not be processed until the extended “long busy” time slot (120 seconds) has expired. This might also be because previous I/O requests are still in a “long busy” state.
4. If you repeatedly submit a FlashCopy FREEZE request for the *same* source volume, this request will fail with a PERM I/O error. This is because the source volume is in an extended “long busy” state.

For details of how to use the FREEZE and THAW parameters with the IXFP SNAP and IXFP DDSR commands, refer to the chapter “Job Control and Attention Routine” in the *z/VSE System Control Statements*,

VSE/Fast Copy (FCOPY) Exploitation of FlashCopy

The *VSE/Fast Copy* utility (described in the manual *z/VSE System Utilities*) exploits the FlashCopy function. It supports **full volume** backup from disk-to-disk. However, it does *not* support the copying of files or extents.

The following VSE/Fast Copy options are supported:
- COPY ALL
- COPY VOLUME
- COPY VOLUME NOCOPY
- COPY ALL NOCOPY
- DDSR

The *DDSR option* which removes a previously-established NOCOPY relation is also supported via the *Remove FlashCopy relation* IUI dialog:
3 (Operations)
7 (Backup/Restore)
7 (Copy a Volume or File)
3 (Remove FlashCopy relation)

For a description of the *Remove FlashCopy relation* dialog, refer to “Backing Up and Restoring Data” in the manual *z/VSE Operation*

The VSE/Fast Copy (FCOPY) uses a FlashCopy when:
- FlashCopy support *is* available in the disk hardware.
- A *full-volume* backup request is for disk-to-*disk* (and not disk-to-*tape*).

If the requested support is *not* available, a normal VSE/Fast Copy backup will be performed.

The following VSE/Fast Copy optional parameters are supported:

```
IV  (input volume)
OV  (output volume)
NV  (new volume)
```

The following VSE/Fast Copy optional parameters are tolerated:

```
NOPROMPT
NOVERIFY
LIST
```

The following VSE/Fast Copy optional parameters force FlashCopy NOT to be used. These parameters apply to COPY VOLUME only:

```
EXCLUDE
NOVSAM
NOEXPIRED
```

Job Stream Examples

This job stream example applies to the COPY ALL as well as to the COPY VOLUME command:

```
// JOB jobname
// ASSGN SYS004,140
// ASSGN SYS005,141
// EXEC FCOPY
COPY VOLUME IV=SYSRES
/*
/&
```

This job stream example shows how the NOCOPY parameter is used to establish a FlashCopy relation to a target disk whose volume-ID is FRA626:

```
// JOB jobname
// ASSGN SYS004,ANYDISK,VOL=DOSRES,SHR
// ASSGN SYS005,ANYDISK,VOL=FRA626,SHR
// EXEC FCOPY
COPY VOLUME NOCOPY -
NOVERIFY NV=FRA626
/*
/&
```

This job stream example shows how the DDSR parameter is used to remove a FlashCopy relation to a target disk whose volume-ID is FRA626:

```
// JOB jobname
// ASSGN SYS005,ANYDISK,VOL=FRA626,SHR
// EXEC FCOPY
DDSR
/*
/&
```

Chapter 19. Managing Non-VSE/VSAM Libraries and User File Labels

To manage libraries in **non-VSE/VSAM** space, z/VSE provides the following skeletons:

- SKLIBDEF (for defining libraries)
- SKLIBEXT (for extending libraries)
- SKLIBDEL (for deleting libraries)

To create standard labels for non-VSE/VSAM user files, z/VSE provides skeleton STDLABUS.

These skeletons are described on the following pages.

To manage libraries in VSE/VSAM space, z/VSE provides dialogs as described in Chapter 17, "Managing VSE/VSAM Files and Catalogs," on page 207.

Defining a VSE User Library in Non-VSE/VSAM Space

The **SKLIBDEF** skeleton defines a VSE user library in non-VSE/VSAM space. The skeleton has two major steps:

1. Add standard label for the new library.
2. Create the library.

The skeleton is shipped in library 59. First copy it to your primary library and edit the copied file. Refer to "Copying Skeletons" on page 11 for information on copying skeletons.

Figure 55 on page 230 shows the skeleton. The variables you should change are highlighted in color. They are described below the figure to help you make the correct changes.

After you make the changes, update the standard label skeleton STDLABUS. Add the label for the library which SKLIBDEF defines. Refer to "Creating Standard Labels for Non-VSE/VSAM User Files" on page 234 for information about STDLABUS.

Note: To access a library you need at least one sublibrary. Use the librarian (LIBR) program to define sublibraries. The topic "Using VSE Libraries" in the manual *z/VSE Guide to System Functions* describes the librarian program LIBR in detail.

```
* $$ JOB JNM=DEFLIBR,CLASS=0,DISP=D
// JOB DEFLIBR DEFINE LIBRARY IN NON VSAM SPACE
// OPTION STDLABEL=ADD
// DLBL --V001--,'--V002--',99/366,SD
// EXTENT ,--V101--,1,--V102--,--V103--,--V104--
// EXTENT ,--V101--,2,--V102--,--V103--,--V104--
/*
// EXEC LIBR
DEFINE LIB=--V001-- R=Y
/*
/&
* $$ EOJ
END OF MEMBER
```

Figure 55. Define VSE User Library in Non-VSE/VSAM Space (SKLIBDEF Skeleton)

Change the **--Vxxx--** variables in the DLBL, EXTENT, and DEFINE LIB statements.

--V001--
: This is the file name of the library. Specify **1 - 7** alphanumeric characters.

  Note: Change --V001-- in both the DLBL and DEFINE LIB statements.

--V002--
: This is the file identification. Specify **1 - 44** alphanumeric characters. Do **not** delete the single quotes (' ') around the variable.

--V101--
: The volume number where the library will reside. It must be **6** characters. The value can be the same or different for each extent.

--V102--
: This is the extent sequence number (**0 - 15**). You can have up to sixteen extents defined on several disks. The disks must all be of the same type.

--V103--
: This is the start location of the library in tracks or blocks. The value can be different for each extent.

--V104--
: This is the amount of space to be allocated for the library on the first volume. The value can be different for each extent. Specify the value in tracks or blocks.

After you have modified the skeleton, enter the following command from the editor's command line:

```
@DTRSEXIT
```

This command calls a macro that deletes specific comments from the skeleton. You should do this *before* filing the skeleton.

Extending a VSE User Library in Non-VSE/VSAM Space

The **SKLIBEXT** skeleton extends a VSE user library in non-VSE/VSAM space. It is recommended that you use a MINI startup for extending a library. This is to ensure that the library is not in an active LIBDEF chain.

The skeleton has five major steps:

1. Back up the library to tape.
2. Delete the library.
3. Delete standard label for the library.

4. Add new standard label for the extended library.
5. Create library and restore the library from tape.

The skeleton is shipped in VSE/ICCF library 59. First copy it to your &iccf primary library and edit the copied skeleton. Refer to “Copying Skeletons” on page 11 for information on copying skeletons.

Figure 56 shows the skeleton. Each section of the skeleton is shown in separate parts of the figure. The variables you should change are highlighted in color. A description of the changes follows each part of the figure.

After you make the changes, update the standard label skeleton STDLABUS. Change the label for the library which SKLIBEXT updates. Refer to “Creating Standard Labels for Non-VSE/VSAM User Files” on page 234 for information about STDLABUS.

```
* $$ JOB JNM=EXTLIBR,CLASS=0,DISP=D
// JOB EXTLIBR EXTEND LIBRARY IN NON VSAM SPACE
*    THIS FUNCTION USES A TAPE FOR OUTPUT
*    MOUNT TAPE REEL --V004-- WITH UNLABELED TAPE ON DEVICE --V003--
*    THEN CONTINUE. IF NOT POSSIBLE CANCEL THIS JOB
*    WARNING: EXISTING TAPE LABEL WILL BE OVERRIDDEN
// PAUSE
// ASSGN SYS005,--V003--
// MTC REW,SYS005
// EXEC LIBR
   BACKUP LIB = --V001--      /* LIBRARY IDENTIFICATION
          RESTORE = ONLINE                      /* RESTORE TYPE
          TAPE = SYS005                         /* TAPEADRESS
/*
// MTC REW,SYS005
// IF $RC > 0 THEN
// GOTO $EOJ
// EXEC LIBR
DELETE LIB=--V001--
/*
// IF $RC > 0 THEN
// GOTO $EOJ
// OPTION STDLABEL=DELETE
--V001--
/*
```

Figure 56. SKLIBEXT Skeleton, Part 1 of 3 (Extend VSE User Library in Non-VSE/VSAM Space)

Change the **--Vxxx--** variables in the comments, the ASSGN, BACKUP LIB, and DELETE LIB statements, and in the statement following the OPTION statement.

--V001--
: This is the file name of the library. Specify **1 - 7** alphanumeric characters.

--V003--
: The tape address (**cuu**) used for the backup.

--V004--
: The volume number of the backup/restore tape. It must be **6** characters.

```
// OPTION STDLABEL=ADD
// DLBL --V001--,'--V002--',99/366,SD

// EXTENT ,--V101--,1,--V102--,--V103--,--V104--
// EXTENT ,--V101--,2,--V102--,--V103--,--V104--
// EXTENT ,--V101--,3,--V102--,--V103--,--V104--
/*
```

Figure 57. SKLIBEXT Skeleton, Part 2 of 3 (Extend VSE User Library in Non-VSE/VSAM Space)

Change the **--Vxxx--** variables in the DLBL and EXTENT statements.

--V001--
: This is the file name of the library. Specify **1 - 7** alphanumeric characters.

--V002--
: This is the file identification. Specify **1 - 44** alphanumeric characters. Do **not** delete the single quotes (' ') around the variable.

--V101--
: The volume number where the library will reside. It must be **6** characters. The value can be the same or different for each extent.

--V102--
: This is the extent sequence number (**0 - 15**). You can have up to sixteen extents defined on several disks. The disks must be all of the same type.

--V103--
: This is the start location of the library in tracks or blocks. The value can be different for each extent.

--V104--
: This is the amount of space to be allocated for the library on the first volume. The value can be different for each extent. Specify the value in tracks or blocks.

```
*    THIS FUNCTION USES AN EXISTING TAPE FOR INPUT
*    MOUNT TAPE REEL --V004-- ON DEVICE --V003-- (THE TAPE IS UNLABELED)
*    THEN CONTINUE. IF NOT POSSIBLE CANCEL THIS JOB
*    REPLY 'DELETE' TO MESSAGE '4433D  EQUAL FILE ID IN VTOC ...'

// PAUSE
// ASSGN SYS005,--V003--
// MTC REW,SYS005
// EXEC LIBR
   RESTORE LIB = --V001--
           LIST  = YES
           REPLACE = YES
           TAPE = SYS005
/*
// MTC RUN,SYS005
/&
* $$ EOJ
```

Figure 58. SKLIBEXT Skeleton, Part 3 of 3 (Extend VSE User Library in Non-VSE/VSAM Space)

Change the **--Vxxx--** variables in the comments and the ASSGN, and RESTORE LIB statements.

--V001--
: This is the file name. Specify **1 - 7** alphanumeric characters.

--V003--
The tape address (**cuu**) used to restore the library.

--V004--
The volume number of the backup/restore tape. It must be **6** characters.

When you run the job, use the same tape and tape device you specify in the skeleton.

After you have modified the skeleton, enter the following command from the editor's command line:

```
@DTRSEXIT
```

This command calls a macro that deletes specific comments from the skeleton. You should do this *before* filing the skeleton.

Deleting a VSE User Library in Non-VSE/VSAM Space

The **SKLIBDEL** skeleton deletes a VSE user library in non-VSE/VSAM space. It is recommended that you use a MINI startup for deleting a library. This is to ensure that the library is not in an active LIBDEF chain.

The skeleton has two major steps:

1. Delete the library.
2. Delete the standard label for the library.

The skeleton is shipped in VSE/ICCF library 59. First copy it to your VSE/ICCF primary library and edit the copied file. Refer to "Copying Skeletons" on page 11 for information on copying skeletons.

Figure 59 shows the skeleton. The variable you should change is highlighted in color. It is described below the figure to help you make the correct changes.

After you make the changes, update the standard label skeleton STDLABUS. Delete the label for the library which SKLIBDEL deletes. Refer to "Creating Standard Labels for Non-VSE/VSAM User Files" on page 234 for information about STDLABUS.

```
* $$ JOB JNM=DELLIBR,CLASS=0,DISP=D
// JOB DELLIBR DELETE LIBRARY IN NON VSAM SPACE
// EXEC LIBR
DELETE LIB=--V001--
/*
// IF $RC > 0 THEN
// GOTO $EOJ
// OPTION STDLABEL=DELETE
--V001--
/*
/&
* $$ EOJ
```

Figure 59. Delete VSE User Library in Non-VSE/VSAM Space (SKLIBDEL Skeleton)

Change the **--V001--** variable in the DELETE LIB statement and the statement following the OPTION statement. --V001-- is the file name. Specify **1 - 7** alphanumeric characters.

After you have modified the skeleton, enter the following command from the editor's command line:

```
@DTRSEXIT
```

This command calls a macro that deletes specific comments from the skeleton. You should do this *before* filing the skeleton.

Creating Standard Labels for Non-VSE/VSAM User Files

You use skeleton **STDLABUS** to create standard labels for files you created in space *not managed by VSE/VSAM*. In addition, you can use STDLABUS to add your partition standard labels.

Note: Labels for VSE/VSAM files are *automatically* created or deleted when using the dialogs for file and catalog management. For additional information on z/VSE label processing, consult the manual *z/VSE Planning*. It describes the standard label procedures STDLABEL, STDLABUP, and STDLABUS used by z/VSE in the topic "Standard Label Procedures".

STDLABUS contains VSE/POWER JECL and JCL statements to catalog the procedure in the system library. It is shipped in VSE/ICCF library 59. First copy it to your VSE/ICCF primary library and edit the copied skeleton. Refer to "Copying Skeletons" on page 11 for information on copying skeletons.

The name of the member in the CATALOG statement **must** be STDLABUS because the system standard label procedure STDLABEL calls STDLABUS to load your standard labels.

Figure 60 on page 236 shows the skeleton. Each section of the skeleton is shown in separate parts of the figure. The values you should change are highlighted. The skeleton contains samples for defining labels for the following types of disk files:

- Sequential
- Direct access
- Index sequential

Each sample is enclosed within asterisks (*) and is noted by a heading. DLBL and EXTENT statements follow each sample. Use these statements to define your labels for that particular type of disk file. The DLBL and EXTENT statements contain variables which you replace. The variables are basically the same for each disk file type. They are described below. Differences for specific types of disk files are noted.

The variables you should change in the DLBL statements are:

YYYYYYY
: This is the file name. Specify 1 - 7 characters.

FFFFFFFF
: This is the file-id. Specify 1 - 44 characters.

YYYY/DDD
: The retention period of the file where YYYY is the year and DDD is the day.

The variables you should change in the EXTENT statements are:

YYYYYY
The SYS number of the file.

VVVVVV
Volume number of the disk where the file resides.

T File type code. This value differs depending on the type of file. For direct access files, you **must** specify **1**.

For sequential disk files, specify:
- **1** - Prime data
- **8** - Prime data with split cylinder (not FBA)

For index sequential disk files, specify:
- **1** - Prime data
- **2** - Overflow
- **4** - Index

S The file sequence number. This value starts with 0.

BBBBBBBB
Starting track or block for the extent of the file.

NNNNNNNN
Number of tracks or blocks for this extent.

PP You need this value for sequential disk files only. Specify the split cylinder start track (for split cylinder only).

```
..$$ JOB JNM=CATALOG,DISP=D,CLASS=0
// JOB CATALOG
                   MAKE SURE SYSLST IS ASSIGNED FOR LIBR              C
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG STDLABUS.PROC  DATA=YES     REPLACE=YES
/. STANDARD LABEL SKELETON FOR YOUR STANDARD LABELS
                                                                      C
         SEQUENTIAL DISK FILE SAMPLES.                                C
                                                                      C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
* // DLBL SAMPSD1,'SAMPLE.SEQUENTL.DISK.FILE.ONE',99/366,SD           C
* // EXTENT SYS004,SAMP01,1,0,2400,1200                               C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
    SINGLE EXTENT SD FILE.                                            C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
* // DLBL SAMPSD2,'SAMPLE.SEQUENTL.DISK.FILE.TWO',99/366,SD           C
* // EXTENT SYS004,SAMP01,8,0,2400,1200,6                             C
* // DLBL SAMPSD3,'SAMPLE.SEQUENTL.DISK.FILE.THREE',99/366,SD         C
* // EXTENT SYS004,SAMP01,8,0,2407,1200,14                            C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
    TWO SINGLE EXTENT SD FILES USING SPLIT CYLINDER ON 3390.          C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
* // DLBL SAMPSD4,'SAMPLE.SEQUENTL.DISK.FILE.FOUR',99/366,SD          C
* // EXTENT SYS004,SAMP01,1,0,2400,600                                C
* // EXTENT SYS004,SAMP01,1,1,2400,600                                C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
    MULTI-EXTENT SD FILE.                                             C
                                                                      C
         SEQUENTIAL DISK FILE SKELETON.                               C
                                                                      C
// DLBL YYYYYYY,'FFFFFFFF.FFFFFFFF.FFFFFFFF.FFFFFFFF',YYYY/DDD,SD     C
// EXTENT YYYYYY,VVVVVV,T,S,BBBBBBBB,NNNNNNNN,PP
```

Figure 60. DTRLABUS Skeleton, Part 1 of 3 (Create Standard Labels)

In the CATALOG statement, the procedure name **must** be STDLABUS. The system standard label procedure STDLABEL calls STDLABUS to load your standard labels.

The three samples are enclosed within asterisks. They are for the following types of sequential disk files:

- Single extent.
- Two single extent using split cylinder on an IBM 3390.
- Multi-extent.

Use the DLBL and EXTENT statements following the three samples to define the labels for your sequential disk files. If you **do not** use these statements, delete them.

```
          DIRECT ACCESS FILE SAMPLE.                                C
                                                                    C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
* // DLBL SAMPDA,'SAMPLE.DIRECT.ACCESS.FILE.A',99/366,DA            C
* // EXTENT SYS002,SAMP01,1,0,100,1200                              C
* // EXTENT SYS002,SAMP01,1,1,1300,1200                             C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
                                                                    C
         DIRECT ACCESS FILE SKELETON.                               C
                                                                    C
// DLBL YYYYYYY,'FFFFFFFF.FFFFFFFF.FFFFFFFF.FFFFFFFF',YYYY/DDD,DA
// EXTENT YYYYYYY,VVVVVV,T,S,BBBBBBBB,NNNNNNNN

         INDEX SEQUENTIAL FILE (CREATE) SAMPLE.                     C
                                                                    C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
* // DLBL SAMPISC,'SAMPLE.INDEX.SEQUENTL.FILE.ONE',99/366,ISC        C
* // EXTENT SYS005,SAMP01,4,0,3700,1200                             C
* // EXTENT SYS005,SAMP01,4,1,4900,1200                             C
* // EXTENT SYS005,SAMP01,1,2,6100,1200                             C
* // EXTENT SYS005,SAMP01,1,3,7300,1200                             C
* // EXTENT SYS005,SAMP01,2,4,8500,1200                             C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
                                                                    C
        INDEX SEQUENTIAL FILE (CREATE) SKELETON.                    C
                                                                    C
// DLBL YYYYYYY,'FFFFFFFF.FFFFFFFF.FFFFFFFF.FFFFFFFF',YYYY/DDD,DA
// EXTENT YYYYYYY,VVVVVV,T,S,BBBBBBBB,NNNNNNNN

         INDEX SEQUENTIAL (EXTENSION) FILE SAMPLE.                  C
                                                                    C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
* // DLBL SAMPISE,'SAMPLE.INDEX.SEQUENTL.FILE.TWO',99/366,ISE        C
* // EXTENT SYS006,SAMP01,4,0,9700,1200                             C
* // EXTENT SYS006,SAMP01,4,1,10900,1200                            C
* // EXTENT SYS006,SAMP01,1,2,12100,1200                            C
* // EXTENT SYS006,SAMP01,1,3,13300,1200                            C
* // EXTENT SYS006,SAMP01,2,4,14500,1200                            C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * C
                                                                    C
         INDEX SEQUENTIAL (EXTENSION) FILE SKELETON.
                                                                    C
// DLBL YYYYYYY,'FFFFFFFF.FFFFFFFF.FFFFFFFF.FFFFFFFF',YYYY/DDD,DA
// EXTENT YYYYYYY,VVVVVV,T,S,BBBBBBBB,NNNNNNNN
```

Figure 61. DTRLABUS Skeleton, Part 2 of 3 (Create Standard Labels)

One sample is enclosed within asterisks for direct access disk files. Use the DLBL and EXTENT statements following the sample to define the labels for your direct access files. If you **do not** use these statements, delete them.

Two samples are enclosed within asterisks for index sequential files (create and extension). Use the DLBL and EXTENT statements following each sample to define the labels for your index sequential (create or extension) files. If you **do not** use these statements, delete them.

```
// OPTION PARSTD
// OPTION PARSTD=F1        FOR STATIC PARTITIONS
// OPTION PARSTD=F2
// OPTION PARSTD=F3
// OPTION PARSTD=F4
// OPTION PARSTD=F5
// OPTION PARSTD=F6
// OPTION PARSTD=F7
// OPTION PARSTD=F8
// OPTION PARSTD=F9
// OPTION PARSTD=FA
// OPTION PARSTD=FB
// OPTION CLASSTD=C        FOR DYNAMIC PARTITIONS
// OPTION CLASSTD=P
// OPTION CLASSTD=Y
// OPTION CLASSTD=Z
/+
CONN S=IJSYSRS.SYSLIB:PRD2.SAVE
COPY STDLABUS.PROC     REPLACE=YES
../*
../&
..$$ EOJ
```

Figure 62. DTRLABUS Skeleton, Part 3 of 3 (Create Standard Labels)

After **each** // OPTION statement, add the standard labels for that particular partition. If you do not specify labels for a partition, the area is cleared.

Note: Do not delete any of the // OPTION statements, even if you have no partition standard labels to add. If you do, startup problems are likely to occur.

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
```

Chapter 20. Implementing Virtual Tape Support

In z/VSE, a *virtual tape* is a file (or dataset) containing a tape image. You can read from or write to a virtual tape in the same way as if it were a physical tape. 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.

This chapter contains these main topics:

Related Topics:

For details of the ...	Refer to ...
• *current restrictions* when working with virtual tapes • hints and tips when working with virtual tapes	URL http://www.ibm.com/systems/z/os/zvse/support/vtape.html
VTAPE-command syntax diagrams and examples: • VTAPE START (to start the Virtual Tape Data Handler) • VTAPE STOP (to stop the Virtual Tape Data Handler) • VTAPE QUERY (to check which virtual tapes are currently active)	"Job Control and Attention Routine" in the *z/VSE System Control Statements*.
installation of optional z/VSE programs using virtual tapes	*z/VSE Installation*.

Overview of Virtual Tape Support

The Virtual Tape Support consists of *three* functional components:

- **Virtual Tape Simulator**

  The Virtual Tape Simulator is part of z/VSE and ready for use after z/VSE installation. The Virtual Tape Simulator:
  - Controls virtual tape processing independent of where the virtual tape is located (VSE/VSAM or remote).
  - Receives any incoming requests and forwards them to the Virtual Tape Data Handler for processing.
  - Provides status information about the function performed.
- **Virtual Tape Data Handler**

  The Virtual Tape Data Handler is part of z/VSE and ready for startup after z/VSE installation. It is required for handling the read or write access to any virtual tape: VSE/VSAM virtual tapes or remote virtual tapes via TCP/IP. When Virtual Tape Support is:
  - *activated* (VTAPE START command or statement), the Virtual Tape Data Handler is loaded into a dynamic partition (the default) or into a static partition.
  - *deactivated*, the partition is released again.
- **Virtual Tape Server**

  The Virtual Tape Server is required for **remote virtual tapes only**. It is the workstation counterpart to the Virtual Tape Data Handler on the z/VSE side with which it communicates via TCP/IP. The Virtual Tape Server must be installed on a workstation with a Java™ platform.

Prerequisites for Using Virtual Tape Support

There are *two* prerequisites for using Virtual Tape Support, as described below.

Ensuring there is Sufficient PFIXed Space in the System GETVIS

Before you use the Virtual Tape Support, you should be aware that the z/VSE I/O Supervisor will allocate a 1 MB buffer for *each* virtual tape in the PFIXed System GETVIS area of your z/VSE system. This may cause problems if you run virtual tapes in parallel, and where the real storage available is not greater than 16 MB (under z/VM, for example).

The 1 MB buffer is used to buffer the tape data before it is:
- Written to the virtual tape.
- Read from the virtual tape.

Therefore, before you use the VTAPE START command, you must ensure that there is sufficient PFIXed space available in the System GETVIS area of your z/VSE system.

To check that sufficient PFIXed System GETVIS space is available, you can issue the AR command:

```
MAP SVA
```

If the system has insufficient PFIXed System GETVIS storage, when the VTAPE START command is issued this message will be displayed:

```
"1YN6t Not Enough Pfixed Getvis Storage to Establish Virtual Tape"
```

Installing the Java Development Kit (JDK)

Before you use the Virtual Tape Support, you must install *Java Development Kit* (JDK) 1.5 or higher on the workstation where you plan to install the Virtual Tape Server.

You can download the *Java Development Kit* (JDK) from the Internet via either of the following URLs:

`http://www.ibm.com/developerworks/java/jdk/index.html`

`http://www.oracle.com/technetwork/java/`

Restrictions When Using Virtual Tapes

In general, the Virtual Tape Support is intended to be transparent to applications, and to provide customers 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 and there are a number of restrictions outlined in this topic. This topic also provides and points to planning information to be considered before you start using virtual tapes at your installation.

The following restrictions apply for virtual tapes:

- The maximum block size of a virtual tape file cannot exceed *65535 bytes*.
- There is no alternate tape support for virtual tapes. To avoid this restriction:
  - For a VSAM virtual tape, you should choose primary and secondary allocations for the virtual tape file that are large enough to avoid an “end of volume” condition.
  - For a VSAM virtual tape, the size limits that apply to an ESDS cluster (4 GB) also apply here.
  - For a remote virtual tape, the size of the remote virtual tape file is assigned according to the PC's file rules. If you receive an “end of volume” condition for your remote virtual tape, you should check these PC file rules.
- The SDAID trace program does not support virtual tapes as output device.
- The DITTO/ESA for VSE program does not support the ERASE TAPE function for virtual tapes. An application that issues an ERASE TAPE function will be cancelled. Instead of using the ERASE TAPE function, you should use the equivalent VSAM and PC file-system functions.
- Multivolume tape files and alternate tapes (ALT option of the ASSGN statement) are not supported for virtual tapes.
- Virtual Tape Support in a stand-alone environment is not supported.
- You are recommended **not** to run virtual tape job streams in VSE/ICCF interactive partitions.
- For remote virtual tapes, the file you need for containing a virtual tape is created *automatically*. However, for a VSE/VSAM virtual tape you must *create the file yourself*. For details, see “VSE/VSAM ESDS File Definition (Skeleton SKVTAPE)” on page 248.
- A remote virtual tape behaves like a physical tape with regard to reading/writing. However, a VSE/VSAM virtual tape behaves differently with regard to writing. For details, see “Writing to VSE/VSAM Virtual Tapes” on page 250.

Restrictions Regarding Label Handling: The following job sequence would cause a deadlock between the partition BG and the Tape Data Handler partition:

```
// OPTION STDLABEL=ADD
// DLBL VTAPFIL,'VTAPFIL.TEST.FILE',,VSAM,CAT=IJSYSCT
// VTAPE START,UNIT=181,LOC=VSAM,FILE='VTAPFIL'
```

The deadlock is caused by the VTAPE command, since the BG partition is concurrently updating system standard labels (that is the `// OPTION STDLABEL` is in effect). To avoid this type of deadlock, you should terminate the standard label processing (by including an `// OPTION USRLABEL` statement before the `VTAPE START` statement).

The correct job control sequence would therefore be:

```
// OPTION STDLABEL=ADD
// DLBL VTAPFIL,'VTAPFIL.TEST.FILE',,VSAM,CAT=IJSYSCT
// OPTION USRLABEL
// VTAPE START,UNIT=181,LOC=VSAM,FILE='VTAPFIL'
```

For the latest information on supported functions and programs, operating characteristics, and possible restrictions refer also to the z/VSE home page (Service and Support page) at:

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

File Names and Other Considerations When Using Virtual Tapes

If the required Linux, UNIX, or Windows file for a remote virtual tape does not exist, it will be automatically created after the corresponding VTAPE START command has been submitted. When assigning file names you must observe certain rules and characteristics as outlined below.

Linux and UNIX Considerations

Linux and UNIX are *case sensitive*, but job streams created on the z/VSE host (using dialogs of the Interactive Interface) are in *capital letters*. It may be therefore necessary to edit such job streams to adapt the filename to the Linux or UNIX conventions.

Windows Considerations

Windows file names may contain blanks, therefore the filename must be enclosed in quotes. A quote within a filename must be coded as two single quotes. For example:

```
FILE='D:\John''s\Virtual Tapes\vt001401.001'
```

Windows file names can have more than 100 characters in length. Since the limit for remote files is 100 characters, you may specify `FILE='filename'` twice or even three times. The filename is concatenated in storage, thus allowing for a file name length of 200 or even 300. The following example is equivalent to the previous example:

```
FILE='D:',FILE='\John''s\Virtual Tapes\',FILE='vt001401.001'
```

Example Job - Here is an example of a job that has been generated by the *Prepare for Installation* dialog:

```
* $$ JOB JNM=INSPRE,DISP=D,PRI=3,                                      C
* $$ NTFY=YES,                                                         C
* $$ LDEST=*,                                                          C
* $$ CLASS=0
// JOB INSPRE  SCAN OPTIONAL PRODUCT TAPE
// LIBDEF PHASE,SEARCH=(PRD1.BASE,IJSYSRS.SYSLIB)
* *
```

```
* *     PREPARE ADDITIONAL PROGRAM INSTALLATION
* *     -  SCAN PROGRAM TAPE
* *
* *  VIRTUAL TAPE SPECIFIED, NO REAL TAPE DRIVE REQUIRED ON
* *               280
// EXEC IJBVTDLG
UNIT=280,
LOC=123.123.123.123,
FILE='DATASET*********************************'
FILE='****************************************'
FILE='****************************************'
FILE='****************************************'
FILE='****************************************'
FILE='****************************************'
FILE='************************'
READ                                                                    /*
// ASSGN SYS006,280
// MTC REW,SYS006
// EXEC DTRIPRE,PARM='VDDR=280'
/*
// ASSGN SYS006,UA
// VTAPE STOP,UNIT=280
/&
* $$ EOJ
```

Before it is submitted, the above job can be stored in a library and edited as required. For example, you might need to use the `SET CASE MIXED` command to produce the statements shown below:

```
:
FILE='my*TEST.file****************************'
FILE='****************************************'
FILE='****************************************'
FILE='****************************************'
FILE='****************************************'
FILE='****************************************'
FILE='************************'
:
```

When the job is finally submitted, a mixed-case file name will therefore be used.

Installing the Virtual Tape Server

Note: The Virtual Tape Server is required for **Remote Virtual Tapes** only.

It must be installed on a workstation with a Java platform. The following steps are required:
- Obtain a copy of the Virtual Tape Server.
- Perform the Virtual Tape Server installation.

Support for uninstalling the Virtual Tape Server is also available.

The Virtual Tape Server is supplied as file **vtapesrv.w** (contained in PRD2.PROD after the VSE Connectors Workstation Code has been installed from the Extended Base Tape) and also as file **vtape510.zip** (on the z/VSE Home Page). For details of how to obtain these files, see "Obtaining a Copy of the Virtual Tape Server" on page 244.

Obtaining a Copy of the Virtual Tape Server

To obtain a copy of the *Virtual Tape Server*, you must decide if you wish to obtain it:
- From the Internet.
- By installing the VSE Connectors Workstation Code component from the Extended Base Tape.

To obtain the Virtual Tape Server *from the Internet*, you should:
1. Start your Web browser and go to the URL

   ```
   http://www.ibm.com/servers/eserver/zseries/zvse/downloads/
   ```
2. From within the Virtual Tape Server section, download the file **vtape*nnn*.zip** (or **vtape*nnn*.zip-*APAR_number*** for the latest APAR level) to the directory where you wish to install the Virtual Tape Server. **Note:** *nnn* refers to the current VSE version (for example, **vtape510.zip**).

To obtain the Virtual Tape Server *by installing the VSE Connectors Workstation Code component*, you should:
1. Install the VSE Connectors Workstation Code component from the Extended Base Tape. After you have installed this component, the Virtual Tape Server W-book **vtapesrv.w** will be stored in z/VSE sublibrary PRD2.PROD.
2. Use the FTP (file transfer program) utility of TCP/IP for VSE/ESA to download **vtapesrv.w** to the directory where you wish to install the Virtual Tape Server.

Note:
1. You must download **vtapesrv.w** in *binary*.
2. Make sure that Unix mode is ***turned off***. Otherwise **vtapesrv.w** will be downloaded in ASCII mode, even when you specify *binary*. *UNIX mode* is one parameter of your VSE FTP daemon. Some FTP clients might *force* UNIX mode to be turned on. The example below shows how a successful transfer of **vtapesrv.w** was made using a (command line) FTP client. The place where the UNIX mode is set, is shown as bold.

   ```
   c:\temp>ftp n.n.n.n   (this is the IP address of the z/VSE system)
   Connected to n.n.n.n

   ⋮
   220 Service ready for new user.
   User (n.n.n.n:(none)): user
   331 User name okay, need password.
   Password:
   230 User logged in, proceed.
   ftp> cd prd2
   250 Requested file action okay, completed.
   ftp> cd prod
   250 Requested file action okay, completed.
   ftp> binary
   200 Command okay.
   ftp> get vtapesrv.w
   200 Command okay.
   150-File: PRD2.PROD.VTAPESRV.W
       Type: Binary Recfm: FB Lrecl:    80 Blksize:      80
       CC=ON  UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON  NAT=NO
   150 File status okay; about to open data connection
   226-Bytes sent:       4,756,400
       Records sent:        59,455
       Transfer Seconds:        16.52 (    290K/Sec)
       File I/O Seconds:         3.94 (  1,548K/Sec)
   226 Closing data connection.
   ```

```
4756400 bytes received in 17,12 seconds (277,91 Kbytes/sec)
ftp> bye
221 Service closing control connection.
c:\temp>ren vtapesrv.w vtape430.zip
```

Performing the Virtual Tape Server Installation

To perform the installation of the Virtual Tape Server, you must:

1. Open a Command Prompt and change to the directory into which you have downloaded the **vtape510.zip** file.
2. If you have downloaded the file from the z/VSE host, rename **vtapesrv.w** to **vtape510.zip**.
3. Extract **vtape510.zip** into a temporary directory (for example, `c:\temp`).
4. Execute the file:
   - **setup.bat** or **setup.cmd** (under Windows)
   - **setup.sh** (under Linux or Unix)

   The installation process now begins. To perform the installation, you are guided through various installation dialogs.

Uninstalling the Virtual Tape Server

There are two possible ways to uninstall the Virtual Tape Server:

- Using the "standard" Windows method, which uses the **Add or Remove Programs** function from within the Windows Control Panel.
- Using the uninstall program that is supplied with the Virtual Tape Server (available for *all platforms*). You can find this program in the **_uninst** sub-directory that is located under the directory where you installed the Virtual Tape Server.

Starting the Virtual Tape Server

To start the Virtual Tape Server you may either click on the icon **Start Server** in the Program Group **Virtual Tape** (available on Windows only), or execute one of the script files **run.bat** or **run.cmd** (Windows) or **run.sh** (Linux or UNIX). The run script adds the *VirtualTape.jar* to the classpath and starts the Virtual Tape Server:

```
set classpath=.;VirtualTape.jar;%classpath%
java com.ibm.vse.vtape.VirtualTapeServer
```

Defining the Tape Device

A virtual tape is added to the system like a physical tape (unit). It must have at least one **cuu** address. For example:

```
ADD cuu,3480
```

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

Note: It is recommended to select for virtual tapes unique *cuu* numbers that are not used by physical tape units in order to avoid any tape handling or tape operation problems.

Starting, Stopping, and Cancelling Virtual Tapes

You start and stop the Virtual Tape Support using the **VTAPE START** and the **VTAPE STOP** command. Examples are shown in the following topics.

The commands can be issued from both, a static or a dynamic partition, and also from a REXX procedure via the JCL host command environment (ADDRESS JCL).

Every job that contains a `VTAPE START,UNIT=`*cuu*`...` statement should contain an `ON $CANCEL GOTO` *label* statement, where *label* references the `VTAPE STOP,UNIT=`*cuu* command. By using this convention, you ensure that virtual tapes will *always* be closed at the end of the job, even if the job is cancelled.

Using the SCOPE=JOB Parameter:

- If you specify the parameter SCOPE=JOB with a VTAPE START command, you can limit the lifetime of a VTAPE definition to the lifetime of the currently-submitted job.
- Using SCOPE=JOB parameter, z/VSE will *automatically* invoke VTAPE STOP processing during the processing of the `/& (EOJ)` statement. You are *not* required to issue a further VTAPE STOP command.
- Examples of jobs that include the use of the SCOPE=JOB parameter are shown in Figure 65 on page 253, Figure 66 on page 254, and Figure 67 on page 254.

Starting and Stopping the Virtual Tape Data Handler

This topic provides some background information on the startup of the Virtual Tape Data Handler. The Virtual Tape Data Handler is started automatically when the first VTAPE START command or statement is submitted. This causes the startup job TAPESRVR, stored in the VSE/POWER reader queue, to be released. By default, the Virtual Tape Data Handler runs in a **dynamic partition** of **class R**.

Note: You should view the Virtual Tape Data Handler as a subsystem of your z/VSE system. Therefore, if you wish to terminate virtual tape processing, you should **not** cancel the Virtual Tape Data Handler.

The Virtual Tape Data Handler requires the C-Runtime Library.

The startup job TAPESRVR is placed in the VSE/POWER reader queue either:

- during initial installation of z/VSE.
- during a cold startup of z/VSE.

z/VSE provides in VSE/ICCF library 59 skeleton SKVTASTJ which includes startup job TAPESRVR (Figure 63 on page 247).

```
* $$ JOB JNM=CATSTVTA,DISP=D,CLASS=0
// JOB CATSTVTA       CATALOG TAPESRVR AND LDVTA, LOAD TAPESRVR
// EXEC LIBR,PARM='MSHP'
ACC S=IJSYSRS.SYSLIB
CATALOG  TAPESRVR.Z        REPLACE=YES
// JOB TAPESRVR START UP VSE TAPE SERVER
// ID USER=VCSRV
// OPTION SYSPARM='00'
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.SCEEBASE)
// EXEC $VTMAIN,SIZE=$VTMAIN
$$/*
$$/&
$$$$ EOJ
/+
CATALOG  LDVTA.PROC     REPLACE=YES DATA=YES
// EXEC DTRIINIT
   LOAD TAPESRVR.Z
/*
/+
/*
// EXEC PROC=LDVTA        TO LOAD  TAPE SERVER  INTO RDR QUEUE
/&
* $$ EOJ
```

Figure 63. Skeleton SKVTASTJ (for Starting the Virtual Tape Data Handler)

The skeleton does the following:

1. Catalogs startup job as TAPESRVR.Z into IJSYSRS.SYSLIB.
2. Catalogs procedure LDVTA which loads TAPESRVR.Z into the reader queue via DTRIINIT.
3. Executes LDVTA.PROC. If you do not want to load the startup job immediately into the active reader queue, delete the line // EXEC PROC=LDVTA.

The job stream loads the Virtual Tape Data Handler into a dynamic partition of class R which occupies 8 MB. 8 MB is the minimum required.

If you create a new TAPESRVR job, you must ensure that this new job is released by the VTAPE START command. To do so, delete the previous TAPESRVR job in the VSE/POWER reader queue.

Here are some guidelines for stopping the Virtual Tape Data Handler:

- If you want to terminate the Tape Data Handler subsystem, use the `VTAPE STOP` command to close all virtual tapes. The Tape Data Handler will then automatically terminate after 30 seconds.
- To check which tape units are defined as virtual, use the `VTAPE QUERY` command.
- If the Tape Data Handler is cancelled because of an error condition (for example, a program check), to avoid possible inconsistencies in the system you should close all open virtual tapes. To see which virtual tapes are open, use the AR `VOLUME` command. A tape device type of `VTAP-00` in the command output indicates that the device is a virtual tape.

Working with VSE/VSAM Virtual Tapes

To work with VSE/VSAM virtual tapes, you must first define the VSE/VSAM ESDS file which is to contain the virtual tape. This can be done via either the:

- Dialog *Define a New File* (**Fast Path 222**).
- VSE/VSAM IDCAMS job stream provided in skeleton SKVTAPE (which is available in VSE/ICCF library 59).

You can then use the JCL command VTAPE to open the VSE/VSAM file as a virtual tape which can be accessed via the *cuu* specified. For example:

```
VTAPE START,UNIT=cuu,LOC=VSAM,FILE='vsamfilename',SCRATCH
```

In the example, the SCRATCH parameter is optional and causes the virtual tape to be cleared before new data is written to it.

```
VTAPE STOP,UNIT=cuu
```

closes the virtual tape file and drops the association between the tape unit *cuu* and the virtual tape file. If this was the only active VTAPE session, the z/VSE partition in which the Virtual Tape Data Handler is running is released after 30 seconds.

Please note that from z/VSE 4.1 onwards, you are no longer required to have the DLBL for the VSAM files that are used with z/VSE VTAPE as *system standard labels*. Instead, you can specify the DLBLs directly in the job from which the VTAPE START command is issued. The VTAPE START command will then transfer the label information to the tape server partition.

For example, using this support you can:

1. Define a new VSAM file (using the IDCAMS DEFINE CLUSTER command).
2. Use the newly-created cluster in the same job that uses with a virtual tape.

You are no longer required to add the DLBL to the system standard labels.

VSE/VSAM ESDS File Definition (Skeleton SKVTAPE)

Skeleton SKVTAPE creates a VSE/VSAM ESDS file that is to contain a virtual tape. Figure 64 on page 249 shows the skeleton.

```
* $$ JOB JNM=SKVTAPE,CLASS=0,DISP=D
// JOB SKVTAPE     CREATE VIRTUAL TAPE FILE
// EXEC IDCAMS,SIZE=AUTO
   DELETE (VSE.VTAPE.FILE) PURGE CL -
          CATALOG(VSESP.USER.CATALOG)
   DEFINE CLUSTER (NAME(VSE.VTAPE.FILE) -
          RECORDS (1000 1000) -
          TO (99366) -
          REUSE -
          NONINDEXED -
          SHAREOPTIONS (1) -
          SPANNED -
          RECORDSIZE (32758 32758) -
          VOLUMES (-V001-)) -
          DATA (NAME(VSE.VTAPE.FILE.@D@) -
          CISZ (32768)) -
          CATALOG (VSESP.USER.CATALOG)
    IF LASTCC NE 0 THEN CANCEL JOB
/*
// OPTION STDLABEL=DELETE
VTAPE1
/*
// OPTION STDLABEL=ADD
// DLBL VTAPE1,'VSE.VTAPE.FILE',99/366,VSAM,CAT=VSESPUC
/*
// EXEC IESVCLUP,SIZE=AUTO             ADD LABEL TO STDLABUP PROC
D                                              VTAPE1
A VSE.VTAPE.FILE                               VTAPE1  VSESPUC
/*
/&
* $$ EOJ
```

Figure 64. Skeleton SKVTAPE

Explanations to skeleton SKVTAPE:

-V001- This variable defines the volume on which the VSE/VSAM virtual tape file is located. The VSAM space is part of the VSE/VSAM user catalog VSESPUC (VSESP.USER.CATALOG).

RECORDS
The number of RECORDS depends on the amount of data the virtual tape is to contain. The definition provided in SKVTAPE results in a file size of about 32MB. If this size does not meet your requirements, modify the definitions in SKVTAPE accordingly.

REUSE
REUSE allows repeated writing to a virtual tape from the beginning. When specifying the SCRATCH parameter in the VTAPE START command, the tape is cleared before new data is written to it. Without the SCRATCH parameter, existing data is overwritten. Refer also to "Writing to VSE/VSAM Virtual Tapes" on page 250.

NOREUSE is for creating a new and empty file for writing data to it once; for backup, for example. NOREUSE together with SCRATCH is invalid and causes an error message.

SHAREOPTIONS
SHAREOPTIONS (1) allows multiple READ operations or one WRITE operation to be active. It is recommended not to change the setting of the share option. Refer to "File Sharing" on page 250 for further details.

RECORDSIZE
The value for the RECORDSIZE parameter is the CISZ value minus 10 bytes (CISZ minus 20 bytes in case of a compressed cluster).

CISZ The recommended value for the Control Interval Size is 32 KB; it should not be smaller than 8 KB.

File Size

The 4GB limit for VSE/VSAM files applies also for VSE/VSAM virtual tape files.

File Name

In the skeleton, **VTAPE1** in the DLBL statement is the filename to be used in the VTAPE START command.

File Sharing

Whatever is defined in SHAREOPTIONS, the Virtual Tape Data Handler accepts only single WRITE **or** multiple READ access to a VSE/VSAM virtual tape from a single z/VSE system. This corresponds to SHAREOPTIONS (1). If you want to access a VSE/VSAM virtual tape from more than one z/VSE system, use SHAREOPTIONS (1) which allows single WRITE **or** multiple READ access only. SHAREOPTIONS (1) avoids unpredictable results where multiple system access is being used.

Writing to VSE/VSAM Virtual Tapes

WRITE can only start at the beginning of a VSE/VSAM virtual tape. No rewrite or overwrite of existing data is possible except from the beginning a tape. For writing to a VSE/VSAM virtual tape more than once, the file definition (skeleton SKVTAPE) must include the REUSE parameter. Once a VTAPE STOP command has been issued, the remaining buffer contents is written to tape, the tape is closed, and an end-of-volume (EOV) indicator is written. This means, no further data can be appended (added) to this particular virtual tape.

Note that VSAM cluster can be defined as compression, but the compression rate can vary a large amount (depending on the type of data).

There are three ways of writing to a VSE/VSAM virtual tape:

1. The VSE/VSAM file is defined with REUSE.

   During OPEN processing the tape is positioned to its beginning before a WRITE operation starts. Existing data is overwritten.
2. The VSE/VSAM file is defined with REUSE and VTAPE START includes the SCRATCH parameter.

   SCRATCH causes the tape to be cleared before WRITE starts at the beginning of the tape.
3. The VSE/VSAM file is defined with NOREUSE.

   You can write to the tape only once (for backup, for example).

   NOREUSE with SCRATCH is invalid and causes an error.

Working with Remote Virtual Tapes

To access a remote virtual tape, a TCP/IP connection must be established between z/VSE and the remote workstation with the **Virtual Tape Server** installed. You start the VTAPE support for a remote virtual tape with a VTAPE command such as the following:

```
 VTAPE START,UNIT=cuu,LOC=ipaddress:portnumber,FILE='filename'
(or)
 VTAPE START,UNIT=cuu,LOC=hostname:portnumber,FILE='filename'
```

The *ipaddress* or *hostname* identifies the Virtual Tape Server workstation on which the remote virtual tape file is located (a Linux, UNIX, or Windows file, for example). The *portnumber* is the TCP/IP port number to be used for the connection. If the virtual tape file does not yet exist, the command automatically causes the creation of the required file (Linux, UNIX, or Windows) using the *filename* specified. The VTAPE command:

```
 VTAPE STOP,UNIT=cuu
```

drops the association between the tape unit *cuu* and the remote virtual tape file. If this was the only active VTAPE session, the z/VSE partition in which the Virtual Tape Data Handler is running is released after 30 seconds.

Note: If a remote virtual tape has been started whose TCP/IP connection is slow, this might have a negative impact on the performance of other virtual tapes. To avoid this problem, you should ensure that the TCP/IP connection to the virtual tape is reliable and fast enough to be able to process high data-transfer rates.

Entering a File Name Under Linux, UNIX, or Windows

If the required Linux, UNIX, or Windows file for a remote virtual tape does not exist yet, it is automatically created when submitting the corresponding VTAPE START command. When assigning file names you must observe certain rules and characteristics.

File names may contain blanks, therefore the *filename* must be enclosed in quotes. A quote within a *filename* must be coded as two single quotes; for example:

```
 FILE='D:\John''s\Virtual Tapes\vt001401.001'
```

File *names* can have more than 100 characters in length, therefore you may specify `FILE='filename'` twice or even three times. However, each file *parameter* cannot be greater than 100 characters in length. Therefore, the *filename* is concatenated in storage, allowing for a file name length of 200 or even 300. The following example is equivalent to the previous example:

```
 FILE='D:',FILE='\John''s\Virtual Tapes\',FILE='vt001401.001'
```

Case Sensitivity Under Linux and UNIX

Linux and UNIX are case sensitive but job streams created on the z/VSE host (dialogs of the Interactive Interface) are in capital letters. It may be therefore necessary to edit such job streams to adapt the *filename* to the Linux or UNIX conventions.

Using Forward or Backward Slashes Under Windows

You can use forward slashes ("/") instead of backward slashes ("\") on Windows (as is the case with Linux and UNIX).

If you wish to use the VTAPE command with Windows, you will probably use backward-slashes to separate the directories. An example would be:

```
C:\vtape\tapeiamge.aws
```

However, the use of backward-slashes might cause code-page errors during the translation from EBCDIC to ASCII. Therefore, when using the VTAPE command with Windows *you are strongly recommended to use forward-slashes*. An example would be:

```
C:/vtape/tapeiamge.aws
```

If you use backward-slashes, the filename on Windows might be treated as a relative-path instead of an absolute-path. As a result, the tape image will be created in the Virtual Tape Server's installation directory! This occurs because Windows does not recognize the path as an absolute-path if backward-slashes are translated into some incorrect characters.

Forward-slashes do not *usually* cause code-page translation errors. In addition, the Java runtime environment will automatically convert forward-slashes into backward-slashes on Windows.

The Virtual Tape Server can accept both forward and backward slashes on Windows.

Further Documentation

The workstation with the Virtual Tape Server installed provides the document **vtape.html**. It includes further details about remote virtual tapes and the related environment. You reach the document by selecting from the Start Menu first *Virtual Tape*, and then *Help.*

Examples of Using Virtual Tapes

Backing Up and Restoring Data

Backup functions often produce tape output. Thus you may consider whether virtual tapes are an alternative for such tasks in your data processing environment. For temporary data or short term backups in a z/VSE environment, for example, VSE/VSAM virtual tapes avoid the overhead of mounting and handling physical tapes.

You can use virtual tapes for all types of backup and restore operations (involving VSAM objects, library objects, history files, volumes, and files). For details, refer to the chapter "Backing Up and Restoring Data" in the manual *z/VSE Operation*.

Following is a backup example using a remote virtual tape. Data on a remote virtual tape can be saved on CD-ROM for archiving or for distribution. In the example, a VSE library backup and a later restore is assumed.

1. Create a job stream to backup a VSE library on a remote virtual tape as shown in the following sample job stream:

```
* $$ JOB JNM=BACKUP,CLASS=0,DISP=D
// JOB BACKUP (Backup a Library to a PC File)
VTAPE START,UNIT=480,LOC=9.164.186.20:2285,                          C
               FILE='D:\VSE Backup\prd2.001',SCOPE=JOB
MTC REW,480
// EXEC LIBR
BACKUP LIB=PRD2 TAPE=480
/*
/&
* $$ EOJ
```

Figure 65. Job Stream Example for Backing Up a z/VSE Library

2. Use the programs available on your Linux or Windows system to copy (burn) the VSE library from the remote virtual tape to a CD-ROM for archiving.

   Depending on the CD-ROM software you have installed, you may create the virtual tape backup of the VSE library directly on the CD-ROM.
3. To restore the VSE library you must create a similar job stream as the one shown for BACKUP, using the Librarian RESTORE command. You can restore the VSE library (stored as virtual tape on CD-ROM) directly from the CD-ROM back to z/VSE.

Transferring Virtual Tape Files

It is possible to transfer virtual tape files between a workstation (remote virtual tape) and the z/VSE host (VSE/VSAM virtual tape) and vice versa with the TCP/IP File Transfer Program (FTP). For a transfer from workstation to host use the **put** command, for a host to workstation transfer the **get** command. Make sure you are transferring the files in **binary** mode.

The example below shows the command sequence for a transfer from server to host:

```
C:\>ftp x.x.x.x                    <-- enter hostname/IP address of your VSE system
Connected to x.x.x.x.

:
220 Service ready for new user.
User (9.164.155.2:(none)): user      <-- enter your user id here
331 User name okay, need password.
Password:                            <-- enter your password here
230 User logged in, proceed.
ftp> bin                             <-- switch to binary mode
200 Command okay.
ftp> quote site lrecl 32758          <-- enter the record size of your VSAM cluster
200 Command okay.
ftp> quote site recfm v              <-- set record format to variable
200 Command okay.
ftp> put tape.image VSE.VTAPE.FILE   <-- enter your filenames
```

Backing Up Data Using the Tivoli Storage Manager

The *Tivoli® Storage Manager* (TSM) is an IBM software product that allows you to backup and restore data over the network. The TSM:
- Manages the storage.
- Controls where the backup data is to be stored.

The TSM runs on a wide range of platforms, including Linux on System z®. If the TSM is already used at your site, z/VSE can be integrated into either new or existing TSM-controlled backup environments.

The TSM manages several *storage pools*, which can reside on disk or tape. The administrator defines the rules which determine:
- How the storage pools are used.
- When data should be migrated from one pool to another.

You can use the *VTAPE command* to back up z/VSE data to the TSM:
- Remote VTAPEs *only* are supported.
- You can encrypt data before it is sent over the network to the TSM server.
- Your established backup jobs on z/VSE can usually be run *almost unchanged*.

Here is a typical *VSAM backup job* that uses the TSM, where *catalog*, *ip-addr*, and *cluster* would be replaced with real values:

```
* $$ JOB JNM=VSAMBKUP,DISP=L,CLASS=0
// JOB VSAMBKUP
* THIS JOB BACKS UP VSAM DATASETS
// DLBL IJSYSUC,' catalog ',,VSAM
VTAPE START,UNIT=181,LOC=ip-addr,FILE='TSM:VSAM.AWS(BACKUP,VSE)',       C
               SCRATCH,SCOPE=JOB
// ASSGN SYS005,181
// EXEC IDCAMS,SIZE=AUTO
BACKUP ( cluster ) -
REW -
NOCOMPACT -
BUFFERS(3)
/*
// ASSGN SYS005,UA
/&
* $$ EOJ
```

Figure 66. Backup Job That Uses the Tivoli Storage Manager and Virtual Tapes

Here is a typical *VSAM restore job* that uses the TSM (which corresponds to the backup job), where *ip-addr*, *cluster*, and *catalog* would be replaced with real values:

```
* $$ JOB JNM=VSAMREST,DISP=L,CLASS=0
// JOB VSAMREST
* THIS JOB RESTORES VSAM DATASETS
VTAPE START,UNIT=181,LOC=ip-addr,FILE='TSM:VSAM.AWS(BACKUP,VSE)',       C
               READ,SCOPE=JOB
// ASSGN SYS004,181
// EXEC IDCAMS,SIZE=AUTO
RESTORE OBJECTS ( ( cluster ) ) -
CATALOG( catalog ) -
REW -
BUFFERS(3)
/*
// ASSGN SYS004,UA
/&
* $$ EOJ
```

Figure 67. Restore Job That Uses the Tivoli Storage Manager and Virtual Tapes

For further details about using the:
- Tivoli Storage Manager, see -
  `http://www.ibm.com/software/tivoli/products/storage-mgr/`
- Tivoli Storage Manager together with VTAPE, see -
  `http://www.ibm.com/systems/z/os/zvse/support/vtape.html`

Chapter 21. Implementing Tape Library Support

This chapter describes how you can configure tape libraries in z/VSE using the *Tape Library Support*.

It consists of these main topics:

Related Topics:

For details of how to ...	Refer to ...
use the *Interactive Interface* to add a tape drive	Chapter 7, "Configuring Disk, Tape, and Printer Devices," on page 87
encrypt tapes using encryption-capable tape drives	Chapter 46, "Implementing Hardware-Based Tape Encryption," on page 521
perform tape library functions (for example, issue a LIBSERV MOUNT or LIBSERV EJECT command), or submit jobs using LIBSERV JCL	"Job Control and Attention Routine" in the *z/VSE System Control Statements*.
use the LBSERV macro	*z/VSE System Macros Reference*.
check (via parameter `DISK-ONLY` of the `QT cuu` command) whether or not a tape library is tapeless	"Job Control and Attention Routine" in the *z/VSE System Control Statements*.

For a summary of the tape libraries that are supported by z/VSE, refer to the chapter "z/VSE 5.1 Hardware Support" in the *z/VSE Planning*.

Overview of Tape Library Support

The z/VSE *Tape Library Support* (TLS) enables z/VSE to access tape libraries via a *channel interface*. As a result:

- The need to use an XPCC or APPC communication protocol is removed.
- The need to set up a VTAM LU6.2 connection when using LCCD is removed.

z/VSE supports these tape libraries:

- IBM TotalStorage 3494 Tape Library
- IBM System Storage TS3500/3584 UltraScalable Tape Library
- IBM System Storage TS7700 Virtualization Engine.
- IBM System Storage tapeless (disk-only) TS7720 Virtualization Engine.
- IBM TS7680 ProtecTIER® Deduplication Gateway for System z.
- IBM TotalStorage Virtual Tape Server (VTS)

For the TS3500/3584 Tape Library:

- The attachment of zSeries hosts to the TS3500/3584 Tape Library is done via the *Enterprise Library Controller* (**ELC**). This attachment takes advantage of the TS3500/3584 Tape Library's support for the:
  - IBM TotalStorage 3592 Tape Drive Model J1A
  - IBM System Storage TS1120 tape drive (also referred to as the 3592 Model E05)
  - IBM System Storage TS1130 tape drive (also referred to as the 3592 Model E06)
  - IBM System Storage TS1140 tape drive (also referred to as the 3592 Model E07)

How Tape Libraries are Configured

Here is an overview of how tape libraries can be configured:

Table 4. Overview of Tape Library Configuration Possibilities

	IBM 3494 Tape Library	IBM TS3500/3584 UltraScalable Tape Library	IBM TS7700 Virtualization Engine [1]	IBM TS7680 ProtecTIER Deduplication Gateway for System z [2]
Which tape drives can be defined/used?	• IBM 3490E • IBM 3590 • IBM 3592 Model J1A • IBM TS1120 (3592 Model E05) • IBM TS1130 (3592 Model E06)	• IBM 3592 Model J1A • IBM TS1120 (3592 Model E05) • IBM TS1130 (3592 Model E06) • IBM TS1140 (3592 Model E07)	• IBM 3490E	• IBM 3592 Model J1A
Run under z/VM using VGS (VSE Guest Server)?	Yes	Yes	Yes	Yes
Run in VSE LPAR mode?	Yes	Only with the: • IBM 3592 Model J1A • IBM TS1120 (3592 Model E05) • IBM TS1130 (3592 Model E06) • IBM TS1140 (3592 Model E07)	Yes	Yes
Run in VSE LPAR mode using TLS (Tape Library Support)?	Yes	Yes	Yes	Yes

Note:

1. The TS7700 1.5 and later support a disk-only configuration of the IBM TS7720 Virtualization Engine.
2. The TS7680 provides a disk-only virtual tape solution, and emulates an IBM tape library and 3592 Model J1A tape drives.
3. The TS1120 is also referred to as the 3592 Model E05. The TS1130 is also referred to as the 3592 Model E06. The TS1140 is also referred to as the 3592 Model E07.

Migrating/Configuring Your z/VSE System for TLS

To prepare your z/VSE system for use with an IBM tape library, you must perform the steps described below:

1. **Add a SYS ATL=TLS Command to Your z/VSE Startup Procedure:** You must add a `SYS ATL=TLS` command to your z/VSE IPL procedure, using the dialog described in "IPL Parameters You Can Modify" on page 15. For details of the SYS command, refer to the manual *z/VSE System Control Statements*, SC34-2637.

2. **Define Customization Options and Control Statements:** You must change the sample job TLSDEF which is provided in ICCF Library 59 to meet your own system requirements. Here is an example job which uses logical addresses 460 to 463, and 580 to 582. All variables are shown in italics.

```
* $$ JOB JNM=TLSDEF,CLASS=0,DISP=D
* $$ LST CLASS=A
// JOB TLSDEF
   EXEC LIBR,PARM='MSHP'
   ACCESS S=IJSYSRS.SYSLIB
   CATALOG TLSDEF.PROC REPLACE=YES
   LIBRARY_ID TAPELIB1 SCRDEF=SCRATCH00 INSERT=SCRATCH00
   LIBRARY_ID TAPELIB2                * SECOND LIB DEF DUAL LIB
   DEVICE_LIST TAPELIB1 460:463       *  DRIVES  460 TO 463
   DEVICE_LIST TAPELIB2 580:582       *  DRIVES  580 TO 582
   QUERY_INV_LISTS LIB=TLSINV         * MASTER INVENTORY FILES
   MANAGE_INV_LISTS LIB=TLSMAN        * MANAGE FROM MASTER
/+
/&
* $$ EOJ
```

These are the keywords and parameters used in job TLSDEF:

LIBRARY_ID

The keyword LIBRARY_ID is followed by the eight-character logical unit name of an attached library. This logical unit name is used as the Library Name field in functional requests from users, and as the z/VSE sub-library name used in inventory requests. If the library name is less than 8 characters, it will be padded to the right with blanks.

- The keyword SCRDEF is followed by the name of the scratch pool to be used as the default for this z/VSE host on this library. Tapes will be mounted from this pool for nonspecific MOUNT SCRATCH requests. If this parameter is not included, SCRATCH00 will be used as the default pool.
- The keyword INSERT allows a target category to be specified for automatic insert processing of new volumes inserted in the library. The target category must be either SCRATCH*nn* (where nn is in the range 00 to 31), or PRIVATE. If this parameter is omitted, no automatic insert processing occurs.

You must define one LIBRARY_ID statement for *each* attached library. If user requests do not specify a library name, the library associated with the first occurrence of this keyword will be used as the default. z/VSE supports a maximum of *eight* libraries.

DEVICE_LIST

Designates the Library_Id and the corresponding devices. When multiple hosts are attached to the same library, this keyword allows device partitioning. This statement may be repeated as required to list all libraries with devices to be used by this host.

QUERY_INV_LISTS
Designates the name (up to seven characters) for the predefined VSE library in which Query Inventory member lists are to be created. If this control card is not found or if it is not coded properly, or if the library and a sub-library for the attached 3494 or 3584 is not defined, Query Inventory requests cannot be processed.

MANAGE_INV_LISTS
Designates the name (up to seven characters) for the predefined library from which Manage Inventory member lists are to be read. If this control card is not found or if it is not coded properly, or if the library and a sub-library for the attached tape library is not defined, Manage Inventory requests cannot be processed.

For details of how to predefine z/VSE libraries for inventory requests, refer to Step 4 (below).

After you have customized the job TLSDEF, submit this job to catalog TLSDEF.PROC into library IJSYSRS.SYSLIB. Your TLSDEF.PROC will then become active the next time an IPL is made of your z/VSE system.

3. **Enter a STDEVOPT Statement in z/VM (Optional)** If your z/VSE system is running under z/VM, you must include this statement in the user directory entry:

```
STDEVOPT LIBRARY CTL
```

Otherwise, the devices will be controlled by z/VM which will return a “command reject” for each request.

4. **Define z/VSE Libraries to Contain Inventory Requests** Before you can start the inventory application, you must create a library into which host copies of tape-library inventory lists can be written. You must create a unique sub-library that has the name of your tape library.

The library into which host copies of tape-library inventory lists can be written requires disk space assigned using DLBL and EXTENT statements. You can add these labels to either the STANDARD or USER STANDARD LABEL procedure. Here is an example of these statements:

```
* DEFINITION FOR THE VSE TLSINV INVENTORY LIBRARY
// DLBL TLSINV,'VSE.TLSINV.QLISTS.LIBRARY',99/365
// EXTENT ,SYSWK1,1,0,9500,100
EXEC LIBR
DEF L=TLSINV
DEF S=TLSINV.TAPELIB1

(Note: the Library and Sublibrary examples are as used in Step 2)
```

Understanding the Format of Inventory Data

Output File Produced by a Query Inventory Request

An 80-byte inventory record output file is created by a Query Inventory request. This file applies to a cartridge and contains the following information:

- External volume label (six characters)
- Media type:
  - CST1 or CST2 for 3490 cartridges.
  - CST3 or CST4 for 3590 cartridges.
  - CST5 for 3592 300 GB cartridges.
  - CST6 for 3592 WORM (Write-Once-Read-Many) 300 GB cartridges.
  - CST7 for 3592 60 GB cartridges.
  - CST8 for 3592 WORM 60 GB cartridges.
  - CST9 for 3592 700 GB.
  - CSTA for 3592 WORM 700 GB.
  - CSTB for 3592 3200 GB.
  - CSTC for 3592 WORM 3200 GB.
  - CSTD for 3592 Economy 400 GB.
- Special attribute byte, represented by an EBCDIC bit string (eight characters)

  Bit 0 If 1, volume is present in library, but inaccessible
  Bit 1 If 1, volume is mounted or queued for mount
  Bit 2 If 1, volume is in eject-pending state
  Bit 3 If 1, volume is in process of ejection
  Bit 4 If 1, volume is misplaced
  Bit 5 If 1, volume has unreadable label or no label
  Bit 6 If 1, volume was used during manual mode
  Bit 7 If 1, volume was manually ejected
- Category name (ten characters)
- Library manager hexadecimal category number, in EBCDIC representation (four characters)

The fields in each record are separated by a blank character to enhance their readability.

A sample file record is:

```
CS0010 CST2 01000000 PRIVATE    FFFF
```

A header with the time of list creation is inserted as the first record in the list.

Input File Submitted by a Manage Inventory Request

A file submitted for use in a Manage Inventory request requires that each six-character external volume serial number in the list start in column 1 of a file record. The remaining space in each 80-character record is ignored as input. This allows for returning the Query Inventory output file as input to the Manage Inventory function.

A sample record in a Manage Inventory input file is:

```
CS0010 CST2 01000000 PRIVATE    FFFF
```

Or, simply

```
CS0010
```

A header record (as described for Query Inventory output) may be optionally present in the input list. Any record starting with an asterisk (*) is not considered a valid input data record and is ignored.

Output Produced by a Manage Inventory Request

After a Manage Inventory request is completed, a return code (or reason code) is supplied to indicate that processing is complete and to report the overall results for processing the request (for example, input was valid, file was found, and so forth). The actual outcome of transferring each volume to a new target category is reflected within the file itself. The file is updated by adding a results message in each file record, starting in column 38. An example of a successful output file record is:

```
CS0010 CST2 01000000 PRIVATE    FFFF    CATEGORY CHANGED TO EJECT
```

An example for an unsuccessful output file record is:

```
AB1234     CATEGORY NOT CHANGED, RSN=3340
```

Naming Conventions for Inventory Files

Table 5 summarizes the naming conventions for inventory files used in the Query Inventory (IQUERY) functions. VSE libraries must be predefined, together with a sub-library for each attached 3494 or 3584.

Table 5. Naming Conventions for Inventory Files

Source Category	Member Name
(omitted)	ALL
PRIVATE	PRIVATE
INSERT	INSERT
SCRATCHnn	SCRnn
SCRATCH	SCRnn (see Note 1)
MANEJECT	MANEJECT
EJECT	EJECT
VOL	VOL

Note:

1. In the above table entry for SCRATCH, *nn* refers to the default scratch pool that was specified using the SCRDEF keyword (see Step 2 of "Migrating/ Configuring Your z/VSE System for TLS" on page 257).
2. There are no naming conventions for the *Manage Inventory* (MINVENT) function. You can specify MEMNAME according to your own naming conventions.

Performing Tape Library Functions

IBM tape library communication is performed using the JCL LIBSERV statements described in Table 6. For details of these commands, refer to the manual *z/VSE System Control Statements*.

Table 6. Overview of LIBSERV Commands Used With an IBM Tape Library Data Server

Command	Description
LIBSERV AQUERY	Verify the location of a volume in all known libraries.
LIBSERV CQUERY	Return the number of volumes in a source category.
LIBSERV CMOUNT	Mount a volume from a source category on a device belonging to a tape library. Access will then be given to the device.

Table 6. Overview of LIBSERV Commands Used With an IBM Tape Library Data Server (continued)

Command	Description
LIBSERV COPYEX	Move copies of logical volumes to an offsite location for a future possible disaster recovery (see "Using the Copy Export Facility for a Disaster Recovery").
LIBSERV DQUERY	Verify the status of a device in a tape library.
LIBSERV RELEASE	Release a device in a tape library.
LIBSERV EJECT	Eject a volume.
LIBSERV IQUERY	Request the inventory data of volumes currently assigned to a category in a tape library. See "Output File Produced by a Query Inventory Request" on page 259.
LIBSERV LQUERY	Return the operational status of a tape library.
LIBSERV MOUNT	Mount a specified volume.
LIBSERV MINVENT	Transfer the volumes in a referenced member name and source category, to a target category.
LIBSERV SETVCAT	Assign a volume to a target category.
LIBSERV SQUERY	Verify the location of a volume in a library.

Using the Copy Export Facility for a Disaster Recovery

This topic describes how you can use the *Copy Export* facility of the IBM TS7700 Virtualization Engine (for example, provided via Model TS7740) to "copy export" selected logical volumes to an offsite location.

This topic contains these sub-topics:
- "Overview of the Copy Export Feature" on page 262
- "Prerequisites for Using the Copy Export Feature" on page 263
- "Restrictions of the Copy Export Feature" on page 263
- "Performing a Copy Export Operation" on page 263

Related Topics:

For details of how to ...	Refer to ...

• set up the Copy Export hardware feature
• perform a disaster recovery (DR)
	• IBM Redbook *IBM Virtualization Engine TS7700 Release 1.7: Tape Virtualization for System z Servers* that you can find at:
	`http://www.redbooks.ibm.com/redbooks/pdfs/sg247712.pdf`
	• IBM White Paper *IBM Virtualization Engine TS7700 Series Copy Export Function, User's Guide Version 1.7* that you can find at:
	`http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101092`

use the LIBSERV COPYEX JCL statement	*z/VSE System Control Statements*, SC34-2637
use the LBSERV COPYEX macro	*z/VSE System Macros Reference*, SC34-2638

Overview of the Copy Export Feature

Using the *Copy Export* feature, you can “export” a copy of selected logical volumes to an offsite location.

- These logical volumes are referred to as a “pool” of volumes.
- On a second TS7700 Virtualization Engine, this “pool” of volumes is then used for disaster recovery (DR) purposes.

A Copy Export operation is performed using one of these methods:

- By tailoring and then submitting job SKCOPYEX, an example of which is shown in Figure 69 on page 265.
- By executing the LIBSERV COPYEX command using a pre-initialized logical volume.

The Copy Export feature makes use of *volume stacking*, which places many logical volumes on a physical volume. In addition, since the data being exported is a copy of the physical volume, the logical volume data remains accessible by the production host systems.

Figure 68 shows the general flow of actions during a Copy Export operation of physical volumes as the “Pool 09” logical volumes.

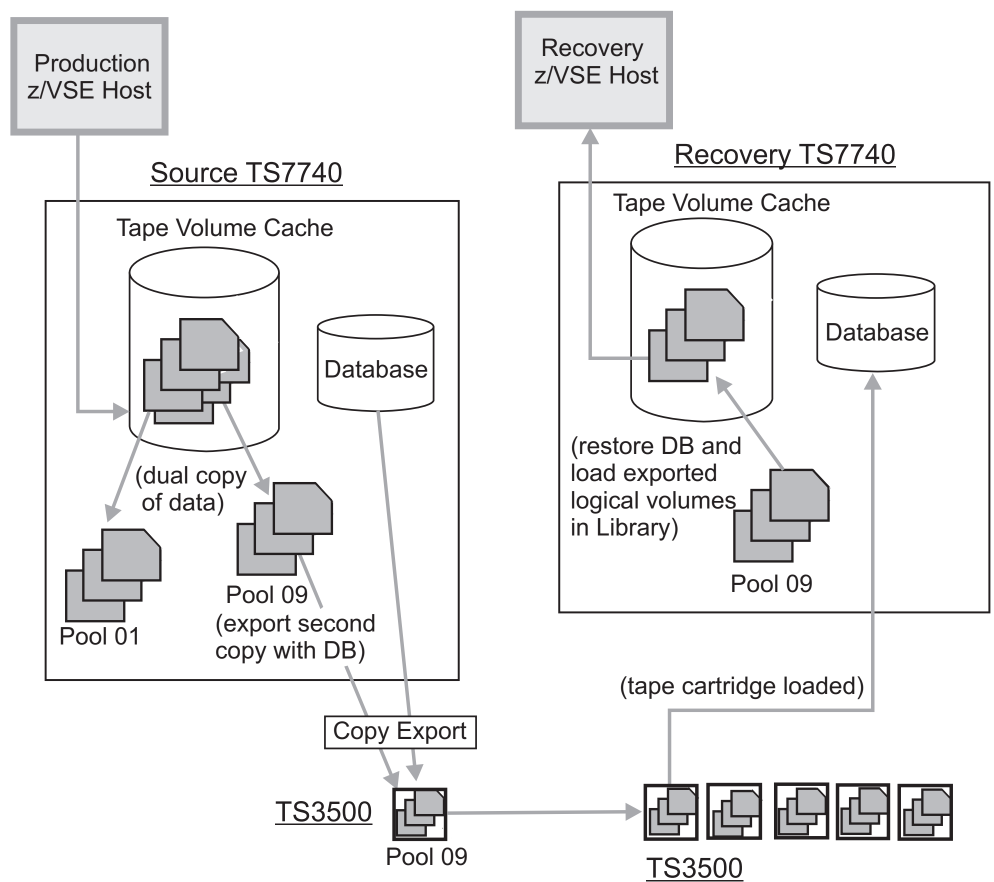


Figure 68. Example of a Copy Export Operation

During a Copy Export operation:

- All logical volumes containing active data in a specified secondary pool are exported as logical volumes from the library associated with the TS7700 that is being used for the Copy Export operation.
- A copy of the current TS7700's database is also exported together with the physical volumes.

To restore access to data on the exported physical volumes:

1. All exported physical volumes are placed into a library that is attached to an empty TS7700.
2. The restore process is started from the recovery TS7700 *Management Interface (MI)* panel.

Prerequisites for Using the Copy Export Feature

These are the prerequisites for using the Copy Export feature:

- The z/VSE operating system must be Version 5 Release 1 or later.
- You must have installed and configured a TS7700 Virtualization Engine (for example, Model TS7740) that supports the Copy Export feature.

Restrictions of the Copy Export Feature

These are the current restrictions that apply when using the Copy Export feature:

- z/VSE does not support *Policy management*. Therefore, you must use the TS7700 Management Interface (MI) panel to assign a range of logical volumes to a management class.
- When running z/VSE under z/VM, the Copy Export feature is not currently supported for use with the VSE Guest Server (VGS) interface. However, if you wish to process a Copy Export when running with VGS, you must:
  1. Tailor the member TLSDEF (stored in VSE/ICCF Library 59).
  2. Submit the job TLSDEF.
  3. IPL your z/VSE system.
  4. Attach a TS7700 device to your z/VSE system.
  5. Run the Copy Export operation.

Performing a Copy Export Operation

These are the actions you take in order to perform a Copy Export operation:

1. Since z/VSE does not support *Policy management*, use the *TS7700 Management Interface* (MI) to assign a range of logical volumes to a management class.
   - If the logical volumes have *not* been inserted, you can use the *Insert Logical Volumes* panel to assign storage constructs to a logical volume.
   - If the logical volumes *have* already been inserted, you can use the *Modify Logical Volumes* panel to assign storage constructs.

     Note: You must perform this action *before* a logical volume is mounted. This would mean that a predetermined set of logical volumes would need to be used for export or all logical volumes must be mounted after modification.

   For details of how to use the *TS7700 Management Interface*, refer to the topic "Implementing Outboard Policy Management for Non-z/OS Hosts" in the Redbook whose URL is given in "Related Topics" on page "Using the Copy Export Facility for a Disaster Recovery" on page 261.

2. Tailor the skeleton SKCOPYEX (shown in Figure 69 on page 265) to meet your requirements. Skeleton SKCOPYEX is stored in VSE/ICCF Library 59. You must supply:
   - A tape device address in the UNIT= statement.
   - A tape library address in the LIB= statement.
   - A logical tape volume serial number in the VOL= statement.
   - A value for a tape label in the TAPE= statement.
   - These *predefined* keywords: `ELFV=CR_F1`, `ELFV=CR_F2`, and `ELFV=CR_F3`. The three predefined keywords (where CR_F is a short-form for ContRol File) provide pool information for the Copy Export operation.

   Note: Also ensure that the ELFV and TAPE keywords used in skeleton Figure 69 on page 265 are consistent with the processing and SYSIPT information.

```
* $$ JOB JNM=SKCOPYEX,CLASS=0,DISP=D
* $$ LST CLASS=A
// JOB SKCOPEX
⋮
* STEP 1 : CREATE COPY EXPORT LIST FILE VOLUME *
* *******************************************************
* PLEASE CHANGE THE ASSGN AND LIBSERV STATEMENT TO MATCH *
* THE CUU ,VOLSER AND TAPE LIBRARY NAME IN THE TS7700    *
* ATTENTION: - ALL DATA ON THE TAPE WILL GET DELETED     *
* - DO NOT SPECIFY COMPACTION MODE IN ASSIGN             *
* *******************************************************
// ASSGN SYS005,CUU
// LIBSERV MOUNT,UNIT=CUU,VOL=VOLSER/W,LIB=LIBRARY
// TLBL COPYTP,'COPY.EXPORT.VOL',,VOLSER
* *******************************************************
* FILE 1 : EXPORT LIST                                   *
* PLEASE CHANGE XX TO CORRESPONDING PHYSICAL VOLUME POOL *
* THAT CONTAINS THE LOGICAL VOLUMES TO EXPORT            *
* *******************************************************
// EXEC IJBCPYEX,SIZE=IJBCPYEX,PARM='TAPE=DD:SYS005-COPYTP ELFV=CR_F1'
EXPORT LIST 03
EXPORT PARAMETERS PHYSICAL POOL TO EXPORT:XX
OPTIONS1, COPY,EJECT
/*
* *******************************************************
* FILE 2 : RESERVED FILE                                 *
* THE RESERVED FILE MUST BE PRESENT(FOR FUTURE USE)      *
* *******************************************************
// EXEC IJBCPYEX,SIZE=IJBCPYEX,PARM='TAPE=DD:SYS005-COPYTP ELFV=CR_F2'
RESERVED FILE
/*
* *******************************************************
* FILE 3 : EXPORT STATUS FILE                            *
* CHECK THIS FILE AFTER THE EXPORT OPERATION IS COMPLETED*
* FOR COPY EXPORT RESULTS                                *
* *******************************************************
// EXEC IJBCPYEX,SIZE=IJBCPYEX,PARM='TAPE=DD:SYS005-COPYTP ELFV=CR_F3'
EXPORT STATUS 01
/*
// LIBSERV RELEASE,UNIT=CUU
// PAUSE
* *******************************************************
* STEP 2 : INITIATE THE COPY EXPORT OPERATION            *
* *******************************************************
* INITIATE COPY EXPORT OPERATION AT THE TS7700           *
* PLEASE CHANGE VOLSER AND LIBRARY                       *
* *******************************************************
// LIBSERV COPYEX,VOL=VOLSER,LIB=LIBRARY
/&
* $$ EOJ
```

Figure 69. Skeleton SKCOPYEX for Performing a Copy Export Operation

3. Submit job SKCOPYEX. This job executes utility program *IJBCPYEX*, which “prepares” a tape to be used for a Copy Export operation by creating an *Export List File Volume* (ELFV) on tape. This process is the prerequisite for the Copy Export operation that is executed by the LIBSERV COPYEX command.

   For a detailed description of how to prepare a tape for a Copy Export operation, refer to the White Paper listed in “Related Topics” on page “Using the Copy Export Facility for a Disaster Recovery” on page 261.

4. Wait for this completion message AOMAP17I to be returned from job SKCOPYEX:

   ```
   AOMAP17I COPYEX OPERATION COMPLETE for VOLID=...... RC=xx
   ```

- If message AOMAP17I does *not* contain a return code of zero, determine the cause of the failure and correct the problem.
- If message AOMAP17I *does* contain a return code of zero, proceed to (5.) below.

5. Tailor skeleton SKCPEXRD (shown in Figure 70) by entering your own values for:
   - A tape device address in the UNIT= statement.
   - A tape library address in the LIB= statement.
   - A logical tape volume serial number in the VOL= statement.

   Skeleton SKCPEXRD is stored in VSE/ICCF Library 59.

   Note: Also ensure that the ELFV and TAPE keywords used in skeleton Figure 70 are consistent with the processing and SYSIPT information.
6. Submit job SKCPEXRD. When the job has completed, the results of the Copy Export operation (the "status") will be written to SYSLST.

```
* $$ JOB JNM=SKCPEXRD,CLASS=0,DISP=D
* $$ LST CLASS=A
// SKCPEXRD
* *******************************************************
* MOUNT VOLUME WITH EXPORT LIST AGAIN TO READ THE EXPORT *
* STATUS FILE (FILE 3 ON THE VOLUME)                     *
* *******************************************************
// ASSGN SYS005,CUU
// LIBSERV MOUNT,UNIT=CUU,VOL=VOLSER,LIB=LIBRARY
// MTC REW,CUU
// TLBL COPYTP,'COPY.EXPORT.VOL',,,,3
* ******************************************************* *
* READ EXPORT STATUS FILE                                 *
*         PLEASE VERIFY THAT COPY EXPORT COMPLETED BY     *
*         MSG AOMAP17I COPYEX OPERATION COMPLETED         *
*         BEFORE YOU CONTINUE                             *
* ******************************************************* *
// PAUSE
// EXEC IJBCPYEX,SIZE=IJBCPYEX,PARM='TAPE=DD:SYS005-COPYTP ELFV=RD_F3'
READ STATUS
/*
// LIBSERV RELEASE,UNIT=CUU
/&
* $$ EOJ
```

Figure 70. Skeleton SKCPEXRD for Obtaining an Export Status File

Part 3. BSM AND LDAP SECURITY

Chapter 22. Roadmap/Overview of BSM-Based Security

This chapter provides a roadmap and overview of the security that is available from *z/VSE V3R1.1 onwards*.

It consists of these main topics:
- "Roadmap for Using BSM-Based Security"
- "General Aspects" on page 273
- "Overview Diagram of BSM-Based Security" on page 274
- "General Concept of Access Control" on page 276

Roadmap for Using BSM-Based Security

To establish an operational security system, you might find it useful to follow this "roadmap":

1. The *general aspects* which you should consider when setting up security for your z/VSE system are described below. This chapter also includes an *overview diagram* of the complete BSM-based security (for z/VSE and CICS) which is shown in "Overview Diagram of BSM-Based Security" on page 274.
2. The *installation tasks* that you must complete are described in the following topics:
   a. The initial tasks you must complete to have security installed on your z/VSE system are described in Chapter 23, "Implementing z/VSE Security Support," on page 277.
   b. The setting up of security options, such as:
      - active resource classes,
      - password rules,
      - audit options.

      are described in "PERFORM | PF Command" on page 367.
   c. The commands that you use to control the security server are described in "Controlling the Security Server Partition" on page 426.
   d. How you migrate CICS transaction security definitions, contained in security table DTSECTXN, to the latest BSM transaction profiles is described in Chapter 24, "Migrating CICS Transaction Security Definitions," on page 281.
   e. How you migrate the 64 transaction security keys (defined for each user) to 64 BSM groups is described in "Performing the Migration" on page 285. This topic also describes how user-IDs can be *connected* to these 64 BSM groups.
   f. How you recreate and/or backup the BSM control file is described in "Recreating Your BSM Control File" on page 288. This process uses the BSTSAVER program. The commands generated by BSTSAVER can also be saved if you wish to migrate to a later level of BSM.
3. How you set up and maintain your *user profiles* is described in these topics:
   a. Chapter 25, "Maintaining User Profiles via BSM Dialogs," on page 289 describes how you define user profiles using *dialogs*.
   b. Chapter 26, "Maintaining User Profiles via Batch Program IESUPDCF," on page 311 describes how you define user profiles using *a batch job*.

Note: Chapter 27, "Maintaining User Profiles in an LDAP Environment," on page 323 describes how you set up your user profiles in an *LDAP environment* (which is not based on the BSM).

4. The *resource customization/administration* tasks that you must complete are described in the following topics:
   a. How you group *resources* (such as CICS transactions, CICS transient data queues, CICS files, and so on) into *resource classes*, is described in Chapter 28, "Resources Classes Stored in the BSM Control File," on page 349.
   b. The syntax rules for the resources you define in the BSM control file are given in "Syntax Rules For Resources Defined in the BSM Control File" on page 349.
   c. How you define resources in the BSM control file using *the BSTADMIN EXEC*, is described in Chapter 29, "Protecting Resources via BSTADMIN Commands," on page 357.
   d. How you define resources in the BSM control file using *dialogs*, is described in Chapter 30, "Protecting Resources via BSM Dialogs," on page 373.
   e. How you protect JCL statements by defining resource profiles for the FACILITY resource class is described in "Using the Tailor-IPL-Procedure Dialog to Tailor Security Parameters" on page 278.
   f. An overview of VSE security based upon the use of table DTSECTAB and the VSE control file is described in Chapter 31, "Overview of DTSECTAB-Based VSE Security," on page 385. This chapter also describes how VSE/POWER jobs are authenticated.
   g. How you customize the IBM-supplied sample table DTSECTAB and activate batch security based on this table, is described in Chapter 32, "Customizing/Activating DTSECTAB-Based Security," on page 391.
   h. How you specify *access rights* in table DTSECTAB, is described in Chapter 33, "Access Rights/Checking in DTSECTAB," on page 401.
   i. The syntax of the DTSECTAB macro (including examples of its use), is described in Chapter 34, "DTSECTAB Macro: Syntax and Examples," on page 413.
   j. How security information is propagated between one or more VSE/POWER batch environments, is described in Chapter 35, "Propagation of VSE/POWER Security Identification," on page 421.
   k. The security items that you should consider in a system where batch security is active (such as avoiding startup problems, performance, tape handling), is described in Chapter 36, "Operating a DTSECTAB-Based Security System," on page 425.
   l. The standard facilities for protecting data (such as IPL Exit, Job Control Exit, Disk-File Protection, Track Hold Option), are described in Chapter 37, "Additional z/VSE Data Protection Facilities," on page 429.
5. The *logging and reporting of security events* is described in the following topics:
   - How you use the DMF (Data Management Facility) to store SMF (System Management Facilities) records created by RACROUTE requests, is described in "Logging and Creating Reports Of Security-Related SMF Records" on page 433.
   - How you use optional program ACLR (VSE/Access Control-Logging and Reporting) to log/report-on access attempts to resources contained in table DTSECTAB, is described in "Using VSE/ACLR to Log/Report Access Attempts to DTSECTAB" on page 439.

General Aspects

If you are an experienced z/VSE user, you may skip the following text.

The information stored in a data processing system is often of vital importance to the organization which uses the system. On one hand, the information is necessary for the members of the organization to do their work. On the other hand, the system may store confidential information, whose disclosure to unauthorized persons could mean considerable damage to an organization.

Corresponding to these two categories of information are two aspects of data protection:

Data Security which means protecting information from unauthorized access and use.

Data Integrity which means protecting information from loss or destruction.

The z/VSE Access Control Function, which is the primary topic of this chapter, solely addresses the first aspect: data security. The Access Control Function is part of the Basic Security Manager of z/VSE.

A computer system operating under control of z/VSE offers a number of protection functions for hardware and software. This chapter deals with the **software** protection functions provided by z/VSE. For hardware protection functions, refer to the appropriate manuals of the hardware you are using.

Security Considerations

A z/VSE system is designed to protect users' data or applications from interference by other users or applications. However, people who deliberately use their knowledge of the internals of the system can gain unauthorized access to data and resources of the system despite built-in security safeguards. Management is responsible for introducing administrative and operational safeguards that help to avoid such exploitations and that ensure system security to a great extent.

To achieve an acceptable level of system security for a z/VSE installation a user should:

- Ensure that knowledgeable and skilled members of the installation's staff will have little or no chance to access certain data or to use or manipulate certain programs;
- Ensure that resources (mainly programs) that can be used to bypass existing security safeguards are protected properly.
- Ensure that IBM's Diagnosis Reference manuals are given only to a limited set of persons.

The Security Administrator

A security administrator must have adequate software protection functions available to meet all security demands. A person appointed to assume the responsibility of safeguarding installation's assets (data and programs) should, therefore, carefully review the available standard functions. A security administrator should do this review in full consideration of the various responsibilities that a person in such a position normally must assume, some of which are indicated below:

- Together with management, prepare a list of sensitive files and of programs that process data in these files.

- Determine who of the installation's staff is authorized to use those programs and data and establish procedures that ensure that authorized persons only, and no one else, will be able to invoke the programs.
- Protect system libraries adequately.
- Minimize the chance that unauthorized access of sensitive data remains undetected.
- Keeping track of the usage of certain protected programs and data (*logging and reporting*).
- Utilize (in addition to z/VSE Access Control) special security functions of z/VSE components and programs.

Passwords and User-IDs

One of the responsibilities of the security administrator is to assign passwords to users and force the users to change them from time to time in order to avoid damage as a result of inadvertent or intentional disclosure.

The password itself should be composed of a random combination of alphanumeric characters. It should not contain any information or be mnemonic.

If passwords must be included in the job stream, special protective measures may be required. For example, the member containing the job stream could be protected such that unauthorized persons are unable to read it.

Overview Diagram of BSM-Based Security

Figure 71 on page 275 provides an overview of the z/VSE security that is based upon the use of the *Basic Security Manager* (BSM). The BSM is *always* activated during z/VSE startup, in order to provide:
- sign-on security,
- CICS transaction security.

Note:
1. If the functionality provided by the BSM does not meet your requirements, you might *instead* be able to use an External Security Manager (ESM) that is supplied by a vendor.
2. You cannot use both the BSM and an ESM in the *same* z/VSE system!

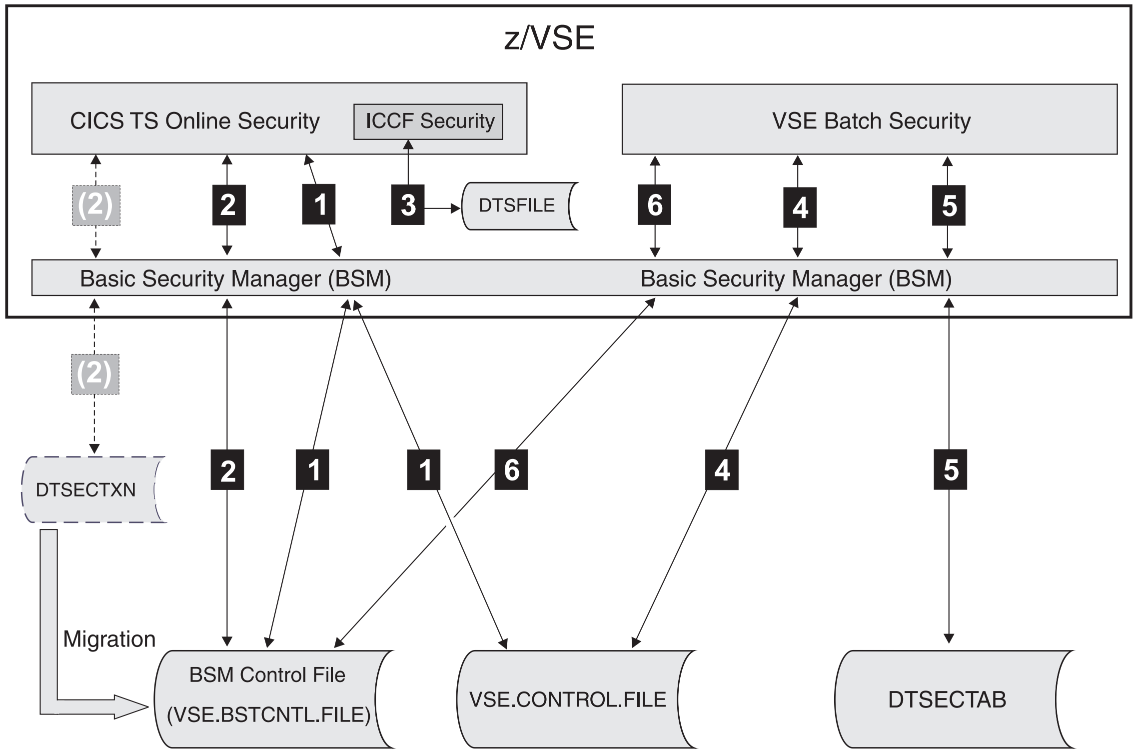


Figure 71. Overview of z/VSE and CICS Security Processing

1 When a user signs on to CICS, CICS issues a RACROUTE request to verify this user-ID, password, and the authorization for this CICS application ID.

1. For the user-ID and password, the BSM checks this information against the entries in the *VSE Control File* (VSE.CONTROL.FILE), which contains all security-related *user profile* information. The check of the CICS application name is done using the CICS application profile belonging to resource class APPL. The resources belonging to class APPL are defined in the *BSM control file* (VSE.BSTCNTL.FILE).
2. If the user verification was successful, the BSM returns an information block that describes this user. CICS retains this user information, and uses it when it checks the authorization to other CICS resources (Step **2**).

2 If a CICS user wishes to access a CICS resource, CICS issues a RACROUTE request containing the name and class of the resource, the required access right, and the user information block for this user. The BSM:

1. Checks this information against the information stored in the BSM control file. The BSM control file contains CICS-specific security information such as CICS transactions, application programs, files, journals, temporary storage queues, transient data queues, transactions initiated by a CICS START command, and general information such as applications, facilities, and user groups.
2. Returns the result of the check to CICS.

(2) Up to z/VSE V3R1.1, CICS transaction security was provided for the CICS TS (not CICS/VSE) via access control table DTSECTXN, which was used to define CICS *transactions* and their security class. From z/VSE V3R1.1 onwards, this method has been succeeded by the use of the BSM concept that includes the use of the BSM control file. Migration programs are provided for you to migrate CICS security data from DTSECTXN to the BSM control file.

3 The VSE/ICCF is an application that runs under the CICS TS. VSE/ICCF manages its own security by checking information about user-IDs and data with the information stored in the DTSFILE.

4 From a batch job, a sign-on request might be sent to the BSM for a user. The BSM:

1. Checks the user against the user information contained in the VSE.CONTROL.FILE.
2. Returns the result to the batch job.

If the user can be signed-on to z/VSE, the z/VSE system retains the user information and uses it when it checks the authorization to z/VSE resources (Step **5**).

5 From a batch job, system routines might issue a security request to access a z/VSE resource (a file, library, sublibrary, or library member) that is defined in the access control table DTSECTAB. The BSM:

1. Checks the information contained in this request against the corresponding information stored in DTSECTAB.
2. Returns the result to the batch job.

6 From a batch job, a program might issue a RACROUTE request to the BSM to access a *general* resource (for example, resource profile `IBMVSE.JCL.ASSIGN.PERM` of the FACILITY resource class) that a user wishes to access. The BSM:

1. Checks the request information against the information stored in the *BSM control file*.
2. Returns the result of the check to the batch job.

General Concept of Access Control

The z/VSE security support allows you to introduce *access control* at your installation and to implement an acceptable degree of data security. It helps you meet requirements of personal accountability and provides support for:

- **User Identification and Authentication**. For details refer to "How Users Are Identified and Authenticated" on page 388.
- **Access Authorization**. For details refer to
  - Chapter 33, "Access Rights/Checking in DTSECTAB," on page 401
  - Chapter 29, "Protecting Resources via BSTADMIN Commands," on page 357
- **Logging and Reporting**.

  For *Logging and Reporting* of VSE, the z/VSE optional program *VSE/Access Control - Logging and Reporting* is available. Its functions are briefly described in Chapter 38, "Logging/Reporting Security Events," on page 433. Console messages are also used to log violations.

  BSM security also allows you to log security events from RACROUTE requests:
  - These events are logged as SMF records using the DMF (*Data Management Facility*) provided by the CICS Transaction Server for VSE/ESA (the CICS TS).
  - You can produce reports from the data using the DMF and the *BSM Report Writer*. For details, see "Logging and Creating Reports Of Security-Related SMF Records" on page 433.

Chapter 23. Implementing z/VSE Security Support

This chapter describes the main tasks you must complete in order to implement security support.

Please note that the *Basic Security Manager* (BSM) is *always* activated during startup in order to provide sign-on security (signing on via the Interactive Interface) and CICS transaction security. This is independent of how you specify the SEC setting in the IPL SYS command (as explained below).

This chapter contains these main topics:

Tasks Required to Implement Security Support

From z/VSE 4.1 onwards, your system is *automatically* set up to use the latest BSM security. This means, during an FSU or initial installation z/VSE will automatically define the:

- Security setup for transactions.
- System-provided files in resource class FCICSFCT.
- Standard facilities for the CICS Report Controller.

However, these are the further tasks you must complete in order to implement security support:

- Decide if you wish to use the Basic Security Manager (BSM) or an External Security Manager (ESM). If the functionality provided by the BSM does not meet your requirements, you might *instead* be able to use an External Security Manager (ESM) that is supplied by a vendor. If you want to use an ESM, you must:
  - Define the name of the ESM initialization routine in the ESM parameter of the IPL SYS command. z/VSE always checks for the ESM parameter setting first. If the parameter is specified then the ESM is activated, otherwise the BSM.
  - Define a security server partition (if the ESM requires one) using the SERVPART parameter of the IPL SYS command. The default is the FB partition.

  Note: You cannot use both the BSM and an ESM in the *same* z/VSE system!
- If you are performing a *Fast Service Upgrade* (FSU), decide if your CICS transaction security should be based upon:
  - the latest BSM security (as described in Chapter 29, "Protecting Resources via BSTADMIN Commands," on page 357).
  - the security table DTSECTXN (which is the older method) as described in Chapter 39, "Protecting CICS Transactions with Access Control Table DTSECTXN," on page 445.
- Specify your security parameters using the IPL SYS command:

  ```
  SEC=NO
  SEC=YES
  SEC=(YES,JCL)
  ```

```
SEC=(YES,NOTAPE)
SEC=(YES,NOTAPE,JCL)
ESM=name
SERVPART=partition
SEC=RECOVER
```

- You can modify these parameters using the *Tailor IPL Procedure* dialog (see "Using the Tailor-IPL-Procedure Dialog to Tailor Security Parameters" for details).
- During initial installation, you are asked whether you want to run your system with "security on". If you respond with YES, this will set SEC=YES in the IPL SYS command.

• Specify your security-related **user profile** information in the VSE.CONTROL.FILE. The *Maintain User Profiles* dialog is used to define access control classes and access control rights.
  - The parameter AUTH for identifying a user as the *security administrator* is not available. Instead, when defining a user profile for a type 1 user (system administrator), this user automatically has "AUTH authorization" and can access all resources with access right ALT (Alter).
  - The CICS transaction security keys in the *Maintain User Profiles* dialog are only valid if you are still using the DTSECTXN table to protect your CICS transactions (as described in Chapter 39, "Protecting CICS Transactions with Access Control Table DTSECTXN," on page 445).
  - If you have migrated your CICS transaction definitions to the BSM control file, you must maintain BSM groups instead of CICS transaction security keys. For details, see Chapter 29, "Protecting Resources via BSTADMIN Commands," on page 357.

  For details of how to use the *Maintain User Profiles* dialog, see Chapter 25, "Maintaining User Profiles via BSM Dialogs," on page 289.

• Define your resource definitions in the security table DTSECTAB (*except for the predefined users DUMMY and FORSEC*). This is described in Chapter 31, "Overview of DTSECTAB-Based VSE Security," on page 385.

Note: User Profiles DUMMY and FORSEC! User profiles are stored in the VSE.CONTROL.FILE (and not in table DTSECTAB) *except for* the predefined users FORSEC and DUMMY. These predefined users are required by z/VSE, and are the only user definitions included in DTSECTAB. User FORSEC is defined in both DTSECTAB and VSE.CONTROL.FILE. **Do not delete these users!**

Using the Tailor-IPL-Procedure Dialog to Tailor Security Parameters

With the *Tailor IPL Procedure* dialog you can modify the security parameters `SEC`, `ESM`, and `SERVPART` of the IPL SYS command.

To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

2 (Resource Definition)

4 (Hardware Configuration and IPL)

2 (Tailor IPL Procedure)

Select the IPL procedure you want to modify and press enter.

```
 TAS$ICM1             TAILOR IPL PROCEDURE: SYS COMMAND

 Enter the required data and press PF5=PROCESS

BUFLD....................... 1           Load printer buffers? 1=yes, 2=no
CHANQ....................... _____       Number of channel queue entries
DASDFP...................... 1           DASD file protection? 1=yes, 2=no
SUBLIB...................... ____        Number of sublibraries

VMCF........................ _           CMS-VSE console interface? 1=yes,
                                         2=no, or blank for system default

SEC......................... 6           access control security? 1=yes, 2=no,
                                         3=NOTAPE, 4=RECOVER, 5=JCL,
                                         6=JCL,NOTAPE
ESM......................... ________    Name of the ESM initialization phase
SERVPART.................... FB          Security server partition (F1,F2, ...
                                         FB)

TRKHLD...................... 12          Number of hold requests
 PF1=HELP      2=REDISPLAY  3=END                 5=PROCESS
               8=FORWARD
```

Figure 72. Tailor IPL Procedure Dialog

SEC Specifies whether DTSECTAB security is to be activated or not.

The following selections are possible:

- If `1 = YES` is specified, the system performs access authorization checking for resources defined in DTSECTAB.
- If `2 = NO` is specified, access control for resources defined in DTSECTAB is **inactive** (job control ID card, for example, is ignored). CICS transaction and sign-on security, however, is still **active**.
- If `3 = NOTAPE` is specified, access control is restricted to DASD files and libraries defined in DTSECTAB.
- `4 = RECOVER` prevents activation of a security manager. It should be used for recovery actions only, which cannot be done while a security manager is active.
- If `5 = JCL` is specified, the system will perform JCL security checking and will use these resource names contained in resource class FACILITY:
  - IBMVSE.JCL.ASSIGN.PERM
  - IBMVSE.JCL.LIBDEF.PERM
  - IBMVSE.JCL.LIBDROP.PERM
  - IBMVSE.JCL.OPTION.PARSTD
  - IBMVSE.JCL.OPTION.STDLABEL

  For details of using JCL security checking, see “Using BSM Dialogs to Protect JCL Operands” on page 382.
- If `6 = JCL,NOTAPE` is specified, both options `5 = JCL` and `3 = NOTAPE` will be active.

ESM Specifies the name of an ESM (External Security Manager) initialization phase. If nothing is specified, the BSM (Basic Security Manager) is activated.

SERVPART
Specifies the static partition to be used for the Security Server; the default is FB. Be careful when selecting another partition for the Security Server, which is not recommended. It must be a static partition which is not controlled by VSE/POWER and needs a corresponding priority.

Applying Security to VSE/ICCF Libraries

VSE/ICCF does *not* use the z/VSE security support to control the logon process and access to VSE/ICCF members.

In VSE/ICCF, access to VSE/ICCF libraries is controlled by defining a library as **public** or as **private**. In addition to a VSE/ICCF user's primary library, up to 8 alternate (private) libraries can be allocated to the user.

If SEC=NO ....

then VSE/ICCF uses tables of its own to control the access to batch resources such as files and programs in VSE libraries. The tables provided are:
- System Program Table
- Load Protection Table
- System File Table

Refer to the IBM manual *VSE/ICCF Administration and Operation* for details about VSE/ICCF libraries and protection tables under “Types of User Libraries” and under “Access Control Facilities”.

If SEC=YES ....

then VSE/ICCF's protection mechanism is bypassed. z/VSE Access Control checks all accesses from VSE/ICCF interactive partitions in the same way as accesses from batch partitions via DTSECTAB.

Access checking for jobs running in interactive partitions uses the ID of the VSE/ICCF terminal user. Therefore, the VSE/ICCF terminal user must have a user profile entry in VSE.CONTROL.FILE. The passwords need not match because the user's authentication was already done at logon.

LIBDEF definitions in the CICS/ICCF partition startup are valid for interactive partitions and the terminal user. Therefore, it is recommended to keep the **permanent LIBDEFs** that are supplied by the pregenerated DTSECTAB (see also “Access Control for LIBDEF Statements” on page 407). In this way, only the universal access rights are granted to interactive partition users.

Dummy Resource IJSYSRS.SYSLIB.DTSUTILA

This is a dummy resource which has an entry in DTSECTAB but is not a real member of any library. Users having READ access to this member are allowed to issue security-relevant DTSUTIL commands, for example ALTER USER. Other users may only read or write to VSE/ICCF members.

Passwords For VSE/ICCF and the Interactive Interface

Normally, a user's password under VSE/ICCF is the same (that is, truncated to 6 characters) as the password under the VSE Interactive Interface.

Chapter 24. Migrating CICS Transaction Security Definitions

This chapter describes how you migrate your CICS transactions and their security keys from DTSECTXN to transaction profiles stored in the *BSM control file*.

The introduction of the *BSM control file* with z/VSE V3R1.1 allows you to protect CICS and general resources using BSM resource profiles stored in the *BSM control file*. From z/VSE 4.1 onwards, the BSM-based security is the *standard security concept* that is established during the initial installation of z/VSE.

The use of the BSM control file succeeds the previous method of protecting CICS *transactions* via the table DTSECTXN. From z/VSE 4.1 onwards, IBM-provided transaction security definitions will *only* be supplied in the format that is used by the BSM control file.

If you have performed:
- an initial installation of z/VSE 5.1, or
- an FSU to z/VSE 5.1 from a VSE system in which you *had already migrated* your security definitions to the latest BSM security,

then the DTSECTXN-based security (that uses transaction security keys) is no longer used. Instead, you *must* use the TCICSTRN resource class and user groups to protect your CICS transactions.

If you have performed an FSU from a VSE system where you *have not migrated* your security definitions to the latest BSM security, you can further use these transaction security keys (1 to 64) to control access to a CICS transaction defined in DTSECTXN.

Note that for compatibility reasons, DTSECTXN can exist in parallel to transaction profiles. The BSM authorization procedure first checks DTSECTXN. If an entry for a CICS transaction is not found, it then uses the transaction profiles.

This chapter contains these main topics:
- "Overview of Migration Steps" on page 282
- "Performing the Migration" on page 285.
- "Recreating Your BSM Control File" on page 288

Related Topic:
- Chapter 40, "Migrating CICS/VSE Security Information to the CICS TS," on page 451

Overview of Migration Steps

The steps you can follow partly depends on:

- The VSE system level from which you installed z/VSE 5.1.
- Whether you performed an FSU (Fast Service Upgrade) or an initial installation.
- Whether you wish to retain the use of your previous security definitions.

Table 7 provides an overview of the steps you follow in order to migrate your CICS transaction security definitions to the latest security concept (shown in Figure 71 on page 275):

Table 7. Overview of Steps to Follow When Migrating Your CICS Transaction Security Definitions

How do you plan to install z/VSE 5.1?	Steps You Should Follow On Your System (Before Installing z/VSE 5.1)	Steps You Must Follow After Installing z/VSE 5.1
Using an **Initial Installation**, and coming from a VSE system that is earlier than VSE/ESA 2.4 that runs CICS/VSE.	Depending upon the type of CICS definitions you are using, perform one or more of the following: • Use the LIBR utility to save to disk/tape the PCT from your previous system. • Generate the migration data set (for details, see "Migrating TRANSEC Definitions Using the Migration Aid" on page 452). • Use the LIBR utility to save to disk/tape the CSD update statements (to be used with DFHCSDUP) from your previous system .	One or more of the following: • Use the LIBR utility to copy the saved PCT definitions to your z/VSE 5.1 system. Next, run DTSECTXS to convert your PCT entries into BSTADMIN control statements. Finally, submit the generated output to update the BSM control file. For details, see "Migrating DFHPCT.A TRANSEC Definitions" on page 456. • Use the LIBR utility to copy the saved migration data set to your z/VSE 5.1 system (for details, see "Migrating TRANSEC Definitions Using the Migration Aid" on page 452). Next, run DTSECTX2 to convert your migration data set definitions into BSTADMIN control statements. Finally, submit the generated output to update the BSM control file. • Use the LIBR utility to copy the saved CSD update statements to your z/VSE 5.1 system. Next, run DTSECTX3 to convert the DFHCSDUP control statements into BSTADMIN control statements. Finally, submit the generated output to update the BSM control file. For details, see "Migrating DFHCSDUP TRANSEC Definitions" on page 458.

Table 7. Overview of Steps to Follow When Migrating Your CICS Transaction Security Definitions (continued)

How do you plan to install z/VSE 5.1?	Steps You Should Follow On Your System (Before Installing z/VSE 5.1)	Steps You Must Follow After Installing z/VSE 5.1

Using an **Initial Installation**, where your system is VSE/ESA 2.4 or later in which **DTSECTXN is used together with the Interactive Interface dialogs**.	Save to disk/tape the copy of DTRISEC.Z (containing the DTSECTXN definitions) from your previous system.	1. Copy the saved DTRISEC.Z into IJSYSRS.SYSLIB.
		2. Run:
		`// EXEC REXX=IPFTABLE,`
		`PARM='IJSYSRS.SYSLIB.DTSECTXS.A`
		`IJSYSRS.SYSLIB.DTRISEC.U'`
		`/*`
		3. Ensure that you have migrated your VSE control file (IESCNTL) using the IESBLDUP utility. Then press **PF6 = Groups** in **Fast Path 211** (*Maintain User Profiles*) to:
		a. define default groups in the BSM control file, and
		b. assign users to these groups.
		4. Press **PF6= Merge** in **Fast Path 285** (*Migrate Security Entries*) to merge the new DTRISEC.U definitions with those from your previous system (see Figure 73 on page 285).
		5. Also use **Fast Path 285** to migrate the DTSECTXN definitions (that were merged in the previous step) to the BSM control file (see Figure 75 on page 287).

Using an **Initial Installation**, where your system is VSE/ESA 2.4 or later in which **DTSECTXN is used without the Interactive Interface dialogs**.	Save to disk/tape a copy of your DTSECTXN definitions from your previous system.	1. Ensure that you have migrated your VSE control file (IESCNTL) using the IESBLDUP utility. Then press **PF6 = Groups** in **Fast Path 211** (*Maintain User Profiles*) to: a. define default groups in the BSM control file, and b. assign users to these groups. 2. Obtain a copy of the job skeleton SKSECVTX from library 59. 3. Submit job SKSECVTX, which catalogs the procedure DTSECVTX.PROC into sub.library PRD2.CONFIG. 4. Read through the instructions contained in procedure DTSECVTX. Then run this procedure with your own values for INFILE and OUTFILE. 5. Run program BSTADMIN, using as input the BSTADMIN statements created by the procedure DTSECVTX. 6. Check the listing produced by the previous step. Providing the listing is correct, activate the new definitions using the BSTADMIN command `PERFORM DATASPACE REFRESH`.

Table 7. Overview of Steps to Follow When Migrating Your CICS Transaction Security Definitions (continued)

How do you plan to install z/VSE 5.1?	Steps You Should Follow On Your System (Before Installing z/VSE 5.1)	Steps You Must Follow After Installing z/VSE 5.1
Using an **Initial Installation**, where your system is z/VSE 3.1.1, 3.1.2, or 3.1.3, in which **DTSECTXN is not used**.	1. Install APARs PK24287 and DY46510 which provide the BSTSAVER utility. 2. Run BSTSAVER to save the contents of the BSM control file (VSE.BSTCNTL.FILE) from your previous system (as described in "Recreating Your BSM Control File" on page 288). You can use skeleton SKBSTSAV in ICCF library 59 to run the BSTSAVER program. The output from the BSTSAVER program consists of BSTADMIN statements.	1. Copy the member created by the BSTSAVER job to your z/VSE 5.1 system. 2. Use skeleton SKBSTSAV to upgrade the BSM control file and to rebuild the security data space. This skeleton runs program BSTADMIN, using as input the BSTADMIN statements created by the BSTSAVER program.
Using an **Initial Installation**, where your system is z/VSE 4.1.x or later in which **DTSECTXN is not used**.	Run BSTSAVER to save the contents of the BSM control file (VSE.BSTCNTL.FILE) from your previous system (as described in "Recreating Your BSM Control File" on page 288). You can use skeleton SKBSTSAV in ICCF library 59 to run the BSTSAVER program. The output from the BSTSAVER program consists of BSTADMIN statements.	1. Copy the member created by the BSTSAVER job to your z/VSE 5.1 system. 2. Use skeleton SKBSTSAV to upgrade the BSM control file and to rebuild the security data space. This skeleton runs program BSTADMIN, using as input the BSTADMIN statements created by the BSTSAVER program.
Using an **FSU**, and coming from z/VSE 4.2.x or later in which **DTSECTXN is used**	None	1. Press **PF6 = Groups** in **Fast Path 211** (*Maintain User Profiles*) to: a. define default groups in the BSM control file, and b. assign users to these groups. 2. Do **not** migrate your DTSECTXN entries using **Fast Path 285** (*Migrate Security Entries*). **Instead**, press **PF6** in **Fast Path 285** to merge the new DTRISEC.U definitions with those from your previous system (see Figure 73 on page 285). 3. If you **do** wish to migrate, use **Fast Path 285** (see Figure 75 on page 287) to migrate your DTSECTXN definitions, merged in previous step, to the BSM control file.
Using an **FSU** from z/VSE 4.2.x or later in which **DTSECTXN is not used**	None	None

Note: For details of how to migrate CICS/VSE resources (except for transaction security definitions) to the latest security concept, refer to Chapter 40, "Migrating CICS/VSE Security Information to the CICS TS," on page 451. The CICS transaction security definitions are migrated using **Fast Path 285** (see Figure 75 on page 287).

Performing the Migration

To migrate CICS transactions and their security keys contained in table DTSECTXN to transaction profiles in the BSM control file, you should:

1. Ensure that parameter SPOOL has been set to YES ("CICS SPOOLER ACTIVE") in the CICS System Initialization Table (SIT).
2. **Create Group Profiles From All User-ID Definitions, and Connect User-IDs to Groups.**
   a. Use **Fast Path 211** to display the *Maintain User Profiles* panel, as shown in Figure 73.

```
IESADMUPL2                 MAINTAIN USER PROFILES
VSE CONTROL FILE
START.... ________
OPTIONS:   1 = ADD             2 = CHANGE             5 = DELETE
                       PASSWORD     REVOKE  USER INITIAL  NAME
   OPT    USERID      VALID UNTIL    DATE   TYPE  NAME    TYPE

    _     $SRV         08/01/05             2   IESERSUP  2
    _     AMAD         08/01/05             1   AMADADM   2
    _     AMA1         08/01/05             1   IESDITTO  1
    _     AMA2         08/01/05             1   AMADADM   2
    _     ASTA         10/21/02 *           1   IESEADM   2
    _     BA01                              1   IESEADM   2
    _     BA02                              1   IESEADM   2
    _     CICSUSER     08/01/05             3   DFLESEL   2
    _     CNSL         08/01/05             1   DUMMY     1
    _     DBDCCICS     08/01/05             1   DUMMY     1
    _     ELKC         02/24/03 *           1   ELKESEL   2
    _     ELKE         07/14/05             1   ELKESEL   2

PF1=HELP                   3=END                              6=GROUPS
PF7=BACKWARD   8=FORWARD    9=PRINT
```

Figure 73. Panel for Mapping All Transaction Security Keys to Groups Profiles

   b. Press PF6 to create a job which (when submitted) will:
      1) Create group profiles from the transaction security keys.
      2) Connect ***all user-IDs*** to their corresponding group profiles.

      This job is stored in the VSE/POWER punch queue and has the file name CICSICCF. You can use **Fast Path 32** to display the contents. Figure 74 on page 286 shows an extract of such a job:

```
// JOB IESTBGRI
⋮
* ADD TRANSEC CLASS MIGRATION GROUPS IN CASE NOT EXIST
 ADDGROUP GROUP01    DATA('TRANSEC CLASS MIGRAT')
 ADDGROUP GROUP02    DATA('TRANSEC CLASS MIGRAT')
 ADDGROUP GROUP03    DATA('TRANSEC CLASS MIGRAT')
⋮
 ADDGROUP GROUP63    DATA('TRANSEC CLASS MIGRAT')
 ADDGROUP GROUP64    DATA('TRANSEC CLASS MIGRAT')
⋮
* BA02      IS SYSTEM ADMINISTRATOR. NOT CONNECTED TO ANY GROUP
 CONNECT GROUP01   CICSUSER
 CONNECT GROUP60   CICSUSER
 CONNECT GROUP61   CICSUSER
 CONNECT GROUP62   CICSUSER
 CONNECT GROUP63   CICSUSER
 CONNECT GROUP64   CICSUSER
* CNSL      IS SYSTEM ADMINISTRATOR. NOT CONNECTED TO ANY GROUP
* DBDCCICS  IS SYSTEM ADMINISTRATOR. NOT CONNECTED TO ANY GROUP
* ELKC      IS SYSTEM ADMINISTRATOR. NOT CONNECTED TO ANY GROUP
* ELKE      IS SYSTEM ADMINISTRATOR. NOT CONNECTED TO ANY GROUP
 CONNECT GROUP01   EOPE
 CONNECT GROUP60   EOPE
 CONNECT GROUP61   EOPE
 CONNECT GROUP62   EOPE
 CONNECT GROUP63   EOPE
 CONNECT GROUP64   EOPE
 CONNECT GROUP01   EPRG
 CONNECT GROUP60   EPRG
 CONNECT GROUP61   EPRG
⋮
```

Figure 74. Job for Building Group Profiles and Connecting User-IDs

   c. Select **Fast Path 323** (*Punch Queue*), and then type a '**4**' ("Copy to Primary Library") next to the job name. The job will then be received from the VSE/POWER punch queue to your Primary Library.
   d. Select **Fast Path 511** (*Primary Library*), and then type a '**1**' ("Edit") next to the job name, to check the contents of the job. If the contents are correct, use Option '**7**' ("Submit") to submit the job to z/VSE. The user-IDs will then be connected to their corresponding group profiles.
3. **Add New User-IDs to Your System.** Now that you have migrated ***all user-IDs*** and connected these user-IDs to their appropriate groups, for ***new user-IDs you should not repeat Step 1***. Instead, you should:
   a. Use **Fast Path 211** to define the new user-IDs.
   b. Authorize the new user-IDs, by connecting them to the appropriate group profiles using **Fast Path 282** (*Security Maintenance*).
4. **Migrate Transaction Security Entries.**
   a. Select **Fast Path 285** (*Migrate Security Entries*) from the Interactive Interface.
   b. You are now prompted to enter a "**1**" to migrate your transaction security keys. If you wish to migrate transaction security entries that are stored in another table, key the name of this table in the field "Migrate Member". Otherwise, the default table DTSECTXN will be selected for migration.

```
TAS$SEC4                 MIGRATE SECURITY ENTRIES

Enter the required data and press ENTER.

The security concept of the Basic Security Manager (BSM) has changed.
You are recommended to migrate your entries and use the dialog
 Maintain Security Profiles.

The  DTSECTXN table as used by this dialog can still be used in parallel to the
new BSM control file.

MIGRATE...................... 1           Do you want to migrate the trans-
                                          action security entries?
                                          Enter 1 for YES.
                                          Enter 2 to proceed with the Define
                                          Transaction Security dialog.

Migrate own security definitions in macro format?
 Migrate Member..........     ________________________________

 PF1=HELP      2=REDISPLAY  3=END

 TO MIGRATE PRESS PF6 IN MAINTAIN USER PROFILE DIALOG.
```

Figure 75. Panel for Migrating DTSECTXN Entries to the BSM Control File

c. After pressing Enter, a job containing BSTADMIN statements is created similar to that shown in Figure 76. Two statements are created for each transaction entry in DTSECTXN (an example of the DTSECTXN structure is shown in Figure 107 on page 450) .

```
// JOB BSTADMIN
ID (PARAMETERS SUPPRESSED)
*
* * This job updates the BSM control file with the initial
* * values and activates them via BSTADMIN.
*
// DLBL BSTCNTL,'VSE.BSTCNTL.FILE',,VSAM,CAT=VSESPUC
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE)
// EXEC BSTADMIN
          ADD TCICSTRN 'emai' UACC(NONE)
          PERMIT TCICSTRN 'emai' ID(GROUP01) ACCESS(READ)
          ADD TCICSTRN 'ftp' UACC(NONE)
          PERMIT TCICSTRN 'ftp' ID(GROUP01) ACCESS(READ)

⋮
```

Figure 76. Job to Map DTSECTXN Entries to BSM Groups

d. The migration job then maps transaction security keys 1 to 64 from DTSECTXN to the BSM group profiles GROUP01 to GROUP64, by:
   1) Migrating the definitions from the *Maintain Transaction Security* dialog.
   2) Migrating mixed-case definitions from member DTSECTXM.A.
   3) Migrating definitions from your own member (if present).
   4) Renaming the previous definition members.
   5) Loading the new BSTADMIN definitions.
   6) Rebuilding the dataspace.
   7) Deleting the previous DTSECTXN phase (providing the previous steps completed successfully).

   To migrate your own transaction security definitions, you can also use these procedures (that are used in the above steps):

DTRMIGRM
Migrates transaction security definitions that are in Macro format.

DTRMIGRT
Migrates transaction security definitions that are in Table format.

Note: After you have migrated your transaction definitions, you are strongly recommended to use **Fast Path 2811** (*Maintain Security Profiles*) to define your transaction security.

Recreating Your BSM Control File

A batch program is available (BSTSAVER) which:

1. Builds BSTADMIN commands from the contents of the BSM control file (VSE.BSTCNTL.FILE).
2. Stores these BSTADMIN commands in a librarian member.

You can use program BSTSAVER to:

- Create a backup of your BSM control file (that is, for security/backup purposes) in readable *text format*.
- Migrate the contents of your BSM control file to your current z/VSE release. For example, when migrating your BSM security to z/VSE 5.1.0 from:
  - z/VSE 3.1.1 (or a later refresh of z/VSE 3.1)
  - z/VSE 4.1 (or a later refresh of z/VSE 4.1)

To run the BSTSAVER program, you can either:

- Create a batch job that contains this statement:
  ```
  // EXEC BSTSAVER,PARM='library.sublibrary.member_name.member_type'
  ```
- Run BSTSAVER from the system console, for example using a command sequence like this:
  ```
  r rdr,pausebg
  0 // exec bstsaver,parm='library.sublibrary.member_name.member_type'
  ...
  0 end
  ```

(where *library.sublibrary.member_name.member_type* is the member containing the BSTADMIN commands that are used for recreating the profiles in your BSM control file).

If you wish to change any commands before restoring member *library.sublibrary.member_name.member_type* to your BSM control file, you can edit this member to do so.

To restore the information from the member *library.sublibrary.member_name.member_type* to your BSM control file, you can use a batch job like this one:

```
// EXEC BSTADMIN
* $$ SLI MEM=member_name.member_type,S=library.sublibrary
/*
// EXEC BSTADMIN
   PERFORM DATASPACE REFRESH
/*
```

Chapter 25. Maintaining User Profiles via BSM Dialogs

This chapter describes how you maintain the security and other information contained in user profiles using BSM dialogs.

This chapter contains these main topics:

Related Topics:

For details of how to ...	Refer to ...
use BSM security server commands	*z/VSE Operation*.
maintain large numbers of user profiles using batch utility IESUPDCF	Chapter 26, "Maintaining User Profiles via Batch Program IESUPDCF," on page 311.
view and change LDAP user profiles	Chapter 27, "Maintaining User Profiles in an LDAP Environment," on page 323.
protect CICS and general resources	Chapter 30, "Protecting Resources via BSM Dialogs," on page 373.

Introduction to Maintaining User Profiles via BSM Dialogs

Every user of the Interactive Interface is defined by a *user profile*. The profile includes a unique user-ID and password which is used to sign on to the Interactive Interface. It also determines what is displayed after the user signs on.

The Interactive Interface provides the *Maintain User Profiles* panel, which is the *central point* for controlling a user's access to the z/VSE system. Starting from this dialog, you can:

- define, update, or delete user profiles.
- ensure that each LDAP user profile is consistent with its corresponding VSE user profile (when LDAP support is active).
- manage a user's connects to groups.

To access the dialog, start with the *z/VSE Function Selection* panel and select Fast Path **211**. The panel shown in Figure 77 on page 290 is then displayed.


```
IESADMUPL2                 MAINTAIN USER PROFILES
VSE CONTROL FILE
START.... ________
OPTIONS:   1 = ADD            2 = CHANGE             5 = DELETE
                       PASSWORD     REVOKE  USER INITIAL  NAME
   OPT    USERID      VALID UNTIL    DATE   TYPE  NAME    TYPE

    _     $SRV         08/01/10             2   IESERSUP  2
    _     AMAD         08/01/10             1   AMADADM   2
    _     AMA1         08/01/10             1   IESDITTO  1
    _     AMA2         08/01/10             1   AMADADM   2
    _     ASTA         10/21/10             1   IESEADM   2
    _     BA01                              1   IESEADM   2
    _     BA02                              1   IESEADM   2
    _     CICSUSER     08/01/10             3   DFLESEL   2
    _     CNSL         08/01/10             1   DUMMY     1
    _     DBDCCICS     08/01/10             1   DUMMY     1
    _     ELKC         12/24/08 *           1   ELKESEL   2
    1     PROG         10/14/15             2   IESEPROG  2

PF1=HELP                   3=END                            6=GROUPS
PF7=BACKWARD   8=FORWARD    9=PRINT
```

Figure 77. Maintain User Profiles Panel

A FULIST displays the user-IDs defined to the system. The options you can choose are shown at the top of the FULIST. Use **PF7** or **PF8** to scroll through the entries. If you want to locate a particular user-ID, enter the user-ID in the START field. The dialog searches for the user-ID and displays it at the beginning of the list.

The options that you can select are explained in the topics that follow.

Adding/Changing a User-ID and Profile Definitions

- *To add a user-ID*, in the *Maintain User Profiles* panel (shown in Figure 77) enter '**1**' in the OPT column of the user-ID that you wish to use as a model. The model provides default values. “Dialog Considerations” on page 309 has information on the default values.
- *To change a user-ID*, in the *Maintain User Profiles* panel (shown in Figure 77) enter '**2**' in the OPT column of the user-ID whose profile you wish to change.

The *Add or Change User Profile* dialog is then displayed which you can use to specify profile information:

- The first two panels of this dialog are used for defining z/VSE user profile information.
- The third and fourth panels of this dialog are used for defining CICS and DTSECTAB-based security information.
- The fifth panel of this dialog is used for defining VSE/ICCF information.

In addition, **two** further panels might be displayed:

- If LDAP support is active, the *Maintain LDAP User Profiles* panel. This dialog allows you to add or change an LDAP user profile that corresponds to the VSE user profile.
- If the user is *not* an administrator (Type 1 user), the *Maintain Security Profiles* panel. This dialog allows you to add or change the user's connects to groups.

Entering z/VSE User Profile Information

For the first two panels, the **z/VSE profile information** is described on the following pages. The example used in this section has **PRG1** as the user-ID to be added and **PROG** as the model user-ID.

```
IESADMUPBA                ADD OR CHANGE USER PROFILE
Base     II       CICS     ResClass ICCF

To CHANGE, alter any of the entries except the userid.

  USERID............ PRG1       1 - 8 characters (4 characters for ICCF users)

  INITIAL PASSWORD... ________   3 - 8 characters

  DAYS.............. 155        0-365 Number of days before password expires
  REVOKE DATE........ 12/31/10   Date when Userid will be revoked (mm/dd/yy)

  USER TYPE.......... 2          1=Administrator, 2=Programmer, 3=General
  INITIAL NAME....... IESEPROG   Initial function performed at signon
  NAME TYPE.......... 2          1=Application, 2=Selection Panel
  SYNONYM MODEL...... ________   Userid to be used as model for synonyms
  PROGRAMMER NAME....                       Supplementary user name



PF1=HELP                    3=END                    5=UPDATE
               8=FORWARD
```

Figure 78. Add or Change User Profile Panel

USERID
The user-ID which identifies the user to the Interactive Interface. It can be 1 to 8 alphanumeric characters long and include the special characters @, #, or $. No blank is allowed.

Note that access to VSE/ICCF depends on the length of the user-ID. A 5 - 8 character user-ID does not have access to VSE/ICCF. Refer also to “Planning Considerations for Using the Interactive Interface” on page 123.

If you are **changing** a user profile, you **cannot** change this field on the panel.

INITIAL PASSWORD
This is the password associated with the user-ID. Specify 3 - 8 alphanumeric characters including the special characters @, #, or $. No blank is allowed.

The user is forced to change the password during the first sign-on.

DAYS The number of days before the password expires. Specify a number between 0 and 365. If you enter 0, the password will **not** expire.

Information about password expiration and how you can change your password is given in “Password Expiration” on page 143.

REVOKE DATE
Enter the date, when the user-ID will be revoked by the system. After this date, a sign-on attempt with this user-ID will be rejected. The valid date format is MM/DD/YY. You can also specify 0 if the user-ID has an unlimited validity.

Note: If the User-ID is revoked you must change the revoke date, not the password.

USER TYPE
Enter one of the following:

- **1** (Administrator)

  This selection provides **VSE/ICCF** administrative authority if the user has a 4-character user-ID and is defined on a CICS subsystem with VSE/ICCF. The **SYSA** profile supplied by z/VSE defines a type 1 user.

  If the user is defined for a CICS subsystem without VSE/ICCF, no VSE/ICCF administrative authority is provided.

- **2** (Programmer and Operator)

  If the user has a 4-character user-ID, this selection provides access to VSE/ICCF, but not VSE/ICCF administrative authority. The **PROG** and **OPER** profiles supplied by z/VSE define type 2 users.

- **3** (General)

  This selection does **not** provide access to VSE/ICCF. It is intended for application end users. z/VSE does not supply a predefined profile for type 3 users.

INITIAL NAME
Name of the selection panel or application invoked when the user signs on to the Interactive Interface.

NAME TYPE
This defines the type of function you specify in the INITIAL NAME field.

- **1** - Application

  The system invokes the application when the user signs on.

- **2** - Selection Panel

  The system displays the selection panel when the user signs on.

SYNONYM MODEL
This defines the user-ID to be used as a model for synonyms. z/VSE provides synonyms for users SYSA, PROG, and OPER. These can be used as models for other users. For further details about synonyms refer to "Maintaining Synonyms" on page 142.

PROGRAMMER NAME
Supplementary user name consisting of up to 20 characters. This field is optional.

After entering the required details, press **PF8** to proceed to the second panel, as shown in Figure 79 on page 293.

```
IESADMUPII                    USER AUTHORIZATION
Base     II       CICS     ResClass ICCF
Answer yes or no to the following questions for userid PRG1
Enter 1 for yes, 2 for no
  NEWS...................... 1   Should user receive news items?
  ESCAPE.................... 2   Can user escape to CICS?
  CONFIRM DELETE............ 2   Does user want a confirmation message?
  VSE PRIMARY SUBLIBRARY..... 1   Does user want a PRIMARY sublibrary?
  SUBMIT TO BATCH........... 1   Can user submit to Batch?
  VSAM FILES................ 1   Can user define VSAM files?
  VSAM CATALOGS............. 2   Can user manage VSAM catalogs?
  OLPD...................... 1   Can user delete OLPD incidents?
  CONSOLE COMMANDS.......... 2   Can user enter all commands?
  CONSOLE OUTPUT............ 1   Can user see all messages?
  BATCH QUEUES.............. 2   Can user manage all POWER jobs?



  DEFAULT USER VSAM CATALOG.. IJSYSCT

PF1=HELP                   3=END                    5=UPDATE
PF7=BACKWARD   8=FORWARD
```

Figure 79. User Authorization Panel for Type 2 User

For the following fields, you can enter:

1 - YES
2 - NO

NEWS

The system displays *news items* to the user.

News items are messages which the system displays when a user signs on. It also displays the messages to users already signed on to the system. You use the *Enter News* dialog to add, change, or delete news items.

ESCAPE

The user can *escape* to CICS. This lets the user leave the Interactive Interface and go into native CICS mode. If a user has this authorization, the selection panels show PF6 and PF9. These PF keys are used for the *escape* facility.

CONFIRM DELETE

This defines whether the user gets a confirmation message when deleting VSE/POWER queue entries, VSE/ICCF library members, or BSM resource definitions.

VSE PRIMARY SUBLIBRARY

This defines whether the user gets assigned a VSE sublibrary named PRIMARY.userid. The PRIMARY sublibrary will be created by using the *Maintain PRIMARY Sublibraries* dialog for any user who has this option set in the profile.

SUBMIT TO BATCH

This defines whether the user is authorized to submit jobs to the batch queues.

VSAM FILES

The user can define and delete VSE/VSAM files, libraries, alternate indexes, and alternate names. This authorization is **not** available for general user (type 3) profiles.

The user can access selections 1, 2, 3, and 4 from the *File and Catalog Management* dialog. These selections are:

- Display or Process a File
- Define a New File
- Define a Library
- Define an Alternate Index or Name

VSAM CATALOGS
The user can process VSE/VSAM catalogs and define and delete VSE/VSAM space. This authorization is **not** available for general user (type 3) profiles.

The user can access selections 1, 5, and 6 from the *File and Catalog Management* dialog. These selections are:
- Display or Process a File
- Display or Process a Catalog, Space
- Define a New User Catalog

OLPD The user can delete Online Problem Determination (OLPD) incident records from the system. This authorization is **not** available for general user (type 3) profiles.

CONSOLE COMMANDS
This allows the user to access a master console and to enter all commands. This is only valid for type 2 users (programmer and operator).

CONSOLE OUTPUT
If this flag is set, the user gets all messages displayed on the console. This authorization is **not** available for type 3 users.

BATCH QUEUES
A type 1 user (administrator) can manage all VSE/POWER jobs of type 1 and type 2 users. This includes displaying, changing, printing, or deleting a VSE/POWER job.

As a type 2 user, you can handle only jobs which you submitted or which are destined for you.

Note: In the dialog, you can set BATCH QUEUES to 1 (yes) also for a type 2 user if needed and assign the same authority as for a type 1 user. This requires, however, that bit 2 in the VSE/ICCF option byte OPTB (which is described later) is set to 1. You should consult the manual *VSE/ICCF Administration and Operation* for the additional authorizations given when changing the setting of bit 2 to 1.

APPLICATION PROFILES
This is only valid for administrator (type 1) profiles. It allows the user to create and maintain application profiles using the *Maintain Application Profiles* dialog.

SELECTION PANELS
This is only valid for administrator (type 1) profiles. It allows the user to create and maintain selection panels using the *Maintain Selection Panels* dialog.

USER PROFILES
This is only valid for administrator (type 1) profiles. It allows the user to create and maintain user profiles using the *Maintain User Profiles* dialog.

DEFAULT USER VSAM CATALOG
This defines the name of the user's default catalog. It is not available for type 3 users.

After entering the required details, press **PF8** to proceed to the third and fourth panels as described in "Adding/Changing CICS Profile and DTSECTAB Information."

Adding/Changing CICS Profile and DTSECTAB Information

For the third and fourth panels, the **CICS profile** and **DTSECTAB access rights** information is described below. Refer to CICS documentation for more details on the values you can specify. The DTSECAB access rights are described in Chapter 31, "Overview of DTSECTAB-Based VSE Security," on page 385.

```
IESADMUPCI                ADD OR CHANGE CICS SEGMENT
Base     II       CICS     ResClass ICCF

   OPERATOR ID............ PR1   Enter 3 character id for user PRG1
   OPERATOR PRIORITY....... 000   Operator priority between 0-255
   XRF SIGNOFF............ 2     Sign off after XRF takeover (1=yes,2=no)
   TIMEOUT................ 00    Minutes until sign off between 0-60

   PRIMARY LANGUAGE........       National language for CICS messages


     Place an 'X' next to the operator classes for this user

   01 X    02 _     03 _     04 _     05 _     06 _     07 _     08 _
   09 _    10 _     11 _     12 _     13 _     14 _     15 _     16 _
   17 _    18 _     19 _     20 _     21 _     22 _     23 _     24 _




PF1=HELP                   3=END                    5=UPDATE
PF7=BACKWARD   8=FORWARD
```

Figure 80. Add or Change CICS Profile Panel

OPERATOR ID
CICS three character operator identification. The ID should be unique.

OPERATOR PRIORITY
The value which CICS uses for the dispatching priorities of the user. Enter a number from 0 to 255.

XRF SIGNOFF

This defines if the user is signed-off after an XRF takeover. Enter 1 for YES, and 2 for NO.

TIMEOUT
Gives the value in minutes used by CICS to initiate sign off after the value specified has elapsed since the latest terminal activity. After such a timeout, you get the *z/VSE Online* panel displayed. You can specify a value from 0 to 60. The value you specify is always rounded up to a multiple of 5 minutes. A value of 0 means no time out. 0 should be specified for VSE/ICCF users.

If you specify a TIMEOUT value for a VSE/ICCF user, the CICS TIMEOUT value for this user should be greater than the sum of all ICCF TIMEOUT values. This ensures that in case of a VSE/ICCF timeout, the user affected is "reset" to a z/VSE panel (from a VSE/ICCF panel) thus enabling a correct working of a possible CICS timeout as well.

PRIMARY LANGUAGE
Specify the language in which CICS messages should be displayed. Specify 'E' for US English or 'J' for Japanese. If you leave this field blank, the CICS default will be used.

OPERATOR CLASSES
Choose the operator classes from 1 to 24. This defines the user to the CICS Transaction Server system. 1 is the default operator class.

After entering the required details, press **PF8** to proceed to the fourth panel, as shown in Figure 81.

On the fourth panel, the CICS transaction security keys and DTSECTAB batch access rights can be specified.

```
 IESADMUPR1          ADD OR CHANGE RESOURCE ACCESS RIGHTS
Base     II       CICS     ResClass ICCF

     Place an 'X' next to the transaction security keys for user PRG1
   01 X   02 X   03 X   04 X   05 X   06 X   07 X   08 X   09 X   10 X   11 X
   12 X   13 X   14 X   15 X   16 X   17 X   18 X   19 X   20 X   21 X   22 X
   23 X   24 X   25 X   26 X   27 X   28 X   29 X   30 X   31 X   32 X   33 X
   34 X   35 X   36 X   37 X   38 X   39 X   40 X   41 X   42 X   43 X   44 X
   45 X   46 X   47 X   48 X   49 X   50 X   51 X   52 X   53 X   54 X   55 X
   56 X   57 X   58 X   59 X   60 X   61 X   62 X   63 X   64 X

     Specify the access rights for 1-32 DTSECTAB access control classes
     ( _=No access, 1=Connect, 2=Read, 3=Update, 4=Alter )
   01 2   02 3   03 3   04 1   05 _   06 _   07 _   08 4   09 _   10 4   11 _
   12 _   13 _   14 _   15 _   16 _   17 _   18 _   19 _   20 _   21 _   22 _
   23 _   24 _   25 _   26 _   27 _   28 _   29 _   30 _   31 _   32 _

   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 81. Add or Change Resource Access Rights Panel

TRANSACTION SECURITY KEYS
If you have migrated your CICS transaction data to the latest security concept that uses the BSM control file, you will not use these security keys to protect your CICS transactions. Instead, you will use *user groups* as described in Chapter 29, “Protecting Resources via BSTADMIN Commands,” on page 357. The migration of security keys to these user groups is described in “Performing the Migration” on page 285.

However, if you have *not* migrated your CICS transaction data to the BSM control file, you can use these transaction security keys to allow access to a CICS transaction defined in DTSECTXN. You can specify from 1 to 64 keys. The default is 1. The Interactive Interface requires the use of CICS security keys 1 and 61. The security keys correspond to the TRANSEC operand in the DTSECTXN macro and the security class in the *Define Transaction Security* dialog.

ACCESS RIGHTS
You can specify the access right for batch access classes 1 to 32. Enter 1 for connect, 2 for read, 3 for update, or 4 for alter. If nothing is specified, no access to a resource with this access class is allowed.

READ DIRECTORY
If you specify 1, this user can read the directory of a library or sublibrary.

B-TRANSIENTS
If you specify 1, this user can catalog, rename or delete a B-transient in a protected sublibrary if the user has the required access right to the sublibrary.

After you have entered z/VSE and CICS/batch information on the four panels, press **PF5** and the dialog updates the VSE.CONTROL.FILE for the user profile.

If you are adding a profile for a user:
- *with access* to VSE/ICCF, z/VSE displays the VSE/ICCF *Maintain User Profiles: Specify Library* panel. Proceed to "Adding/Changing VSE/ICCF Profile Information."
- *without* access to VSE/ICCF and if LDAP support *is active*, z/VSE displays the *Maintain LDAP User Profiles* panel. Proceed to "Adding an LDAP User-ID to Correspond to the VSE User-ID" on page 299.
- *without* access to VSE/ICCF and if LDAP support *is not active*, z/VSE displays the *Maintain Security Profiles* panel in which you can add or change the group connects information. Proceed to "Adding/Changing the Group Connects for a VSE User-ID" on page 302.

Adding/Changing VSE/ICCF Profile Information

If you are adding or changing a user-ID for an administrator (**Type 1 user**) or programmer (**Type 2 user**), you have now the *option* of updating VSE/ICCF profile information (via the **Transfer Control** panel when you are *changing* a user-ID). Press the appropriate PF key:

```
PF5 - YES    (You want to update the VSE/ICCF profile).
PF6 - NO     (You do not want to update the VSE/ICCF profile).
```

Note that access to VSE/ICCF is only possible for users with a 4-character user-ID.

The VSE/ICCF default values should be acceptable for most users. You should not change the default values unless you have a specific reason to do so. See also "Additional Considerations When Maintaining User Profiles via Dialogs" on page 309.

If you press PF6 (NO), the update process is complete. The dialog redisplays the FULIST of user-IDs.

For the remaining panels, the **VSE/ICCF profile information** is described below.

The dialog displays the **Specify Library** panel. In the LIBRARY field, enter the library number for the user's VSE/ICCF primary library. For further information about VSE/ICCF libraries refer to "VSE/ICCF Library Considerations" on page 310. You can accept the remaining z/VSE defaults for VSE/ICCF information or change the defaults. In the DEFAULTS field, enter one of the following:

```
1 - YES  (You do accept the defaults.)
2 - NO   (You do not accept the defaults.)
```

If you enter 2 (NO), you are asked for additional VSE/ICCF information. In general, the default values should be acceptable for most users. You should carefully consider any VSE/ICCF changes that you make and use the recommended values. This is to ensure that the Interactive Interface operates correctly. For more detailed information on VSE/ICCF options, refer to the manual *VSE/ICCF Administration and Operation*.

- VSE/ICCF option bytes: OPTA, OPTB, and OPTC

  The default option byte settings depend on the z/VSE user type. For administrator (type 1) profiles, the defaults are:

  OPTA - 01110001

  OPTB - 11111010

  OPTC - 01000000

  For programmer and operator (type 2) profiles, the defaults are:

  OPTA - 00000100

  OPTB - 10000000

  OPTC - 01000000

  The default settings are usually satisfactory for most users.

  In the OPTA, OPTB, and OPTC bytes, you can change certain bits identified by an asterisk (*) below.

  Note: As a general rule, you should only change bits identified by an *. If you change any other bits, the Interactive Interface may not work correctly for that the user. An exception is the following situation:

  If you decide to have several type 1 users sharing one common VSE/ICCF library, you should set bit 5 of the OPTA byte to ensure that all functions of the Interactive Interface work correctly.

  – User Type 1 (Administrator)

    OPTA - 011*00*1 (You can only change bits 3,6)

    OPTB - **111010 (You can only change bits 0,1)

    OPTC - **000*0* (You can only change bits 0,1,5,7)

  – User Type 2 (Operator or Programmer)

    OPTA - 000*01*0 (You can only change bits 3,6)

    OPTB - ***00000 (You can only change bits 0,1,2)

    OPTC - **000*0* (You can only change bits 0,1,5,7)

- **VSE/ICCF security keys:**
  1 to 32 keys.

- **Alternate VSE/ICCF libraries:**
  Enter up to eight additional private VSE/ICCF libraries that the user can access (in addition to the primary library and public libraries).

- **CLASS:**
  Specify the default interactive partition (alphabetic).

- **MAXSTATE:**
  The value must be between 500 and 9999.

- **MAXPRINT:**
  The value cannot be greater than 9999.

- **MAXPUNCH:**
  The value cannot be greater than 32,767.

- **LINESIZE:**
  A value from 1-80.

- **TIMELIM:**
  The value cannot be greater than 32,767.
- **TIMEMAXEX:**
  The value cannot be greater than 65,535.
- **DEL:** You should **not** change the default.
- **TAB:** You should **not** change the default.
- **BS:** You should **not** change the default.
- **ESC:** You should **not** change the default.
- **END:**
  You should **not** change the default.
- **HEX:** You should **not** change the default.
- **LOGONRTN:**
  You should **not** change the default.
- **TIMEOUT:**
  You should **not** change the default.

If there is a need to change the above settings in full screen editor mode of VSE/ICCF, for example the HEX or TAB option, you must use the SET command of VSE/ICCF. If you have completed editing, you should reset the changed values to their defaults to ensure that afterwards all functions work correctly again.

- If LDAP support *is active*, after pressing **Enter** z/VSE displays the *Maintain LDAP User Profiles* panel. Proceed to "Adding an LDAP User-ID to Correspond to the VSE User-ID."
- If the user is *not* an administrator (Type 1 user), after pressing **Enter** z/VSE displays the *Maintain Security Profiles* panel in which you can add or change the group connects information. Proceed to "Adding/Changing the Group Connects for a VSE User-ID" on page 302.

Adding an LDAP User-ID to Correspond to the VSE User-ID

To use an LDAP logon, you can now define the LDAP user-ID which corresponds to the VSE user-ID added in the previous step. If you wish to update the details of the LDAP user-ID at a later time (described in "Using Dialogs to Maintain LDAP User Mappings" on page 335), press **PF3**. Otherwise, follow the instructions below.

1. To *add* an LDAP user-ID, type a '**1**' in the first OPT field as shown in Figure 82 on page 300, and press **Enter**.

```
IESADMLUPM         MAINTAIN LDAP USER PROFILES

START.... ________________________________________________________________
VSE USERID.... PRG1
OPTIONS:   1 = ADD
                                                                      USER
OPT  LDAP USERID                                                      TYPE
 1














PF1=HELP                      3=END
                              9=PRINT        10=EXPORT
MAKE NECESSARY CHANGES TO VSE USER-ID 'PRG1    '.
```

Figure 82. Adding an LDAP User-ID in the Maintain LDAP User Profiles Panel

If an LDAP user-ID for the VSE user-ID already exists, it is displayed.

To *change* an LDAP user-ID, type a '**2**' in the OPT field for the LDAP user-ID you wish to change and press **Enter**.

2. The *Add or Change LDAP User Profile* panel is then displayed. Figure 83 shows an example when adding an LDAP user profile for user-ID **PRG1**. Make your selections and enter values in the fields shown.

```
IESADMLUPA         ADD OR CHANGE LDAP USER PROFILE

LDAP USERID.. prg1_LDAP_user_ID_______________________________________________

DESCRIPTION.. This is the LDAP User-ID corresponding to VSE User-ID PRG1______

VSE USERID......... PRG1____   Assigned VSE user-ID. 1-8 characters

VSE PASSWORD....... PRG1PWD    Specifies VSE password. 3-8 characters

GENERATE PASSWORD.. 2          1 - Forces generation of random VSE password
                               2 - Use current password
PASSWORD PATTERN... ________   Specifies a pattern for password generation
                               Required if password is generated
                               d - decimal digit (0-9)
                               c - character (A-Z)
                               a - decimal digit (0-9) or character (A-Z)
                               x - special character (@, # or $)
                               other - place is filled with specified character
                               blank - place is not filled with a character.

PF1=HELP                 3=END                        5=PROCESS
```

Figure 83. Panel Used for Updating an LDAP User Profile

Where:

LDAP USERID
The LDAP user-ID to be added or changed. This can be up to 64 characters. This parameter is case-sensitive.

DESCRIPTION
A free-text description. This can be up to 64 characters. This parameter is optional and case-sensitive.

VSE USERID
The VSE user-ID that is assigned to that user. It can contain between 1 and 8 alphanumeric or special characters, where the special characters can be @, #, or $. This parameter is required when TYPE=LDAP. This parameter is automatically translated to upper case.

VSE PASSWORD
The VSE password for the VSE user-ID (between 3 and 8 characters).

For *Add LDAP User Profile*: You can either:

- *Leave this field empty* - if you specified a '**1**' in the GENERATE PASSWORD field. A new VSE password is generated using the parameters defined in the PASSWORD PATTERN field. This VSE password is encrypted and stored in the LDAP mapping file.
- *Enter a VSE password* - if you specified a '**2**' in the GENERATE PASSWORD field. The entered VSE password is automatically translated to upper case. The VSE password is then encrypted and stored in the LDAP mapping file.

For *Change LDAP User Profile*: You can either:

- *Leave this field empty*:
  - If you specified a '**1**' in the GENERATE PASSWORD field, a new VSE password is generated using the parameters defined in the PASSWORD PATTERN field. This VSE password is encrypted and stored in the LDAP mapping file.
  - If you specified a '**2**' in the GENERATE PASSWORD field, the existing (previously saved) VSE password is read from the LDAP mapping file and is used for the operation.
- *Enter your existing or a new VSE password*. The password is automatically translated to upper case. The VSE password is then encrypted and stored in the LDAP mapping file.

GENERATE PASSWORD
Enter either:

- **1** which instructs the LDAP service program to generate a *random* VSE password (GENPWD). The generated password is stored in the LDAP mapping file. For details of how to specify a pattern for password generation, see parameter PASSWORD PATTERN. **Note:** If you generate a random VSE password, any previous VSE password (that you chose yourself) *will be overwritten!*
- **2** which means the VSE password (that you entered in the VSE PASSWORD field) will be used for logon.

PASSWORD PATTERN
A pattern of between 3 and 8 characters that is used when generating a password. The following characters can be used:

- d – denotes that this place is to be filled with a decimal digit (0-9).
- c – denotes that this place is to be filled with a character (AZ).
- a – denotes that this place is to be filled with either a decimal digit (0-9) or with a character (A-Z).
- x – denotes that this place is to be filled with special character @, #, or $.
- other – denotes that this place is to be filled with another specified character.
- blank – denotes that this place is not filled with a character.

3. After entering all required information in Figure 83 on page 300, press **PF5** to proceed with your request. You are returned to the *Maintain LDAP User Profiles* panel, and a confirmation message (either "User Profile Was Added Successfully" or "User Profile Was Changed Successfully") is displayed.
4. Press **PF3=END** to leave the *Maintain LDAP User Profiles* panel.

- If the user-ID is **Type 1**, the next step (in which group connects are defined) is skipped. This is because an administrator *automatically* has access to all groups. z/VSE now re-displays the *Maintain User Profiles* panel together with a message that the user has been added or changed successfully.
- If the user-ID is **not Type 1**, the *Maintain Security Profiles* panel is displayed where you can add or change this user-ID's connects to groups. Proceed to "Adding/Changing the Group Connects for a VSE User-ID."

Adding/Changing the Group Connects for a VSE User-ID

Note: If you wish, you can add or change the details of group connects at a later time (described in Chapter 30, "Protecting Resources via BSM Dialogs," on page 373).

If a user-ID was added the *Maintain Security Profiles* panel shows the added user-ID (in the example, **PRG1**) in the `USERID CONNECTED?` column together with *all* Basic Security Manager groups, as shown in Figure 84:

- The groups to which the new user-ID (in the example, **PRG1**) is *already* connected are identified with a '*'.
- The groups to which the *model user-ID* (in the example, **PROG**) is connected are identified with an **M**. The model user-ID was selected in Figure 77 on page 290.

```
IESADMBSLG                 MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS:    GROUP
START....
OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = USER LIST
           8 = CONNECT     9 = REMOVE       USERID
   OPT    GROUP NAME   DESCRIPTION          CONNECTED?
                                            PRG1
    _     GROUP01      TRANSEC CLASS MIGRAT    M
    _     GROUP02      TRANSEC CLASS MIGRAT
    _     GROUP03      TRANSEC CLASS MIGRAT    *
    _     GROUP04      TRANSEC CLASS MIGRAT    M
    _     GROUP05      TRANSEC CLASS MIGRAT
    _     GROUP06      TRANSEC CLASS MIGRAT    M
    _     GROUP07      TRANSEC CLASS MIGRAT    *
    _     GROUP08      TRANSEC CLASS MIGRAT    M
    _     GROUP09      TRANSEC CLASS MIGRAT    M
    _     GROUP10      TRANSEC CLASS MIGRAT    M
    _     GROUP11      TRANSEC CLASS MIGRAT    *
    _     GROUP12      TRANSEC CLASS MIGRAT    M

PF1=HELP                   3=END      6=CONNECT MODEL
              8=FORWARD    9=PRINT   10=REMOVE ALL
CHANGE THE SECURITY PROFILE OF USERID ACCORDING TO THE MODEL PROG    .
```

Figure 84. Adding Group Connect Information for a New VSE User-ID

In Figure 84, ***for all groups*** you can now:

- Press **PF6** to *build the same connects* as used by the model user-ID (marked with an **M**).
- Press **PF10** to *remove all existing connects* (marked with a '*').

- Press **PF3** to accept the existing connects (marked with a '*'). The connects marked with an **M** will *not* be built.

In Figure 84 on page 302, ***for a single group*** you can now:
- Type **1** (ADD) in the OPT column to add a new group.
- Type **2** (CHANGE) in the OPT column to change an existing group.
- Type **5** (DELETE) in the OPT column to delete an existing group.
- Type **8** (CONNECT) in the OPT column to connect to a group.
- Type **9** (REMOVE) in the OPT column to remove the user-ID from a group.

If you press **Enter**, **PF7**, or **PF8**, the options you have entered will be executed and the results are displayed.

After completing your changes, press **PF3**. z/VSE re-displays the *Maintain User Profiles* panel together with a message confirming that the user-ID has been successfully updated.

If a user-ID was changed the *Maintain Security Profiles* panel shows the user-ID whose connect information is to be changed (in the example, **PRG2**) in the USERID CONNECTED? column together with the Basic Security Manager groups to which the user-ID is currently connected (identified with a '*'), as shown in Figure 85.

```
IESADMBSLG                  MAINTAIN SECURITY PROFILES
 BSM RESOURCE CLASS:    GROUP
 START....
 OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = USER LIST
            8 = CONNECT     9 = REMOVE       USERID
    OPT    GROUP NAME   DESCRIPTION          CONNECTED?
                                             PRG2
     _     GROUP01      TRANSEC CLASS MIGRAT    *
     _     GROUP02      TRANSEC CLASS MIGRAT
     _     GROUP03      TRANSEC CLASS MIGRAT    *
     _     GROUP04      TRANSEC CLASS MIGRAT    *
     _     GROUP05      TRANSEC CLASS MIGRAT
     _     GROUP06      TRANSEC CLASS MIGRAT    *
     _     GROUP07      TRANSEC CLASS MIGRAT    *
     _     GROUP08      TRANSEC CLASS MIGRAT    *
     _     GROUP09      TRANSEC CLASS MIGRAT    *
     _     GROUP10      TRANSEC CLASS MIGRAT    *
     _     GROUP11      TRANSEC CLASS MIGRAT    *
     _     GROUP12      TRANSEC CLASS MIGRAT    *

 PF1=HELP                    3=END
               8=FORWARD    9=PRINT   10=REMOVE ALL
MAKE NECESSARY CHANGES TO THE SECURITY PROFILE OF THE CHANGED USERID.
```

Figure 85. Changing Group Connect Information for an Existing VSE User-ID

In Figure 85, ***for all groups***, you can now press **PF10** to *remove all existing connects*.

In Figure 85, ***for a single group*** you can now:
- Accept the existing connects (marked with a '*') by pressing **PF3**.
- Type **1** (ADD) in the OPT column to add a new group.
- Type **2** (CHANGE) in the OPT column to change an existing group.
- Type **5** (DELETE) in the OPT column to delete a group.
- Type **8** (CONNECT) in the OPT column to connect to a group.

- Type **9** (REMOVE) in the OPT column to remove a connect to a group.

If you press **Enter**, **PF7**, or **PF8**, the options you have entered will be executed and the results are displayed.

After completing your changes, press **PF3**. z/VSE re-displays the *Maintain User Profiles* panel together with a message confirming that the user-ID has been successfully updated.

Deleting a User-ID and Profile Definitions

To delete a user-ID, in the *Maintain User Profiles* panel (shown in Figure 77 on page 290)

1. Type '**5**' in the OPT column for the user-ID and press **Enter**. The next step depends upon whether or not the user-ID is LDAP-enabled.
2. When deleting a user-ID that is LDAP-enabled z/VSE displays the *Maintain LDAP User Profiles* panel containing the LDAP user-ID you wish to delete.
   a. Type a '**5**' in the OPT column for this user-ID, and press **Enter**. z/VSE displays a message confirming that the LDAP user profile has been deleted.
   b. On pressing **PF3**, the *Maintain Security Profiles* panel is displayed as shown in Figure 84 on page 302. You can now either:
      - Remove *all* existing connects by pressing **PF10**.
      - Remove specific group connects by typing a '**9**' in the OPT column for one or more groups, and pressing **Enter**.
      - Leave all existing group connects unchanged by pressing **PF3**.

   In all other cases z/VSE displays the *Maintain Security Profiles* panel as shown in Figure 84 on page 302. You can now either:
   - Remove *all* existing connects by pressing **PF10**.
   - Remove specific group connects by typing a '**9**' in the OPT column for one or more groups, and pressing **Enter**.
   - Leave all existing group connects unchanged by pressing **PF3**.
3. After leaving the *Maintain Security Profiles* panel, z/VSE displays the *Maintain User Profiles* panel together with a message confirming that the deletion was successful. z/VSE has now deleted the user profile record from the VSE control file (IESCNTL), and the profile definitions from the BSM control file (BSTCNTL). If the user has access to VSE/ICCF:
   - The VSE/ICCF DTSFILE entry for type 1 and 2 users with a 4-character user-ID is deleted by a batch job created and submitted by the dialog.
   - If the VSE/ICCF user owns a PRIMARY sublibrary, the administrator is asked to confirm the deletion of this user-ID.

Note: After deleting the user-ID, you can use BSM reports to check that you have deleted all related profile definitions from the BSM control file. See “Generating BSM Cross Reference Reports” on page 305.

Generating a Job to Create BSM Groups

To create BSM groups, in the *Maintain User Profiles* panel (shown in Figure 77 on page 290) press **PF6**. A job is then generated with the name IESTBGRI. You can use job IESTBGRI to:

- Create BSM groups.
- Connect these groups to specific user-IDs.

Job IESTBGRI is stored in the punch queue with the file name **CICSICCF**. For details, see "Performing the Migration" on page 285.

Creating a Status of User-IDs Using the Dialog

To create a status report of user profiles that are stored in the VSE Control File, in the *Maintain User Profiles* panel (shown in Figure 77 on page 290) press **PF9**. A status report is then created using the reporting tool IESXSPR, and stored in the VSE/POWER List Queue. The job name of this List Queue entry is IESXSUSP.

The skeleton IESXSUSP is provided in VSE/ICCF Library 59. This skeleton contains the source code of the report format. To create your own report layouts, you can modify this source code.

If you change the skeleton IESXSUSP, you must activate the related phase using the CEMT SET PROG(IESXSUSP) NEWCOPY command.

Maintaining CICS User Profiles without VSE/ICCF

You can also update user profiles for CICS users in an environment without VSE/ICCF (in case of a second CICS, for example), or if:

- VSE/ICCF has been terminated.
- The VSE/ICCF DTSFILE has been disconnected.
- The system administrator is a non-VSE/ICCF user.

Generating BSM Cross Reference Reports

The *BSM Cross Reference reports* provide information about:

- User-IDs
- Groups
- Access control classes
- Resources which can be accessed via the UACC definition in the profile.
- Undefined user-IDs found in groups and access lists of resource profiles.

Using the BSM Cross Reference reports, you can manage the *profile definitions* that are stored in the:

- VSE control file (IESCNTL)
- Table DTSECTAB
- BSM control file (BSTCNTL)

BSM Cross Reference reports are especially useful for checking that after you delete a user-ID, you have also deleted *all* related profile definitions from access lists and groups.

You can generate your BSM Cross Reference reports using either the:

- BSTXREF service, described in "Using the BSTXREF Service."
- *BSM Cross Reference Report* panel (Fast Path **286**), described in "Using the BSM Cross Reference Report Dialog" on page 307.

Using the BSTXREF Service

To use the BSM cross reference service, start BSTXREF by using this statement:

```
EXEC BSTXREF,PARM='parameters'
```

You can call BSTXREF from either a:
- Console job (such as PAUSEBG).
- Batch job.

The generated report (that has the job name BSTXREF) is created in the VSE/POWER list queue.

These are the BSTXREF parameters you can enter (***in UPPER CASE only***):

PARM='USERID=*USER* **[,L]'**
The user information for user-ID *USER* will be listed, which includes:
- Whether or not this user-ID is defined in the VSE control file (IESCNTL).
- Whether or not this user-ID is an administrator.
- The groups to which this user-ID is connected.
- The resource profiles that contain this user-ID on their access lists. If you specify an '**L**' (to generate a detailed report), all resource profiles are additionally listed where this user-ID is authorized via a group.
- All ACCs (access control classes) and access rights of this user-ID.
- If *batch security is active* (SYS SEC=YES), then:
  - If you specify an '**L**' (to generate a detailed report), all DTSECTAB resource entries that have an ACC (access control class) of this user-ID on their list of ACCs.
  - If this user-ID is defined in the DTSECTAB, the following DTSECTAB information will be listed:
    - Whether or not the user-ID is type "administrator".
    - All ACCs (access control classes) and access rights of this user-ID.
    - All resources where an ACC of this user-ID is contained in the ACC list.

PARM='USERID=*[,L]'
The user information for *all user-IDs* will be listed.

PARM='GROUP=*GROUP_NAME***'**
The group information for group *GROUP_NAME* will be listed. This consists of:
- A description of the group.
- The user-IDs that are connected to this group.
- The resource profiles whose access lists contain this group.

PARM='GROUP=*'
The group information for *all groups* will be listed.

PARM='ACC=1|...|32'
All user-IDs will be listed that have the specified access control class. If batch security is active (SYS SEC=YES), then:
- All user-IDs defined in DTSECTAB that have this ACC will be listed.

- All resource entries from DTSECTAB that have this ACC contained in their ACC list will be listed.

PARM='ACC=*'
All access control classes will be listed.

PARM='UACC'
All resource profiles will be listed that have a UACC which is *not* NONE. If batch security is active (SYS SEC=YES), all resources defined in DTSECTAB that have a UACC which is neither NONE nor CONNECT will be listed.

PARM='INCONS[,L]'
All user-ID inconsistencies will be listed. These are user-IDs found in groups or on access lists of resource profiles, but that are not defined in the VSE control file (IESCNTL) as a user-ID. If you specified an '**L**' (to generate a detailed report), all resource profiles are additionally listed where this user-ID is authorized via a group.

Using the BSM Cross Reference Report Dialog

To use this dialog, you must:

1. Display the main panel of the *BSM Cross Reference Report* dialog. To so do, start with the *z/VSE Function Selection* panel and select Fast Path **286**.

```
IESADMBSXT             BSM CROSS REFERENCE REPORT


OPTIONS:  1 = REPORT   2 = DETAILED REPORT

   OPT    REPORT NAME

    _     Information about user-ID *_______
    _     Information about group *_______
    _     Information about access control class *_ (1..32)
    _     Information about all user-ID inconsistencies
    _     Information about UACC that allow resource access


   * = ALL





PF1=HELP                    3=END
```

2. For one of the above options, select the level of detail you require (**1** = obtain a summary report, **2** = obtain a detailed report).
3. For the options below you must enter additional information.
   - For *Information about user-ID*, enter either a specific user-ID, or an asterisk (the default) to obtain a report of all user-IDs.
   - For *Information about group*, enter either a specific group name, or an asterisk (the default) to obtain a report of all groups.
   - For *Information about access control class*, enter the number of an access control class (ACC) between 1 and 32, or an asterisk (the default) to obtain a report of all ACCs.
4. Press ENTER, and a member CICSICCF is created in your punch queue. Use Fast Path **32** (*Manage Batch Queues*) and **3 = Punch Queue** to locate member CICSICCF (that contains the job BSTXREF).
   a. Enter option **4** to copy member CICSICCF to your z/VSE Primary Library.

b. Use Fast Path **51** (*Program Development Library*) to view member CICSICCF.
c. Using option **7**, you can submit job BSTXREF.
d. Job BSTXREF is then created in the z/VSE List Queue. It contains the BSM Cross Reference Report.
e. Use Fast Path **32** (*Manage Batch Queues*) and **1 = List Queue** to view the display, change, print, or delete the report (BSTXREF).

The contents of the report (BSTXREF) vary according to the selection you made:

Information about user-ID
The generated report provides this information:
- Whether or not this user-ID is defined in the VSE control file (IESCNTL).
- Whether or not this user-ID is an administrator.
- The groups to which this user-ID is connected.
- The resource profiles that contain this user-ID on their access lists.
- Whether or not this user-ID is defined in the DTSECTAB table. If this user-ID *is defined* in the DTSECTAB, this information (taken from DTSECTAB) will be listed:
  – Whether or not the user-ID is type "administrator".
  – All ACCs (access control classes) and access rights of this user-ID.
  – All resources where an ACC of this user-ID is contained in the ACC list.
- In you request a *detailed report*, this additional information is provided:
  – The resource profiles for which this user-ID is authorized via a group.
  – The DTSECTAB resource entries which have an access control class for this user on their list of ACCs.

Information about group
The generated report provides this information:
- A description of the group.
- The user-IDs that are connected to this group.
- The resource profiles whose access lists contain this group.

Information about access control class
The generated report lists all users for whom the specified access control class (ACC) has been defined.

Information about all user-ID inconsistencies
The generated report lists all user-ID inconsistencies. If you request a *detailed report*, the resource profiles in which this user-ID is authorized via a group are also listed.

Information about UACC that allow resource access
The generated report lists all resource profiles that have a UACC which is *not* NONE.

Additional Considerations When Maintaining User Profiles via Dialogs

Creating a Status Report of User-IDs Using IESBLDUP

To maintain user profiles, you need an up-to-date record of the users defined to your system. You can print such a status report with the migration utility program (IESBLDUP), as described in Chapter 25, "Maintaining User Profiles via BSM Dialogs," on page 289.

You can also create a status report using the PRINT function of the *Maintain User Profiles* panel. For details, see "Creating a Status of User-IDs Using the Dialog" on page 305.

Also refer to the topic "IESBLDUP Utility" in the manual *z/VSE System Utilities* for a general description of the program and for a job stream example.

Dialog Considerations

1. If you change a user profile which is currently being used on the system, any new options do **not** immediately take effect. The user must sign off and sign on again to take advantage of new or changed options.
2. When you add a new user, the user-ID that you enter the option number next to is used as a model. The values defined for the model are used as defaults for the new profile you are defining.

   With this, you can add new profiles using existing profiles as models. z/VSE provides user profiles for type 1 and type 2 users. If you do not need to change the defaults, you simply have to enter a new user-ID and password.

   If some z/VSE defaults are not satisfactory, define a new profile and enter your own values. You can then use the new profile as a model to define other users on the system.
3. Observe the following when you add or change a profile:
   - When you add a user, select a model profile which has the same profile type (1, 2, or 3) and the same length of the user-ID (4 characters or 5 to 8 characters) that you want for the new user.
   - If you change a type 1 profile to a type 2 or 3, the options that do not apply to the new user type (2 or 3) are set to 0.
   - If you add or change a profile and you change the user type, you **must update** the VSE/ICCF information. On the *Specify Library* panel, specify 2 (NO) indicating that you do not accept the defaults.

     Note:

     a. Refer to "Planning Considerations for Using the Interactive Interface" on page 123 for details about VSE/ICCF dependencies.
     b. If you do not update the VSE/ICCF information, the defaults for the original user profile types 1 and 2 are used as defaults for the new user-ID. This could result in incorrect authorization values for the new user.
4. If you delete a type 1 or 2 user profile, you must disconnect the VSE/ICCF DTSFILE. A message requests you to do so when the job stream created is being processed.
5. z/VSE ensures that only one user can access the *Maintain User Profiles* dialog at one time.

6. Any VSE/ICCF information you enter is saved until you leave the dialog. When you finish the dialog, it makes all updates to the VSE/ICCF DTSFILE at the same time. Because of this, you can maintain several user profiles at once without waiting for the system to make the DTSFILE updates one at a time. However, when you leave the dialog, you may notice a delay while the dialog updates the DTSFILE.

VSE/ICCF Library Considerations

You can allocate VSE/ICCF libraries 3 to 7, 11 to 49, and 70 to 199 as user libraries. Other libraries are for use by the system. Further planning details about VSE/ICCF libraries are provided in the *z/VSE Planning* manual under “VSE/ICCF Libraries”. Refer also to “Reformatting the VSE/ICCF DTSFILE” on page 174.

Programmer (type 2) profiles cannot access library 1. You should not define their VSE/ICCF primary library as library 1. By default bit setting, they have **read access** only to public libraries 50 - 69. However, this is only true if a user is working with the Interactive Interface. When using the command mode of VSE/ICCF, a user can switch to library 51 and read from as well as write to that library.

Note also that a FULIST for type 2 users does not display members that have been defined as shared through the VSE/ICCF utility DTSUTIL. Such members are only accessible for the owner and the system administrator.

VSE/ICCF Interactive Partitions

The manual *z/VSE Planning* has information about the characteristics and layout of the predefined VSE/ICCF interactive partitions under “VSE/ICCF Interactive Partition Layout and Characteristics”. VSE/ICCF interactive partition requirements and eligibility for concurrent execution are important considerations when you create your own panel hierarchy. If you increase the size of existing interactive partitions or add class A and B partitions, you also should make a corresponding increase to the size of the CICS/ICCF (F2) partition. Interactive partitions reside in the partition GETVIS area of the CICS/ICCF partition.

VSE/ICCF DTSFILE Considerations

When the system updates the DTSFILE in an interactive partition, it uses the VSE/ICCF utility program DTSUTIL. Output from the job is put in VSE/ICCF library member U$xxxx.P (xxxx is your user-ID). The member is in your default primary library. The system replaces the contents of U$xxxx.P each time you run this task.

If there is a power failure or other system interruptions before the dialog ends and updates to the DTSFILE are complete, the contents of the z/VSE control file and the DTSFILE may not match. If you think that this has occurred, do the following:

1. Access the *Maintain User Profiles* dialog again.
2. Select the CHANGE option (2) for the user profile(s) you were working with.
3. Request an update (PF5) of the VSE/ICCF information. This is necessary to ensure that all profile information is consistent.

If you do these steps, the information in the DTSFILE and the z/VSE control file will agree with each other.

Chapter 26. Maintaining User Profiles via Batch Program IESUPDCF

This chapter describes the batch utility program **IESUPDCF**, which allows the system administrator to maintain user profiles in the VSE Control File (IESCNTL) and in the VSE/ICCF DTSFILE. Using this program, you can **ADD, ALT**er, and **DEL**ete user profiles. IESUPDCF helps you save time when configuring user profiles.

This chapter contains these main topics:

Preparing to Use Batch Program IESUPDCF

The following topic describes the procedures you should perform before using IESUPDCF.

Planning for User Profiles

With z/VSE you can use three types of user profiles.

A VSE/ICCF (short form: ICCF) user profile is a type 1 or type 2 user profile with a 4 character user-ID. It is defined in the VSE control file (IESCNTL) and also in the VSE/ICCF DTSFILE.

Model profiles for type 1 and type 2 user profiles are provided:

Type 1 User Profile
: Valid for the System Administrator. Access to all z/VSE functions, including ICCF.

Type 2 User Profile
: Valid for Operators and Programmers. Access to most of the z/VSE functions, including ICCF.

Type 3 User Profile
: Valid for general users (and Type 1 and Type 2 users with a user-ID of 4 to 8 characters). Access to selected functions, but not to ICCF.

Information for ICCF users is recorded in the IESCNTL and in the DTSFILE. Information for type 3 user profiles is only recorded in the IESCNTL file. For the following discussion you should know that ICCF-related definitions (PASSWORD and LIBRARY) are recorded in two places: in the DTSFILE and in the IESCNTL file.

"Skeleton IESUPDCF" on page 318 shows skeleton IESUPDCF. It is shipped in ICCF library 59. You have to change this skeleton to add, alter, or delete user profiles. Before you change skeleton IESUPDCF, you should carefully plan for the types of users you want to create.

Preparing Skeleton IESUPDCF

You have to prepare skeleton IESUPDCF according to your needs. This may include:

- Set the **ICCF** parameter for all users referred to in the job.
- Insert **ADD** statements for adding user profiles.
- Insert **ALT**er statements for altering user profiles.
- Insert **DEL**ete statements for deleting user profiles.

The following topics have more details.

Setting the ICCF Parameter in Skeleton IESUPDCF

With the setting of the ICCF parameter, you control the generation of job DTRUPD, which updates the DTSFILE. You must enter either Yes, No, or Ignore. There is no default.

```
►►──ICCF=──┬─Yes────┬──────────────────────────────────────────────►◄
           ├─No─────┤
           └─Ignore─┘
```

ICCF=YES
IESUPDCF updates user profiles in the control file (IESCNTL). For ICCF users, IESUPDCF updates user profiles in the DTSFILE. Therefore, a new job DTRUPD is generated.

The following describes how specifying ICCF=YES affects the ADD, ALTER, and DELete statements:

ADD The new user is added to the IESCNTL control file. The definitions of the model user profile are used as default.

If the model profile is for an ICCF user and the new user-ID is 4 characters long, then the new user will also be an ICCF user. Thus, the DTSUTIL statement is generated for job DTRUPD.

ALTer The user definition is altered in the IESCNTL control file. If the user profile is for an ICCF user, a DTSUTIL statement is generated for job DTRUPD.

DELete
The user definition in the IESCNTL control file is deleted. If the user profile is an ICCF user, the DTSUTIL statement for job DTRUPD is generated.

ICCF=No
No update of the DTSFILE is performed. This means that you cannot ADD or DELete ICCF users. In addition, you cannot ALTer the password or the ICCF library of ICCF users.

The following describes how specifying ICCF=NO affects the ADD, ALTer, and DELete statements:

ADD If the model profile is for an ICCF user, then the ADD statement is ignored, and an error message is inserted into the listing. If the model profile is not for an ICCF user, the new user is added to the IESCNTL control file.

ALTer ICCF-related definitions (PASSWORD and LIBRARY) are not altered in the IESCNTL control file.

DELete
If the user profile is an ICCF user, the statement is ignored and an error message is inserted into the listing. For type 3 user profiles, the definition in the control file is deleted.

ICCF=Ignore
You can ADD, ALTer, or DELete any user in the control file (IESCNTL). For VSE/ICCF users, however, the DTSFILE is not updated. IGNORE must be used if the control file is not related to an ICCF subsystem.

The following describes how specifying ICCF=IGNORE affects the ADD, ALTer, and DELete statements:

ADD The new user is added to the control file.

ALTer All specified parameters are altered in the control file. If specified, the LIBRARY parameter is ignored, since it is only relevant for ICCF subsystems.

DELete
The user definition in the control file is deleted, independent of the user profile type.

Refer also to item 2 in skeleton IESUPDCF, shown in "Skeleton IESUPDCF" on page 318.

Adding a User-ID in Skeleton IESUPDCF

To **ADD** a user-ID, you insert the following statement into skeleton IESUPDCF:

```
►►──ADD user_id,password,profile──┬─────────────────┬──┬───────────┬──────────────────────►
                                  └─,Catalog=catname─┘  └─,Days=nnn─┘

►──┬─────────────────┬──┬──────────────────────────┬──┬─────────────────────┬─────────────►
   └─,Library=libname─┘  └─,Initial=──┬─nnnnnnnn(A)─┬─┘  └─,Operator=oper_id─┘
                                      └─nnnnnnnn(S)─┘

►──┬─────────────────────────┬──┬────────────────────┬──┬──────────────────┬──────────────►
   └─,PGMID=programmer_id─┘  └─,PRiority=priority─┘  └─,Synonym=user_id─┘

►──┬─────────────┬──┬─────────────────┬──┬─────────────────┬──┬─────────────────┬─────────►
   └─,Timeout=mm─┘  └─,APM=──┬─Yes─┬─┘  └─,BAT=──┬─Yes─┬─┘  └─,BQA=──┬─Yes─┬─┘
                             └─No──┘             └─No──┘             └─No──┘

►──┬─────────────────┬──┬─────────────────┬──┬─────────────────┬──┬─────────────────┬─────►
   └─,CMD=──┬─Yes─┬─┘  └─,COD=──┬─Yes─┬─┘  └─,COU=──┬─Yes─┬─┘  └─,DVF=──┬─Yes─┬─┘
            └─No──┘             └─No──┘             └─No──┘             └─No──┘

►──┬─────────────────┬──┬─────────────────┬──┬──────────────────┬─────────────────────────►
   └─,ESC=──┬─Yes─┬─┘  └─,MVC=──┬─Yes─┬─┘  └─,Natlang=──┬─E─┬─┘
            └─No──┘             └─No──┘                 └─J─┘

►──┬──────────────────┬──┬──────────────────┬──┬─────────────────┬────────────────────────►
   └─,NEWS=──┬─Yes─┬─┘  └─,OLPD=──┬─Yes─┬─┘  └─,PSL=──┬─Yes─┬─┘
             └─No──┘              └─No──┘             └─No──┘

►──┬────────────────────────┬──┬─────────────────┬──┬─────────────────┬───────────────────►
   └─,Revoke=──┬─mm/dd/yy─┬─┘  └─,SPM=──┬─Yes─┬─┘  └─,UPM=──┬─Yes─┬─┘
               └─0────────┘             └─No──┘             └─No──┘

►──┬─────────────────┬────────────────────────────────────────────────────────────────────►◄
   └─,XRF=──┬─Yes─┬─┘
            └─No──┘
```

The first three parameters are mandatory; the rest are optional. **Do not change the order of the mandatory parameters**.

Mandatory Parameters

Note: Each ADD statement may use one or more physical lines. A continuation line is indicated by the continuation character "–" as the last character in the previous line. The continuation character must be preceded by a blank or a comma. The required parameters must be specified together with the ADD statement on *one* line.

user-id
: The user-ID which identifies the user to the system. It must be 4-8 alphanumeric characters long and may include the characters @, #, or $. Blanks are not allowed.

  Note: For ICCF users, the user-ID can only be 4 characters long.

password
: This is the password associated with the user-ID. It can be 3 - 8 alphanumeric characters long and may include the characters @, #, or $. Blanks are not allowed.

profile
: This is the identification (user-ID) of a user already defined to the system

and used as a model for the new user. It must be 4-8 characters long. Using optional parameters you can alter the defaults for the new user-ID.

Optional Parameters

This topic describes the optional parameters you can use with either the ADD statement, or the ALTer statement (described on page "Altering a User-ID in Skeleton IESUPDCF" on page 317).

Catalog=*catname*
: The name of the user's default VSE/VSAM catalog (IJSYSCT). This parameter is not available for type 3 users.

Days=*nnn*
: The number of days before the user's password expires. Specify a number between 0 and 365. If you enter 0, the password will **not** expire.

Library=*libname*
: The user's primary ICCF library. This value may be 4 digits in length. When specifying:
- ICCF=IGNORE, the LIBRARY parameter is ignored, since it is only relevant for ICCF subsystems.
- ICCF=NO, you cannot change the library for an VSE/ICCF user.

Initial=*nnnnnnnn(A)***|***nnnnnnnn(S)*
: Initial function performed at sign on. You can use up to eight alphanumeric characters. The value must be followed by the type specification:
- (A) - if the initial function is an application, or
- (S) - if the initial function is a selection panel.

For example:

`INITIAL=FUNCNAME(A)` for an application.

Operator=*oper-id*
: 3-character operator identification for CICS. The ID must be unique.

PGMID=*Programmer-id*
: Up to 20-character programmer name.

PWD|PAssword=*password*
: It can be 3 - 8 alphanumeric characters long and may include the characters @, #, or $. Blanks are not allowed. You cannot change the password for an ICCF user when specifying ICCF=NO. This parameter is only to be used together with the ALTer statement.

PRiority=*priority*
: The value which CICS uses for the dispatching priorities of the user. Enter a number from **0 - 255**. 0 is the highest priority; 255 the lowest.

Synonym=*user-id*
: This defines the user-ID to be used as a model for synonyms. z/VSE provides synonyms for users SYSA, PROG, and OPER. These can be used as models for other users.

Timeout=*mm*
: Gives the value in minutes used by CICS to initiate sign off after the value specified has elapsed since the latest terminal activity. You can specify a value from 0 to 60. The value you specify is always rounded up to a multiple of 5 minutes. A value of 0 means no time out. 0 should be specified for ICCF users.

APM=Yes|No
When set to “Yes”, you can create and maintain application profiles. This is only valid for the administrator (type 1) user profiles.

BAT=Yes|No
When set to “Yes”, you can submit jobs for batch processing.

BQA=Yes|No
When set to “Yes”, you can manage all VSE/POWER jobs of an ICCF user.

CMD=Yes|No
When set to “Yes”, you can enter system console commands from the *System Console* dialog. This authorization is **not** available for general (type 3) user profiles.

COD=Yes|No
When set to “Yes”, you will get a confirmation message when deleting VSE/POWER queue entries, VSE/ICCF library members, or BSM resource definitions.

COU=Yes|No
When set to “Yes”, all console output is shown.

DVF=Yes|No
When set to “Yes”, you can define and delete VSE/VSAM files, libraries, alternate indexes, and alternate names. This authorization is **not** possible for general (type 3) user profiles.

ESC=Yes|No
When set to “Yes”, you can *escape* to CICS. This lets you leave the Interactive Interface and work directly with CICS.

MVC=Yes|No
When set to “Yes”, you can process VSE/VSAM catalogs and define and delete VSE/VSAM space. This authorization is **not** available for general (type 3) user profiles.

Natlang=E|J
National language indicator for this user.E = English, J = Japanese

NEWS=Yes|No
When set to “Yes”, the system displays *news items* to you. News items are messages which the system displays when you sign on or when you are already signed on.

OLPD=Yes|No
When set to “Yes”, you can delete Online Problem Determination (OLPD) incident records from the system. This authorization is **not** available for general (type 3) user profiles.

PSL=Yes|No
When set to “Yes”, you will have a private sublibrary (primary user-ID).

Revoke=*mm/dd/yy***|0**
Revoke date when the user-ID will be revoked. If zero is specified, the user-ID will never get revoked.

SPM=Yes|No
When set to “Yes”, you can create and maintain selection panels. This is only valid for the administrator (type 1) user profiles.

UPM=Yes|No
: When set to "Yes", you can create and maintain user profiles. This is only valid for the administrator (type 1) user profiles.

XRF=Yes|No
: When set to "Yes", the user gets signed off after XRF takeover. Otherwise, the user stays signed on.

Altering a User-ID in Skeleton IESUPDCF

To **ALT**er a user-ID, you insert the following statement into skeleton IESUPDCF:

```
►►──ALTer userid──┬──────────────────┬──┬─────────────┬──┬──────────────────┬──►
                  └─,Catalog=catname─┘  └─,Days=nnn───┘  └─,Library=libname─┘

►─┬─────────────────────────────┬──┬────────────────────┬──┬────────────────────┬──►
  └─,Initial=──┬─nnnnnnnn(A)─┬──┘  └─,Operator=oper-id──┘  ├─,PWD=password──────┤
               └─nnnnnnnn(S)─┘                             └─PAssword=password──┘

►─┬──────────────────────────┬──┬──────────────────────┬──┬───────────────────┬──►
  └─,PGMID=programmer-id─────┘  └─,PRiority=priority───┘  └─,Synonym=user-id──┘

►─┬─────────────┬──┬───────────────┬──┬───────────────┬──┬───────────────┬──►
  └─,Timeout=mm─┘  └─,APM=──┬─Yes─┬─┘  └─,BAT=──┬─Yes─┬─┘  └─,BQA=──┬─Yes─┬─┘
                            └─No──┘            └─No──┘            └─No──┘

►─┬───────────────┬──┬───────────────┬──┬───────────────┬──┬───────────────┬──►
  └─,CMD=──┬─Yes─┬─┘  └─,COD=──┬─Yes─┬─┘  └─,COU=──┬─Yes─┬─┘  └─,DVF=──┬─Yes─┬─┘
           └─No──┘            └─No──┘            └─No──┘            └─No──┘

►─┬───────────────┬──┬───────────────┬──┬─────────────────┬──────────────────►
  └─,ESC=──┬─Yes─┬─┘  └─,MVC=──┬─Yes─┬─┘  └─,Natlang=──┬─E─┬─┘
           └─No──┘            └─No──┘               └─J─┘

►─┬────────────────┬──┬────────────────┬──┬───────────────┬──────────────────►
  └─,NEWS=──┬─Yes─┬─┘  └─,OLPD=──┬─Yes─┬─┘  └─,PSL=──┬─Yes─┬─┘
            └─No──┘             └─No──┘            └─No──┘

►─┬───────────────────────────┬──┬───────────────┬──┬───────────────┬──────►
  └─,Revoke=──┬─mm/dd/yy─┬────┘  └─,SPM=──┬─Yes─┬─┘  └─,UPM=──┬─Yes─┬─┘
              └─0────────┘               └─No──┘            └─No──┘

►─┬───────────────┬─────────────────────────────────────────────────────►◄
  └─,XRF=──┬─Yes─┬─┘
           └─No──┘
```

ALTer checks the user types and performs those changes allowed for the specific user types. For the ALTer statement the first parameter (*userid*) is mandatory. The user-ID identifies the user to the system. It must be 4-8 alphanumeric characters long and may include the characters @, #, or $. Blanks are not allowed. You also must specify at least **one** additional parameter from the optional parameters shown in the ALTer statement syntax. The optional parameters you can use are described in detail under "Optional Parameters" on page 315. You may have more than one ALTer statement for the same user profile.

Each ALTer statement may use one or more physical lines. A continuation line is indicated by the continuation character "–" as the last character in the previous line. The continuation character must be preceded by a blank or a comma. The *user-id* must be specified on the same line as the ALTer statement.

Deleting a User-ID in Skeleton IESUPDCF

To **DEL**ete a user-ID, you insert the following statement into skeleton IESUPDCF:

```
►►──DELete user_id──────────────────────────────────────────────────────────►◄
```

For the DELete statement, the user-ID identifies the user which is to be deleted. It must be 4-8 alphanumeric characters long and may include the characters @, #, or $. Blanks are not allowed.

Skeleton IESUPDCF

The example below shows skeleton IESUPDCF, shipped in ICCF library 59. Use this skeleton to ADD, ALTer, or DELete your user profiles.

```
* $$ JOB JNM=IESUPDCF,CLASS=0,DISP=D
* $$ PUN DISP=I,CLASS=0,PRI=9
// JOB IESUPDCF
// OPTION NOLOG
*
*  THIS SKELETON MAY BE USED BY THE ADMINISTRATOR TO GENERATE A
*  JOB FOR BATCH USER PROFILE MAINTENANCE.
*    1. IF THE CONTROL FILE BELONGS TO A CICS WITHOUT ICCF AND THIS
*       CICS DOES NOT SHARE THE CONTROL FILE WITH CICS/ICCF,
*       ADJUST THE '// DLBL' STATEMENT TO MAINTAIN
*       USER PROFILES IN THE RELATED CONTROL FILE.
*    2. SUPPLY AN OPERAND FOR THE ICCF PARAMETER, VALID OPERANDS ARE:
*         Yes    ... UPDATE USER PROFILES IN CONTROL FILE (CICS) AND
*                    IN THE DTSFILE (ICCF).
*         No     ... UPDATE USER PROFILES IN CONTROL FILE ONLY.
*                    INHIBIT CHANGES TO ICCF RELATED INFORMATION.
*         Ignore ... UPDATE USER PROFILES IN CONTROL FILE ONLY.
*                    THIS VALUE MUST BE USED IF THE CONTROL FILE
*                    IS USED IN CICS SUBSYSTEMS RUNNING WITHOUT ICCF.
*    3. INSERT THE ADD, ALTER AND DELETE STATEMENTS THAT YOU NEED TO
*       MAINTAIN USER PROFILES.
*          SAMPLE STATEMENTS:
*          ==================
*            * TEXT   ... A COMMENT LINE
*            ADd     USERID,PASSWD,PROFILE(,OPTIONAL PARAMETERS)
*            ALter   USERID(,OPTIONAL PARAMETERS)
*            Delete  USERID
*          EXPLANATION OF PARAMETERS:
*          ==========================
*           1. REQUIRED AND POSITIONAL PARAMETERS:
*           -------------------------------------
*            USERID  ... THE ID OF THE USER ( ADD, ALTER, DELETE )
*                        ( 4-8 CHARACTER / 4 CHARACTER FOR ICCF USER )
*            PASSWD  ... THE PASSWORD OF THE USER ( ADD )
*                        ( 3-8 CHARACTERS )
*            PROFILE ... THE ID OF THE USER USED AS PROFILE FOR
*                        THE NEW USER  ( ADD )
*                        ( 4-8 CHARACTER / 4 CHARACTER FOR ICCF USER )
*
*           2. OPTIONAL PARAMETERS IN ADD/ALTER STATEMENT:
*           ----------------------------------------------
*            Catalog=   ... THE DEFAULT CATALOG OF THE USER
*                       EXAMPLE: CAT=VSESPUC
*            Days=      ... NUMBER OF DAYS IN EXPIRATION INTERVAL
*                       EXAMPLE: DAYS=20  ( RANGE: 0-365)
*            Library=   ... Primary ICCF library ( only ICCF users )
*                       EXAMPLE: LIB=20
*            Initial=   ... Initial function at SIGNON
*                       EXAMPLE: INIT=APPLNAME(A) ... FOR APPLICATION
*                                INIT=SELNAME(S)  ... FOR SELECTION P.
*            Natlang=   ... NATIONALLANGUAGE_INDICATOR
*                       EXAMPLE: NAT=E ( for English)
*            PGMID=     ... PROGRAMMER ID (maximum 20 characters)
```

```
*                      EXAMPLE: PGMID=G_SMITH
*           Operator=  OPERATOR ID
*                      EXAMPLE: OPER=OPE
*           PWD=       ... USER PASSWORD
*           PAssword=  EXAMPLE: PWD=PASSWD  ( 3-8 Characters )
*           PRiority=  ... OPERATOR PRIORITY
*                      EXAMPLE: PRIOR=5  ( RANGE: 0-255 )
*           Revoke=    ... REVOKE_DATE
*                      EXAMPLE: R=01/31/01 ( Format mm/dd/yy )
*           Synonym=   ... SYNONYMS MODEL
*                      EXAMPLE: SYNONYM=SYNS  ( 4-8 CHARACTERS )
*           Timeout=   ... TIMEOUT INTERVAL
*                      EXAMPLE: TIME=20  ( VALUES: 0,5,10,...,60 )
*           APM=Yes|No   ... APPLICATION PROFILE MAINTENANCE
*           BQA=Yes|No   ... MANAGE ALL BATCH QUEUES
*           CMD=Yes|No   ... ENTER CONSOLE COMMANDS
*           COU=Yes|No   ... FULL OUTPUT ON SYSTEM CONSOLE
*           COD=Yes|No   ... CONFIRM ON DELETE
*           DVF=Yes|No   ... DEFINE VSAM FILES
*           ESC=Yes|No   ... ESCAPE TO CICS
*           MVC=Yes|No   ... MANAGE VSAM CATALOGS
*           NEWS=Yes|No  ... DISPLAY NEWS TO USER
*           OLPD=Yes|No  ... DELETE OLPD INCIDENTS
*           PSL=Yes|No   ... OWNS A PRIVATE SUBLIBRARY
*           BAT=Yes|No   ... SUBMIT TO BATCH
*           SPM=Yes|No   ... SELECTION PANEL MAINTENANCE
*           UPM=Yes|No   ... USER PROFILE MAINTENANCE
*           XRF=Yes|No   ... XRF SIGNOFF
*           -                ... CONTINUATION CHARACTER
*    4. DELETE BLOCK 'UPDPL', IF YOU DO NOT WANT TO MAINTAIN
*       THE PRIMARY LIBRARY.
* ***************************************************************
* IESCNTL MUST BE CLOSED IF UPDATES ARE DONE. PERFORM
*   CEMT SET FILE(IESCNTL) CLOSE   IN EACH CICS WITH THE
*                                  INTERACTIVE INTERFACE ACTIVE
*   MSG FB,DATA=CLOSECNTL          TO CLOSE THE FILE IN BSM
*
* IMPORTANT:
*     IF NEW USERS ARE ADDED, IN ORDER TO DEFINE THIS USERS
*     TO THE BSTCNTL BASED SECURITY, YOU HAVE TO PRESS PF6 (GROUPS)
*     ON PANEL 'USER PROFILE MAINTENANCE' FASTPATH 211.
*     THE USERS CAN ALSO BE ADDED USING BSTADMIN:
*         // EXEC BSTADMIN
*            CONNECT GROUPxx   user
*         /*
*     IF USERS ARE DELETED, THIS USERS SHOULD BE REMOVED FROM THE
*     GROUP:
*         // EXEC BSTADMIN
*            REMOVE GROUPxx   user
*         /*
*
* ******************************************************************
*
* ====> UPDATE NEXT LINE IF NECESSARY (SEE 1.)
// DLBL IESCNTL,'VSE.CONTROL.FILE',,VSAM,CAT=VSESPUC
*
// EXEC PROC=DTRICCF
// EXEC IESUPDCF,SIZE=64K
*
* ====> SUPPLY AN OPERAND FOR THE ICCF PARAMETER (SEE 2.)
ICCF=
*
* ====> INSERT STATEMENTS HERE (NO COMMENT '*' IN FIRST COLUMN, SEE 3.)
* /*
// IF $RC=0 THEN
// GOTO STEP2
// IF $RC=4 THEN
// GOTO ERROR
// IF $RC>6 THEN
// GOTO END
// LOG
```

```
* ===> JOB 'DTRUPD' CREATED, ENSURE THAT THIS JOB IS EXECUTED NEXT
// NOLOG
// IF $RC=2 THEN
// GOTO STEP2
/. ERROR
// LOG
* ===> ERRORS IN INPUT DATA, STATEMENT(S) FLAGGED IN LISTING
// NOLOG
/. STEP2
*
* ====> DELETE BLOCK 'UPDPL', IF REQUIRED (SEE 4.)
* ************ BEGIN OF BLOCK 'UPDPL' ****************************
// EXEC PROC=IESUPDPL
* /*
* ************ END OF BLOCK 'UPDPL' ******************************
*
/. END
* /&
* $$ EOJ
```

Using Batch Program IESUPDCF to Maintain User Profiles

After making changes in skeleton IESUPDCF, submit the job for processing. Once the job is processed, check the output listing to see whether the job DTRUPD was created. Please note that this job will only be created when you specify ICCF=YES. If so:

- Check the system console, since job DTRUPD prompts you to disconnect the DTSFILE and waits for a response.
- Disconnect the DTSFILE (**/DISC DTSFILE**) and reply to the suspended job.
- Reconnect the DTSFILE after the job has terminated (**/CON DTSFILE**).
- If you have specified ICCF=YES, you will have two listings with the name IESUPDCF. (With ICCF=NO or ICCF=IGNORE you will get one IESUPDCF listing). Check both of them for flagged statements and return codes.
- After users have been added, run the BSTADMIN statements.

Return Codes Issued by IESUPDCF

0 No error. Job DTRUPD was not generated.

2 No error. Job DTRUPD was generated.

User action:

- Ensure that job (DTRUPD) is started immediately.
- Disconnect the DTSFILE when prompted on the system console.

4 The program has detected one or more invalid user statements in the job. The invalid statements are flagged in the listing. All valid statements are processed. Job DTRUPD was not generated.

User action:

- Examine the job listing.
- Correct the flagged job statements.
- Delete statements that are not flagged from the job, because they have been processed before.
- Submit the corrected job again.

6 The program has detected one or more invalid user statements in the job. The invalid statements are flagged in the listing. All valid statements are executed. Job DTRUPD was generated.

User action:
- Submit job DTRUPD.
- Examine the job listing.
- Correct the flagged job statements.
- Delete statements that are not flagged from the job, because they have been processed before.
- Submit the corrected job again.

8 The ICCF statements were ignored. ICCF=NO was specified in the job, but there was at least one statement that tried to alter an ICCF user definition. This statement was ignored.

User action:

If the erroneous statement is to be processed:
- Specify ICCF=YES.
- Delete all statements that are not flagged, because they have been processed before.
- Submit the corrected job again.

16 The program has been canceled due to severe errors.

User action:
- Examine the listing to determine the reason. The error might have been caused by one of the following:

  CDLOAD
  The program was unable to load the DTSFILE I/O routine DTSFILRT.

  CONTROL FILE
  A VSE/VSAM macro caused an error.

  GETVIS
  The partition GETVIS area is too small for the job.

Example of Completed Skeleton IESUPDCF

The following is an example of a completed IESUPDCF skeleton, which shows ADDing, ALTering, and DELeting users.

```
* $$ JOB JNM=IESUPDCF,CLASS=0,DISP=D
* $$ PUN DISP=I,CLASS=0,PRI=9
// JOB IESUPDCF
// OPTION NOLOG
*
*  THIS SKELETON MAY BE USED BY THE ADMINISTRATOR TO GENERATE A
*  JOB FOR BATCH USER PROFILE MAINTENANCE.
*       ...
*       ... Description is deleted.
*       ... (See skeleton IESUPDCF)
*
* ====> UPDATE NEXT LINE IF NECESSARY (SEE 1.)
// DLBL IESCNTL,'VSE.CONTROL.FILE',,VSAM,CAT=VSESPUC
*
// EXEC PROC=DTRICCF
// EXEC IESUPDCF,SIZE=64K
*
* ====> SUPPLY AN OPERAND FOR THE ICCF PARAMETER (SEE 2.)
ICCF=YES
*
* ====> INSERT STATEMENTS HERE (NO COMMENT '*' IN FIRST COLUMN, SEE 3.)
ADD NEWUSR,PASSWD,OLDUSR, -
    DAYS=30,TIMEOUT=15, -
```

```
    CPW=YES,PSL=YES
ALT MYUSER, PWD=NEWPWD, -
    DAYS=30,TIMEOUT=15, -
    CPW=YES,PSL=YES
DEL OLDUSR
/*
// IF $RC=0 THEN
// GOTO STEP2
// IF $RC=4 THEN
// GOTO ERROR
// IF $RC>6 THEN
// GOTO END
// LOG
* ===> JOB 'DTRUPD' CREATED, ENSURE THAT THIS JOB IS EXECUTED NEXT
// NOLOG
// IF $RC=2 THEN
// GOTO STEP2
/. ERROR
// LOG
* ===> ERRORS IN INPUT DATA, STATEMENT(S) FLAGGED IN LISTING
// NOLOG
/. STEP2
*
* ====> DELETE BLOCK 'UPDPL', IF REQUIRED (SEE 4.)
* ************ BEGIN OF BLOCK 'UPDPL' ****************************
// EXEC PROC=IESUPDPL
* /*
* ************ END OF BLOCK 'UPDPL' **********************************
/. END
/&
* $$ EOJ
```

Chapter 27. Maintaining User Profiles in an LDAP Environment

This chapter describes how you maintain your z/VSE user profiles in an **LDAP environment**. LDAP is the abbreviation for "Lightweight Directory Access Protocol".

It contains these main topics:

Note:

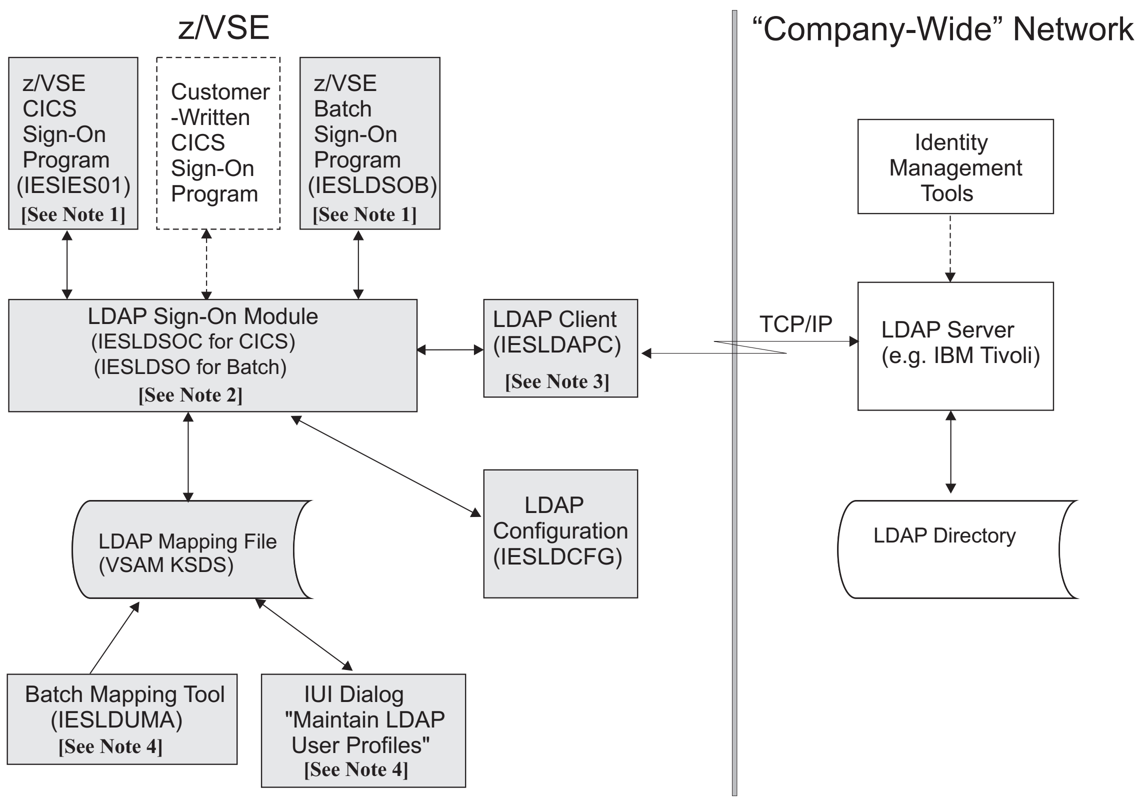
1. The LDAP support described in this chapter can be used together with the *Basic Security Manager* (BSM). If the functionality provided by the BSM does not meet your requirements, you might *instead* be able to use an External Security Manager (ESM) that is supplied by a vendor.
2. You cannot enter an LDAP long-user-ID and long-password in a VSE/POWER Job statement. You can only enter a *z/VSE* short-user-ID and short-password using this VSE/POWER Job statement: `* $$ JOB ... SEC=(userid,password)`.

Related Topics:

For details of how to ...	Refer to ...
define user profiles for use with BSM-based security	Chapter 25, "Maintaining User Profiles via BSM Dialogs," on page 289
define an LDAP user-ID and profile during the process of adding a VSE user-ID	"Adding/Changing a User-ID and Profile Definitions" on page 290
maintain large numbers of user profiles in batch, for use with BSM-based security	Chapter 26, "Maintaining User Profiles via Batch Program IESUPDCF," on page 311

For a *general description* of LDAP support and how it can benefit your company, refer to "Security Support" in the *z/VSE Planning*, SC34-2635.

Overview of LDAP Sign-On Processing



Notes:
1. The CICS or batch sign-on program calls the LDAP sign-on module, providing LDAP-authentication is enabled "system-wide".
2. The LDAP sign-on module authenticates the user via the LDAP Client API. It also looks up the user in the LDAP mapping file, and returns the short-user-ID (z/VSE user-ID) and short-password.
3. The LDAP client implements the LDAP protocol (specified by RFC2251) and connection handling. It also provides a programming interface in accordance with RFC1823.
4. To manage user mappings in the LDAP mapping file you can either use the IUI dialog "Maintain LDAP User Profiles" or the (batch) LDAP Mapping Tool. For example, you can enable/disable LDAP-authentication for specific users.

Figure 86. Overview of LDAP Sign-On Processing

LDAP sign-on is based upon the use of the *LDAP mapping file*. This VSAM KSDS file is used to store the user-ID mappings, and is *automatically* defined if you performed an FSU to z/VSE 5.1. An FSU from one modification level of z/VSE 5.1 to a later level will keep the LDAP mapping file unchanged.

The LDAP mapping file contains:
- Records containing user-IDs that *are* to be used for *LDAP-authentication*, in which a mapping of a long-user-ID (used in the LDAP environment) to a short-user-ID (used in z/VSE) is done. These user-IDs are referred to as being *LDAP-enabled*.
- Records containing user-IDs that are *not* used for *LDAP-authentication* (for example, the SYSA user-ID). These user-IDs are referred to as being *not LDAP-enabled*, and these users can sign on to z/VSE even if the LDAP server is not operational.

The processing flow for an LDAP sign-on via the LDAP sign-on panel is as follows.

1. When a user starts his/her terminal session either:
   a. The z/VSE sign-on program (IESIES01) is started, or
   b. A customer-written sign-on program is started.
2. The z/VSE sign-on program (IESIES01) checks if LDAP-authentication is enabled:
   a. If LDAP-authentication *is* enabled an LDAP sign-on panel (the z/VSE LDAP panel is shown in Figure 87) is displayed, containing the long-user-ID and long-password fields.

```
IESADMSO3                          z/VSE SIGN ON
    5609-ZV5 and Other Materials (C) Copyright IBM Corp. 2011 and other dates



                                 ++
                                ++    VV    VV    SSSSS    EEEEEEE
                               ++     VV    VV   SSSSSSS   EEEEEEE
                 zzzzzz       ++      VV    VV  SS         EE
                 zzzzz       ++       VV    VV   SSSSSS    EEEEEE
                   zz       ++        VV    VV    SSSSSS   EEEEEE
                  zz       ++          VV  VV         SS  EE
                 zzzzzz   ++            VVVV     SSSSSSS   EEEEEEE
                zzzzzzz  ++              VV       SSSSS    EEEEEEE

        Your terminal is      and its name in the network is D3000001
        Today is 10/11/2011   To sign on to DBDCCICS  --  enter your:

USER-ID. ________________________________________________________________
PASSWORD

PF1=HELP       2=TUTORIAL   3=TO VM      4=REMOTE APPLICATIONS     6=ESCAPE(U)
                            9=Escape(m) 10=NEW PASSWORD           12=LOGON HERE
```

Figure 87. LDAP Sign-On Panel

   b. If LDAP-authentication is *not* enabled, a non-LDAP sign-on panel is displayed. **The remainder of this chapter no longer applies.**

   Note: LDAP-authentication is enabled using the skeleton SKLDCFG described in Table 8 on page 330.
3. The z/VSE sign-on program links (via an `EXEC CICS LINK` command) to the LDAP sign-on module *IESLDSOC*. The sign-on program uses the COMMAREA (described in Table 9 on page 344) to pass the long-user-ID and long-password to the LDAP sign-on module.
4. The LDAP sign-on module locates the *LDAP mapping file*. The LDAP sign-on module then searches for the record containing the user-ID mapping. Depending upon whether or not the record is found, one of the following applies:
   - If the record containing the user-ID mapping is *not found* and LDAP is being operated in:
     – *strict mode*, the sign-on attempt is rejected.
     – *non-strict mode* and the user-ID and password are both less than or equal to 8 characters, a mapping of user-IDs does *not* take place. The sign-on attempt is then sent "unchanged" to the security manager (for example, to the Basic Security Manager in z/VSE), as described in Step 5 (below).
     – *non-strict mode* and the user-ID and/or password are greater than 8 characters, the sign-on attempt is rejected.

Note: *strict mode* and *non-strict mode* are described in "Deciding if Strict-User-Mappings Are to be Used" on page 328.

- If the record containing the user-ID mapping *is found* and the user-ID *is not LDAP-enabled*, a mapping of user-IDs does *not* take place. The sign-on attempt is then sent "unchanged" to the security manager (for example, to the Basic Security Manager in z/VSE), as described in Step 5 (below).
- If the record containing the user-ID mapping *is found* and the user-ID *is LDAP-enabled*, an *LDAP-authentication* is performed with the remote LDAP server:
  - If the LDAP-authentication *is successful*, a mapping of user-IDs occurs. The sign-on attempt is sent to the security manager (for example, to the Basic Security Manager in z/VSE) as described in Step 5 (below). The sign-on attempt will use the short-user-ID and short-password obtained from the mapping,
  - If the LDAP-authentication *is not successful*, the sign-on attempt is rejected.

5. Providing the sign-on attempt has not already been rejected, the LDAP sign-on module (IESLDSOC) returns the short-user-ID and short-password (the "z/VSE" user-ID and password) to the z/VSE sign-on program via the COMMAREA.
6. Depending upon whether the sign-on request was accepted or rejected in the previous steps, the z/VSE sign-on program (IESIES01) continues to process the sign-on request.

The processing flow for an LDAP sign-on via a batch job is as follows.

1. The batch job must be modified to include these statements:
   ```
   // EXEC IESLDSOB
   USER=xxx...
   PWD=xxx...
   /*
   ```
2. The z/VSE sign-on program (IESLDSOB) calls the LDAP sign-on module *IESLDSO*.
3. The LDAP sign-on module locates the *LDAP mapping file*. The LDAP sign-on module then searches for the record containing the user-ID mapping. Depending upon whether or not the record is found, one of the following applies:
   - If the record containing the user-ID mapping is *not found* and LDAP is being operated in:
     - *strict mode*, the sign-on attempt is rejected.
     - *non-strict mode* and the user-ID and password are both less than or equal to 8 characters, a mapping of user-IDs does *not* take place. The sign-on attempt is then sent "unchanged" to the security manager (for example, to the Basic Security Manager in z/VSE), as described in Step 5 (below).
     - *non-strict mode* and the user-ID and/or password are greater than 8 characters, the sign-on attempt is rejected.

     Note: *strict mode* and *non-strict mode* are described in "Deciding if Strict-User-Mappings Are to be Used" on page 328.
   - If the record containing the user-ID mapping *is found* and the user-ID *is not LDAP-enabled*, a mapping of user-IDs does *not* take place. The sign-on attempt is then sent "unchanged" to the security manager (for example, to the Basic Security Manager in z/VSE) as described in Step 5 (below).

- If the record containing the user-ID mapping *is found* and the user-ID *is LDAP-enabled*, an *LDAP-authentication* is performed with the remote LDAP server:
  – If the LDAP-authentication *is successful*, a mapping of user-IDs occurs. The sign-on attempt is sent to the security manager (for example, to the Basic Security Manager in z/VSE) as described in Step 4 (below). The sign-on attempt will use the short-user-ID and short-password obtained from the mapping,
  – If the LDAP-authentication *is not successful*, the sign-on attempt is rejected.

4. Providing the sign-on attempt has not already been rejected, the LDAP sign-on module (IESLDSO) returns the short-user-ID and short-password (the "z/VSE" user-ID and password) to the z/VSE sign-on program (IESLDSOB).
5. Depending upon whether the sign-on request was accepted or rejected in the previous steps, the z/VSE sign-on program (IESLDSOB) continues to process the sign-on request using the short "z/VSE" user-ID and password in the same way as when processing an `// ID` statement.

LDAP Sign-On: Prerequisites and Getting Started

To get started with LDAP sign-on, you must have:

- Set up an LDAP environment within your company.
- Set up your network so that z/VSE can connect to an LDAP server.
- Enabled "system-wide" LDAP sign-on processing within z/VSE. For details, see field 'FLAGS' in Table 8 on page 330.
- Decided whether or not to enable *strict-mode* and set the field 'FLAGS' accordingly. For details, see:
  – "Deciding if Strict-User-Mappings Are to be Used" on page 328.
  – Field 'FLAGS' in Table 8 on page 330.
- Decided whether or not to use *password-hashing* and set the field 'CACHE_EXPIRATION' accordingly. For details, see:
  – "Deciding if Password-Caching is to be Used" on page 328.
  – Field 'CACHE_EXPIRATION' in Table 8 on page 330.
- Decided on which *LDAP-authentication method* you wish to use and set the field 'AUTH_METHOD' accordingly. For details, see:
  – "Choosing an LDAP Authentication Method" on page 329.
  – Field 'AUTH_METHOD' in Table 8 on page 330.
- Added an entry in the *LDAP mapping file* for each user who will be using an LDAP sign-on. For details, see "Rules for Using LDAP-Enabled User-IDs" on page 334.
- Tailored the remaining fields (to those described above) that are contained in member SKLDCFG. For details, see "Tailoring the LDAP Configuration Member SKLDCFG" on page 330.

Note:

1. IBM does *not* provide you with an LDAP server that runs under z/VSE! However, you can use any suitable LDAP server that is used within your company (for example, the *IBM Tivoli Directory Server* or the *IBM Tivoli Identity Manager*).
2. The LDAP mapping file IESLDUM (a VSE/VSAM KSDS file) is defined *automatically* if you performed an FSU to z/VSE 5.1. An FSU from one modification level of z/VSE 5.1 to a later level will keep the LDAP mapping file unchanged.

Deciding if Strict-User-Mappings Are to be Used

You can configure the LDAP sign-on support to operate in one of the following modes:

- *Strict mode*, in which *all* users (those using a long-user-ID *and* those using a short-user-ID) are defined in the LDAP mapping file.
- *Non-strict mode*, in which *only* those users using a long-user-ID are defined in the LDAP mapping file. However, if a user signs on using a short-user-ID, the sign-on request will nevertheless be processed *providing* this short-user-ID is recognized by the security manager (for example, by the Basic Security Manager). The advantage of using non-strict mode is that users who should be able to sign on to z/VSE when the LDAP server is not operational (for example, the SYSA user-ID), do not have to be defined in the LDAP mapping file.

For details of how to set *strict mode* or *non-strict mode*, see field 'FLAGS' in Table 8 on page 330.

Deciding if Password-Caching is to be Used

Password-caching is *optional*. Without the use of password-caching, each LDAP sign-on attempt must communicate with the LDAP server. This can add “overhead” to your z/VSE system. This is especially true if you are using SSL for secure communication.

To *disable* password-caching, you must set the field 'CACHE_EXPIRATION' to zero, as described in Table 8 on page 330.

Using password-caching, a *cache* is used to store a hash of the last LDAP password. The LDAP password can then be verified *locally* using the LDAP-password hash stored in the *LDAP mapping file*. This means, to verify the LDAP password there is no need to authenticate with the LDAP server.

This is how password-verification works:

1. A SHA-256 hash is generated from the user's password and is stored in the LDAP mapping file.
2. When the user attempts to perform an LDAP sign-on, z/VSE checks if the password that has been entered is valid or not. To do so, a SHA-256 hash is generated from the password that has been entered by the user.
3. z/VSE compares the generated-hash with the hash stored in the LDAP mapping file.
4. If the hash values are the same, the password is valid. If the hash values are not the same, z/VSE rejects the LDAP sign-on.

Note: There is no way of recovering the password from the hash.

If password-caching *is* enabled, you must also configure the *period of validity* for the password hash-values stored in the LDAP mapping file. This is also done using the field 'CACHE_EXPIRATION', as described in Table 8 on page 330.

- If a user performs an LDAP sign-on *within* the time-limit for the password's validity, an LDAP-authentication with the LDAP server is *not* performed. Instead, the password is verified locally using the stored password hash.
- If a user performs an LDAP sign-on *after* the time-limit for the password's validity, an LDAP-authentication with the LDAP server is performed. If the

LDAP-authentication is successful, z/VSE updates the password hash and marks it as valid for the time-limit you have defined.

Choosing an LDAP Authentication Method

You must choose from one of two possible LDAP-authentication methods described below, and then set the field 'AUTH_METHOD' in Table 8 on page 330 accordingly.

The two LDAP authentication methods described here are well known in the computer industry. For example, Unix- and Linux-based systems provide support for so called “Pluggable Authentication Modules” (PAM). You can use one of the modules (PAM-LDAP) to authenticate with an LDAP server (since PAM-LDAP uses exactly the same processing that is described here).

The method you choose largely depends upon how LDAP has been set up within your installation:

- *Direct BIND with LDAP user-ID and password*. This method uses a pattern to build a LDAP distinguished name (DN) with the user-ID (for example, `cn=%u,dc=ibm,dc=com`, where `%u` is replaced with the user-ID). The distinguished name is then used to perform a BIND operation with the password. This method requires that the user-ID is part of the LDAP distinguished name. To use this method, you must set the field 'AUTH_METHOD' to '1'.
- *Search for distinguished name using attribute*. This method is used when the user-ID is *not* part of the distinguished name. Therefore, an LDAP SEARCH operation is first performed, to search for the associated entry. The search uses a filter such as `uid=%u`, where `uid` is the name of an attribute and `%u` is replaced with the user-ID. The distinguished name of the search result is used to perform the final BIND operation with the given password. To use this method, you must set the field 'AUTH_METHOD' to '2'.

Tailoring the LDAP Configuration Member SKLDCFG

Tailor skeleton SKLDCFG (contained in ICCF library 59) according to the configuration you wish to implement. These are the fields contained in skeleton SKLDCFG:

Table 8. Fields Contained in the LDAP Configuration Member SKLDCFG

Field	Description
FLAGS	The following bits are used: • X'00000001' - LDAP-authentication is enabled (system-wide). When this bit is ON, the LDAP sign-on panel (shown in Figure 87 on page 325) is displayed. • X'00000002' - SSL is enabled. When this bit is ON, secure communication is used. The SSL options must also be configured. • X'00000004' - "Strict" user mapping is to be used: – If this bit is ON, *all users* must be defined in the LDAP mapping file, including users that are not LDAP-enabled (for example, SYSA). If a user tries to sign in that is not defined in the LDAP mapping file, a "User not found" error message will be returned. – If this bit is OFF, z/VSE first attempts to locate the user in the LDAP mapping file. If not found, the user is processed as if he/she were not LDAP-enabled. Therefore, the user-ID and password are returned without any mapping taking place. • X'00000008' - Failure hash support is active. A second password expiration time can be used to control how long a hash is valid if none of the LDAP servers can be reached. • X'00000010' - uppercase mode is enabled. When enabled, the LDAP user ID will be translated to uppercase before it is used to perform the signon. The LDAP user mapping administration tool and the LDAP dialogs also process this flag when creating user mapping records. • X'80000000' - Tracing is enabled.
USER_MAP_FILE_DLBL	DLBL name of the LDAP mapping file (for example, IESLDUM).
EBCDIC_CODEPAGE	EBCDIC code page name (in LE/VSE `iconv` format). The LDAP protocol uses the UTF-8 code page. Therefore, any textual data has to be translated into EBCDIC. This setting is important for special characters such as an '@' that is used in a user-ID or password. An example of an EBCDIC code page is IBM-1047.
CACHE_EXPIRATION	Validity period (in minutes) for caching the LDAP password. To disable caching, set this value to zero.
LDAP_SERVERS	One or multiple IP addresses or hostnames of LDAP servers, separated by blanks. The servers are contacted in order of specification. If the first one is not available, the second one is tried, and so on. The specification can optionally contain the port (*hostname:port*). If no port is specified, the *default* ports are used (that means, for non-SSL: 389, for SSL: 636).
KEYRING_LIBRARY	The keyring library and sublibrary used for SSL keys and certificates. This field is only used when SSL is enabled (see field 'FLAGS').
KEYNAME	Name of the key member used for SSL. This field is only used when SSL is enabled (see field 'FLAGS').
CIPHER_SPEC	The cipher specs used for SSL handshake. The value contains of one or multiple 2 character codes: for example, `010208090A62`. This field is only used when SSL is enabled (see field FLAGS, above).
SESSION_TIMEOUT	The SSL session timeout in seconds. This field is only used when SSL is enabled (see field 'FLAGS').

Table 8. Fields Contained in the LDAP Configuration Member SKLDCFG (continued)

Field	Description
AUTH_METHOD	LDAP-authentication method: **1** **Direct**: The LDAP user-ID is used directly with the BIND operation. The field DN_BIND_PATTERN is used to build the distinguished name for BIND. **2** **Search**: The LDAP user-ID cannot be used directly for BIND. Instead, a SEARCH is first performed using the attribute that is specified in field USER_ATTRIBUTE. The search result's "distinguished name" is then used for a BIND.
DN_BIND_PATTERN	Pattern used for building the "distinguished name" when using the direct authentication method.
BIND_DN	"Distinguished name" used for BIND, when the search authentication method is used. This "distinguished name" is used to BIND before the SEARCH operation is performed. When this field is left blank, an anonymous BIND is performed.
BIND_PWD	Password used for BIND when the search authentication method is used. This password is used to BIND before the SEARCH operation is performed. When this field is left blank, an anonymous BIND is performed.
USER_ATTRIBUTE	Name of the attribute containing the user-ID used with the search authentication method. An LDAP SEARCH operation is performed using an LDAP filter, such as: • (%a=%u) • (&(%f)(%a=%u)) – when field ADD_SEARCH_FILTER is not blank: – "%a" is replaced with the attribute name. – "%u" is replaced with the LDAP user-ID. – "%f" is replaced with the additional search filter specified in field ADD_SEARCH_FILTER.
BASE_DN	The base distinguished name for performing the SEARCH operation. The search results are dependent on the search scope (see field 'SEARCH_SCOPE').
SEARCH_DEREF	This option specifies how references should be handled when performing the search. The following values are used: **0** Dereference never **1** Dereference searching **2** Dereference finding **3** Dereference always
SEARCH_SCOPE	The search scope specifies how the search should be performed. The following values are used: **0** Base **1** One level **2** Sub-tree
ADD_SEARCH_FILTER	Additional search filter that is concatenated to the search filter using "AND". See also field 'USER_ATTRIBUTE'.
SEARCH_TIMEOUT	This option specifies the time limit in seconds for the SEARCH operation. A value of zero means no time limit.
FAILURE_CACHE_EXPI	The "failure cache expiration" option specifies the validity period in minutes for failure cases. This field has a similar function to the CACHE_EXPIRATION field.

Example of an LDAP Configuration Member

This topic provides you with a practical example that is taken from the *IBM Blue Pages* LDAP configuration. The IBM Blue Pages contain records of IBM employees, which can be updated both locally and from a central administrative location, and can only be accessed from within the IBM internal network.

```
* $$ JOB JNM=LDCONFIG,CLASS=A,DISP=D
// JOB LDCONFIG GENERATE LDAP SIGNON CONFIG PHASE
* *******************************************************************
* ASSEMBLE AND LINK THE LDAP CONFIG PHASE                           *
*                                                                   *
* NOTE: THE CONTENTS OF THIS MEMBER IS CASE SENSITIVE !             *
* *******************************************************************
// LIBDEF *,CATALOG=PRD2.CONFIG
// LIBDEF *,SEARCH=PRD1.BASE
// OPTION ERRS,SXREF,SYM,NODECK,CATAL,LISTX
   PHASE IESLDCFG,*,SVA
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
IESLDCFG CSECT
IESLDCFG AMODE ANY
IESLDCFG RMODE ANY
* *******************************************************************
* GENERAL SETTINGS:
* *******************************************************************
*
* FLAGS. POSSIBLE VALUES SEE EQUATES BELOW
*
FLAGS              DC     XL4'00000001'
LDAP_AUTH_ENABLED  EQU    X'00000001'    LDAP AUTH IS ENABLED
USE_SSL            EQU    X'00000002'    SSL IS ENABLED
STRICT_MODE        EQU    X'00000004'    STRICT USER MAPPING
FAILURE_HASH       EQU    X'00000008'    FAILURE PASSWORD HASH USED
UPPERCASE_MODE     EQU    X'00000010'    UPPERCASE LDAP USERID & PWD
TRACE              EQU    X'80000000'    ENABLE TRACING
*
* USER MAPPING FILE NAME (DLBL)
*
USER_MAP_FILE_DLBL DC     CL8'IESLDUM'
*
* EBCDIC CODEPAGE (IMPORTANT FOR @ CHAR). THE LDAP PROTOCOL USES UTF-8
* FOR DATA TRANSFER. THE LDAP CLIENT TRANSLATES THIS TO THE SPECIFIED
* EBCDIC CODEPAGE. (DEFAULT IS IBM-1047).
*
EBCDIC_CODEPAGE    DC     CL16'IBM-1047'
*
* VALIDITY PERIOD IN MINUTES FOR CACHING THE LDAP PASSWORD.
* TO DISABLE CACHING OF LDAP PASSWORDS SET THIS VALUE TO ZERO.
*
CACHE_EXPIRATION   DC     F'0'
*
* *******************************************************************
* LDAP SERVER SETTINGS
* *******************************************************************
*
* LDAP SERVER IP OR HOSTNAME.
*
* SPECIFICATION CONTAINS THE IP ADDRESS OR HOSTNAME OF ONE OR MORE
* LDAP SERVERS IN THE FORM <SERVER>:<PORT>. MULTIPLE SERVER ADDRESSES
* ARE SEPARATED BY BLANKS. IF THE PORT NUMBER IS OMITED, THE DEFAULT
* PORT NUMBERS ARE USED:
*    389  - NON-SSL
*    636  - SSL
* PLEASE NOTE THAT MICROSOFT ACTIVE DIRECTORY MAY USE DIFFERENT PORT
* NUMBERS. THE AD GLOBAL CATALOG SERVER USES PORT 3268 PER DEFAULT.
*
LDAP_SERVERS       DC     CL256'my.ldap.server:389'
*
* *******************************************************************
```

```
* SSL SPECIFIC SETTINGS. IF SPECIFIED, A SSL CONNECTION IS USED TO
* CONNECT TO THE LDAP SERVER.
* ******************************************************************
*
* KEYRING LIBRARY
*
KEYRING_LIBRARY    DC     CL16'CRYPTO.KEYRING'
*
* NAME OF THE KEY MATERIAL IN THE KEYRING LIBRARY
*
KEYNAME            DC     CL8'LDAPKEY'
*
* SSL CIPHER SPECS TO USE
*
CIPHER_SPEC        DC     CL64'010208090A62'
*
* SSL SESSION TIMEOUT IN SECONDS
*
SESSION_TIMEOUT    DC     F'86440'
*
* ******************************************************************
* LDAP AUTHENTICATION SETTINGS
* ******************************************************************
*
* AUTHENTICATION METHOD:
* 1.) DIRECT: THE LDAP USER ID IS USED DIRECTLY AS THE USER NAME
*             PASSED TO BIND. THE DN_BIND_PATTERN BELOW IS USED TO
*             BUILD THE DISTINGUISHED NAME FOR BIND.
* 2.) SEARCH: THE LDAP USER ID IS NOT USED DIRECTLY FOR BIND.
*             INSTEAD A SEARCH IS PERFORMED FOR THE USER ID USING
*             A SPECIFIC ATTRIBUTE. THE DISTINGUISHED NAME OF THE
*             SEARCH RESULT IS USED TO PERFORM THE BIND.
*
AUTH_METHOD        DC     F'2'
AUTH_DIRECT        EQU    1         USE DIRECT BIND WITH USER ID
AUTH_SEARCH        EQU    2         USE SEARCH ON ATTRIBUTE
*
* DISTINGUISHED NAME PATTERN. AN OCCURANCE OF '%u' IS REPLACED
* WITH THE USER NAME.
*
DN_BIND_PATTERN    DC     CL64'cn=%u,dc=mycompany,dc=com'
*
* DISTINGUISHED NAME USED FOR BIND WHEN PERFORMING THE SEARCH.
* LEAVE IT BLANK FOR ANONYMOUS BIND
*
BIND_DN            DC     CL64''
*
* PASSWORD USED FORBIND WHEN PERFORMING THE SEARCH.
* LEAVE IT BLANK FOR ANONYMOUS BIND
*
BIND_PWD           DC     CL64''
*
* USER ID ATTRIBUTE NAME USED WHEN PERFORMING THE SEARCH.
*
USER_ATTRIBUTE     DC     CL64'emailaddress'
*
* BASE DISTINGUISHED NAME USED WHEN PERFORMING THE SEARCH.
*
BASE_DN            DC     CL64'ou=myorgunit,o=mycompany.com'
*
* THE DEREFERENCING OPTION USED WHEN DOING THE SEARCH
*
SEARCH_DEREF       DC     F'0'
DEREF_NEVER        EQU    0         DEREF NEVER
DEREF_SEARCHING    EQU    1         DEREF SEARCHING
DEREF_FINDING      EQU    2         DEREF FINDING
DEREF_ALWAYS       EQU    3         DEREF_ALWAYS
*
* THE SCOPE USED WHEN DOING THE SEARCH
*
SEARCH_SCOPE       DC     F'2'
```

```
SCOPE_BASE         EQU    0   SEARCH THE OBJECT ITSELF
SCOPE_ONELEVEL     EQU    1   SEARCH THE OBJECT'S IMMEDIATE CHILDREN
SCOPE_SUBTREE      EQU    2   SEARCH THE OBJECT AND ALL ITS DESCENDENTS
*
* ADDITIONAL SEARCH FILTER USED WHEN PERFORMING THE SEARCH.
* LEAVE IT BLANK IF NO ADDITIONAL FILTER IS REQUIRED
*
ADD_SEARCH_FILTER  DC     CL128''
*
* THE TIMEOUT OPTION USED WHEN DOING THE SEARCH. A VALUE
* OF 0 MEANS NO TIME LIMIT. THE TIMEOUT IS SPECIFGIED IN SECONDS.
*
SEARCH_TIMEOUT     DC     F'0'
*
*
* VALIDITY PERIOD IN MINUTES FOR CACHING THE LDAP PASSWORD. THIS
* PERIOD IS USED WHEN THE LDAP SERVER CAN NOT BE REACHED.
* TO DISABLE FAILURE CACHING OF LDAP PASSWORDS SET THIS VALUE TO ZERO.
* TO ENABLE FAILURE CACHEING; YOU ALSO NEED TO SET THE BIT FAILURE_HASH
* IN THE FLAGS FIELD.
*
FAILURE_CACHE_EXPI DC     F'0'
*
* ********************************************************************
* END OF SETTINGS
* ********************************************************************
         END
/*
// IF $MRC GT 4 THEN
// GOTO NOLINK
// EXEC LNKEDT,PARM='MSHP'
* ********************************************************************
* DONT FORGET TO NEWCOPY IESLDCFG IN ORDER TO ACTIVATE IT:           *
*      CEMT SET PROG(IESLDCFG) NEWCOPY                               *
* ********************************************************************
/. NOLINK
/*
/&
* $$ EOJ
```

Rules for Using LDAP-Enabled User-IDs

User-IDs that *are* to be used for LDAP-authentication are referred to as being *LDAP-enabled*.

User-IDs that are *not* used for LDAP-authentication are referred to as being *not LDAP-enabled*. These users can sign on to z/VSE even if the LDAP server is not operational.

- After a user-ID has been LDAP-enabled, this user should *not* be able to perform an LDAP sign-on using his/her short-user-ID (used in z/VSE). Doing so would bypass the company's security policies that are enforced by the LDAP-authentication.
- When enabling a user-ID for LDAP authentication, a new z/VSE password is randomly generated, and a password-change request is issued (using RACROUTE).
  - The randomly-generated password is stored (encrypted) in the *LDAP mapping file*.
  - The user will never know the randomly-generated password. Therefore, he/she will not be able to perform an LDAP sign-on using the short-user-ID (used in z/VSE).
- You should set short-user-IDs (used in z/VSE) that *are* LDAP-enabled to *non-expiring* (as described in “Password Expiration” on page 143 and “Entering

z/VSE User Profile Information” on page 291). For such short-user-IDs, password expiration should be enforced by the *LDAP server* based on the long-user-ID and long-password.

You should add these users as *not* LDAP-enabled:
- Administrators (such as SYSA)
- Operators
- Security administrators

To do so, use the `TYPE=VSE` setting (described in “ADD Command” on page 340 and “CHANGE Command” on page 341).

Choosing a Method for Maintaining LDAP User Mappings

To maintain LDAP user mappings in the LDAP mapping file, these tasks are available:
- add LDAP user mappings,
- change LDAP user mappings,
- delete LDAP user mappings,
- list LDAP user mappings,
- export all definitions for migrating LDAP user mappings to a new system.

To perform the above tasks, you can use either:
- Interactive User Interface dialogs (described in “Using Dialogs to Maintain LDAP User Mappings”).
- the LDAP mapping tool (described in “Using the LDAP Mapping Tool to Maintain LDAP User Mappings” on page 339).

You are strongly recommended to use dialogs and not the (batch) LDAP mapping tool, since dialogs are not only easier-to-use but also reduce the possibility of error.

Using Dialogs to Maintain LDAP User Mappings

The *Maintain LDAP User Profiles* dialog allows you to browse and modify the entries in the LDAP mapping file. Using this dialog, you can perform *all* of the tasks that can be performed using the LDAP mapping tool. However, the *Maintain LDAP User Profiles* dialog is easier to use than the LDAP mapping tool.

Note: The dialogs described here are *automatically* linked-to during the procedure to add, change, or delete a user profile. For details, see “Adding/Changing a User-ID and Profile Definitions” on page 290.

To invoke the *Maintain LDAP User Profiles* dialog, the *administrator* uses Fast Path **217**. The panel shown in Figure 88 on page 336 is then displayed.

```
IESADMLUPM         MAINTAIN LDAP USER PROFILES

START.... ________________________________________________________________
VSE USERID.... ________
OPTIONS:   1 = ADD       2 = CHANGE     3 = DISPLAY    5 = DELETE
                                                                      USER
OPT  LDAP USERID                                                      TYPE
 _   LDAP_User_2                                                      LDAP
 _   LDAP_User_3                                                      LDAP
 _   LDAP_User_4                                                      LDAP
 _   LDAP_User_5                                                      LDAP
 _   LDAP_User_6                                                      LDAP
 _   LDAP_User_7                                                      LDAP
 _   LDAP_User_8                                                      LDAP
 _   LDAP_User_9                                                      LDAP
 2   LINE__for__LDAP__User__Name__Case__Sensitive__and__64_chars_long LDAP
 _   VSEUSER1                                                         VSE
 _   VSEUSER2                                                         VSE
 _   VSEUSER3                                                         VSE

PF1=HELP                      3=END
                              9=PRINT        10=EXPORT
```

Figure 88. Panel Used for Maintaining User Profiles in the LDAP Mapping File

1. You can now optionally enter:
   - A value in the **START** field to display all LDAP user-IDs whose values are greater than or equal to the START value.
   - A value in the **VSE USERID** field to only display LDAP user-IDs that are linked to this specific VSE user-ID.
2. After pressing ENTER, Figure 88 then lists the:
   - *LDAP user-IDs* that you selected, which can be up to 64 characters long and case-sensitive.
   - *User type* of each LDAP user-ID, which can be either:
     – LDAP (for LDAP-enabled users).
     – VSE (for LDAP-disabled users).

If you enter **1 = ADD** or **2 = CHANGE** against an LDAP user-ID in Figure 88, the *Add or Change LDAP User Profile* panel is displayed.

```
IESADMLUPA         ADD OR CHANGE LDAP USER PROFILE

LDAP USERID.. ________________________________________________________________

DESCRIPTION.. ________________________________________________________________

VSE USERID......... ________  Assigned VSE user-id. 1-8 characters

VSE PASSWORD....... ________  Specifies VSE password. 3-8 characters

GENERATE PASSWORD.. _         1 - Forces generation of random VSE password
                              2 - Use current password
PASSWORD PATTERN... ________  Specifies a pattern for password generation
                              Required if password is generated
                              d - decimal digit (0-9)
                              c - character (A-Z)
                              a - decimal digit (0-9) or character (A-Z)
                              x - special character (@, # or $)
                              other - place is filled with specified character
                              blank - place is not filled with a character.

PF1=HELP                    3=END                        5=PROCESS
```

Figure 89. Panel Used for Adding/Updating an LDAP User Profile

Where:

LDAP USERID
The LDAP user-ID to be added or changed. This can be up to 64 characters. This parameter is case-sensitive.

DESCRIPTION
A free-text description. This can be up to 64 characters. This parameter is optional and case-sensitive.

VSE USERID
The VSE user-ID that is assigned to that user. It can contain between 1 and 8 alphanumeric or special characters, where the special characters can be @, #, or $. This parameter is required when TYPE=LDAP. This parameter is automatically translated to upper case.

VSE PASSWORD
The VSE password for the VSE user-ID (between 3 and 8 characters).

For *Add LDAP User Profile*: You can either:

- *Leave this field empty* - if you specified a '**1**' in the GENERATE PASSWORD field. A new VSE password is generated using the parameters defined in the PASSWORD PATTERN field. This VSE password is encrypted and stored in the LDAP mapping file.
- *Enter a VSE password* - if you specified a '**2**' in the GENERATE PASSWORD field. The entered VSE password is automatically translated to upper case. The VSE password is then encrypted and stored in the LDAP mapping file.

For *Change LDAP User Profile*: You can either:

- *Leave this field empty*:
  - If you specified a '**1**' in the GENERATE PASSWORD field, a new VSE password is generated using the parameters defined in the PASSWORD PATTERN field. This VSE password is encrypted and stored in the LDAP mapping file.
  - If you specified a '**2**' in the GENERATE PASSWORD field, the existing (previously saved) VSE password is read from the LDAP mapping file and is used for the operation.
- *Enter your existing or a new VSE password*. The password is automatically translated to upper case. The VSE password is then encrypted and stored in the LDAP mapping file.

GENERATE PASSWORD
Enter either:

- **1** which instructs the LDAP service program to generate a *random* VSE password (GENPWD). The generated password is stored in the LDAP mapping file. For details of how to specify a pattern for password generation, see parameter PASSWORD PATTERN. **Note:** If you generate a random VSE password, any previous VSE password (that you chose yourself) *will be overwritten!*
- **2** which means the VSE password (that you entered in the VSE PASSWORD field) will be used for logon.

PASSWORD PATTERN
A pattern of between 3 and 8 characters that is used when generating a password. The following characters can be used:

- d – denotes that this place is to be filled with a decimal digit (0-9).
- c – denotes that this place is to be filled with a character (AZ).

- a – denotes that this place is to be filled with either a decimal digit (0-9) or with a character (A-Z).
- x – denotes that this place is to be filled with special character @, #, or $.
- other – denotes that this place is to be filled with another specified character.
- blank – denotes that this place is not filled with a character.

After making your selections and entering values in the displayed fields, press **PF5** to proceed with your request. The message “User Profile Was Added Successfully” is displayed.

If you enter **3 = DISPLAY** against an LDAP user-ID in Figure 88 on page 336, the *Display LDAP User Profile* panel is displayed. It shows the information about an LDAP user-ID that is stored in the LDAP mapping file.

```
IESADMLUPD            DISPLAY LDAP USER PROFILE


LDAP USERID.. LDAP_USER_N1

DESCRIPTION.. This user is system admin

LDAP ENABLED....... 1         1 - LDAP authentication is enabled; 0 - disabled

VSE USERID......... ADN1      Assigned VSE userid

HASH EXPIRATION.... 20090101  Date when password hash expires (yyyymmdd)
                    150000    Time when password hash expires (hhmmss)
LAST LDAP SIGNON... 20081001  Date of last sign-on using online LDAP(yyyymmdd)
                    112043    Time of last sign-on using online LDAP(hhmmss)
LAST HASH SIGNON... 20081002  Date of last sign-on using cached LDAP(yyyymmdd)
                    101533    Time of last sign-on using cached LDAP(hhmmss)
FAILURE HASH EXPIR. 20090101  Date when hash expires in failure case(yyyymmdd)
                    150000    Time when hash expires in failure case(hhmmss)


PF1=HELP                   3=END
```

Figure 90. Panel Used for Displaying Details of an LDAP User Profile

If you enter **5 = DELETE** against an LDAP user-ID in Figure 88 on page 336, z/VSE will request that you confirm the deletion of this user-ID *providing* the `CONFIRM DELETE` option is set to YES in the profile for this user-ID (using Fast Path **211**, *Maintain User Profiles*). Otherwise, the LDAP user-ID will be deleted without a warning message.

If you press **PF9=PRINT** in Figure 88 on page 336, z/VSE will send a listing to the VSE/POWER list queue. It has the same name as the CICS startup job (default CICSICCF) and contains details of all LDAP users.

If you press **PF10=EXPORT** in Figure 88 on page 336, z/VSE will create a job with the same name as the CICS startup job (default CICSICCF) in the VSE/POWER punch queue. Job CICSICCF can be used for *migrating* LDAP users.

Using the LDAP Mapping Tool to Maintain LDAP User Mappings

The LDAP mapping tool is a batch tool that is invoked via an `// EXEC IESLDUMA` statement. It reads control statements from SYSIPT. Since this tool provides security-sensitive services, it can only be used by *administrator* type users.

- If batch security is active in z/VSE (via the `SYS SEC=YES` statement), you must specify an ID statement in the job running IESLDUMA.
- If batch security is *not* active in z/VSE, you must specify the user-ID and password of an *administrator* using the ID command (as the first command when executing IESLDUMA).

Syntax of the LDAP Mapping Tool:

```
// ID USER=xxx,PWD=xxx
// EXEC IESLDUMA,PARM='parameters'
control statements
⋮
/*
```

If a command does not fit on one line, it can be continued on the next line(s). To indicate a continuation, the line must end with a '-' character (minus). The continuation character can be in any position.

These are the ***parameters*** you can specify with the LDAP mapping tool:

Parameter
Description

SYSLST=DD:SYS*nnn*
Specifies an alternative destination for messages printed on SYSLST. This parameter is optional.

DEBUG
If specified, DEBUG is turned ON. You should only set DEBUG to be ON if problems exist.

These are the ***control statements*** you can specify with the LDAP mapping tool. **Note:** Some parameters of these control statements are *case sensitive!* Make sure you switch to *mixed-case mode* in your editor (if you are using VSE/ICCF, enter `CASE M` in the VSE/ICCF command line).

ID Command

The **ID** command must be the first command in a sequence of commands, unless you are using the ID statement in the job. It authorizes an administrator to perform user mapping modifications:

```
►►──ID──USER='user-ID'──PWD='password'──────────────────────────►◄
```

Where:

USER='*user-ID***'**
Specifies an administrator user-ID. This parameter is required when batch security is not active. Parameter PWD has to be specified in conjunction with USER.

PWD='*password*'
Specifies the administrator's password. This parameter is required when batch security is not active. Parameter USER has to be specified in conjunction with PWD.

ADD Command

The **ADD** command adds a new user mapping to the LDAP mapping file:

```
►►──ADD──LDAPUSER='xxx'──TYPE=VSE──┬────────────┬──────────────────────►◄
                                   └─DESC='xxx'─┘

►►──ADD──LDAPUSER='xxx'──TYPE=LDAP──┬────────────┬──VSEUSER='xxx'──────►
                                    └─DESC='xxx'─┘

►──┬─VSEPWD='xxx'────────────────────────────────┬─────────────────────►◄
   └─GENPWD─OLDPWD='xxx'──┬────────────────────┬─┘
                          └─PWDPATTERN='xxx'───┘
```

Where:

LDAPUSER='*xxx*'
Specifies the LDAP user-ID to be added. This can be up to 64 characters. This parameter is case sensitive.

TYPE=VSE | LDAP
Specifies the type of the user:
- **VSE**: User is *not* LDAP-enabled.
- **LDAP**: User is LDAP-enabled

DESC='*xxx*'
Specifies a free-text description. This can be up to 64 characters. This parameter is optional and case-sensitive.

VSEUSER='*xxx*'
Specifies the VSE user-ID of up to 8 characters that is assigned to that user. This parameter is required when TYPE=LDAP.

VSEPWD='*xxx*'
Specifies the VSE password of up to 8 characters. Either VSEPWD or GENPWD can be specified. If VSEPWD is specified, the user's password is not changed.

GENPWD
Forces the VSE password to be generated by random. Either VSEPWD or GENPWD can be specified. If GENPWD is specified, the VSE user's password is changed via RACROUTE call. See parameter PWDPATTERN for details of specifying a pattern for password generation.

PWDPATTERN='*xxx*'
Specifies a pattern that is used when generating a password. The following characters can be used:
- d – denotes that this place is to be filled with a decimal digit (0-9).
- c – denotes that this place is to be filled with a character (AZ).
- a – denotes that this place is to be filled with either a decimal digit (0-9) or with a character (A-Z).
- x – denotes that this place is to be filled with special character @, #, or $.

- other – denotes that this place is to be filled with another specified character.
- blank – denotes that this place is not filled with a character.

OLDPWD='*xxx***'**
Specifies the current password of the VSE user-ID. This parameter is required when GENPWD is specified.

CHANGE Command

The **CHANGE** command changes a user in the LDAP mapping file:

```
►►──CHANGE──LDAPUSER='xxx'──TYPE=VSE──┬──────────────┬────────────────────────►◄
                                      └─DESC='xxx'───┘


►►──CHANGE──LDAPUSER='xxx'──TYPE=LDAP──┬──────────────┬──┬─────────────────┬───►
                                       └─DESC='xxx'───┘  └─VSEUSER='xxx'───┘

►──┬────────────────────────────────────────────────┬──────────────────────────►◄
   ├─VSEPWD='xxx'───────────────────────────────────┤
   └─GENPWD──┬───────────────┬──┬──────────────────┬┘
             └─OLDPWD='xxx'──┘  └─PWDPATTERN='xxx'─┘
```

Where:

LDAPUSER='*xxx***'**
Specifies the LDAP user-ID to be changed. This can be up to 64 characters. This parameter is case sensitive.

TYPE=VSE | LDAP
Specifies the type of the user:
- **VSE**: User is not LDAP-enabled.
- **LDAP**: User is LDAP-enabled

If omitted, the type is not changed.

DESC='*xxx***'**
Specifies a free-text description. This can be up to 64 characters. This parameter is optional and case-sensitive. If omitted, the description is not changed.

VSEUSER='*xxx***'**
Specifies the VSE user-ID of up to 8 characters that is assigned to that user. If omitted, the VSE user-ID assignment is not changed.

VSEPWD='*xxx***'**
Specifies the VSE password of up to 8 characters. Either VSEPWD or GENPWD can be specified.
- If VSEPWD is specified, the user's password is not changed.
- If both VSEPWD and GENPWD are omitted, the user's password is not changed.

GENPWD
Forces the VSE password to be generated by random. Either VSEPWD or GENPWD can be specified, or this parameter can be omitted.
- If GENPWD is specified, the VSE user's password is changed via RACROUTE call. See parameter PWDPATTERN for details of specifying a pattern for password generation.

- If both VSEPWD and GENPWD are omitted, the user's password is not changed.

OLDPWD='*xxx*'
Specifies the current VSE password for the VSE user-ID.
- OLDWPD can be specified when GENPWD is specified.
- If OLDWPD is not specified, the previous VSE password is read from the user's entry in the LDAP mapping file.

PWDPATTERN='*xxx*'
Specifies a pattern that is used when generating a password. The following characters can be used:
- d – denotes that this place is to be filled with a decimal digit (0-9).
- c – denotes that this place is to be filled with a character (AZ).
- a – denotes that this place is to be filled with either a decimal digit (0-9) or with a character (A-Z).
- x – denotes that this place is to be filled with special character @, #, or $.
- other – denotes that this place is to be filled with another specified character.
- blank – denotes that this place is not filled with a character.

DELETE Command

The **DELETE** command deletes a user mapping from the LDAP mapping file:

```
►►──DELETE──LDAPUSER='xxx'──────────────────────────────────────────►◄
```

Where:

LDAPUSER='*xxx*'
Specifies the LDAP user-ID to be deleted. This can be up to 64 characters. This parameter is case-sensitive.

LIST Command

The **LIST** command lists all (the default) or specific users contained in the LDAP mapping file:

```
►►──LIST──┬────────────────┬──┬─────────────────┬──┬────────────────┬──────►◄
          └─LDAPUSER='xxx'─┘  └─TYPE=──┬─VSE──┬─┘  └─VSEUSER='xxx'──┘
                                       └─LDAP─┘
```

Where:

LDAPUSER='*xxx*'
Specifies the LDAP user-IDs to be listed. They can be up to 64 characters. This parameter is case sensitive. You can use wildcards (* and ?). If this parameter is omitted, all LDAP users are listed.

TYPE=VSE | LDAP
Specifies the type of the users to be listed:
- **VSE**: List only non-LDAP-enabled users.
- **LDAP**: List only LDAP-enabled users.

VSEUSER='*xxx*'
Specifies the VSE user-ID of up to 8 characters that is assigned to that user. If the parameter is omitted, all VSE users are listed.

EXPORT Command

The **EXPORT** command exports all or certain users from the LDAP mapping file. The users are exported as control statements so that the output can be directly used for migration. **Note:** The output contains the VSE passwords in *readable format*.

```
►►──EXPORT──┬──────────────────┬──┬───────────────────┬──┬──────────────────┬──────────►
            └─LDAPUSER='xxx'───┘  └─TYPE=──┬─VSE──┬───┘  └─VSEUSER='xxx'────┘
                                           └─LDAP─┘

►──┬──────────────────┬──┬─────────────────────────────────┬──────────────────────────►◄
   └─OUTPUT='DD:nnn'──┘  └─GENPWD──┬────────────────────┬───┘
                                   └─PWDPATTERN='xxx'───┘
```

Where:

LDAPUSER='*xxx*'
: Specifies the LDAP user-IDs to be exported. They can be up to 64 characters. This parameter is case sensitive. You can use wildcards (* and ?). If this parameter is omitted, all users are exported.

TYPE=VSE | LDAP
: Specifies the type of the users to be exported:
- **VSE**: Export only non-LDAP-enabled users.
- **LDAP**: Export only LDAP-enabled users.

VSEUSER='*xxx*'
: Specifies the VSE user-ID of up to 8 characters that is assigned to that user. If the parameter is omitted, all users are exported.

OUTPUT='DD:*nnn*'
: Specifies the output device or file. If this parameter is omitted, the users are exported to SYSPUN.
- DD:SYS*nnn* – denotes a logical unit
- DD:*lib*.*slib*(*member*.*type*) – denotes a library member

GENPWD
: If specified, the export function will generate control statements that use the GENPWD option. Since the exported data contains passwords in clear-text format, these passwords can be treated as compromised. With GENPWD, the passwords will be changed when importing the exported data and the passwords are no longer known.

PWDPATTERN='*xxx*'
: Specifies the pattern that is used in the exported control statements. Only valid if the GENPWD option is specified. If not specified, no PWDPATTERN parameter will be used in the exported data.

Example of How to Specify Control Statements

Here is an example of how to specify control statements using the LDAP mapping tool.

```
// JOB RUNLDUMA
// EXEC IESLDUMA
ID USER='SYSA' PWD='password'
ADD LDAPUSER='hugo@de.ibm.com' VSEUSER='HUGO' -
TYPE=LDAP GENPWD PWDPATTERN='aaaaaaaa' OLDPWD='INITPWD' -
DESC='Hugo Maier'
ADD LDAPUSER='HUGO' TYPE=VSE DESC='Hugo Schmidt'
DELETE LDAPUSER='HUGO'
CHANGE LDAPUSER='hugo@de.ibm.com' VSEUSER='FRAN' -
TYPE=LDAP -
DESC='Hugo Maier'
LIST
EXPORT
/*
/&
```

Using Your Own LDAP Sign-On Program

If required, you can create your own LDAP sign-on program. Your sign-on program's *callable interface* must use an `EXEC CICS LINK` command which includes the CICS Communication Area (COMMAREA) used by the IBM-supplied sign-on module IESLDSOC:

- The *input to* sign-on module IESLDSOC is the:
  - long LDAP user-ID (up to 64 characters), and
  - long LDAP password (up to 64 characters).
- The *output from* sign-on module IESLDSOC is the:
  - short z/VSE user-ID (between 4 and 8 characters), and
  - short z/VSE password (up to 8 characters).

To proceed with sign-on processing (for example, using an `EXEC CICS SIGNON` command), your own LDAP sign-on program uses the information *returned from* sign-on module IESLDSOC . Table 9 describes the COMMAREA layout (length 152 bytes) that is used when calling IESLDSOC:

Table 9. COMMAREA Used When Calling Sign-On Module IESLDSOC

Field	Length	Description
LDAPUserID	64 chars	(Input) The LDAP user-ID
LDAPPassword	64 chars	(Input) The LDAP password
ReturnCode	4 bytes	(Output) Return code of sign-on processing
FeedbackCode	4 bytes	(Output) Applicable for certain return codes (for example, LDAP error codes)
VSEUserID	8 chars	(Output) The assigned z/VSE user-ID. Only set if sign-on was successful.
VSEPassword	8 chars	(Output) The corresponding z/VSE password. Only set if sign-on was successful.

Return/Feedback Codes Generated During LDAP Sign-On

If LDAP-authentication fails, appropriate *error messages* will be displayed on the Interactive User Interface sign-on screen.

If a *severe error* occurs, the following message is issued to SYSLST by the LDAP sign-on program:

```
IESC3001E SEVERE ERROR WHILE PERFORMING LDAP AUTHENTICATION
          RC='nn' FDBK='nn' OPERATION='nnnn'
```

For details, refer to *z/VSE Messages and Codes, Volume 2*, SC34-2633.

These are the *return codes* that might be generated during LDAP sign-on processing:

Return Code
Explanation

0 OK, no error.
1 Fetch/CDLOAD of IESLDSO.PHASE has failed.
2 Invalid parameter (for example, NULL pointer, or area-too-short).
3 A buffer is too small. An internal operation resulted in more data than buffer space is available.
4 Configuration error. Either the configuration IESLDCFG.PHASE is not available or invalid.
5 Error during SHA-256 processing.
6 I/O error when accessing the LDAP mapping file.
7 Invalid authentication method.
8 SSL initialization has failed.
9 An LDAP error has occurred. See feedback code for LDAP return code.
10 The connection to an LDAP server has failed.
11 An LDAP BIND operation has failed. See feedback code for LDAP return code.
12 The user has not been found in the LDAP mapping file or has not been found in the LDAP directory.
13 The authorization has failed for the specified user and password.
14 The specified user-ID is not LDAP-enabled (for example, SYSA).

These are the *feedback codes* that might be generated during LDAP sign-on processing:

Feedback Code
Explanation

X'00' (decimal 0)
LDAP_SUCCESS

X'01' (decimal 1)
LDAP_OPERATIONS_ERROR

X'02' (decimal 2)
LDAP_PROTOCOL_ERROR

X'03' (decimal 3)
LDAP_TIMELIMIT_EXCEEDED

X'04' (decimal 4)
LDAP_SIZELIMIT_EXCEEDED

X'05' (decimal 5)
LDAP_COMPARE_FALSE

X'06' (decimal 6)
LDAP_COMPARE_TRUE

X'07' (decimal 7)
LDAP_AUTH_METHOD_NOT_SUPPORTED
X'08' (decimal 8)
LDAP_STRONG_AUTH_REQUIRED
X'09' (decimal 9)
LDAP_PARTIAL_RESULTS
X'0a' (decimal 10)
LDAP_REFERRAL
X'0b' (decimal 11)
LDAP_ADMINLIMIT_EXCEEDED
X'0c' (decimal 12)
LDAP_UNAVAILABLE_CRITICAL_EXTENSION
X'0d' (decimal 13)
LDAP_CONFIDENTIALITY_REQUIRED
X'0e' (decimal 14)
LDAP_SASL_BIND_IN_PROGRESS
X'10' (decimal 16)
LDAP_NO_SUCH_ATTRIBUTE
X'11' (decimal 17)
LDAP_UNDEFINED_TYPE
X'12' (decimal 18)
LDAP_INAPPROPRIATE_MATCHING
X'13' (decimal 19)
LDAP_CONSTRAINT_VIOLATION
X'14' (decimal 20)
LDAP_TYPE_OR_VALUE_EXISTS
X'15' (decimal 21)
LDAP_INVALID_SYNTAX
X'20' (decimal 32)
LDAP_NO_SUCH_OBJECT
X'21' (decimal 33)
LDAP_ALIAS_PROBLEM
X'22' (decimal 34)
LDAP_INVALID_DN_SYNTAX
X'23' (decimal 35)
LDAP_IS_LEAF
X'24' (decimal 36)
LDAP_ALIAS_DEREF_PROBLEM
X'30' (decimal 48)
LDAP_INAPPROPRIATE_AUTH
X'31' (decimal 49)
LDAP_INVALID_CREDENTIALS
X'32' (decimal 50)
LDAP_INSUFFICIENT_ACCESS
X'33' (decimal 51)
LDAP_BUSY
X'34' (decimal 52)
LDAP_UNAVAILABLE
X'35' (decimal 53)
LDAP_UNWILLING_TO_PERFORM
X'36' (decimal 54)
LDAP_LOOP_DETECT
X'3c' (decimal 60)
LDAP_SORT_CONTROL_MISSING
X'3d' (decimal 61)
LDAP_INDEX_RANGE_ERROR

X'40' (decimal 64)
LDAP_NAMING_VIOLATION
X'41' (decimal 65)
LDAP_OBJECT_CLASS_VIOLATION
X'42' (decimal 66)
LDAP_NOT_ALLOWED_ON_NONLEAF
X'43' (decimal 67)
LDAP_NOT_ALLOWED_ON_RDN
X'44' (decimal 68)
LDAP_ALREADY_EXISTS
X'45' (decimal 69)
LDAP_NO_OBJECT_CLASS_MODS
X'46' (decimal 70)
LDAP_RESULTS_TOO_LARGE
X'47' (decimal 71)
LDAP_AFFECTS_MULTIPLE_DSAS
X'50' (decimal 80)
LDAP_OTHER
X'51' (decimal 81)
LDAP_SERVER_DOWN
X'52' (decimal 82)
LDAP_LOCAL_ERROR
X'53' (decimal 83)
LDAP_ENCODING_ERROR
X'54' (decimal 84)
LDAP_DECODING_ERROR
X'55' (decimal 85)
LDAP_TIMEOUT
X'56' (decimal 86)
LDAP_AUTH_UNKNOWN
X'57' (decimal 87)
LDAP_FILTER_ERROR
X'58' (decimal 88)
LDAP_USER_CANCELLED
X'59' (decimal 89)
LDAP_PARAM_ERROR
X'5a' (decimal 90)
LDAP_NO_MEMORY
X'5b' (decimal 91)
LDAP_CONNECT_ERROR
X'5c' (decimal 92)
LDAP_NOT_SUPPORTED
X'5d' (decimal 93)
LDAP_CONTROL_NOT_FOUND
X'5e' (decimal 94)
LDAP_NO_RESULTS_RETURNED
X'5f' (decimal 95)
LDAP_MORE_RESULTS_TO_RETURN
X'60' (decimal 96)
LDAP_CLIENT_LOOP
X'61' (decimal 97)
LDAP_REFERRAL_LIMIT_EXCEEDED
X'70' (decimal 112)
LDAP_SSL_ALREADY_INITIALIZED
X'71' (decimal 113)
LDAP_SSL_INITIALIZE_FAILED

X'72' (decimal 114)
LDAP_SSL_INITIALIZE_NOT_CALLED

X'73' (decimal 115)
LDAP_SSL_PARAM_ERROR

X'74' (decimal 116)
LDAP_SSL_HANDSHAKE_FAILED

X'75' (decimal 117)
LDAP_SSL_GET_CIPHER_FAILED

X'76' (decimal 118)
LDAP_SSL_NOT_AVAILABLE

X'77' (decimal 119)
LDAP_SSL_KEYRING_NOT_FOUND

X'78' (decimal 120)
LDAP_SSL_PASSWORD_NOT_SPECIFIED

Chapter 28. Resources Classes Stored in the BSM Control File

This chapter provides "reference-type of information" describing the resource classes used to access resources that are *stored in the BSM control file* (VSE.BSTCNTL.FILE). The syntax of resource names is based on the RACF® class descriptor table (CDT), but with a few changes. The maximum length for resource names supported by BSM is 246 bytes.

This chapter contains these main topics:

Related Topic:

For details of how ...	Refer to the ...
the BSM resource classes are used with the CICS TS (including practical examples)	*CICS Transaction Server for VSE/ESA, Security Guide*, SC33-1942-03 (or later).

Syntax Rules For Resources Defined in the BSM Control File

During a resource check the BSM takes the resource name and looks for the matching profile name. These names are case sensitive.

Due to the various rules for the profile names in the different resource classes, there are **3 general rules**:

1. Profile names, which contain only alphanumeric characters or the characters # (X'7B'), $ (X'5B'), and @ (X'7C') in their name-parts, can be entered as *lower-case characters without single quotes*. They will be converted to upper case characters.
2. If a profile name contains special characters, the profile name *must be enclosed in single quotes*.
3. If the profile name is entered in single quotes, by default *a conversion to upper case will not take place*. You are responsible for using upper and lower case letters as required. You will find details of any exceptions in the resource class description.

Resource Class ACICSPCT

Description
Used by the BSM to process CICS-started transactions and EXEC CICS commands. The related CICS parameter is XPCT.

Format
<prefix.>transaction

Max. length from CDT
13

These are the syntax rules for using the ACICSPCT resource class:

prefix
The user-ID which was used in the CICS start job:

Length
Between 4 and 8 characters, as is required by the *Maintain User Profiles* dialog.

Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). Like user-IDs, prefixes must be stored as uppercase characters.

transaction
The CICS transaction name:

Length
Between 1 and 4 characters.

Acceptable Characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < and >. For further details, refer to the manual *CICS Transaction Server for VSE/ESA, Resource Definition Guide*.

Resource Class APPL

Description
Used by the BSM to control the access to applications.

Format
application

Max. length from CDT
8

These are the syntax rules for using the APPL resource class:

application
The name of the protected application:

Length
Between 1 and 8 characters.

Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). Lowercase characters are converted to uppercase. You might also refer to the manual *CICS Transaction Server for VSE/ESA, Resource Definition Guide*, SC33-1653, for relevant information about the CONNECTION resource.

Resource Class DCICSDCT

Description
Used by the BSM to control access to CICS transient data queues. The related CICS parameter is XDCT.

Format
<prefix.>transdatqueue

Max. length from CDT
13

These are the syntax rules for using the DCICSDCT resource class:

prefix The user-ID which was used in the CICS start job:

Length
Between 4 and 8 characters, as is required by the *Maintain User Profiles* dialog.

Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). Like user-IDs, prefixes must be stored as uppercase characters.

transdatqueue
The CICS transient data queue name:

Length
Between 1 and 4 characters.

Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). You cannot use special characters, lowercase, or mixed case characters as is defined for the `DESTID=` parameter in CICS. For further details, refer to the description of DFHDCT in the manual *CICS Transaction Server for VSE/ESA, Resource Definition Guide*, SC33-1653-05 or later.

Resource Class FACILITY

Description
Used by the BSM for various purposes.

Format
facility1<.facility2>...<.facilityn>

Max. length from CDT
39

These are the syntax rules for using the FACILITY resource class:

facility
The name of the protected resource:

Length
Between 1 and 39 characters.

Acceptable Characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < and >. The character '.' is used as the separator of the facility name parts.

Resource Class FCICSFCT

Description
Used by the BSM to protect files managed by CICS file control. The related CICS parameter is XFCT.

Format
<prefix.>filename

Max. length from VSE
16

These are the syntax rules for using the FCICSFCT resource class:

prefix The user-ID which was used in the CICS start job:

Length
Between 4 and 8 characters, as is required by the *Maintain User Profiles* dialog.

Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). Like user-IDs, prefixes must be stored as uppercase characters.

filename
The CICS file name.

Length
Between 1 and 7 characters from CICS, and between 1 and 8 characters from RACF. The BSM supports between 1 and 7 characters as for the VSE file name.

Acceptable Characters
The same as for VSE file names. That is, between 1 and 7 alphanumeric characters, plus # (X'7B'), $ (X'5B'), and @ (X'7C'). The first character must be alphabetic or #, $, or @.

Resource Class JCICSJCT

Description
Used by the BSM to control access to CICS journals. The related CICS parameter is XJCT.

Format
<prefix.>journal

Max. length from BSM
15

These are the syntax rules for using the JCICSJCT resource class:

prefix The user-ID which was used in the CICS start job:

Length
Between 4 and 8 characters, as is required by the *Maintain User Profiles* dialog.

Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). Like user-IDs, prefixes must be stored as uppercase characters.

journal
The CICS journal names DFHJ01 to DFHJ99. The BSM uses a length of 6 characters, which differs from the length of 7 characters from the RACF CDT.

Length
6 characters from CICS.
Acceptable Characters
Always DFHJnn, where *nn* is between 01 and 99.

Resource Class MCICSPPT

Description
Used by the BSM to control access to CICS programs that have been invoked by other CICS application programs. The related CICS parameter is XPPT.

Format
<prefix.>program

Max. length from CDT
17

These are the syntax rules for using the MCICSPPT resource class:

prefix The user-ID which was used in the CICS start job:
Length
Between 4 and 8 characters, as is required by the *Maintain User Profiles* dialog.
Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). Like user-IDs, prefixes must be stored as uppercase characters.

program
The CICS program names (also refer to the topic describing BSM enhancements in the *CICS Transaction Server for VSE/ESA Security Guide*, SC33-1942-03 or later):
Length
Between 1 and 8 characters.
Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C').

Resource Class SCICSTST

Description
Used by the BSM to control access to CICS temporary storage queue. The related CICS parameter is XTST.

Format
<prefix.>tsqueue

Max. length from CDT
25

These are the syntax rules for using the SCICSTST resource class:

prefix The user-ID which was used in the CICS start job:
Length
Between 4 and 8 characters, as is required by the *Maintain User Profiles* dialog.
Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). Like user-IDs, prefixes must be stored as uppercase characters.

tsqueue
The CICS temporary storage queue name:
Length
Between 1 and 16 characters (also refer to the topic describing the WRITE TS command in the *CICS Transaction Server for VSE/ESA, Enhancements Guide*, GC34-5763-05 or later).
Acceptable Characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < and >. For further details, refer to the manual *CICS Transaction Server for VSE/ESA, Resource Definition Guide*, SC33-1653-05 or later.

Note:

1. This is the BSM checking rule: If there is a dot between the fourth and last character, and the string in front of the dot corresponds to the syntax rules for a VSE user-ID, the entry will be treated as prefixed. That means, if no matching BSM profile is found, to identify the access right the BSM will search for a profile that does not have a prefix.
2. If you include a temporary storage queue that contains hexadecimal characters in its name, unpredictable results may occur.
3. If a temporary storage queue name contains an imbedded blank, an ESM that you might be using may truncate the resource name to that blank.

Resource Class TCICSTRN

Description
Used by the BSM to control access to CICS transactions. The related CICS parameter is XTRAN.

Format
<prefix.>transaction

Max. length from CDT
13

These are the syntax rules for using the TCICSTRN resource class:

prefix The user-ID which was used in the CICS start job:
Length
Between 4 and 8 characters, as is required by the *Maintain User Profiles* dialog.
Acceptable Characters
Alphanumeric characters plus # (X'7B'), $ (X'5B'), and @ (X'7C'). Like user-IDs, prefixes must be stored as uppercase characters.

transaction
The name of the CICS transaction:
Length
Between 1 and 4 characters.
Acceptable Characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < and >. For further details, refer to the manual *CICS Transaction Server for VSE/ESA, Resource Definition Guide*, SC33-1653-05 or later.

WebSphere MQ for z/VSE Resource Classes

The Basic Security Manager supports the following resource classes that are used by *WebSphere MQ for z/VSE* Version 3 onwards:

Table 10. WebSphere MQ for z/VSE Resource Classes Supported by the BSM

Resource Class	Max. Length of Resource Name
MQADMIN	62
MQCMDS	22
MQCONN	10
MQQUEUE	53
MQNLIST	53

For details of how to protect WebSphere MQ for z/VSE resources using the BSTADMIN batch program, refer to the *WebSphere® MQ for z/VSE V3.0, System Management Guide*, GC34-6981.

Additional Resource Classes

The Basic Security Manager supports these additional resource classes:

Table 11. Additional Resource Classes Supported by the BSM

Resource Class	Max. Length of Resource Name	Description
SURROGAT	17	Used by the CICS Transaction Server for VSE/ESA

Chapter 29. Protecting Resources via BSTADMIN Commands

This chapter describes how you use **BSTADMIN** commands to protect CICS and other resources.

You can call BSTADMIN from either:
- The system console, for example using a command sequence such as:
```
r rdr,pausebg
0 // id user=sysa,pwd=xxxxxxxx
0 // exec bstadmin
0 li tcicstrn iesn
...
0 end
```
- A batch job.

Note:
1. If the functionality provided by the BSM does not meet your requirements, you might *instead* be able to use an External Security Manager (ESM) that is supplied by a vendor.
2. You *cannot* use both the BSM and an ESM with the *same* z/VSE system.
3. The partition in which BSTADMIN is to run must have *at least 256K* of virtual storage allocated to it.

This chapter contains these main topics:

- “Overview of BSM BSTADMIN Commands and Their Syntax” on page 358
- “ADD | AD Command” on page 361
- “CHANGE | CH Command” on page 362
- “DELETE | DE Command” on page 363
- “PERMIT | PE Command” on page 364
- “ADDGROUP | AG Command” on page 365
- “CHNGROUP | CG Command” on page 365
- “DELGROUP | DG Command” on page 365
- “CONNECT | CO Command” on page 365
- “REMOVE | RE Command” on page 366
- “LIST | LI Command” on page 366
- “LISTG | LG Command” on page 367
- “LISTU | LU Command” on page 367
- “PERFORM | PF Command” on page 367
- “USERID | ID Command” on page 370
- “STATUS | ST Command” on page 370
- “Return Codes That Might Occur When Using BSTADMIN” on page 371

Related Topics:

For details of how to ...	Refer to ...
use *Interactive Interface dialogs* to protect CICS and other resources	Chapter 30, “Protecting Resources via BSM Dialogs,” on page 373.
use BSM security to protect *operands* of specific JCL statements	“Using BSM Dialogs to Protect JCL Operands” on page 382.

For details of how to ...	Refer to ...
protect *CICS resources* using BSM resource classes, including practical examples	*CICS Security Guide*, SC33-1942-03 or later.
use BSM resource classes (reference information)	Chapter 28, "Resources Classes Stored in the BSM Control File," on page 349.
use security server commands	*z/VSE Operation*
use batch BSTADMIN commands to protect access to the IDCAMS SNAP command	*VSE/VSAM User's Guide and Application Programming*, SC33-8316.

Overview of BSM BSTADMIN Commands and Their Syntax

For most of the BSTADMIN commands shown in Figure 91 on page 359, the BSM must be active. If a command can be used when the BSM is *not* active (that means, z/VSE has been IPL'd with `SYS SEC=RECOVER`), an appropriate comment is provided in the command description.

Note: When logging with DMF: If you use logging with DMF, for performance reasons you should specify parameter `OS390` in the `EXEC BSTADMIN` statement of Figure 91 on page 359. This is required so that BSTADMIN can communicate *directly* with the DMF partition. The first statement should therefore look like this:

```
EXEC BSTADMIN,OS390
```

```
EXEC BSTADMIN

Commands:
  ADD|AD      class-name profile-name [GEN|NOGEN]
                                      [AUDIT(audit-level1[(access-level)]
                                           [,audit-level2[(access-level)]]])]
                                      [UACC(uacc)]
                                      [DATA('installation-data')]
  CHANGE|CH   class-name profile-name [GEN|NOGEN]
                                      [AUDIT(audit-level1[(access-level)]
                                           [,audit-level2[(access-level)]]])]
                                      [UACC(uacc)]
                                      [DATA('installation-data')]
  DELETE|DE   class-name profile-name [GEN|NOGEN]
  PERMIT|PE   class-name profile-name [GEN|NOGEN] ID(name) ACCESS(access)|DELETE

  ADDGROUP|AG group [DATA('installation-data')]
  CHNGROUP|CG group [DATA('installation-data')]
  DELGROUP|DG group
  CONNECT|CO  group user-id
  REMOVE|RE   group user-id

  LIST|LI     class-name profile-name|* [GEN|NOGEN]
  LISTG|LG    group-name|*
  LISTU|LU    user-id

  PERFORM|PF  [AUDIT ADMINACC|NOADMINACC] |
              [CLASS(class-name) ACTIVE|INACTIVE] |
                        [CMDAUDIT|NOCMDAUDIT]]
              [SETOPT CMDUSERID|NOCMDUSERID]
              [DATASPACE REFRESH|SIZE(nK|nM)] |
              [PASSWORD [HISTORY|NOHISTORY]
                        [LENGTH(minimum-pw-length)]
                        [REVOKE(number-invalid-pws)|NOREVOKE]
                        [WARNING(days-before-pw-expires)|NOWARNING]]

  USERID|ID   USER(user-id) PASSWORD(password)

  STATUS|ST
```

Figure 91. Summary of Basic Security Manager BSTADMIN Commands

How You Enter a Command Continuation

To continue a command on the next line, you enter a dash (“–”) as the last non-blank character in the current line. In the following example, a LIST command is entered with continuations:

```
0 li -
BG-0000 BST902A CONTINUE
0 tcicstrn -
BG-0000 BST902A CONTINUE
0 iesn
BG 0000 CLASS      NAME
BG 0000 -----      ----
BG 0000 TCICSTRN   IESN
BG 0000
BG 0000 UNIVERSAL ACCESS
BG 0000 ----------------
BG 0000      NONE
BG 0000
BG 0000 INSTALLATION DATA
BG 0000 -----------------
BG 0000 NONE
BG 0000
BG 0000 AUDITING
BG 0000 --------
```

```
BG 0000 FAILURES(READ)
BG 0000
BG 0000 USER      ACCESS
BG 0000 ----      ------
BG 0000 GROUP61   READ
BG 0000 JIM       READ
BG 0000
BG 0000 BST904I RETURN CODE OF LIST IS 00
BG-0000 BST901A ENTER COMMAND OR END
```

How You Enter Generic Names

A generic profile name will usually be defined when resources of the same
- resource class, and
- access right

begin with the same characters in their names. This "common part" of the resource names can be used as a generic profile name.

For example, the security definition:

```
ADD TCICSTRN ABC GEN
PERMIT TCICSTRN ABC GEN ID(GROUP03) ACCESS(READ)
```

would be applied to all transactions with a name beginning with ABC (for example, ABC1, ABC2, ABCX, ...), if no discrete (non-generic) definition exists. In this example, possible generic names could also be AB and A. The BSM uses this rule: the longest generic profile name which matches the name of the accessed resource is used to identify the access right.

If there is a discrete profile for a resource, it will be used instead of generic profiles. For example, the security definition:

```
ADD TCICSTRN ABC3
PERMIT TCICSTRN ABC3 ID(GROUP04) ACCESS(READ)
```

would be used for transaction ABC3 instead of the ABC definition.

Definitions for CICS resources can have prefixes. If a prefix is specified in the generic profile name, it must be fully specified. Only the resource-name part can be truncated.

If a generic profile for all resources of a resource class is required, two single quotes must be specified instead of the resource name. For example,

```
ADD TCICSTRN '' GEN
PERMIT TCICSTRN '' GEN ID(GROUP04) ACCESS(READ)
```

This can also be defined for a single CICS using the prefix. For example,

```
ADD TCICSTRN DBDCCICS. GEN
PERMIT TCICSTRN DBDCCICS. GEN ID(GROUP05) ACCESS(READ)
```

How You Enter Comment Lines

Comment lines can be inserted between commands. A comment line starts with an asterisk, and is followed by a blank. For example:

```
* ***************************************
* Define generic transaction ABC
* ***************************************
ADD TCICSTRN ABC GEN
PERMIT TCICSTRN ABC GEN ID(GROUP03) ACCESS(READ)
* ***************************************
* Show security status
* ***************************************
ST
```

⋮

ADD | AD Command

Use the ADD command to add to BSM all resources belonging to resource classes which are supported by the BSM control file. These are the parameters for use with `ADD | AD`:

class-name
: Specifies the 1 to 8 character name of the class to which the resource belongs.

profile-name
: Specifies the name of the discrete or generic profile you want to add to the specified class. The syntax of the resource names depend on the specified resource class (described in Chapter 28, "Resources Classes Stored in the BSM Control File," on page 349). If the resource name contains special characters and/or is case-sensitive, the profile name must be enclosed in single quotes. Single quotes as part of the profile name have to be entered as two single quotes.

GEN
: Specifies that the profile name is a generic name. For details, see "How You Enter Generic Names" on page 360.

NOGEN
: Specifies that the profile name is fully specified. Usually it stays for a discrete resource. If nothing is specified, NOGEN will be assumed.

AUDIT(*audit-level1***[(***access-level***)][,***audit-level2***[(***access-level***)]])**
: Specifies the access attempts that should be logged (for details, see "Logging and Creating Reports Of Security-Related SMF Records" on page 433).

  Note: If you omit the AUDIT parameter, the BSM will use the default **FAILURES(READ)**.

  The *audit-level1* and *audit-level2* specify which access attempts should be logged. These are the values you can enter:

  ALL
  : Specifies that both authorized accesses and detected unauthorized access-attempts should be logged.

  FAILURES
  : Specifies that detected unauthorized access-attempts should be logged (the **Default**).

  NONE
  : Specifies that no logging should be done. An *access-level* cannot be specified.

SUCCESS
Specifies that authorized access-attempts should be logged.

Note: If you define the same audit-level twice, the access-level of the *most recent* definition will be used.

The *access-level* specifies which access levels should be logged. These are the values you can enter:

ALTER
Specifies that only ALTER access-level attempts should be logged.

READD
Specifies that access-attempts at any level should be logged (the **Default**).

UPDATE
Specifies that access-attempts at the UPDATE and ALTER level should be logged.

UACC(*uacc***)**
Specifies the universal access authority to be associated with this resource. The values you can enter for “universal access authorities” (*uacc*) are: ALTER, UPDATE, READ, or NONE. If you do not enter a value for this parameter, NONE will be assumed.

DATA(*'installation-data'***)**
You can use this parameter to specify up to 20 characters of installation-defined data that is to be stored in the profile. The data must be enclosed in quotes. *You can list this information by using the LIST command.*

Here is an example of using the ADD command to define files in the FCICSFCT resource class:

```
ADD FCICSFCT file1 UACC(NONE)
ADD FCICSFCT file2 UACC(NONE)
```

CHANGE | CH Command

Use the CHANGE command to change the profile of a resource. This resource must belong to a class that is used by the BSM control file. These are the parameters for use with CHANGE | CH:

class-name
Specifies the 1 to 8 character name of the class to which the resource belongs. For a list of resource classes supported by BSM, see Chapter 28, “Resources Classes Stored in the BSM Control File,” on page 349.

profile-name
Specifies the name of the discrete or generic profile you want to change. The name you specify must be the name of an existing discrete or generic profile in the specified class. If the resource name contains special characters and/or is case-sensitive, the profile name must be enclosed in single quotes. Single quotes as part of the profile name have to be entered as two single quotes.

GEN Specifies that the profile name is a generic name. For details, see “How You Enter Generic Names” on page 360.

NOGEN
Specifies that the profile name is fully specified. Usually it stays for a discrete resource. If nothing is specified, NOGEN will be assumed.

AUDIT(*audit-level1***[(***access-level***)][,***audit-level2***[(***access-level***)]])**
Specifies the access attempts that should be logged (for details, see "Logging and Creating Reports Of Security-Related SMF Records" on page 433).

The *audit-level1* and *audit-level2* specify which access attempts should be logged. These are the values you can enter:

ALL Specifies that both authorized accesses and detected unauthorized access-attempts should be logged.

FAILURES
Specifies that detected unauthorized access-attempts should be logged (the **Default**).

NONE
Specifies that no logging should be done. An *access-level* cannot be specified.

SUCCESS
Specifies that authorized access-attempts should be logged.

Note: If you have defined the same audit-level twice, the access-level of the *most recent* definition will be used.

The *access-level* specifies which access levels should be logged. These are the values you can enter:

ALTER
Specifies that only ALTER access-level attempts should be logged.

READ Specifies that access-attempts at any level should be logged (the **Default**).

UPDATE
Specifies that access-attempts at the UPDATE and ALTER level should be logged.

Note: If you omit the AUDIT parameter, the BSM will use the default **FAILURES(READ)**.

UACC(*uacc***)**
Specifies the universal access authority to be associated with this resource. The values you can enter for "universal access authorities" (*uacc*) are: ALTER, UPDATE, READ, or NONE.

DATA(*'installation-data'***)**
You can use this parameter to specify up to 20 characters of installation-defined data that is to be stored in the profile. The data must be enclosed in quotes. *You can list this information using the LIST command.*

DELETE | DE Command

Use the DELETE command to delete BSM resource profiles from the BSM control file. These are the parameters for use with `DELETE | DE`:

class-name
Specifies the 1 to 8 character name of the class to which the resource belongs. For a list of resource classes supported by BSM, see Chapter 28, "Resources Classes Stored in the BSM Control File," on page 349.

profile-name
Specifies the name of the discrete or generic profile BSM is to delete from specified class. The name you specify must be the name of an existing discrete or generic profile in the specified class. If the resource name

contains special characters and/or is case-sensitive, the profile name must be enclosed in single quotes. Single quotes as part of the profile name have to be entered as two single quotes.

GEN Specifies that the profile name is a generic name. For details, see "How You Enter Generic Names" on page 360.

NOGEN
Specifies that the profile name is fully specified. Usually it stays for a discrete resource. If nothing is specified, NOGEN will be assumed.

PERMIT | PE Command

Use the PERMIT command to maintain the list of user-IDs and groups that are authorized to access a specific resource that belongs to a resource class. The resource class must be supported by BSTADMIN. These are the parameters for use with `PERMIT | PE`:

class-name
Specifies the 1 to 8 character name of the class to which the resource belongs. For a list of resource classes supported by BSM, see Chapter 28, "Resources Classes Stored in the BSM Control File," on page 349.

profile-name
Specifies the name of an existing discrete or generic profile in the specified class whose access list you want to modify. If the resource name contains special characters and/or is case-sensitive, the profile name must be enclosed in single quotes. Single quotes as part of the profile name have to be entered as two single quotes.

GEN Specifies that the profile name is a generic name. For details, see "How You Enter Generic Names" on page 360.

NOGEN
Specifies that the profile name is fully specified. Usually it stays for a discrete resource. If nothing is specified, NOGEN will be assumed.

ID(*name***)**
Specifies which user-ID or group should have a defined access authority to this resource.

Note: If a user-ID *and* a group to which the user-ID is connected are *both* contained in the *same access list*, only the access right of the user-ID will be used (and *not* the access right of the group).

ACCESS(*access***)**
Specifies the access authority to this resource for the specified ID. The access authorities are ALTER, UPDATE, READ, and NONE. Independent of the PERMIT definition an administrators will always have access to a resource.

DELETE
For the specified ID, specifies that the access authority to this resource should be deleted.

Here is an example of using the `PERMIT` command to authorize users to read-from or write-to `file1` and `file2`:

```
PERMIT FCICSFCT file1 ID(group1) ACCESS(UPDATE)
PERMIT FCICSFCT file2 ID(group1) ACCESS(READ)
```

ADDGROUP | AG Command

Use the ADDGROUP command to define a new group. These are the parameters for use with `ADDGROUP | AG`:

group Specifies the 1 to 8 alphanumeric-character name of the group whose profile should be added to the BSM control file.
- The characters # (X'7B'), $ (X'5B'), and @ (X'7C') can be part of the group name. Lower case characters will be converted to upper case.
- The name of the group must be unique and must not currently exist as group name or user-ID in the BSM control file.

DATA('*installation-data***')**
You can use this parameter to specify up to 20 characters of installation-defined data that is to be stored in the profile. The data must be enclosed in quotes. *You can list this information using the LIST command.*

CHNGROUP | CG Command

Use the CHNGROUP command to change the installation-defined data associated with a group. These are the parameters for use with `CHNGROUP | CG`:

group Specifies the 1 to 8 alphanumeric-character name of the group whose profile should be changed. The characters # (X'7B'), $ (X'5B'), and @ (X'7C') can be part of the group name. Lower case characters will be converted to upper case.

DATA('*installation-data***')**
You can use this parameter to specify up to 20 characters of installation-defined data that is to be stored in the profile. The data must be enclosed in quotes. *You can list this information using the LIST command.*

DELGROUP | DG Command

Use the DELGROUP command to delete a group and its connections to user-IDs from the BSM control file. This is the parameter for use with `DELGROUP | DG`:

group Specifies the 1 to 8 alphanumeric-character name of the group which should be deleted. The characters # (X'7B'), $ (X'5B'), and @ (X'7C') can be part of the group name. Lower case characters will be converted to upper case.

CONNECT | CO Command

Use the CONNECT command to add a user-ID to an existing group. These are the parameters for use with `CONNECT | CO`:

group Specifies the 1 to 8 alphanumeric character name of the group to which the user-ID should be connected. The characters # (X'7B'), $ (X'5B'), and @ (X'7C') can be part of the group name. Lower case characters will be converted to upper case.

user-id Specifies the user-ID which is to be connected to the group.

REMOVE | RE Command

Use the REMOVE command to remove a user-ID from an existing group. These are the parameters for use with `REMOVE | RE`:

group Specifies the 1 to 8 alphanumeric character name of the group from which a user-ID should be removed. The characters # (X'7B'), $ (X'5B'), and @ (X'7C') can be part of the group name. Lower case characters will be converted to upper case.

user-id Specifies the user-ID which is to be removed from the group.

LIST | LI Command

Use the LIST command to list information about resources that are defined in the BSM control file. These are the parameters for use with `LIST | LI`:

class-name
Specifies the 1 to 8 character name of the class to which the resource belongs. For a list of resource classes supported by BSM, see Chapter 28, "Resources Classes Stored in the BSM Control File," on page 349.

profile-name
Specifies the name of an existing discrete or generic profile in the specified class about which information is to be displayed. If the resource name contains special characters and/or is case-sensitive, the profile name must be enclosed in single quotes. Single quotes as part of the profile name have to be entered as two single quotes.

* Specifies that all profiles of this resource class should be listed.

GEN Specifies that the profile name is a generic name (see "How You Enter Generic Names" on page 360). If '*' was entered instead of a profile name, all generic profiles of the specified resource class will be listed.

NOGEN
Specifies that the profile name is fully specified. Usually it stays for a discrete resource.
- If nothing is specified, NOGEN will be assumed.
- If '*' was entered instead of a profile name and NOGEN was specified, all non-generic (discrete) resources profiles of the specified class will be displayed.

Here is an example of using the `LIST` command and the output it produces:

```
0 li tcicstrn iesn
BG 0000 CLASS NAME
BG 0000 ----- ----
BG 0000 TCICSTRN IESN
BG 0000
BG 0000 UNIVERSAL ACCESS
BG 0000 ----------------
BG 0000 NONE
BG 0000
BG 0000 INSTALLATION DATA
BG 0000 -----------------
BG 0000 NONE
BG 0000
BG 0000 AUDITING
BG 0000 --------
BG 0000 SUCCESS(READ),FAILURES(ALTER)
BG 0000
BG 0000 USER ACCESS
```

```
BG 0000 ---- ------
BG 0000 GROUP61 READ
BG 0000 JIM READ
BG 0000
BG 0000 BST904I RETURN CODE OF LIST IS 00
BG-0000 BST901A ENTER COMMAND OR END
```

LISTG | LG Command

Use the LISTG command to list the user-IDs which belong to a specified group. These are the parameters for use with `LISTG | LG`:

group Specifies the group for which the connected user-IDs should be listed.

* Specifies that all groups should be listed together with their connected user-IDs.

Here is an example of using the `LISTG` command and the output it produces:

```
0 lg group61

BG 0000 INFORMATION FOR GROUP GROUP61
BG 0000     INSTALLATION DATA = NONE
BG 0000     USER(S)=
BG 0000     ANNA
BG 0000     BERTA
BG 0000     CICSUSER
BG 0000     HUGO
BG 0000     TONY
BG 0000 BST904I RETURN CODE OF LISTG IS 00
BG-0000 BST901A ENTER COMMAND OR END
```

LISTU | LU Command

Use the LISTU command to list the names of the groups to which a specific user-ID belongs. These is the parameter for use with `LISTU | LU`:

user-id Specifies the user-ID for which the group information should be listed.

Here is an example of using the `LISTU` command and the output it produces:

```
0 lu anna

BG 0000 INFORMATION FOR USER ANNA
BG 0000     GROUP(S)=
BG 0000     GROUP61
BG 0000     TEST1
BG 0000 BST904I RETURN CODE OF LISTU IS 00
BG-0000 BST901A ENTER COMMAND OR END
```

PERFORM | PF Command

Use the PERFORM command to activate or deactivate resources classes in the BSM, or to refresh the contents of the data space. These are the parameters for use with `PERFORM | PF`:

AUDIT
Specifies the system-wide BSM auditing options.

ADMINACC
Specifies that the BSM should log *all* accesses to resources made by the system administrator (user type 1). For resources that are

defined in DTSECTAB, you should use the program ACLR (for details, see Chapter 38, "Logging/Reporting Security Events," on page 433).

NOADMINACC (default)
Specifies that the BSM should **not** log accesses to resources made by the system administrator (user type 1). However, if the resource profile specifies AUDIT(ALL) or AUDIT(SUCCESS), accesses by the administrator will nevertheless be logged.

CLASS(*class-name***)**
Specifies the 1 to 8 character name of the class to which the resource belongs. The variable *class-name* can be followed by:

ACTIVE
Specifies that the resource class should be set to "active", and that access-authority evaluation should be done for resources of this class.

INACTIVE
Specifies that the resource class should be set to "inactive", and that no access-authority evaluation should be done for resources of this class. **Warning!**: Do not make a resource "inactive" if the resource class is being used by an active CICS system. **This will probably cause the CICS system to crash!**

CMDAUDIT
Specifies that command-auditing should be performed for this resource class. The BSM will then log *all* changes that are made (using BSTADMIN commands) to resource profiles of this resource class.

NOCMDAUDIT
Specifies that command-auditing should no longer be performed for this resource class. The BSM will then not log changes that are made (using BSTADMIN commands) to resource profiles of this resource class.

SETOPT
Specifies the general BSM options:

CMDUSERID
Specifies that in an environment where batch security is *not active* (SYS SEC=NO), the user of BSTADMIN has to first identify himself/herself via a user-ID and password, before he/she can enter any other BSTADMIN command.

NOCMDUSERID
Specifies that in an environment where batch security is *not active* (SYS SEC=NO), the user of BSTADMIN can enter any BSTADMIN command *without* having to first identify himself/herself via a user-ID and password. This is the **default**.

DATASPACE
Specifies actions that are related to the data space. The keyword DATASPACE can be followed by:

REFRESH
Specifies that the contents of the data space should be rebuilt from information contained in the BSM control file.

Note: The user issuing the `PERFORM DATASPACE REFRESH` command or issuing the `RACROUTE REQUEST=LIST` macro has the responsibility to ensure that no multitasking that results in the issuing of a `RACROUTE REQUEST=AUTH`, `RACROUTE REQUEST=FASTAUTH`, `RACROUTE`

REQUEST=VERIFY, RACROUTE REQUEST=LIST macro, or PERFORM DATASPACE REFRESH command occurs at the same time as the PERFORM DATASPACE REFRESH command.

SIZE Specifies how large the data space will be allocated. It can be specified as kilobytes (1K to 99999K) or as megabytes (1M to 2048M) . If a SIZE value has *never been specified*, the system default size for a single data space will be used. For further details of system settings for data spaces, you can use the QUERY DSPACE command.

Note: The PERFORM command with the SIZE option can be used during recovery when the BSM is not active.

PASSWORD
Specifies actions for the monitoring and checking of passwords. The keyword PASSWORD can be followed by:

HISTORY
Specifies that for each user-ID, the BSM should save the previous passwords and compare these with an intended new password. If there is a match with one of these previous passwords, or with the current password, the BSM will reject the intended new password.

NOHISTORY
Specifies that the new password information should only be compared with the current password.

LENGTH(*minimum-pw-length***)**
Specifies the minimum password length. The BSM does not permit a password that is longer than 8 alphanumeric characters. You can enter a minimum of 3 characters for a password. This is also the default.

REVOKE(*number-invalid-pws***)**
Specifies the number (1 to 254) of consecutive incorrect password attempts that the BSM will allow before it revokes the user-ID on the next incorrect attempt. The default value is 5.

NOREVOKE
Specifies that the BSM should ignore the number of consecutive invalid password attempts.

WARNING(*days-before-pw-expires***)**
Specifies the number of days (1 to 255) before a password expires, that the signon program should issue a warning message to a user. The default is 7 days.

Note: The signon program of the z/VSE Interactive Interface only supports an internal maximum value of 7 days!

NOWARNING
Specifies that CICS should not issue a warning message for password expiration.

Note: Ensure that you have removed the // EXEC IESIRCVT statement in the procedure USERBG.PROC. To do so, use skeleton SKUSERBG (located in ICCF library 59). If this statement is contained in USERBG.PROC, the settings specified by the PERFORM PASSWORD ... command will be overwritten by the IESIRCVT settings!

USERID | ID Command

Use the USERID command to identify yourself to BSTADMIN using a user-ID and password. To use all other BSTADMIN commands, the user must be an administrator (type 1 user-ID). The USERID command is required in an environment where *no batch security is active* (SYS SEC=NO) and the BSM option CMDUSERID is set. It is *ignored* in all other cases.

USER(*user-ID***)**
: Specifies the user-ID that identifies the user who wishes to use BSTADMIN commands.

PASSWORD(*password***)**
: Specifies the password with which to authenticate the user who wishes to use BSTADMIN commands.

STATUS | ST Command

Use the STATUS command to obtain the status information of the BSM.

Here is an example of using the STATUS command and the output it produces:

```
0 st
BG 0000 CLASS     ACTIVE    CMDAUDIT
BG 0000 -----     ------    --------
BG 0000 USER      YES       NO
BG 0000 GROUP     YES       NO
BG 0000 DATASET   NO        NO
BG 0000 VSELIB    NO        NO
BG 0000 VSESLIB   NO        NO
BG 0000 VSEMEM    NO        NO
BG 0000 TCICSTRN  YES       NO
BG 0000 ACICSPCT  YES       NO
BG 0000 DCICSDCT  YES       NO
BG 0000 FCICSFCT  YES       NO
BG 0000 JCICSJCT  YES       NO
BG 0000 MCICSPPT  YES       NO
BG 0000 SCICSTST  YES       NO
BG 0000 APPL      YES       NO
BG 0000 FACILITY  YES       NO
BG 0000 MQADMIN   YES       NO
BG 0000 MQCMDS    YES       NO
BG 0000 MQCONN    YES       NO
BG 0000 MQNLIST   YES       NO
BG 0000 MQQUEUE   YES       NO
BG 0000 SURROGAT  YES       NO
BG 0000
BG 0000 PASSWORD PROCESSING OPTIONS:
BG 0000   12 GENERATIONS OF PREVIOUS PASSWORDS BEING MAINTAINED.
BG 0000   AFTER   3 CONSECUTIVE UNSUCCESSFUL PASSWORD ATTEMPTS,
BG 0000       A USERID WILL BE REVOKED.
BG 0000   PASSWORD EXPIRATION WARNING LEVEL IS   7 DAYS.
BG 0000   A PASSWORD CAN HAVE   3 TO   8 CHARACTERS.
BG 0000
BG 0000 AUDIT OPTIONS:
BG 0000   ADMINISTRATOR ACCESSES TO RESOURCES ARE LOGGED
BG 0000
BG 0000 GENERAL OPTIONS:
BG 0000   NO USER ID IS REQUIRED TO USE BSTADMIN WITHOUT BATCH SECURITY
BG 0000   BSTADMIN IS USING USER ID SYSA FOR AUTHORIZATION
BG 0000
BG 0000 LOGGING STATUS:
BG 0000   SMF LOGGER FOR DTSECTAB RESOURCES IS INSTALLED AND ACTIVE
BG 0000   DMF STATUS IS: ACTIVE
BG 0000
```

```
BG 0000 DATA SPACE STATUS:
BG 0000   CURRENT DATA SPACE SIZE IS     960K.
BG 0000   USAGE OF DATA SPACE STORAGE IS  16%.
BG 0000   DATA PART SIZE IS     159K.
BG 0000   SIZE OF PREVIOUS DATA SPACE WAS     960K.
BG 0000   USAGE OF PREVIOUS DATA SPACE WAS  16%.
BG 0000   DATA PART SIZE WAS     159K.
BG 0000
BG 0000 BST904I RETURN CODE OF STATUS IS 00
BG-0000 BST901A ENTER COMMAND OR END
```

The above listing of classes shows the names of the classes as they are used *in RACROUTE requests*. The user profiles that are defined in the VSE.CONTROL.FILE belong to the class USER.

For the resources defined in table DTSECTAB, the following class names are used:
- DATASET (VSE files that have the format *volid.fileid*)
- VSELIB (VSE libraries that have the format *volid.fileid.libname*)
- VSESLIB (VSE sublibraries that have the format *libname.sublibname*)
- VSEMEM (VSE sublibrary members that have the format *libname.sublibname.membername*)

Return Codes That Might Occur When Using BSTADMIN

These are the return codes that might be generated by the BSTADMIN program.

Return Code (Dec)	Return Code (Hex)	Description
0	0	The requested operation was successful.
8	8	A user error (that is, syntax error) occurred. BSTADMIN continues with the next command.
12	C	An internal processing error occurred. BSTADMIN terminates processing.
16	10	The required BSM was not active, or another severe problem exists. BSTADMIN terminates processing.
77	4D	The phase usage of BSTADMIN or BSTADMII is incorrect, or incorrect **rmode** objects have been linked. BSTADMIN terminates processing.
87	57	BSTADMIN is already active in this partition. BSTADMIN terminates processing.
88	58	BSTADMIN is not supported for the VSE release you are currently running. BSTADMIN terminates processing.
89	59	The request is not supported for use with dialogs. BSTADMIN terminates processing.

Chapter 30. Protecting Resources via BSM Dialogs

This chapter shows how BSM dialogs can be used to protect CICS and other resources. It begins with a short scenario that illustrates the main concepts, and then goes on to describe how you can implement security via BSM dialogs for specific applications.

Note: The Fast Path **285**, the *Define Transaction Security (DTSECTXN)* dialog, has been retained for migration purposes and in case you wish to continue using the earlier security concept based upon the DTSECTXN table. Details of how to migrate your CICS transaction data are given in "Performing the Migration" on page 285.

This chapter contains these main topics:

Related Topics:

For details of how to ...	Refer to ...
use the BSTADMIN EXEC to protect CICS and other resources	Chapter 29, "Protecting Resources via BSTADMIN Commands," on page 357.
connect a user-ID to BSM security groups	"Adding/Changing the Group Connects for a VSE User-ID" on page 302.
use BSM resource classes (reference information)	Chapter 28, "Resources Classes Stored in the BSM Control File," on page 349.
protect CICS resources using BSM resource classes, including practical examples	*CICS Security Guide*, SC33-1942-03 or later.
use batch BSTADMIN commands to protect access to the IDCAMS SNAP command	*VSE/VSAM User's Guide and Application Programming*, SC33-8316.

Scenario to Demonstrate the Use of BSM Dialogs

Security Environment to be Created in the Scenario

This is the security environment that we establish in the scenario:

Table 12. BSM Resource Profiles Used in the Scenario

BSM Resource Class	Resource Name (and Description)	Access/User list with UACC(READ)
GROUP	PRODCGRP (the group of users who can access the Production CICS system)	IVAN, APPLUSR1, ...
GROUP	DBDCCGRP (the group of users who can access the Development CICS system)	DEV1, DEV2, ...
APPL	DBDCCICS (the Development CICS system)	CICSUSER, DBDCCGRP
APPL	PRODCICS (the Production CICS system)	CICSUSER, PRODCGRP

- CICSUSER is included in the scenario because this user is required as a default for CICS.
- IVAN will be given access to PRODCICS. Afterwards, if IVAN then signs on to z/VSE, the name of the application he wishes to access will be sent to the BSM. Providing the resource class APPL is activated in the BSM and a resource profile for the subject-application exists, the BSM verifies that IVAN has a minimum of read-access to the application profile. If not, an appropriate message is sent to his sign-on panel.

Note: For some resource classes, the resource profile is *case-sensitive*. Here is an example of how incorrect information will be generated if you enter a resource profile in the wrong case:

1. In the *Maintain Security Profiles* panel (Fast Path **2811**) and for resource class TCICSTRN, you wish to display a list of CICS transactions starting with the CEDA transaction.
2. You enter **ceda** (in lower case) in the field `START....`
3. z/VSE does *not* display a list of transactions starting with CEDA. Instead, z/VSE displays a list of transactions starting (for example) with emai, ftp, iccf, and so on (all in lower case).

Step 1: Add Group Profiles

In the first step of the scenario, we add the group profiles PRODCGRP and DBDCCGRP.

1. Use Fast Path **28** to display the *Security Maintenance* dialog. Next, select Option '2' (“BSM Group Maintenance”) and press Enter.

```
IESADMSL.IESEBSEC           SECURITY MAINTENANCE
                                                       APPLID: DBDCCICS
  Enter the number of your selection and press the ENTER key:

        1  BSM Resource Profile Maintenance
        2  BSM Group Maintenance
        3  BSM Security Rebuild
        4  Maintain Certificate - User ID List
        5  Define Transaction Security (DTSECTXN)










PF1=HELP                    3=END        4=RETURN              6=ESCAPE(U)
                            9=Escape(m)

==> 2                                            Path: 28
```

2. The *Maintain Security Profiles* dialog is displayed for BSM resource class GROUP. (In the example below, “TRANSEC CLASS MIGRAT” indicates that these groups were migrated from CICS security-keys). Now enter a '1' anywhere in the OPT column and press Enter.

```
IESADMBSLG                  MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS:    GROUP
START.... GROUP01
OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = USER LIST
           8 = CONNECT     9 = REMOVE       USERID
   OPT    GROUP NAME   DESCRIPTION          CONNECTED?
                                            ________
    1     GROUP01      TRANSEC CLASS MIGRAT
    _     GROUP02      TRANSEC CLASS MIGRAT
    _     GROUP03      TRANSEC CLASS MIGRAT
    _     GROUP04      TRANSEC CLASS MIGRAT
    _     GROUP05      TRANSEC CLASS MIGRAT
    _     GROUP06      TRANSEC CLASS MIGRAT
    _     GROUP07      TRANSEC CLASS MIGRAT
    _     GROUP08      TRANSEC CLASS MIGRAT
    _     GROUP09      TRANSEC CLASS MIGRAT
    _     GROUP10      TRANSEC CLASS MIGRAT
    _     GROUP11      TRANSEC CLASS MIGRAT
    _     GROUP12      TRANSEC CLASS MIGRAT

PF1=HELP                    3=END
              8=FORWARD    9=PRINT   10=REMOVE ALL
```

Note: The *Maintain Security Profiles* panel is described in detail in “Adding/Changing the Group Connects for a VSE User-ID” on page 302.

3. The *Maintain Security Profiles* dialog is displayed. Now enter the group-name PRODCGRP and description, and press PF5 (Update). The BSM control file is then updated with these details. PRODCGRP then becomes an “instance” of resource class GROUP.

```
IESADMBSAG                  MAINTAIN SECURITY PROFILES
BSM    CLASS: GROUP

Add Group:


  GROUP NAME......... PRODCGRP        1 - 8 characters

  DESCRIPTION........ Production Group     Optional remark










PF1=HELP                    3=END                     5=UPDATE
```

4. Add group DBDCCGRP (with description “Development Group”) by repeating actions **1**, **2**, and **3** of this step.

Step 2: Add Users to Groups

In this second step, we add IVAN and APPLUSR1 to the group PRODCGRP. This is required so that IVAN and APPLUSR1 can access PRODCICS (the Production CICS system). Then we add user-ID APPLUSR1 to the group PRODCGRP, and finally user-IDs DEV1 and DEV2 to the group DBDCCGRP.

1. In the *Maintain Security Profiles* dialog, we enter a '6' (User List) next to the group PRODCGRP and press Enter.

```
IESADMBSLG                  MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS:    GROUP
START.... GROUP54
OPTIONS:   1 = ADD        2 = CHANGE      5 = DELETE      6 = USER LIST
           8 = CONNECT    9 = REMOVE       USERID
   OPT    GROUP NAME   DESCRIPTION         CONNECTED?
                                           ________
    _     GROUP54      TRANSEC CLASS MIGRAT
    _     GROUP55      TRANSEC CLASS MIGRAT
    _     GROUP56      TRANSEC CLASS MIGRAT
    _     GROUP57      TRANSEC CLASS MIGRAT
    _     GROUP58      TRANSEC CLASS MIGRAT
    _     GROUP59      TRANSEC CLASS MIGRAT
    _     GROUP60      TRANSEC CLASS MIGRAT
    _     GROUP61      TRANSEC CLASS MIGRAT
    _     GROUP62      TRANSEC CLASS MIGRAT
    _     GROUP63      TRANSEC CLASS MIGRAT
    _     GROUP64      TRANSEC CLASS MIGRAT
    6     PRODCGRP     Production Group

PF1=HELP                    3=END
PF7=BACKWARD   8=FORWARD    9=PRINT   10=REMOVE ALL
```

2. The *Maintain User List* dialog is displayed. Now enter a '1' (Add) in the OPT field and press Enter.

```
IESADMBSLU                 MAINTAIN USER LIST
BSM    CLASS:    GROUP       GROUP: PRODCGRP
START....
OPTIONS:   1 = ADD                         5 = DELETE

   OPT    USERID

    1

PF1=HELP                   3=END
PF7=BACKWARD   8=FORWARD
```

3. The *Maintain Security Profiles* dialog is displayed. We now add IVAN to the group PRODCGRP and press PF5 (Update).

```
IESADMBSAU                 MAINTAIN SECURITY PROFILES
BSM    CLASS: GROUP

Connect Userid to group:


  GROUP NAME......... PRODCGRP       Group name

  USERID............. IVAN           1 - 8 characters

PF1=HELP                   3=END                    5=UPDATE
```

4. Add the user-ID APPLUSR1 to the group PRODCGRP. To do so, we repeat actions **2** and **3** of this Step.
5. Add user-IDs DEV1 and DEV2 to the group DBDCCGRP (to complete the first two rows of Table 12 on page 374). To do so, we repeat actions **1**, **2**, and **3** of this Step for group DBDCCGRP.

Note: If the groups in this scenario had *already existed*, we could have used Option '8' (Connect) to connect the user-IDs to the groups. See “Connecting a User to Groups via Option 8” on page 380.

Step 3: Add Resource Profiles and Give Access Rights

In this step, we first add APPL resource profiles for application DBDCCICS. Then we add user-ID CICSUSER and group DBDCCGRP to the access list for DBDCCICS. Finally, CICSUSER and group DBDCCGRP are given an access-right of READ.

1. Use Fast Path **2818** to display the *Maintain Security Profiles* dialog for resource class APPL. Now we enter a '1' anywhere in the OPT column and press Enter. The *Maintain Security Profiles* dialog *for resource class APPL* is displayed. Now enter details of resource DBDCCICS and press PF5 (Update). DBDCCICS then becomes an "instance" of resource class APPL.

```
IESADMBSAE                 MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS:    APPL

Add Profile:


  PREFIX............ ________         CICS region

  RESOURCE NAME...... DBDCCICS
                                      Maximum length is  8 characters.
  GENERIC...........                  (1=yes, 2=no)

  UNIVERSAL ACCESS... _               (_=None, 2=Read, 3=Update, 4=Alter)

  AUDIT-LEVEL 1 ..... 1               (_=None, 1=Failure, 2=Success, 3=All)
  ACCESS-LEVEL 1 .... 2               (2=Read, 3=Update, 4=Alter), _=default

  AUDIT-LEVEL 2 .....                 (_=None, 1=Failure, 2=Success, 3=All)
  ACCESS-LEVEL 2 ....                 (2=Read, 3=Update, 4=Alter), _=default
  DESCRIPTION........                     Optional remark
PF1=HELP                   3=END                    5=UPDATE
```

2. The *Maintain Security Profiles* dialog for resource class APPL is re-displayed. Now we must find the entry DBDCCICS in the list of profile names that are displayed. Enter option '6' next to DBDCCICS to display the access list for this resource. The *Maintain Access List* dialog is displayed. Now enter a '1' (ADD) in the OPT column to display the "Add Userid or Groupid" function.
3. Enter details of CICSUSER (who is given read-access to resource profile DBDCCICS), and press PF5 (Update).

```
IESADMBSAA                 MAINTAIN ACCESS LIST
BSM    CLASS: APPL        PROFILE:  DBDCCICS

Add Userid or Groupid:


  NAME............... CICSUSER        Userid or Groupid

  ACCESS............ 2                (_=None,
                                       2=Read, 3=Update, 4=Alter)










PF1=HELP                   3=END                    5=UPDATE
```

4. Add group DBDCCGRP to the access list for resource DBDCCICS, by repeating action **3** of this Step. When completed, the *Maintain Access List* dialog is

displayed showing user-ID CICSUSER and group DBDCCGRP. Both have access '2' (read-only) for resource profile DBDCCICS.

```
IESADMBSLA                  MAINTAIN ACCESS LIST
BSM    CLASS: APPL         PROFILE:  DBDCCICS
START....                                  NUMBER OF ENTRIES ON LIST:  00002
OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE

   OPT    NAME    ACC

    _     CICSUSER 2
    _     DBDCCGRP 2














PF1=HELP                    3=END
PF7=BACKWARD   8=FORWARD
```

The next actions in Step 3 consists of:

- Adding APPL resource profiles for application PRODCICS.
- Adding user-ID CICSUSER and group PRODCGRP to the access list for application PRODCICS.
- Giving CICSUSER and PRODCGRP an access-right of READ.

To perform these actions, we repeat actions **1**, **2**, **3**, and **4** of this Step.

Step 4: Activate the Security Setup

In this last step, we activate:

- Application profiles:
  - DBDCCICS
  - PRODCICS
- Group profiles:
  - DBDCCGRP
  - PRODCGRP

To do so:

1. Use Fast Path **28** to display the *Security Maintenance* panel.
2. Select option '3' (“BSM Security Rebuild”). The activation of the application and group profiles will now automatically proceed.
3. The message “Security Information Was Successfully Rebuilt” is displayed when the process has completed.

The completion of Steps 1 to 4 means that the BSM security environment shown in Table 12 on page 374 has been successfully created!

Connecting a User to Groups via Option 8

In "Step 2: Add Users to Groups" on page 376, three actions were required to connect user-ID IVAN to the group PRODCGRP via option '6' (User List). However, for *existing* groups you can use a single action to connect a user to one or more groups. This is done using Option '8' (Connect) of the *Maintain Security Profiles* panel.

In the example provided here, the user-ID DEV3 is a programmer (Type 2 user) and will be connected to these groups:
- DBDCCGRP
- GROUP06
- GROUP07
- GROUP08
- GROUP11

1. We type an '8' (Connect) in the OPT column for the five groups listed above.

```
IESADMBSLG                  MAINTAIN SECURITY PROFILES
 BSM RESOURCE CLASS:    GROUP
 START....
 OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = USER LIST
            8 = CONNECT     9 = REMOVE       USERID
    OPT    GROUP NAME   DESCRIPTION          CONNECTED?
                                             DEV3____
     8     DBDCCGRP     DEVELOPMENT CICS
     _     GROUP01      TRANSEC CLASS MIGRAT
     _     GROUP02      TRANSEC CLASS MIGRAT
     _     GROUP03      TRANSEC CLASS MIGRAT
     _     GROUP04      TRANSEC CLASS MIGRAT
     _     GROUP05      TRANSEC CLASS MIGRAT
     8     GROUP06      TRANSEC CLASS MIGRAT
     8     GROUP07      TRANSEC CLASS MIGRAT
     8     GROUP08      TRANSEC CLASS MIGRAT
     _     GROUP09      TRANSEC CLASS MIGRAT
     _     GROUP10      TRANSEC CLASS MIGRAT
     8     GROUP11      TRANSEC CLASS MIGRAT

 PF1=HELP                   3=END
 PF7=BACKWARD   8=FORWARD    9=PRINT   10=REMOVE ALL
```

2. After pressing ENTER, the user-ID is connected to the five groups. An asterisk will be displayed in the USERID CONNECTED? column for each of the five groups.

Removing User Connects to Groups via Option 9

The *Maintain Security Profiles* panel allows you to use Option '9' (Remove) to remove the connects of a user-ID to one or more groups.

In the example provided here, the user-ID DEV3 is a programmer (Type 2 user) whose connects to these groups will be removed:
- DBDCCGRP
- GROUP08

1. We type a '9' (Remove) in the OPT column for the two groups listed above.

```
IESADMBSLG                  MAINTAIN SECURITY PROFILES
 BSM RESOURCE CLASS:    GROUP
 START....
 OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = USER LIST
            8 = CONNECT     9 = REMOVE       USERID
    OPT    GROUP NAME   DESCRIPTION          CONNECTED?
                                             DEV3
     9     DBDCCGRP     DEVELOPMENT CICS        *
     _     GROUP01      TRANSEC CLASS MIGRAT
     _     GROUP02      TRANSEC CLASS MIGRAT
     _     GROUP03      TRANSEC CLASS MIGRAT
     _     GROUP04      TRANSEC CLASS MIGRAT
     _     GROUP05      TRANSEC CLASS MIGRAT
     _     GROUP06      TRANSEC CLASS MIGRAT    *
     _     GROUP07      TRANSEC CLASS MIGRAT    *
     9     GROUP08      TRANSEC CLASS MIGRAT    *
     _     GROUP09      TRANSEC CLASS MIGRAT
     _     GROUP10      TRANSEC CLASS MIGRAT
     _     GROUP11      TRANSEC CLASS MIGRAT    *

 PF1=HELP                   3=END
 PF7=BACKWARD   8=FORWARD    9=PRINT   10=REMOVE ALL
```

2. After pressing ENTER, the connects are removed. The asterisks will be removed from the `USERID CONNECTED?` column for each of the groups.

Removing User Connects to All Groups via PF10

The *Maintain Security Profiles* panel allows you to use PF10 (Remove All) to remove the connects of a user-ID to *all* groups to which the user-ID is currently connected.

In the example provided here, the user-ID `DEV3` is a programmer (Type 2 user) and will be removed from *all* groups to which this user-ID is currently connected. These groups are:

- GROUP06
- GROUP07
- GROUP11
- GROUP22
- GROUP46

(GROUP 22 and GROUP46 would be displayed in the panel below by pressing PF8).

1. We press PF10 (Remove All) in the panel below.

```
IESADMBSLG                  MAINTAIN SECURITY PROFILES
 BSM RESOURCE CLASS:    GROUP
 START....
 OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = USER LIST
            8 = CONNECT     9 = REMOVE       USERID
    OPT    GROUP NAME   DESCRIPTION          CONNECTED?
                                             DEV3
     _     DBDCCGRP     DEVELOPMENT CICS
     _     GROUP01      TRANSEC CLASS MIGRAT
     _     GROUP02      TRANSEC CLASS MIGRAT
     _     GROUP03      TRANSEC CLASS MIGRAT
     _     GROUP04      TRANSEC CLASS MIGRAT
     _     GROUP05      TRANSEC CLASS MIGRAT
     _     GROUP06      TRANSEC CLASS MIGRAT    *
     _     GROUP07      TRANSEC CLASS MIGRAT    *
     _     GROUP08      TRANSEC CLASS MIGRAT
     _     GROUP09      TRANSEC CLASS MIGRAT
     _     GROUP10      TRANSEC CLASS MIGRAT
     _     GROUP11      TRANSEC CLASS MIGRAT    *

 PF1=HELP                   3=END
 PF7=BACKWARD   8=FORWARD    9=PRINT   10=REMOVE ALL
```

2. The connects are removed, and the asterisks will be removed from the USERID CONNECTED? column for each of the groups.

Using BSM Dialogs to Protect JCL Operands

You can use BSM security to protect *operands* of specific JCL statements. For example, you can protect the PERM operand of the ASSGN and LIBDEF statements.

IBM provides five resource profiles that are used for JCL statement checking:
- IBMVSE.JCL.ASSGN.**PERM**
- IBMVSE.JCL.LIBDEF.**PERM**
- IBMVSE.JCL.LIBDROP.**PERM**
- IBMVSE.JCL.OPTION.**PARSTD**
- IBMVSE.JCL.OPTION.**STDLABEL**

(In the above resource profiles, the operands are shown as highlighted).

JCL statement checking is activated using the SEC parameter of the z/VSE IPL procedure. For details, see "Using the Tailor-IPL-Procedure Dialog to Tailor Security Parameters" on page 278.

Note:
1. To perform JCL statement checking:
   - JCL security must be enabled.
   - The minimum access right for Universal Access or user-IDs/groups must be READ.
2. As an alternative to using resource profiles, you could use *generic* profiles. For details, see "How You Enter Generic Names" on page 360.

After JCL security is enabled, each time a user submits a batch job the statements contained in the batch job (for example, the PERM operand of an ASSIGN or LIBDEF statement) will be checked against access lists. For each resource profile, these user-IDs/groups will be authorized to execute the JCL statement:
- User-IDs of type 1 (Administrator).
- User-IDs/groups contained in the access list used with the security profile.

If a security profile does not exist, *all* user-IDs/groups will be authorized to execute the JCL statement.

An example of setting up JCL statement checking now follows. To give all users contained in group S1JCLGRP read-access to the resource profiles used for JCL statement checking, we must:
1. Use Fast Path **2819** to display the *Maintain Security Profiles* panel. Then enter IBMVSE in the **START** field and press Enter. The resource profiles used for JCL statement checking are then displayed.

```
IESADMBSLE                 MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS:    FACILITY                            ACTIVE
START....  IBMVSE                                   (CASE SENSITIVE)
OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = ACCESS LIST

   OPT     PROFILE NAME                     DESCRIPTION       UNIVERSAL AUDIT
                                          >                      ACCESS VALUE
    _      IBMVSE.JCL.ASSGN.PERM            FOR JCL SECURITY                12
    _      IBMVSE.JCL.LIBDEF.PERM           FOR JCL SECURITY                12
    _      IBMVSE.JCL.LIBDROP.PERM          FOR JCL SECURITY                12
    _      IBMVSE.JCL.OPTION.PARSTD         FOR JCL SECURITY                12
    _      IBMVSE.JCL.OPTION.STDLABEL       FOR JCL SECURITY                12










PF1=HELP                   3=END
               8=FORWARD    9=PRINT                 11=NAME RIGHT
```

2. Type 6 (Access List) in the OPT column for the profile IBMVSE.JCL.ASSGN.PERM and press Enter. The *Maintain Access List* panel is displayed.
3. Now enter a '1' (ADD) in the OPT column to display the “Add Userid or Groupid” function. Enter the details for S1JCLGRP as shown below and press PF5 (Update).

```
IESADMBSAA                 MAINTAIN ACCESS LIST
BSM    CLASS: FACILITY     PROFILE:  IBMVSE.JCL.ASSGN.PERM

Add Userid or Groupid:



  NAME............... S1JCLGRP          Userid or Groupid

  ACCESS............ 2                  (_=None,
                                         2=Read, 3=Update, 4=Alter)











PF1=HELP                   3=END                    5=UPDATE
```

4. Repeat actions 2 and 3 of this Step for the other four resource profiles to give S1JCLGRP read-access to these resource profiles.
5. To confirm that read-access has been successfully granted, we use Fast Path **2819** and type 6 (Access List) in the OPT column for any of the profiles (in this example, for IBMVSE.JCL.ASSGN.PERM). Press Enter and the *Maintain Access List* panel should confirm that access has been successfully defined.

```
IESADMBSLA                  MAINTAIN ACCESS LIST
BSM    CLASS: FACILITY     PROFILE:  IBMVSE.JCL.ASSGN.PERM
START....                                 NUMBER OF ENTRIES ON LIST:  00002
OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE

   OPT    NAME    ACC

    _     S1JCLGRP 2















PF1=HELP                    3=END
PF7=BACKWARD   8=FORWARD
```

Note that instead of using dialogs, you can use batch BSTADMIN commands to set up JCL statement checking (described in Chapter 29, “Protecting Resources via BSTADMIN Commands,” on page 357).

Chapter 31. Overview of DTSECTAB-Based VSE Security

This chapter describes how the *z/VSE Access Control Function* is used together with table *DTSECTAB* to control access to files, libraries, sublibraries, and library members. It also describes how access control to *batch jobs* is done.

The z/VSE Access Control Function is supplied with the *Basic Security Manager* (BSM).

This chapter contains these main topics:

Related Topics:

For details of ...	Refer to ...
how to activate VSE security based on table DTSECTAB	Chapter 32, "Customizing/Activating DTSECTAB-Based Security," on page 391.
the contents of DTSECTAB (as delivered by IBM)	"Content of Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59)" on page 395.
the syntax of the DTSECTAB macro, and examples of its use	Chapter 34, "DTSECTAB Macro: Syntax and Examples," on page 413.
how security information is propagated between one or more VSE/POWER batch environments	Chapter 35, "Propagation of VSE/POWER Security Identification," on page 421.

How Security Checking Is Performed

Security checking is performed on two distinct levels:

1. User **identification** and **authentication**:
   - User identification: this is done by checking the **user ID**. Is this user-ID known to the system?
   - User authentication: is the user really the person that owns this user ID? This is checked either via an **explicit password** supplied with a job, or it is checked via an indication that the password had been validated at some earlier stage. This may have been done during sign-on, for example, before the job was submitted. In this case, no further password check is necessary.
2. **Access authorization**:

   is the user permitted to **access a particular** resource such as a file, library, sublibrary, or member?

   This is done by comparing the
   - **User profile** information in the VSE.CONTROL.FILE with the

- **Resource profile** information in DTSECTAB.

How User Profile Information Is Used

User profiles are stored in the *VSE.CONTROL.FILE*. A user profile specifies, for an individual user, the **access rights** to resources. You define user profiles via the Maintain User Profile Dialog. See Chapter 25, “Maintaining User Profiles via BSM Dialogs,” on page 289 for details.

The ACC parameter defines access control **classes** together with associated **access rights**. A user's access control **class** can have one of the following access rights:

- ALT (Alter)
- UPD (Update)
- READ (Read-only)
- CON (Connect)

These access rights are ordered hierarchically: ALTER implies UPDATE, UPDATE implies READ, READ implies CONNECT.

The definition of access rights will be discussed in detail in “How Access Rights Are Used” on page 401. Refer also to Figure 81 on page 296 where the "Add or Change Resource Access Rights" panel of the *Maintain User Profiles* dialog is shown.

A user that is defined as system administrator (type 1 user) in the user profile has unrestricted access with access right of ALT to all protected resources.

Which Resources Can Be Protected in DTSECTAB?

Resources to be protected must be defined in DTSECTAB. The following resources can be protected:

1. All **libraries, sublibraries**, and all their **members**.

   Members are protected at **member name** level. That is, within one sublibrary, members of different types with the same name are protected under the same resource profile. For example, if a user has an access right to member name PROG1, that right applies to PROG1.A, PROG1.E, PROG1.OBJ, PROG1.PHASE, and PROG1.PROC.
2. **Files** as outlined below:
   - All VSE/VSAM KSDS, RRDS, VRDS and ESDS accessed directly via an ACB macro, and all VSE/VSAM-managed SAM files accessed via either a DTFSD or an ACB macro, or by appropriate file definition statements of the IBM compiler(s) used at your installation. Note that the file's VOLSER and catalog are not checked.

     When VSE/VSAM data is accessed via a path, the path name is used for access checking.

     The file's catalog can only be checked if the cluster is defined with the authorization parameter, and a VSE/VSAM *user security verification routine* (USVR) is coded. The catalog name can be passed to the USVR exit after the entry point name in the authorization parameter. You find detailed information on the USVR exit in the IBM manual *VSE/VSAM Commands* under “User Security-Verification Routine”.

- All non-VSAM disk files and standard-labeled tape files that are defined by a file description macro (DTFxx), or by appropriate file definition statements of the IBM compiler(s) used at your installation.

You cannot protect through DTSECTAB entries:

- Unlabeled tapes
- Tapes with non-standard labels

Defining Resources in DTSECTAB

For each resource to be protected, the security administrator defines one or more access control classes in the corresponding resource profile. In general, resources without an entry in DTSECTAB are not protected.

Access control classes are numbers between 1 and 32 which are assigned to the resource.

A typical definition might look as follows:

```
DTSECTAB TYPE=SUBLIB,NAME=AUX.PR$302,ACC=(8,9)
```

Sublibrary AUX.PR$302 is defined as a resource that can be accessed by users who have access control classes 8 and 9 defined in their user profile.

Authorization of a particular user to access a resource is determined by a match of the access control classes. In our example "An Example of Using Access Rights" on page 404, user ENDU with access control class 1 through 8 is allowed to access sublibrary AUX.PR$302 due to the match on class 8. The access right is limited to UPDATE. An attempt by ENDU to ALTER (rename or delete) the sublibrary would be an access violation.

Access Control via classes establishes an **individual access right** for the user. For resources "library, sublibrary, and member", additionally a **universal access right (UACC)** can be specified. It grants **all** users of the system, irrespective of the classes specified in user profiles (if such profiles exist), the defined access right of ALT, UPD, READ or CON. For a resource with a **universal** access right, **individual** access rights are meaningful only if they are higher than the universal access right, because at least the UACC will be granted to any user.

You find detailed information about protection of resources in Chapter 33, "Access Rights/Checking in DTSECTAB," on page 401.

Using the IBM-Provided DTSECTAB

The predefined table DTSECTAB contains mainly *system-defined resources*. It does not use any classes, the resources are protected via **universal access rights** only. In this way the pregenerated definitions do not interfere with the user's installation-specific class definitions.

You can build upon this table if you need to tailor the given support. You may, for example, extend the set of resources to include your own resources. Or, you may want to establish a set of **access control classes** to implement your own rules of differentiation between individual users.

More details on the predefined support are given in Chapter 32, "Customizing/Activating DTSECTAB-Based Security," on page 391.

How Users Are Identified and Authenticated

In a secured z/VSE system, batch jobs that are submitted for processing are checked for identification and authentication (security identification):

- User identification: is the user known to the system? That is, does the user have a user profile in the VSE.CONTROL.FILE?
- Authentication: is the user really the person that owns this user ID?

Security identification is supplied in three ways:

1. During sign-on to a z/VSE subsystem such as the z/VSE Interactive Interface or VSE/ICCF. Jobs submitted from here run under the sign-on user ID.
2. An explicit security identification in the SEC parameter of the VSE/POWER JECL statement * $$ JOB of a submitted job.

   An equivalent identification can be given for jobs submitted via a z/VSE-internal interface, the “VSE/POWER Spool Access Support”. It is described in the IBM manual *VSE/POWER Application Programming* in the topic “Introduction to Spool-Access Support”.
3. An explicit security identification in the // ID job control statement of a job submitted.

If you submit a job, you need not explicitly enter user-ID and password with each submission, as explained in “How VSE/POWER Jobs are Authenticated” on page 389.

Security Information in the JECL Statement * $$ JOB

The parameter SEC in the * $$ JOB statement of VSE/POWER specifies user-ID and password of the VSE/POWER job to be submitted:

```
* $$ JOB ... SEC=(user-ID,password)
```

The SEC parameter is optional. However, if specified, it must contain both user-ID and password.

The security information in the * $$ JOB statement is valid for the entire sequence of z/VSE jobs included in a VSE/POWER job stream.

For a complete description of the * $$ JOB statement refer to the manual *VSE/POWER Administration and Operation* under “* $$ JOB: Marking the Start of a VSE/POWER Job”.

Security Information in the JCL Statement // ID

The job control statement // ID carries the same information as the JECL statement * $$ JOB, that is: user-ID and password.

The information is valid for one z/VSE job: it covers the job where it is included, but not any other job that might follow.

A // ID statement overrides the VSE/POWER security information for the length of the z/VSE job. After that, VSE/POWER's security information becomes effective again.

// ID statements should be avoided because users with access to jobs in the VSE/POWER reader queue can see both, user-ID and password. Specifying the user-ID and password in the *** $$ JOB** statement is the better and recommended solution. Retrieving a job from the VSE/POWER reader queue generally does not reveal user-ID and password.

However, there are situations where the statement is needed, for example

- In z/VSE startup procedures. Please refer to "Access Control for Startup Procedures" on page 410.
- In PAUSExx jobs. Please refer to "Tasks to be Done after Initial Installation" on page 392.
- In jobs that accomplish the transferring of jobs and files between systems. z/VSE dialogs may create a // ID statement if the remote system is at *backlevel* (z/VSE prior to VSE/ESA 1.3). Please refer to "Transfer of Jobs or Files/Members between Systems" on page 424.

How VSE/POWER Jobs are Authenticated

The user-ID, which z/VSE knows (for example from sign-on), is sufficient for user authentication if a batch job is submitted from one of the five sources:

1. z/VSE Interactive Interface
2. VSE/ICCF
3. A workstation via the SEND/RECEIVE command interface
4. A job with explicit user-ID and password specification
5. Another authenticated job.

Therefore, a user who submits a job from any of the above sources does not need to care about the user-ID or the password for this job.

A job that is submitted on behalf of a user whose user-ID and password have been validated earlier is called an **authenticated job**.

Note: If a job is to run with another user profile than the one of the submitter, user-ID and password must be supplied. In this case, not the // ID statement but the *** $$ JOB** statement should be used for the reasons outlined above.

The subject of user-ID propagation is discussed in more detail in Chapter 35, "Propagation of VSE/POWER Security Identification," on page 421.

Chapter 32. Customizing/Activating DTSECTAB-Based Security

This chapter describes how you can customize the IBM-supplied table DTSECTAB, and then activate access-control security based on this table.

It also lists the contents of table DTSECTAB, as shipped by IBM.

As shipped, z/VSE provides basic security support for access-control which is *ready to use*. This support includes the following functions and resources:
- Activation of the Basic Security Manager (BSM) during initial installation.
- Startup procedures adapted for a system with the Basic Security Manager active.
- A predefined access control table (DTSECTAB) for resource protection which is automatically generated during initial installation, if you answered with YES during initial installation.
- PRIMARY sublibraries for predefined users SYSA, OPER, and PROG.

This chapter contains these main topics:

Related Topics:

For details of ...	Refer to ...
how to specify access rights in table DTSECTAB, and how access checking is used to process these access rights	Chapter 33, "Access Rights/Checking in DTSECTAB," on page 401.
the syntax of the DTSECTAB macro, and examples of its use	Chapter 34, "DTSECTAB Macro: Syntax and Examples," on page 413.

Activating Security for Batch Resources

You can activate basic security for batch resources either during initial installation, or later by using the *Tailor IPL Procedure* dialog as described under "Using the Tailor-IPL-Procedure Dialog to Tailor Security Parameters" on page 278.

If the support is activated **during initial installation**, the installation program updates the IPL procedure $IPLESA and adds the statement:

```
SYS SEC=(YES,NOTAPE)
```

where:
- NOTAPE - means that files on disk are protected but not files on tape.

- SEC=YES - activates both the protection of disk and the protection of tape files.

In a system with security on, the predefined access control table DTSECTAB becomes active at the first IPL after initial installation.

This ensures a correct startup for a system with security active. You should use the predefined DTSECTAB as a base when later adding your own entries. The table provides the necessary protection of system libraries. Startup would not work if system libraries were not properly protected.

Tasks to be Done after Initial Installation

During initial installation, the system defines the users shown in the table in Table 1 on page 5.

The first IPL *after initial installation* with SEC=YES activates the predefined basic security support. A system administrator (type 1 user, SYSA) has unlimited access to all resources and should now do the following:

- Change the passwords of user-IDs FORSEC, SYSA, OPER, $SRV, and PROG in file VSE.CONTROL.FILE.

  Note: These user-IDs are explained in Table 1 on page 5.
- For the user-ID FORSEC, also change the password in DTSECTAB.
- For users FORSEC, PROG, OPER and $SRV, you must perform a logon immediately after installation, in order to change the password. Afterwards, use the *Maintain Primary Sublibraries* dialog to define PRIMARY sublibraries for the users SYSA, PROG, and OPER. The password for user POST needs to be changed after the initial installation process has been completed. Change the password via the dialog *Maintain User Profiles*, (for details refer to Chapter 25, “Maintaining User Profiles via BSM Dialogs,” on page 289).
- Change and submit skeleton SKCICS and possibly SKCICS2 in VSE/ICCF library 59. Log on as system administrator, and submit the job. This submission catalogs the startup job CICSICCF as member CICSICCF.Z in the system sublibrary IJSYSRS.SYSLIB and places the job into the VSE/POWER reader queue. The job "inherits" the security attributes of the submitter and changes them via the ID statement to the special task user-ID for the CICS region. Make sure this is the only CICS startup job by deleting the original from the reader queue.

As PAUSExx jobs do not have any access rights, you can provide access rights via the // ID statement if required.

Considerations for User-IDs FORSEC and DUMMY

Users FORSEC and DUMMY are available for system purposes only. They are the *only users defined in DTSECTAB*.

FORSEC

is defined as **system administrator** in the z/VSE system that is delivered to you. Its purpose is to provide appropriate access rights **at system startup**.

Note: Important! You should NEVER remove user FORSEC from either DTSECTAB or the VSE.CONTROL.FILE

If your installation uses the optional program *VSE/Access Control - Logging and Reporting (VSE/ACLR)*, you may want to reduce the access rights of this user by specifying access control **classes.** This greatly reduces the number of logging records because **administrator** accesses are always logged (not only violations). Therefore, activate the Logging and Reporting program only after you have reduced the access rights of user FORSEC in DTSECTAB and the VSE control file. Refer also to Chapter 38, “Logging/Reporting Security Events,” on page 433.

The manual *VSE/Access Control-Logging and Reporting: Program Reference and Operations Guide* provides details about VSE/Access Control - Logging and Reporting.

DUMMY

has no special access rights. This user-ID is included in certain startup procedures before certain jobs (such as PAUSEBG, PAUSEF1, ...) are submitted. User-ID DUMMY serves to inhibit the inheritance of the user FORSEC's access rights (which are administrator rights) to those jobs.

Note: Important! Never remove user DUMMY from DTSECTAB

Pregenerated Access Control Table DTSECTAB

The z/VSE system that is delivered to you has a predefined access control table DTSECTAB. The table provides a basic level of security; it does not use any access control classes. It is automatically generated and ready for use immediately after initial installation.

You should **never remove or change** any of the supplied entries without careful consideration. You may, of course, add entries by using the methods described below.

Predefined Member DTSECTRC (Containing DTSECTAB)

DTSECTRC contains security information about protected **z/VSE system resources** (plus definitions of special users: FORSEC and DUMMY). Its content is shown under “Content of Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59)” on page 395. (The exact content may have changed since this manual was printed. Please look at your copy in *VSE/ICCF library 59*, member DTSECTRC.)

You may add entries for protected z/VSE system resources as needed or change those provided by adding access control classes. Programs that can potentially bypass access control, should also be protected at member level. Be sure that no person other than the security administrator is given an access right to sensitive members.

Note: Except for users FORSEC and DUMMY, users should only be defined in VSE.CONTROL.FILE (user SYSA).

“Maintaining the Access Control Table DTSECTAB” on page 394 describes how to maintain DTSECTAB.

Maintaining the Access Control Table DTSECTAB

VSE Access Control offers two major groups of access rights for resources:

- **Universal** access rights via the UACC parameter, and/or
- **Individual** access rights via the ACC parameter.

Three scenarios will be discussed in the following:

1. Only the pregenerated security support is used: you do not add **resources** of your own to DTSECTAB, just **users** in the VSE.CONTROL.FILE.
2. You add users plus resources of your own. The resources are protected with universal access rights only. Access control classes are not used.
3. You add users plus resources of your own. The resources are protected by universal access rights as well as by individual access rights via classes.

In each of these three cases, a new DTSECTAB phase must be cataloged. z/VSE activates the new table when the next user identification and authentication is to be done. The source version of DTSECTAB is called ***DTSECTRC***.

Note: If you make changes to table DTSECTAB, you are recommended to IPL your z/VSE to ensure that these changes will be included in z/VSE components that store parts of the z/VSE security data.

Scenario 1. Predefined Security Support Only

You confine yourself to the resources defined in the pregenerated DTSECTAB. These resources are defined with UACC only; access control classes are not used. You just add/delete users. To do this, use one of the two methods:

- Define a user in the VSE.CONTROL.FILE, with the *Maintain User Profiles* dialog.
- Modify skeleton IESUPDCF (a member in library 59) and submit the job to batch. This job updates the VSE.CONTROL.FILE.

Scenario 2. Add Resources Using the UACC Parameter Only

You add resources in addition of those defined in the pregenerated DTSECTAB. The resources are protected by universal access rights only. This implies that you need not modify the user profiles in the VSE.CONTROL.FILE.

Proceed as follows:

- Get yourself a copy of DTSECTRC from VSE/ICCF library 59 into your private VSE/ICCF library. **Whenever you change DTSECTAB, use the member you copied into your private VSE/ICCF library.**
- Update member DTSECTRC in your private VSE/ICCF library by adding definitions for the resources you want to protect.
- Submit DTSECTRC from your private VSE/ICCF library.

Scenario 3. Add Resources Using the ACC Parameter

You add **resources** in addition to those defined in the pregenerated DTSECTAB. Some of the resources are protected via **access control classes**. This forces you to modify user profiles in the VSE.CONTROL.FILE because the ACC parameter must be included.

Proceed as follows for users:

- Use the *Maintain User Profiles* dialog as described under “Adding/Changing a User-ID and Profile Definitions” on page 290.

For resources, proceed in the same way as described under “Scenario 2. Add Resources Using the UACC Parameter Only” on page 394.

Applying IBM Service to DTSECTRC

Since DTSECTRC is code supplied by IBM, it *can be affected by IBM service*.

After a service PTF or FSU (Fast Service Upgrade) has been applied, you will have the latest IBM-supplied version of DTSECTRC in VSE/ICCF library 59. Consider that you may have to **update your version in sublibrary PRD2.SAVE** accordingly.

Protecting the Access Control Table DTSECTAB Itself

To prevent manipulation and misuse of the information stored in DTSECTAB, the IBM-supplied DTSECTAB has entries which serve to protect the DTSECTAB itself. Please refer to Figure 92 on page 396.

The system encrypts sensitive information in the table to provide additional protection.

Content of Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59)

The supplied DTSECTAB is shown in Figure 92 on page 396.

```
*****************************************************
*                                                   *
*   5686-CF7 (C) COPYRIGHT IBM CORP. 1984, 2004     *
*                                                   *
*****************************************************
         TITLE 'DTSECTAB - SECURITY TABLE FOR RESOURCES'
********************************************************************
         PUNCH ' CATALOG DTSECTRC.OBJ  REP=YES'
         SPACE 3
*-------------------------------------------------------------------*
*                                                                   *
*       STATIC PART OF DTSECTAB                                     *
*                                                                   *
*-------------------------------------------------------------------*
*       THIS PART IS SHIPPED AS A-BOOK IN IJSYSRS.SYSLIB.DTSECTRC.   *
*       IF CHANGED, THE USER SHOULD PUT HIS VERSION UNDER THE SAME   *
*       NAME IN PRD2.SAVE, AS IBM SERVICE IS DONE ON THE MEMBERS     *
*       CONTAINED IN IJSYSRS.SYSLIB.                                *
*       (THE JOB TO BUILD A DTSECTAB LOOKS FIRST IN PRD2.SAVE FOR    *
*       DTSECTRC).                                                  *
*-------------------------------------------------------------------*
*-------------------------------------------------------------------*
*                IBM SUPPLIED USERS                                 *
*-------------------------------------------------------------------*
*** USER DUMMY HAS NO SPECIAL SECURITY RIGHTS.USED TO RESET INHERITANCE
*** IT AVOIDS GETTING TOO MANY RIGHTS WHILE LOADING POWER JOBS DURING
*** ASI.
*** YOU SHOULD NOT DEFINE AN II USER WITH THE NAME 'DUMMY'.
*-------------------------------------------------------------------*
         DTSECTAB TYPE=USER,                                           C
               NAME=DUMMY,                                             C
               PASSWRD=DUMMY,                                          C
               AUTH=NO,                                                C
               SUBTYPE=INITIAL
         SPACE 3
*-------------------------------------------------------------------*
*** USER FORSEC HAS ALL ACCESS RIGHTS. THEREFORE, THE PASSWORD NEEDS
*** TO BE CHANGED AFTER INITIAL INSTALLATION.
*-------------------------------------------------------------------*
         DTSECTAB TYPE=USER,                                           C
               NAME=FORSEC,                                            C
               PASSWRD=FORSEC,                                         C
               READDIR=YES,                                            C
               MCONS=YES,                                              C
               AUTH=YES,                                               C
               RIGHT=BTRANS
*-------------------------------------------------------------------*
*                 END OF IBM SUPPLIED USERS                         *
*-------------------------------------------------------------------*
*-------------------------------------------------------------------*
*                                                                   *
*       FOLLOWING IS THE Z/VSE 3.1 SUPPLIED PART OF THE DTSECTAB     *
*       THAT DEFINES A MINIMUM SET OF RESOURCES TO BE PROTECTED.     *
*                                                                   *
*-------------------------------------------------------------------*
```

Figure 92. Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59), Part 1 of 6

```
****** LIBRARIES*******************************************************
******   IJSYSRS
         DTSECTAB TYPE=LIB,                                            C
               NAME=DOSRES.VSE.SYSRES.LIBRARY.IJSYSRS,                 C
               UACC=CON
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=IJSYSRS.SYSLIB,                                    C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.*,                                  C
               UACC=READ
******   CPUVAR* IS USED BY VARIOUS JOBSTREAMS TO SAVE PARAMETERS
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.CPUVAR*,                            C
               UACC=UPD
******   ALLOW PROGRAMMER TO ADD HIS/HER OWN VSAM FILE VIA II DIALOGS
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.STDLABUP,                           C
               UACC=UPD
******   CLRDK DESTROYS THE DATA ON A DISK.
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.CLRDK
******   ICKDSF DESTROYS DATA ON A DISK.
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.ICKDSF
******   IKQVDU CHANGES THE FORMAT 1 LABEL
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.IKQVDU
******   DTSANALS SHOULD BE EXECUTED BY AUTHORIZED PERSONS ONLY
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.DTSANALS
******   DTSUTILA IS A RESOURCE THAT PROTECTS SECURITY SENSITIVE
*        DTSUTIL COMMANDS FROM BEING EXECUTED BY NON-ADMINISTRATORS
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.DTSUTILA
******   SECURITY PROGRAMS/TABLES
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.DTSEC*
******   PROGRAMS TO MANIPULATE THE CONTROL FILE
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.IESUPDCF
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.IESBLDUP
******   PROGRAMS TO MANIPULATE THE LDAP USER MAPPING FILE
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.IESLDUMA
         DTSECTAB TAPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.IESLDSO
         DTSECTAB TAPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.IESLDSOC
******   PROGRAMS TO ALLOW POWER Q MANIPULATION
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.DTRIJMGR
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.IPW$$DD
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=IJSYSRS.SYSLIB.BSTADMIN
```

Figure 93. Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59), Part 2 of 6

```
******   THE NEXT ENTRIES ARE USED TO PROTECT THE PASSWORDS IN ...
*        (PROGRAMS THAT ACCES THE VSE.CONTROL.FILE)
         DTSECTAB TYPE=MEMBER,                                          C
               NAME=IJSYSRS.SYSLIB.CICSICCF
         DTSECTAB TYPE=MEMBER,                                          C
               NAME=IJSYSRS.SYSLIB.CICS2
******   IJSYSR2 IS ALIAS NAME OF IJSYSRS, USED BY SERVICE DIALOGS
*        WHICH CAN ONLY INVOKED BY SYSTEM ADMINISTRATOR
         DTSECTAB TYPE=LIB,                                             C
               NAME=DOSRES.VSE.SYSRES.LIBRARY.IJSYSR2,                  C
               UACC=CON
******   IJSYSR1 IS ALIAS NAME OF IJSYSRS, USED BY SERVICE DIALOGS
*        WHICH CAN ONLY INVOKED BY SYSTEM ADMINISTRATOR
         DTSECTAB TYPE=LIB,                                             C
               NAME=SYSWK1.SYS.NEW.RES.IJSYSR1,                         C
               UACC=CON
******   PRD1
         DTSECTAB TYPE=LIB,                                             C
               NAME=*.VSE.PRD1.LIBRARY.PRD1,                            C
               UACC=CON
******   PRD1.BASE
         DTSECTAB TYPE=SUBLIB,                                          C
               NAME=PRD1.BASE,                                          C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                          C
               NAME=PRD1.BASE.*,                                        C
               UACC=READ
******   PRD1.BASED SERVICE SUBLIBRARY
         DTSECTAB TYPE=SUBLIB,                                          C
               NAME=PRD1.BASED,                                         C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                          C
               NAME=PRD1.BASED.*,                                       C
               UACC=READ
******   PRD1.MACLIB
         DTSECTAB TYPE=SUBLIB,                                          C
               NAME=PRD1.MACLIB,                                        C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                          C
               NAME=PRD1.MACLIB.*,                                      C
               UACC=READ
******   PRD1.MACLIBD SERVICE SUBLIBRARY
         DTSECTAB TYPE=SUBLIB,                                          C
               NAME=PRD1.MACLIBD,                                       C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                          C
               NAME=PRD1.MACLIBD.*,                                     C
               UACC=READ
******   AVOID THAT ANYONE CAN MANIPULATE FILES USING DITTO
         DTSECTAB TYPE=MEMBER,                                          C
               NAME=PRD1.BASE.DITTO
******   PRDPRIM IS ALIAS NAME OF PRD1, USED BY SERVICE DIALOGS
*        WHICH CAN ONLY BE INVOKED BY SYSTEM ADMINISTRATOR
         DTSECTAB TYPE=LIB,                                             C
               NAME=*.VSE.PRD1.LIBRARY.PRDPRIM,                         C
               UACC=CON
******   PRD2
         DTSECTAB TYPE=LIB,                                             C
               NAME=*.VSE.PRD2.LIBRARY.PRD2,                            C
               UACC=CON
         DTSECTAB TYPE=SUBLIB,                                          C
               NAME=PRD2.*,                                             C
               UACC=READ
```

Figure 94. Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59), Part 3 of 6

```
******  PRD2.SCEEBASE LE CODE LIBRARY
        DTSECTAB TYPE=SUBLIB,                                       C
              NAME=PRD2.SCEEBASE,                                   C
              UACC=CON
        DTSECTAB TYPE=MEMBER,                                       C
              NAME=PRD2.SCEEBASE.*,                                 C
              UACC=READ
******  PRD2.SCEEBASD LE SERVICE LIBRARY
        DTSECTAB TYPE=SUBLIB,                                       C
              NAME=PRD2.SCEEBASD,                                   C
              UACC=CON
        DTSECTAB TYPE=MEMBER,                                       C
              NAME=PRD2.SCEEBASD.*,                                 C
              UACC=READ
******  PRD2.DBASE PRODUCT LIBRARY DATABASES
        DTSECTAB TYPE=SUBLIB,                                       C
              NAME=PRD2.DBASE,                                      C
              UACC=CON
        DTSECTAB TYPE=MEMBER,                                       C
              NAME=PRD2.DBASE.*,                                    C
              UACC=READ
******  PRD2.PROD PRODUCT LIBRARY IN GENERAL
        DTSECTAB TYPE=SUBLIB,                                       C
              NAME=PRD2.PROD,                                       C
              UACC=CON
        DTSECTAB TYPE=MEMBER,                                       C
              NAME=PRD2.PROD.*,                                     C
              UACC=READ
******  PRD2.COMM PRODUCT LIBRARY COMMUNICATION PRODUCTS 1
        DTSECTAB TYPE=SUBLIB,                                       C
              NAME=PRD2.COMM,                                       C
              UACC=CON
        DTSECTAB TYPE=MEMBER,                                       C
              NAME=PRD2.COMM.*,                                     C
              UACC=READ
******  PRD2.COMM2 PRODUCT LIBRARY COMMUNICATION PRODUCTS 2
        DTSECTAB TYPE=SUBLIB,                                       C
              NAME=PRD2.COMM2,                                      C
              UACC=CON
        DTSECTAB TYPE=MEMBER,                                       C
              NAME=PRD2.COMM2.*,                                    C
              UACC=READ
******  PRD2.AFP PRODUCT LIBRARY ADVANCED PRINTER
        DTSECTAB TYPE=SUBLIB,                                       C
              NAME=PRD2.AFP,                                        C
              UACC=CON
        DTSECTAB TYPE=MEMBER,                                       C
              NAME=PRD2.AFP.*,                                      C
              UACC=READ
******  PRD2.DB2750 PRODUCT LIBRARY DB2 7.5.0
        DTSECTAB TYPE=SUBLIB,                                       C
              NAME=PRD2.DB2750,                                     C
              UACC=CON
        DTSECTAB TYPE=MEMBER,                                       C
              NAME=PRD2.DB2750.*,                                   C
              UACC=READ
```

Figure 95. Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59), Part 4 of 6

```
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=PRD2.DB2750C,                                      C
               UACC=CON
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=PRD2.DB2750C.*,                                    C
               UACC=READ
******   PRD2.DB2STP PRODUCT LIBRARY DB2 STORED PROCEDURES
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=PRD2.DB2STP,                                       C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=PRD2.DB2STP.*,                                     C
               UACC=READ
******   SRV$SYS IS USED BY BACKUP JOBSTREAMS TO SAVE PARAMETERS
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=PRD2.CONFIG.SRV$SYS,                               C
               UACC=UPD
******   BASIC START NEEDS THIS SUBLIB
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=PRD2.SAVE,                                         C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=PRD2.SAVE.*,                                       C
               UACC=UPD
******   ONLY THE SA IS ALLOWED TO READ THE SECURITY RELATED MEMBERS
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=PRD2.SAVE.DTSEC*
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=PRD2.SAVE.DTRI*
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=PRD2.CONFIG,                                       C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=PRD2.CONFIG.*,                                     C
               UACC=UPD
******   ONLY THE SA IS ALLOWED TO READ THE SECURITY RELATED MEMBERS
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=PRD2.CONFIG.DTSEC*
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=PRD2.BSXCPU*,                                      C
               UACC=ALT
******   PRIMARY LIBRARY
         DTSECTAB TYPE=LIB,                                            C
               NAME=*.VSE.PRIMARY.LIBRARY.PRIMARY,                     C
               UACC=CON
******   THE $$C SUBLIB SHOULD BE USED TO EXCHANGE DATA BETWEEN USERS
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=PRIMARY.$$C,                                       C
               UACC=UPD
******   CRYPTO LIBRARY SSL KEYS
         DTSECTAB TYPE=LIB,                                            C
               NAME=*.VSE.CRYPTO.LIBRARY.CRYPTO,                       C
               UACC=CON
******   CRYPTO.KEYRING
         DTSECTAB TYPE=SUBLIB,                                         C
               NAME=CRYPTO.KEYRING,                                    C
               UACC=CON
         DTSECTAB TYPE=MEMBER,                                         C
               NAME=CRYPTO.KEYRING.*
```

Figure 96. Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59), Part 5 of 6

```
******  FILES ********************************************************
        DTSECTAB TYPE=FILE,                                          C
               NAME=*.VSE.CONTROL.FILE
        DTSECTAB TYPE=FILE,                                          C
               NAME=*.VSE.BSTCNTL.FILE
        DTSECTAB TYPE=FILE,                                          C
               NAME=*.VSE.LDAP.MAPPING.FILE
        DTSECTAB TYPE=FILE,                                          C
               NAME=DOSRES.VSE.POWER.QUEUE.FILE
        DTSECTAB TYPE=FILE,                                          C
               NAME=SYSWK1.VSE.POWER.DATA.FILE,                      C
               SUBTYPE=FINAL
         EJECT
```

Figure 97. Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59), Part 6 of 6

Chapter 33. Access Rights/Checking in DTSECTAB

This chapter provides detailed information about:
- how you specify *access rights* in the table DTSECTAB, and
- how *access checking* is used to process these access rights.

It contains these main topics:
- "How Access Rights Are Used"
- "Access Control for Libraries" on page 405
- "Access Control for Startup Procedures" on page 410
- "System Phases, B-Transients, Link Area, SVA and LTA" on page 411

Related Topics:

For details of ...	Refer to ...
how to activate VSE security based on table DTSECTAB	Chapter 32, "Customizing/Activating DTSECTAB-Based Security," on page 391.
the contents of DTSECTAB (as delivered by IBM)	"Content of Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59)" on page 395.
the syntax of the DTSECTAB macro, and examples of its use	Chapter 34, "DTSECTAB Macro: Syntax and Examples," on page 413.

How Access Rights Are Used

Table 13 shows how access rights are used in relation to protected resources. Detailed explanations are provided in the topics that follow.

Access rights are to be defined for a user's access control classes through the *Maintain User Profiles* dialog.

Table 13. Access Rights for Libraries, Sublibraries and Members

Access Right	Library	Sublibrary	Member
ALT	Create and delete.	Create, delete and rename.	Create, delete and rename.

Table 13. Access Rights for Libraries, Sublibraries and Members (continued)

Access Right	Library	Sublibrary	Member
UPD	Update contents. Create, delete and rename (ALT) sublibraries in it.	Update contents. Catalog, delete and rename (ALT) members in it.	Update contents. Add, delete and change lines.
READ	Read only for library and all sublibraries in it.	Read only for sublibrary and all members in it.	Read only.
CON	Access to sublibraries in it, if user has access right for these sublibraries individually.	Access to members in it, if user has access right for these members individually.	Not Applicable.

The meaning of the access rights is as follows:
- ALT = Alter
- UPD = Update
- READ = Read
- CON = Connect

Please recall that ALT implies UPD, UPD implies READ, READ implies CON (where applicable).

Note: A user must have at least access right CON to a protected sublibrary in order to access it by a LIBDEF statement.

The above table shows access rights for libraries, sublibraries and members. Access rights for **files** are as follows:
- Both ALT and UPD provide the right to **create, delete, rename** a file, and to **add, delete,** and **change** records.
- READ means 'read-only'.
- CON is not applicable to a file.

For **DASD files**, the Access Control Function determines during OPEN processing the required access right depending on the ACB (access control block) of a VSE/VSAM or DTF file. For example, to open a DTFSD file for INPUT, only an access right of READ must be defined in DTSECTAB. If the same file is opened for OUTPUT, an access right of UPD is required in DTSECTAB.

The following table lists the access rights required for DASD files. Note that for DTFDI files access checking is already done when the ASSGN statement is being processed.

Table 14. Access Rights Required for ACB or DTF Open Processing

DASD File:	Access Right Required
ACB; MACRF defines (..,OUT)	UPD
ACB; MACRF does not define (..,OUT)	READ
DTFDA (every case)	UPD
DTFIS (every case)	UPD
DTFPH (every case)	UPD

Table 14. Access Rights Required for ACB or DTF Open Processing (continued)

DASD File:	Access Right Required
DTFDI DEVADDR=SYSIPT|SYSRDR	READ
DTFDI DEVADDR=SYSPCH|SYSLST	UPD
DTFSD TYPEFLE=INPUT	READ
DTFSD TYPEFLE=OUTPUT	UPD
DTFSD UPDATE=YES	UPD
DTFSD TYPEFLE=WORK	UPD
DTFSD TYPEFLE=WORKIN	READ
DTFSD TYPEFLE=WORKUP	UPD
DTFSD TYPEFLE=WORKMOD	UPD

Two Kinds of Access Rights

The security administrator, who is defined as type 1 user (system administrator) in the user profile, has the highest access right to all resources.

The following text refers to a user who is **not** the security administrator. For such users, access to a protected resource is controlled by one of the following access rights:

1. Universal Access Right
2. A match of an access control class.

1. **Universal Access Right**

It grants **all users** of the system a particular access right to a library, sublibrary or member (files cannot have a universal access right). A universal access right is defined in the UACC parameter of LIB, SUBLIB or MEMBER-type calls in the DTSECTAB macro. For example, the macro call

```
        DTSECTAB TYPE=MEMBER,                                        C
              NAME=IJSYSRS.SYSLIB.STDLABUP,                          C
              UACC=UPD
```

authorizes all users of the system to update the contents of member STDLABUP in the system sublibrary IJSYSRS.SYSLIB.

Access to a protected resource is allowed if the universal access right of the resource is sufficient for the requested access. In the example above, UACC=UPD is sufficient when a program attempts to read or change member STDLABUP.

2. Access by **Access Control Class**

In a resource profile, one or more of 32 access control classes can be assigned to a resource (ACC parameter of the DTSECTAB macro).

The user profiles also refer to these classes and are defined with the *Maintain User Profile* dialog. In addition, the user profile specifies which access right the user has for a particular class: CON, READ, UPD, or ALT. The ACC parameter thus defines the range of the user's authorization for the specified access class or group of classes as long as the universal access right for the resource is not sufficient.

A resource that has neither a UACC nor an access control class defined can only be accessed by the system administrator (type 1 user).

An Example of Using Access Rights

The example assumes a type 3 user with the following definitions in the corresponding user profile:

```
USERID=ENDU
PASSWORD=XB3L25
ACCESS CLASS=1-8
ACCESS RIGHT=UPD
```

The access request concerns the following resource:

```
DTSECTAB TYPE=SUBLIB,NAME=AUX.PR$302,ACC=(8,9),UACC=READ
```

The Access Control function first checks whether the resource has a sufficient universal access right. If this is not the case, it compares the user profile entry with the resource profile entry. This check is done in two steps:

1. A check for a match of the **access control class**.

   If there is a match between the access control classes of the user profile entry and the profile entry of the resource to be accessed, processing is allowed to continue. Otherwise, a security violation is indicated. This check is done for the requested resources.
2. A check for the user's **access right** in the user profile and the **type of access** attempted by a job or program.

   If the access right for this class in the user profile is sufficient for the type of access attempted, processing is allowed to continue. Otherwise, a security violation is indicated.

In the above example, user ENDU with access control class 1 through 8 is allowed to access sublibrary AUX.PR$302 due to the match on class 8. The access right is limited to UPDATE. An attempt by ENDU to ALTER (rename or delete) the sublibrary would be an access violation.

In case of an access violation, the job or user program is canceled, or execution of the function is skipped. The violation is recorded (*logged*) on the log data set if the optional program *VSE/Access Control-Logging and Reporting* is installed.

Access control classes are also used to determine whether **allowed** accesses to resources defined in DTSECTAB are to be logged. Please refer to the description of the LOG parameter on 415.

In case there is more than one match between access control classes, the higher access right becomes effective.

Diagram of Access-Checking Flow

Figure 98 on page 405 shows the concept of access authorization checking (universal access rights and type 1 user access rights are not taken into account).

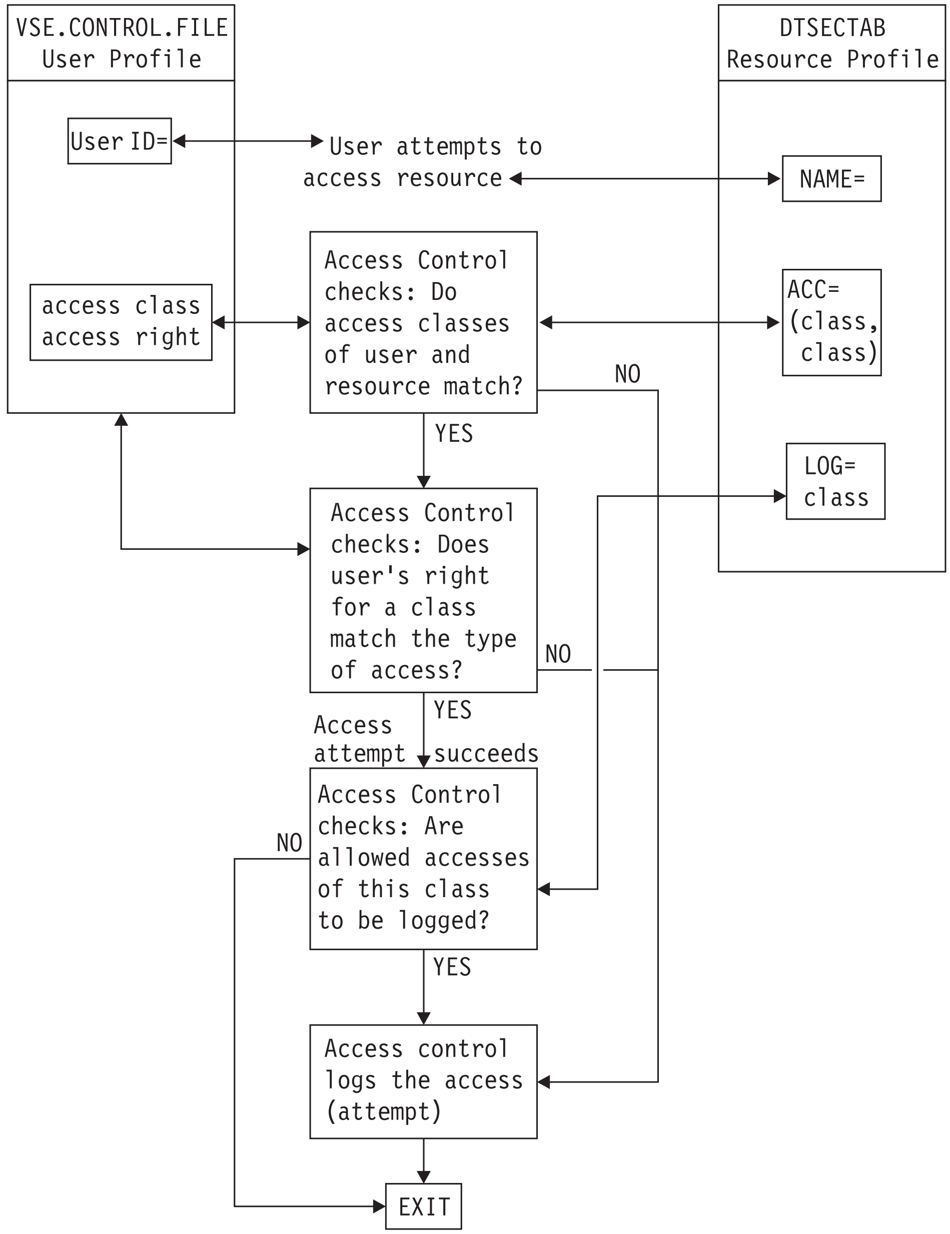


Figure 98. Access Authorization Checking with Access Control Classes Defined in DTSECTAB

Access Control for Libraries

The Access Control Function follows the hierarchy of the VSE library structure. If a library is protected, all sublibraries and members in it are protected automatically. If a library is **not** protected, its sublibraries and members **cannot** be protected.

The possession of any access right (except CON) to a library entity implies at least the same access right to the entities below it in the hierarchy. Consider this example:

```
Sublibrary REP1.DEV............ access right UPD
Member     REP1.DEV.PROG1...... access right READ
```

Member PROG1 automatically inherits the access right of UPD from sublibrary REP1.DEV. Its own specification of READ is overridden.

The Access Right of CON

The access right of CON is valid for libraries and sublibraries only. It determines whether a user may access the library/sublibrary at all, for example by a LIBDEF.

The access right of CON is **not** conferred to the next lower level. Therefore, in a library protected with CON, a sublibrary that has no entry for itself cannot be accessed.

Consider the following example:

```
Library PRD1 .............. access right CON

Sublibrary PRD1.BASE....... access right CON
Sublibrary PRD1.CONFIG..... no access right specified, or
                            not defined in DTSECTAB at all
```

The user can do a LIBDEF for sublibrary PRD1.BASE but **not** for sublibrary PRD1.CONFIG.

CON allows no higher access right such as READ or UPD. In particular, to be able to read or update members of a sublibrary that is protected with the access right of CON, you must provide profile entries in the access control table for all its members (be sure to use generic member notation). Otherwise, only the security administrator has access to these members.

Hierarchical Access Checking

For access rights other than CON, library protection is strictly hierarchical. If the user has an access right for a library, he has the same right for all sublibraries and all members therein.

When a protected resource in the library hierarchy is to be accessed, the access control checking is done at each level in the hierarchy, until a sufficient access right is found (see Figure 103 on page 441). For this reason,

- Access checking on the lower level is only done if the inherited access right from the higher level is not sufficient.
- Defining access rights at the lower level makes only sense if the lower level's right is higher than the right inherited from the higher level.

When a member is to be accessed, first the access right to the library is checked, then to the sublibrary. These rights are determined when the library and the sublibrary are accessed (for example by a LIBDEF). If their rights are not sufficient, the member itself will be checked. The user attempting the access must have at least the access right CON (connect) to the library and the sublibrary. If, for example, a user is attempting read access to the member, and has the READ access right to the **library** in which the member is stored, Access Control allows the access, and does no checking at sublibrary or member level.

The same is true if you attempt to access a protected sublibrary. In this case, you must at least have the access right CON (connect) to the library in which the sublibrary resides, and checking stops if you have a sufficient access right to that library.

Impact on Logging

Via the LOG parameter in the resource definition, you can specify for which classes successful accesses to a protected resource are to be logged. Access violations are always logged. An access granted via a universal access right is never logged.

The method of hierarchical access checking, as described before, has an impact on logging. The access to a member is only logged if access checking reaches the member level, that is: if there is no inheriting of a sufficient access right.

Consider the following example.

```
EXEC LIBR
ACCESS S=IJSYSRS.SYSLIB
LIST $IPLESA.PROC
```

- The access right for library IJSYSRS is established as the maximum of universal and (the job submitter's) individual access rights. This right (if larger than CON) is inherited by sublibrary IJSYSRS.SYSLIB.
- The access right for sublibrary IJSYSRS.SYSLIB is established as the maximum between the inherited right and the maximum of universal and individual access rights.

  This right (if larger than CON) is inherited by all members in IJSYSRS.SYSLIB.

For the

```
LIST $IPLESA.PROC
```

statement, an access right of READ is required.

- If the inherited access right is READ or higher, then no more access checking is done and access is allowed. The access is not logged.
- Else, the maximum of universal and individual access rights must be READ or higher to access the member, otherwise an access violation occurs.

  The access is logged if the UACC is not sufficient and the access class is specified in the LOG parameter.

Access Control for LIBDEF Statements

For certain functions (for example FETCH/LOAD, the linkage editor, or job control), access to sublibraries is requested by LIBDEF job control statements.

The system regards a LIBDEF statement as an attempt to access the sublibrary or sublibraries that it specifies. LIBDEF statements are of two types:

- Permanent (PERM), valid for all jobs in the partition in which they are entered;
- Temporary (TEMP), valid only within the job in which they are entered.

Access checking and the granting of access rights is done as described in the preceding topics.

Please note for **permanent** LIBDEFs:

For any sublibrary (except IJSYSRS.SYSLIB) which you intend to specify in a permanent LIBDEF statement, plus its containing library, you **must** specify a **universal access right** (UACC=CON or higher) in the resource profile.

Only the **universal** access right to the sublibrary is granted. This is because a permanent LIBDEF is still effective after completion of the job that established the LIBDEF. Another user's job can use this LIBDEF.

Please note for **temporary** LIBDEFs:

For a temporary LIBDEF statement, **individual** access rights of the user or **universal** access rights (whichever are higher) are granted.

When access to a sublibrary is via a **temporary** LIBDEF, the normal rules for checking apply. If the universal access right is sufficient, the access is allowed; if not, the individual access right is checked, and the access is allowed if this is sufficient.

This means, that a user who wants an individual access right to be granted for a sublibrary must use a temporary LIBDEF.

Access Checking for Source Library Inclusion (SLI)

Checking for the access right of READ (to a member) becomes necessary when a VSE/POWER job contains an * $$ SLI statement which includes a member from a **VSE library**.

There are two formats of * $$ SLI:

1. * $$ SLI MEM=
2. * $$ SLI MEM=... ,S=lib.sublib...

The first format specifies only the member name whereas the second format has also the **sublibrary** specified that contains the member.

Format 1 (with member name only)

In this case, the VSE/POWER partition (via LIBDEF) must have the access right of at least CON for the member's library and sublibrary. If the VSE/POWER partition's access rights are not higher than CON, **the job** must have an access right of at least READ to the member.

In the z/VSE system that is shipped to you, the startup procedure for the VSE/POWER partition contains permanent LIBDEFs. Therefore only the universal access rights are established.

It is recommended that you retain the **permanent** LIBDEFs. Keeping these universal rights low allows you to carefully set the higher access rights on member level.

Format 2 (member plus sublibrary)

In this case, no access checking against the VSE/POWER partition takes place. Rather, the job containing the SLI statement must provide the proper access right.

Special Access Checking for Librarian Commands

Normally, a user can access a member in a sublibrary if that user has the appropriate access right for the member and the connect (CON) right for the sublibrary.

However,

- Librarian commands with a **generic** member specification require the access right of READ (or higher) for the sublibrary, in addition to the appropriate access rights for the members. Alternatively, the access right of CON to a library or sublibrary and a user profile entry of READDIR=YES are sufficient for reading the respective directory.
- A Librarian TEST command specifying a sublibrary or member name always requires the access right of READ for the **library** in which the sublibrary or member resides.
- A Librarian SEARCH command with OUTPUT=FULL always requires the access right of READ for the **library** to be searched, even if only a sublibrary has been specified.

Protection of the System Library and System Sublibrary

The system library IJSYSRS and the system sublibrary IJSYSRS.SYSLIB are treated in a special way because they are accessed for the first time at IPL time before label information is available. z/VSE uses the FORMAT-1 standard disk file label for access checking.

By default, the system **library**, IJSYSRS, has the universal access right of connect (UACC=CON) while the system **sublibrary**, IJSYSRS.SYSLIB, has the universal access right of read (UACC=READ).

The **default** also applies if the (sub)library has no entry in the access control table DTSECTAB. If the (sub)library has an entry in DTSECTAB which does not specifically defines a universal access right, the universal access right is set to CON.

Therefore, regardless of what you specify in your DTSECTAB, IJSYSRS and IJSYSRS.SYSLIB have at least a universal access right of connect (UACC=CON).

The access control table DTSECTAB that is delivered with your z/VSE system defines universal access rights of connect (UACC=CON) for all system sublibraries. This allows to exercise selective control over certain programs (for example DITTO/ESA for VSE in PRD1.BASE).

Protection of PRIMARY Library and Sublibraries

During initial installation, a VSE library named PRIMARY is automatically created. This library will later contain PRIMARY.userid sublibraries for all users defined in z/VSE's control file.

PRIMARY sublibraries allow for **private user libraries** that offer a similar kind of protection as the *primary libraries* in a VSE/ICCF environment.

Accessing PRIMARY Sublibraries

The name of such a sublibrary is always PRIMARY.userid.

No entries in the access control table DTSECTAB are required for PRIMARY **sublibraries**. An entry for the PRIMARY library is sufficient and is provided in the pregenerated table of your z/VSE system.

Without explicit authorization in the access control table DTSECTAB, only the owner user-ID and a type 1 user (system administrator) can access the data stored in such a sublibrary. Note that such a user can access all resources.

The owning user automatically has the UPDATE right. The access right can be increased to ALTER by appropriate entries in DTSECTAB. It is not possible to reduce in DTSECTAB a user's UPDATE right to a sublibrary owned by this user.

For a type 2 user (programmer) and a type 3 user (operator), Access Control checks whether the user-ID of the requestor and the name of the PRIMARY sublibrary match. If user-ID and name do not match, no access is possible without appropriate DTSECTAB entries.

PRIMARY Sublibraries for Predefined Users SYSA, OPER and PROG

The above users have by default a primary sublibrary when the system is newly installed.

PRIMARY.$$C Common Sublibrary

A special sublibrary, named PRIMARY.$$C, is also available to allow data exchange among different users. Sublibrary PRIMARY.$$C is protected in the pregenerated DTSECTAB with the universal access right of update (UPD). Therefore, all users have read/write access to this sublibrary.

Access Control for Startup Procedures

Startup procedures, such as $0JCL, include // ID statements for user FORSEC for accessing protected resources. The statements are processed if the system has security on SEC=YES. They are ignored if SEC=NO is specified. The password is not necessary during system startup and is therefore omitted from the // ID statement.

For each partition, the startup procedure is loaded without access checking. Neither a universal nor an individual access right is necessary.

Startup Procedures with Access Rights of a Particular User

The following applies to startup processing for static partitions.

1. If the `// ID` statement in an ASI procedure contains no password, then password verification is skipped.
2. All access checks from now on till end-of-job (/&) are done using the information from the specified user's profile.
3. Batch jobs that are submitted to VSE/POWER inherit the user-ID for which startup is running. If desired, this user-ID propagation can be overridden by including a // ID statement with another user-ID and the respective password in the job streams or by coding this user information in the * $$ JOB statement of the submitted jobs.

Access Control and CICS Region Prefix

In an environment with more than one CICS region, the CICS prefixing allows an installation to prevent users on one CICS region to access resources of a different CICS region that has a different prefix.

The BSM supports the CICS prefixing of transaction names (see Chapter 39, “Protecting CICS Transactions with Access Control Table DTSECTXN,” on page 445).

CICS uses the user-ID under which the CICS region is running as prefix. If the system is started with SYS SEC=YES, this user-ID is the ID of the job submitter, the POWER job card, or the // ID statement. This may result in short (4 character) prefixes which are normally not very meaningful or it exposes the password. To avoid this, special task user-IDs are provided by BSM. They should be specified in the // ID statement without password. Jobs, which have such a user-ID in the ID statement have to be submitted from an administrator (type 1 user-ID).

IBM ships three special task user-IDs in the VSE control file:
- DBDCCICS (for use with CICS)
- PRODCICS (for use with CICS)
- VCSRV (for use with the VSE Connector Server and TCP/IP)

The initial name that is specified in the *Maintain User Profile* dialog is DUMMY. A logon to the Interactive Interface is *not* possible using any of these user-IDs.

If no DTSECTAB security is active (SYS SEC=NO), the BSM takes the user-ID from the // ID statement in the CICS startup job. This // ID statement does not require a password. If no // ID statement is found, the user-ID FORSEC is used as default.

System Phases, B-Transients, Link Area, SVA and LTA

In the pregenerated access control table DTSECTAB, the entries

```
DTSECTAB TYPE=SUBLIB,                                          C
      NAME=IJSYSRS.SYSLIB,                                     C
      UACC=CON
DTSECTAB TYPE=MEMBER,                                          C
      NAME=IJSYSRS.SYSLIB.*,                                   C
      UACC=READ
```

ensure that system phases can be executed at any time and for any user. If for some reason you lower the UACC in the second call (TYPE=MEMBER), you should increase the UACC in the first call (TYPE=SUBLIB) to at least READ.

Considerations for B-Transients

B-Transients are a special kind of system phases. In a system with security active, they can be loaded from protected libraries only. The attempt to load a B-transient from an unprotected library would cause an access violation.

Likewise, **in a system with security active, B-transients can be cataloged only in protected libraries**.

Libraries that contain B-transients of the z/VSE **base programs** are automatically protected by appropriate entries in the pregenerated access control table DTSECTAB.

Considerations for Link Area, SVA, and LTA

The Link Area (used when link-editing with OPTION LINK), the Shared Virtual Area (SVA), and the Logical Transient Area (LTA) cannot be protected explicitly by entries in DTSECTAB.

The **Link Area** is considered as an unprotected library. This means that if OPTION LINK is used, and a name beginning with $$B... appears in the PHASE statement, an access violation occurs.

For the execution of phases, the **SVA** is regarded as a resource with the universal access right READ.

The **LTA** is activated and used by special phases, the B-transients. The rules for B-transients apply (see the preceding topic). B-transients which do not conform to these rules cannot be activated, and cause an access violation when called.

Chapter 34. DTSECTAB Macro: Syntax and Examples

This chapter describes the syntax of the DTSECTAB macro and includes many examples of its use.

The DTSECTAB macro is used to build the access control table DTSECTAB and to define resource entries in it. You must specify one entry for each resource to be protected. If generic specification of resource names is used, an arbitrary number of resources can be protected with one table entry.

For each entry to be specified a DTSECTAB macro call is required. The first macro call must include the SUBTYPE=INITIAL parameter, the last one the SUBTYPE=FINAL parameter.

This chapter contains these main topics:

- “Format of DTSECTAB Macro for Defining Resources”
- “Generic Protection of Resources” on page 415
- “Examples of DTSECTAB Resource Entries” on page 416

Related Topics:

For details of ...	Refer to ...
how to activate VSE security based on table DTSECTAB	Chapter 32, “Customizing/Activating DTSECTAB-Based Security,” on page 391.
the contents of DTSECTAB (as delivered by IBM)	“Content of Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59)” on page 395.

Format of DTSECTAB Macro for Defining Resources

```
►►──DTSECTAB TYPE=──────────────────────────────────────────────────────────►

►──┬─FILE─,NAME=──┬─volid.fileid─┬──────────────────────────────┬───────────►
   │              ├─volid.*──────┤                              │
   │              └─*.fileid─────┘                              │
   ├─LIB─,NAME=──┬─volid.fileid.libname─┬──┬───────────────┬────┤
   │             └─*.fileid.libname─────┘  └─,UACC=right───┘    │
   ├─SUBLIB─,NAME=──┬─libname.sublibname─┬──┬───────────────┬───┤
   │                └─libname.*──────────┘  └─,UACC=right───┘   │
   └─MEMBER─,NAME=──┬─libname.sublibname.membername─┬──┬───────────────┬─┘
                    └─libname.sublibname.*──────────┘  └─,UACC=right───┘

►──┬───────────────────────────────────┬──┬───────────────────────────────────┬──►
   │         ┌─,───────────────┐       │  │         ┌─,───────────────┐       │
   │         ▼                 │       │  │         ▼                 │       │
   └─,ACC=(──┬─class───────┬───┴─)─────┘  └─,LOG=(──┬─class───────┬───┴─)─────┘
             └─class-class─┘                        └─class-class─┘
```

```
►─┬─────────────────────────┬──────────────────────────────────────►◄
  └─,SUBTYPE=─┬─INITIAL─┬───┘
              └─FINAL───┘
```

Mandatory Parameters:

TYPE Specifies for which type of resource this profile is intended. The possible values are:

FILE for a file profile,

LIB for a library profile,

SUBLIB
for a sublibrary profile

MEMBER
for a member profile.

NAME
Specifies the name of the resource. The format of the value must correspond to the value of the TYPE parameter, as follows:

TYPE **Format of NAME Value**

FILE volid.fileid

LIB volid.fileid.libname

SUBLIB
libname.sublibname

MEMBER
libname.sublibname.membername

The elements of the resource names are:

volid The 1 to 6-character volume identifier of the volume on which the file or library resides. For files on tape, for files or libraries in VSE/VSAM-managed space, and for VSE/VSAM files code an asterisk (*) for **volid**.

fileid The 1 to 44-character fileid (the second operand of the DLBL statement) or the 17-byte tape-fileid (TLBL) as defined for the file or library.

libname
The library name used in the librarian DEFINE command which defined the library.

sublibname
The sublibrary name used in the librarian DEFINE command which defined the sublibrary.

membername
The member name under which the member was cataloged. The member type is not used in the access control table. Members of all types having the same name share the same access control profile.

Note:

1. A library is, in practice, a file which has been put under the control of the librarian program using a librarian DEFINE command. Be careful not to submit a FILE-type macro in addition to the LIB-type macro for a library.
2. “libname” must be used only once in the system, so that the NAME parameters for sublibrary entries can be made unique.
3. The combination “volid.fileid” must be used only once in the system.
4. The “file-id” must be identical to the file-id in the DLBL or TLBL statement. It may contain periods; the access control function does not confuse these with the periods delimiting the file-id from the other operands.
5. Specification and padding of “volid” and “file-id” are as described for the “DLBL”, “TLBL”, and “EXTENT” statements in the IBM manual *z/VSE System Control Statements*.
6. For a multi-extent, multi-volume library residing in non-VSAM space, only one entry with TYPE=LIB is needed. The VOLID must be of the volume on which the first extent of the library resides.

Optional Parameters:

ACC Specifies the access control class(es) to be assigned to the named resource. For **class**, specify a decimal number from 1 to 32. You can also specify a list of classes, for example (1,3,5), or a range of classes, for example (1-3). Access is allowed to those users who have one or more of the specified classes in the ACC parameter of their user profile.

UACC For **libraries, sublibraries and members only:** Specifies an access right which **all** users of the system are to have to the named resource. This right is granted irrespective of the classes specified in the users' profiles. The possible values for **right** are:

ALT, UPD, READ and CON.

See Table 13 on page 401 for explanations of the access rights.

LOG Specifies which class(es) of **successful accesses** are to be logged. Access **violations** are always logged.

The same syntax as for the ACC parameter applies. Only classes that appear in the ACC parameter are meaningful here.

SUBTYPE
This parameter is required in the first and the last call of the DTSECTAB macro, and is not allowed in the intermediate calls. The value **INITIAL** must be used in the first call, and the value **FINAL** in the last call.

Generic Protection of Resources

The rules for single resources, as explained above, also apply to groups of resources. In the DTSECTAB macro, you can make a **generic** specification in the NAME parameter. To do this, code an asterisk (*) for the **last** element of the resource name. For example, ...NAME=000111.*... would apply the access control profile to **all files** on the volume with the serial number 000111. Coding ...NAME=LIB1.SUBA.*... would apply the profile to **all members** in sublibrary SUBA of library LIB1.

Note that the asterisk (*) is a reserved character and must not be used in resource names. Therefore, do not use the asterisk in a file name or in a volume serial number.

To apply an access control profile to, for example, all files on volume 000111 whose file-ids begin with PAY..., code:

```
NAME=000111.PAY*
```

The common part of the resource names can be any length, up to one character less than the allowed maximum length. In other words, the generic name, including the asterisk, is subject to the same length restrictions as a normal resource name.

Note: You can give a group of members the same access class(es) by generic specification in the DTSECTAB macro. However, this is not sufficient to access these members with a generic specification in a Librarian command. You still need the access right of READ or higher to the sublibrary or, for reading directory information, the access right of CON plus READDIR=YES in the user profile (see "Special Access Checking for Librarian Commands" on page 409).

Generic protection is not available at library level.

A special case arises when there are two or more generic specifications starting with the same characters. For example, two entries of type FILE might have the parameters:

```
NAME=000999.PAY*,ACC=1
```

and

```
NAME=000999.PAYR*,ACC=2
```

If an access to the file PAYROLL on volume 000999 is attempted, the Access Control Function will check for (and allow) access class 2 and reject access class 1. The rule is that the longest generic name which matches with the name of the accessed resource is used.

If the full name of the resource to be accessed (in this example, PAYROLL) is found in an entry, this entry is, of course, used for checking. Or take as another example the two specifications

```
IJSYSRS.SYSLIB.*,UACC=READ
IJSYSRS.SYSLIB.DITTO
```

All members of the SYSLIB sublibrary have the universal access right of READ, but the program DITTO is excluded from this rule.

Examples of DTSECTAB Resource Entries

File Entries:

```
DTSECTAB TYPE=FILE,                X
      NAME=999999.PAYROLL,         X
      ACC=(16),                    X
      LOG=(16)
```

This entry can be used to protect non-VSAM disk files. The protected name consists of the six-character volume-ID of the storage medium used, followed by the 44-byte file-id, taken from the DLBL statement.

```
DTSECTAB TYPE=FILE,                X
      NAME=*.PAYROLL,              X
      ACC=(16),                    X
      LOG=(16)
```

This entry can be used to protect tape files or SAM files in VSE/VSAM-managed space. The protected resource name must consist of an asterisk (*) followed by the 44-byte (disk) or 17-byte (tape) file-id, taken from the DLBL or TLBL statement, respectively.

In both examples, only users with the access class 16 or AUTH=YES in their access control profile can access the file PAYROLL. Successful accesses will be logged (LOG parameter).

Library Entries:

```
DTSECTAB TYPE=LIB,                 X
      NAME=888888.P.C.TEST.LIB1,   X
      ACC=(1-8,32),                X
      LOG=(1-8)
```

Library entries are used for protecting libraries. Protecting a library means also protecting the sublibraries and members in it. The name consists of the 6-character volume-ID of the storage medium used, followed by the file-id from the DLBL statement defining the disk file, and the filename from the current DLBL statement associated with the library.

Users with access classes in the range 1 to 8 and 32 can access this library (ACC parameter), and all entities in it. Accesses made via classes 1 to 8 will be logged.

It is possible to allow users who need to access only some of the sublibraries in this library to do so. This is done in the following example by using the universal access right connect (UACC=CON):

```
DTSECTAB TYPE=LIB,                 X
      NAME=888888.TEST.LIB1,       X
      ACC=(1-8,32),                X
      UACC=CON,                    X
      LOG=(1-8)
```

If a sublibrary in this library is given, for example, the access class 10, a user with only access class 10 in his profile could access that sublibrary (but no other sublibrary in this library).

Note: The examples above are for libraries in non-VSAM space. For libraries in VSE/VSAM-managed space, code an asterisk (*) in place of the volume identifier.

Sublibrary Entries:

```
DTSECTAB TYPE=SUBLIB,              X
      NAME=LIB1.SUBA,              X
      ACC=(4-12),                  X
      LOG=(4-12)
```

Sublibrary entries are used for defining access rights to sublibraries. The name consists of the name of the library, followed by the name of the sublibrary. The sublibrary name is the name used in the librarian DEFINE command which created the sublibrary.

Users with access classes in the range 4 to 12 can access this sublibrary (ACC parameter), and all members in it. Accesses made via classes 4 to 12 will be logged.

It is possible to allow users who need to access only some of the members in this sublibrary to do so. This is done in this example by using the universal access right connect (UACC=CON):

```
DTSECTAB TYPE=SUBLIB,               X
      NAME=LIB1.SUBA,               X
      ACC=(4-12),                   X
      UACC=CON,                     X
      LOG=(4-12)
```

If a member is given the access class, for example, 3, a user with only access class 3 in the user profile could access that member.

Member Entries:

```
DTSECTAB TYPE=MEMBER,                X
      NAME=LIB1.SUBA.PAY*,           X
      ACC=(9-15),                    X
      LOG=(9-10,15)

DTSECTAB TYPE=MEMBER,                X
      NAME=LIB1.SUBX.DITTO,          X
      SUBTYPE=FINAL
```

Member entries define access rights to members in sublibraries.

The value in the NAME parameter consists of the library name, sublibrary name and member name. The member type is not specified. All types of member (phases, object modules, procedures and so on) with the same name have the same access profile. This means, for example, that if a programmer has the “alter” access right to the source book of a program, he will automatically have the same right to the object module and the phase form of the program. The programmer can also catalog a procedure of the same name to set up the job control statements for the program. The first example has a generic name specification. This means that the access profile is the same for all source books, object modules, phases and procedures in this sublibrary whose names begin with PAY.

Note that generic protection does not necessarily allow generic access to members in Librarian commands (see “Generic Protection of Resources” on page 415).

SUBTYPE=FINAL, as in the second example, must be specified for the last macro call in the assembler input stream.

Example of DTSECTAB Entries for Library Control

The following DTSECTAB entries demonstrate the various protection levels available for controlling the access to libraries, sublibraries, and their members. The example is based on sublibrary REP1.BASE, which is located in VSE/VSAM-managed space. This sublibrary is assumed to have stored a phase with the name PROG1.

Four DTSECTAB entries are shown followed by explanatory text.

1. Protect the library:
```
DTSECTAB TYPE=LIB,                                X
         NAME=*.VSE.REP1.LIBRARY.REP1,            X
         UACC=CON
```
2. Protect the sublibrary:
```
DTSECTAB TYPE=SUBLIB,                             X
         NAME=REP1.BASE,                          X
         UACC=CON
```
3. Allow each user to read all members:
```
DTSECTAB TYPE=MEMBER,                             X
         NAME=REP1.BASE.*,                        X
         UACC=READ
```
4. Restrict access to PROG1 to the security administrator:
```
DTSECTAB TYPE=MEMBER,                             X
         NAME=REP1.BASE.PROG1                     X
```

The **first two** entries protect sublibrary REP1.BASE and give all users the access right of CON. (By substituting or adding to UACC=CON in the entry for the sublibrary an ACC= statement, you could give selected users higher access rights to REP1.BASE).

Entry **three** allows all users read access to all members in sublibrary REP1.BASE.

Entry **four** restricts access to PROG1 to the security administrator. The member is protected against any access from other users because neither a universal nor an individual access right is defined.

Entry **four** could be expanded to look like this:
```
DTSECTAB TYPE=MEMBER,                             X
         NAME=REP1.BASE.PROG1,                    X
         ACC=(10,12)
```

This restricts access to PROG1 to selected users (other than a security administrator). These users must have access control classes 10 or 12 defined in their user profile.

Chapter 35. Propagation of VSE/POWER Security Identification

This chapter describes how security information is propagated between one or more **VSE/POWER batch environments**.

A user who submits a batch job from the z/VSE Interactive Interface (or from a VSE/ICCF terminal or through the SEND command from a workstation) does not have to pass user-ID and password for the job submitted. The system automatically makes sure that the job will run with the user's profile information. Only if the job is to run with another user profile, the submitter must specify the other user's security identification.

This chapter contains these main topics:

VSE/POWER Authenticated Jobs

A job is considered authenticated if the user-ID and password of the submitter were checked successfully before the job was submitted. This type of job thus is called an **authenticated job**. Only the **user-ID** of the submitter is associated with the job.

An authenticated job retains its status even when being transferred via
- A PNET network to another VSE/POWER system
- VSE/POWER shared spooling to another system
- A POFFLOAD tape to another system

under the condition that the originating system and the executing system belong to the **same security zone**. (The concept of *security zone* is described below, under “Propagating Security Identification between VSE/POWER Subsystems” on page 422.)

The authenticated job is submitted internally via the
- VSE/POWER Spool Access Support or Extended Device Support (XPCC interface), or
- VSE/POWER spooled punch output (* $$ PUN DISP=I).

The user-ID is propagated from the submitting job into the authenticated job.

Examples:
- A job that is generated as * $$ PUN DISP=I output inherits the security identification from the job which generates the punch output.
- The utility DTRIINIT (which loads programs into the RDR queue via Spool Access Support) propagates the security identification of the utility job to the jobs that are being loaded.

If the submitting job contains a // ID statement, the propagated user-ID is taken from that statement. Note that when using a VSE/POWER JECL statement * $$

PUN with DISP=I, the // ID statement must precede the * $$ PUN. The submitting job may override the propagated security identification by punching a * $$ JOB statement which contains user-ID and password for the job being created.

Note that a job runs even if the job's user-ID does not exist in the VSE.CONTROL.FILE, but can access resources with a (sufficient) universal access right, or which are unprotected.

From a system with security inactive (SEC=NO), propagation of security identification is not possible.

Propagating Security Identification between VSE/POWER Subsystems

The CPU on which a batch job is submitted need not be the same as the CPU which is to execute the job. This is the case when a job is submitted via

- A PNET network to another VSE/POWER system
- VSE/POWER shared spooling to another system
- A POFFLOAD tape to another system

For the propagation of security information to an authenticated job, it is important whether or not the two systems belong to the same *security zone*. When an authenticated job runs in a security zone other than the originating zone, it runs without security authorization.

Security Zone

A security zone consists of a group of systems where a given user-ID that occurs on any of these systems identifies the same user.

If all user profiles on the submitting system and the executing system are unique (that is: a user-ID identifies the same person on each system), the submitter needs not be concerned about passing security identification to the other system because the (local) user-ID does not belong to another person on the other system. The security administrator should define the two systems as belonging to the same **security zone**.

Within one security zone, authenticated jobs keep their status. The security zone is defined to VSE/POWER in the SECNODE parameter of one of the following:

- the POWER generation macro:
  ```
  POWER SECNODE=zonename
  ```
- VSE/POWER SET statement:
  ```
  SET SECNODE=zonename
  ```

For more details refer to the IBM manual *VSE/POWER Administration and Operation* under “POWER Generation Macro” and “SET: Setting VSE/POWER Startup Control Values”.

Equal SECNODE names on multiple systems mean: each user-ID describes the same person on those systems where the user-ID is defined in the VSE.CONTROL.FILE.

If, on the other hand, the security administrator cannot guarantee that user profiles are unique on the two systems, different SECNODE names must be defined on the two systems. In this case, a submitter must explicitly pass along security identification in the * $$ JOB statement in order to access protected resources with insufficient universal access rights in another security zone. This ensures that the job does not run with the user profile of a different person.

General Rules for VSE/POWER Subsystems

The preceding discussion assumed that the submitting system and the executing system are both IPLed with security active (SEC=YES).

Consider the following combinations:

Submitting System SEC=YES - Executing System SEC=YES:

- If the SECNODE IDs are equal, the submitting system propagates the security identification to the executing system unless it is explicitly overridden by the submitter.
- If the SECNODE IDs are not equal, the submitter should explicitly supply security information in the * $$ JOB statement. The implicitly propagated security identification is ignored on the executing system, and further propagation cannot take place.

Submitting System SEC=YES - Executing System SEC=NO:

The security information transferred by the submitting system is ignored.

Submitting System SEC=NO - Executing System SEC=YES:

No security information is implicitly propagated with the submitted job. The submitter should explicitly supply security information in the * $$ JOB statement.

Submitting System SEC=NO - Executing System SEC=NO:

No security information is propagated with the submitted job.

The preceding text outlined the general rules that govern security authorization between systems. The following topics deal with two special VSE/POWER environments:

- Shared Spooling
- Job/File transfer from one system to another.

Security Checking under VSE/POWER Shared Spooling

In a VSE/POWER shared spooling environment, typically the security zone would be equal for all sharing systems. This allows users to have their jobs run on **any** of these systems. If for some special reason any of the shared systems has its own security zone, VSE/POWER's job scheduling will take this into account. VSE/POWER attempts to execute authenticated jobs on systems with matching SECNODE names.

For detailed information (in particular about the SHARED parameter of the POWER macro) refer to the IBM manual *VSE/POWER Administration and Operation* under “POWER Generation Macro”.

Transfer of Jobs or Files/Members between Systems

For a VSE/POWER PNET network, z/VSE provides dialogs for:
- Submitting a job to another system in the network
- Transferring/retrieving VSAM files or VSE/ICCF members.

When the local system and/or the remote system run with security active, access control also needs to be considered. The system in which you use the dialog and submit the job is called *local system*. In particular, the propagation of security information is affected by:
- Differing security zones
- The possible participation of a *backlevel system* (this can be a VSE system prior to VSE/ESA 1.3, or a VSE/SP system).

Under “Submitting Jobs to Other Systems”, the IBM manual *z/VSE SNA Networking Support* provides details and describes how you work with the dialog.

Chapter 36. Operating a DTSECTAB-Based Security System

This chapter describes various security items that you should consider in a z/VSE system with DTSECTAB-based (batch) security "active".

It consists of these main topics:

Related Topics:

For details of ...	Refer to ...
how to activate VSE security based on table DTSECTAB	Chapter 32, "Customizing/Activating DTSECTAB-Based Security," on page 391.
the contents of DTSECTAB (as delivered by IBM)	"Content of Pregenerated DTSECTAB (DTSECTRC in VSE/ICCF Library 59)" on page 395.
the syntax of the DTSECTAB macro, and examples of its use	Chapter 34, "DTSECTAB Macro: Syntax and Examples," on page 413.

Some General Rules

- The console must be physically protected as a **physical object** because anybody with physical access to the console can IPL the system with SEC=NO.
- Do not place jobs that access secured resources into the RDR queue while batch security is not yet active. Such jobs could fail after you IPL your system with batch security active. This is because the propagation of security information (user-ID!) does not happen if batch security is not active.

  To avoid this potential problem, place explicit security information into the job.
- When using VSE/ICCF via the Interactive Interface, passwords may differ between the z/VSE Interactive Interface and VSE/ICCF without causing any harm. However, when invoking the VSE/ICCF utility DTSBATCH from the console, you must know the VSE/ICCF password.

Avoiding Startup Problems

You can start up your system even after a problem came up. Interrupt IPL processing with STOP=SVA and specify SEC=NO in the IPL SYS command. This lets your system work without batch security active.

Refer to the IBM manual *z/VSE Guide to System Functions* under "Interrupt and Restart the IPL Process" for information on how to interrupt IPL processing.

For recovery reasons it might be necessary to initialize your system without security support. In this case specify SEC=RECOVER in the IPL SYS command.

Performance Considerations

The impact on performance when using the z/VSE security support depends on how many resources are protected, how frequently they are accessed and by how many users, and whether all accesses are to be logged, or only the attempted violations.

Performance can be improved by efficient use of universal access rights. For example, a sublibrary whose members can be read (called for program execution) by anybody suggests itself for being protected only by universal access rights. In this case, access checking ends at sublibrary level; no time is wasted by access checking for each and every member.

Tape Handling

Tapes can be excluded from resource protection by specifying NOTAPE in the IPL SYS parameter SEC:

```
SEC=(YES,NOTAPE)
```

This is appropriate when, for example, you have a "Tape Manager" program installed that provides security checking for tapes.

In a system where tapes are part of the resource protection scheme, only standard-labeled tapes can be protected, and only under the following restriction. Specification of REWIND=NORWD in the associated DTFMT is allowed only if the tape is positioned at load point. Therefore, the tapes must be single-file. Otherwise the system could not verify whether the volume is on the tape.

In a system with batch security active, the following operational rules should be observed:

- The REWIND option must be specified for the particular magnetic tape volumes in the following VSE/VSAM commands: EXPORT, EXPORTRA, IMPORT, IMPORTRA, PRINT and REPRO. This means that multifile volumes on unlabeled tapes are not supported.
- The operator is not allowed to enter IGNORE as a reply to certain warning messages that relate to tape processing (these messages have numbers of type 41xx).

Controlling the Security Server Partition

The Security Server uses the VSE.CONTROL.FILE as data repository for user profiles. The Security Server runs per default in the **FB** partition.

For controlling the Security Server partition, specific MSG commands are available. These commands can be entered from the system console through:

```
MSG xx,DATA=command
```

where *xx* is the partition where your security server runs.

You can display a list of all available commands by entering at the console:

```
msg nn,data=? (or HELP or blank)
```

For further details of these commands, refer to the manual *z/VSE Operation*.

If you have installed an External Security Manager (ESM), refer to the documentation shipped with this product.

Chapter 37. Additional z/VSE Data Protection Facilities

This chapter describes the **standard facilities** for protecting data that are provided by z/VSE.

It contains these main topics:

Using the IPL Exit to Check After IPL

After the IPL procedure has been completed, control is passed to the phase $SYSOPEN. This phase, a dummy phase in the IBM-supplied system sublibrary, can be replaced by a user exit routine to perform various checks that may be important for the security and integrity of the system. For more information on the IPL exit refer to the IBM manual *z/VSE Guide to System Functions* under "Writing an IPL Exit Routine".

Note: You can use the IPL Exit, for example, to perform integrity and security checks. You might wish to check if the correct system and data packs have been mounted.

Using the Job Control Exit to Check Job Control Statements

This standard facility can be used for access control, but it requires a great deal of programming effort. It is better to use the access control table DTSECTAB for overall data protection in the system. You may, however, want to use the exit in special cases.

The job control exit transfers control to the phase $JOBEXIT each time a job control statement has been read. $JOBEXIT is a dummy phase in the IBM-supplied system sublibrary. It is loaded into the SVA automatically at IPL. You can replace the dummy phase by a user exit routine to perform various access control checks.

For example, to check the usage authorization for a program named PRIVLGE, the exit routine could be designed to examine the // EXEC statement for a specific code in the comment field as shown below:

```
User specification:
   // EXEC PRIVLGE   (no code specified as a comment)

Replacement by the user exit routine:
   // EXEC ERROR1
```

where program ERROR1 might simply issue a message indicating that the job cannot be processed due to a missing usage code.

The routine might also be used to examine an installation-defined identification code in the JOB statement, and compare this code with an access code associated with a file to be accessed.

The exit routine is not allowed to perform any I/O operations, to issue any SVCs, or to cancel the job.

For more information on the job control exit refer to the IBM manual *z/VSE Guide to System Functions* under “Writing a Job Control Exit Routine”. This topic also covers the setup of **multiple job control exits**.

Using Labeling to Identify/Date Files

Labeling helps to ensure that the correct data is mounted for processing, and assists in protecting data against
- inadvertent destruction, and
- unauthorized access and usage.

This protection is provided for files stored on magnetic tape or disk, where each file is identified by one or more file labels. In addition, each volume of data is identified by one volume label.

Volume and file labels are mandatory for disks, and optional for tapes. For security reasons, however, it is strongly recommended to label all tapes. Note, however, that some IBM programs require tapes without standard labels.

The TLBL statement is used for specifying label information for a magnetic tape, the DLBL and EXTENT statements for specifying it for a disk device. The VSE system label processing routines check whether the correct volume is mounted, and whether the retention period or expiration date has been reached. This protects files from being overwritten and destroyed prematurely.

The contents of the label must be specified when a file is created. The same label information must be available when the file is processed, to enable the system to compare the actual label of the data being processed with the label information submitted by the user. If a mismatch is detected the job is terminated.

For more information on labeling refer to the manual *z/VSE Guide to System Functions* under “Job Control for Label Information”.

Using Data Secured Files to Protect Files on Disk

The DSF operand in the DLBL statement indicates that a data secured file is to be created or processed. If a data secured file is to be accessed, a warning message is issued on the console. The operator then has to decide whether this file can be accessed by the program causing the message, and enter a reply at the console to allow the access. All these warning messages would make up a record of file accesses. While this method may have provided sufficient protection of and privacy for an installation's data in the past, it may not meet the protection and privacy standards of the future. Using the Access Control Function of VSE is the better method.

Using DASD File Protection to Protect Files on Disk

The DASD File Protection facility prevents programs from writing data outside the limits of their disk files. This might happen if, for example, a randomizing algorithm produces an unexpected disk address which is outside the file limits. Other files on a disk volume are thus protected against unintended destruction. However, if two disk files have been opened in the same partition and use the same programmer logical unit, these two files are not protected against destroying each other's data.

Disk file protection is activated if the DASDFP=YES operand is used in the IPL SYS command. DASDFP=YES is set in the IBM provided setting.

Using the Track Hold Option to Prevent Concurrent Updates

This facility is available only for disk volumes not under control of VSE/VSAM. In a multiprogramming environment, it prevents two or more programs from concurrently updating the same record of a track on a CKD-type disk volume, or the same record within a block on an FBA-type disk volume. In an installation that makes effective use of this facility, all of an installation's programs should specify track hold in their DTFs.

To specify the TRKHLD parameter:
1. Select Fast Path **242** (*Tailor IPL Procedure*).
2. Select *Modify SYS command parameters*.
3. Enter a value in the TRKHLD field.

Using Lock Management to Lock Resources Using Assembler Macros

In a multitasking environment, a mechanism is needed to prevent a task from using the resources of another task in an uncontrolled way, so as to avoid the destruction and erroneous updating of data. The lock management protects user-defined and system resources against concurrent use by different tasks in different partitions on one or more servers. Two levels of sharing are available when using the IBM-supplied LOCK and UNLOCK macros:
- Exclusive usage of a resource.
- Shared usage of a resource.

The following resources may be protected:

Files, libraries, catalogs, disk volumes and control blocks.

The resources are defined by symbolic names. Any symbolic name may be used; however, a naming convention should be established for the installation, and should be adhered to by all programmers using the LOCK and UNLOCK macros. A file name, a volume-id (VOLID), or a disk file begin address are examples of symbolic names.

The DTL and GENDTL macros are available to define a resource for share control. The DTL macro builds a lock control block, which the operating system needs to control the sharing of the particular resource during assembly; the GENDTL macro builds this block dynamically during execution.

Once the lock control block is defined for a resource, the operating system can efficiently control exclusive or shared access to the resource in accordance with the DTL or GENDTL macro. The MODDTL macro allows a lock control block to be modified dynamically.

A successful lock request (via the LOCK macro) means that the resource is locked for the task or partition issuing the request. With the UNLOCK macro the program can either release the resource completely, or in conjunction with the MODDTL macro, weaken the lock control from "exclusive control" to a shared status.

For more information on the various macros refer to the manuals *z/VSE System Macros Reference* and *z/VSE System Macros User's Guide*.

Protecting VSE/VSAM Files via Passwords

VSE/VSAM allows you to define passwords for accessing VSE/VSAM objects like:

- Clusters
- Alternate Indexes
- Components (Data and Index)
- Paths
- Catalogs

To gain access to a protected object, a program or the operator must provide the password defined for it. You define passwords with the Access Method Services DEFINE command. Passwords can be defined for different levels of access:

1. Read access
2. Update access
3. Control-interval access
4. Full access

Moreover, you can supply a user security-verification routine to double-check the authority of a program accessing a file.

Note: The job streams created by the Interactive Interface of z/VSE (for File and Catalog Management) do not include VSE/VSAM data protection parameters. If you want to define passwords, for example, you must edit the job streams and add the required password parameter yourself.

For a detailed description of VSE/VSAM data protection, refer to the manual *VSE/VSAM User's Guide and Application Programming* under "Data Protection and Integrity Options." For a description of the DEFINE command refer to the manual *VSE/VSAM Commands* under "Defining a Catalog." In a z/VSE user profile you can assign the default catalog a user is allowed to access. In addition, you can assign authority for managing VSE/VSAM files or catalogs or both. Refer to Chapter 25, "Maintaining User Profiles via BSM Dialogs," on page 289 for details about maintaining user profiles.

Chapter 38. Logging/Reporting Security Events

This chapter describes how security events can be logged and reported.

It contains these main topics:

Logging and Creating Reports Of Security-Related SMF Records

This topic describes how you use *System Management Facility* (SMF) records together with the *Data Management Facility* (DMF) to log access attempts to resources protected by the:

- BSM Control File (for CICS and other resources),
- DTSECTAB table (for batch-related security).

SMF is the part of the z/OS® operating system that collects records data in an installation in order to provide system and subsystem information. SMF is not available on z/VSE. However, z/VSE's *Basic Security Manager* (BSM) uses the layout of SMF records for security events:

1. The CICS Transaction Server issues a RACROUTE request each time a user wishes to access a resource.
2. The BSM processes these RACROUTE requests and creates SMF records.
3. The DMF (supplied with the CICS Transaction Server for VSE/ESA) collects and stores these SMF records.

If you wish to collect and store SMF records for CICS-related security only, follow the instructions provided in "Configuring Your System to Use the DMF" on page 434.

If you wish to collect and store SMF records for CICS-related security *and* batch-related security, first follow the instructions provided in "Using SMF/DMF to Log Access Attempts to DTSECTAB Resources" before those provided in "Configuring Your System to Use the DMF" on page 434.

Using SMF/DMF to Log Access Attempts to DTSECTAB Resources

If you wish to use SMF records together with the DMF to log access attempts to *DTSECTAB resources*, you must:

1. Obtain a copy of skeleton SKSECLOG from VSE/ICCF library 59.
2. Use SKSECLOG to build the $SVALOG.PHASE.
3. Catalog $SVALOG.PHASE in library IJSYSRS.SYSLIB.

z/VSE uses the member $SVALOG.PHASE to load DSPLLOG.PHASE into the SVA and store the address in SCYVECTB.

There are two versions of the member DSPLLOG.PHASE:

- One version of DSPLLOG.PHASE is active when the *BSM* is used to log access attempts to DTSECTAB resources via SMF records.
- Another version of DSPLLOG.PHASE is distributed with the *VSE/ACLR* program. This version is also used to log access attempts to DTSECTAB resource via its *own format* (as described in "Using VSE/ACLR to Log/Report Access Attempts to DTSECTAB" on page 439).

Because the same member name ($SVALOG.PHASE) is used for both of the above options, this ensures that *either* BSM SMF logging *or* VSE/ACLR logging can be active at any one time.

SMF records and the DMF are used in this way:
1. Member DSPLLOG.PHASE (that is supplied with the BSM) sends the SMF records to the BSM Security Server running in partition FB.
2. The BSM Security Server transfers the SMF records to the DMF.
3. The DMF writes the SMF records to the VSAM data set.

The resource manager can check accesses to DTSECTAB resources using either the:
- SECHECK macro.
- RACROUTE macro.

If a RACROUTE macro is used, up to *four* records can be written to the logging file. These four entries report on access to the library, sublibrary, member, and RACROUTE processing. You should take this into account when reading reports of such accesses to DTSECTAB resources.

Configuring Your System to Use the DMF

To use the DMF, you must:
1. Configure the DMF and its utilities to meet your requirements. The tasks that you must complete include creating the data sets required by the DMF, and configuring the DMF dump utility DFHDFOU. The DMF dump utility processes the data sets used by DMF, and allows you to selectively copy records from the DMF data sets to tape/disk for subsequent processing. For details of how to perform these tasks, refer to the manual *CICS Transaction Server for VSE/ESA, Operations and Utilities*, SC33-1654.
2. Ensure you have specified parameter `OS390` in the startup procedure for FB ($BJCL.PROC):
   ```
   // EXEC BSTPSTS,DSPACE=3M,OS390
   ```
   Parameter `OS390` is required to allow the security server to communicate with the DMF partition.
3. Take copies of jobs DFHDMFSP (to prepare the DMF) and SKDMFST (to start the DMF) from ICCF library 59, and tailor them for your environment.
4. Start the DMF *as quickly as possible* after VSE/POWER is active, so that the DMF can begin to collect data. To start DMF, you can add the statement:
   ```
   PWR PRELEASE RDR,DMFSTART
   ```
   to the USERBG.PROC *before* CICS or other jobs are started.

   Note: If you are collecting SMF records for DTSECTAB, for performance reasons you are recommended to:
   - Run the DMF in class **Z** with only one job active in this partition.
   - Increase the priority of class Z using, for example:

```
PRTY BG,FA,F9,F8,F6,F5,F4,F2,F7,FB=Z,F3,F1
```

Overview of the DMF Logging and Reporting Process

Figure 99 provides an overview of how the DMF is used to log records and create reports.

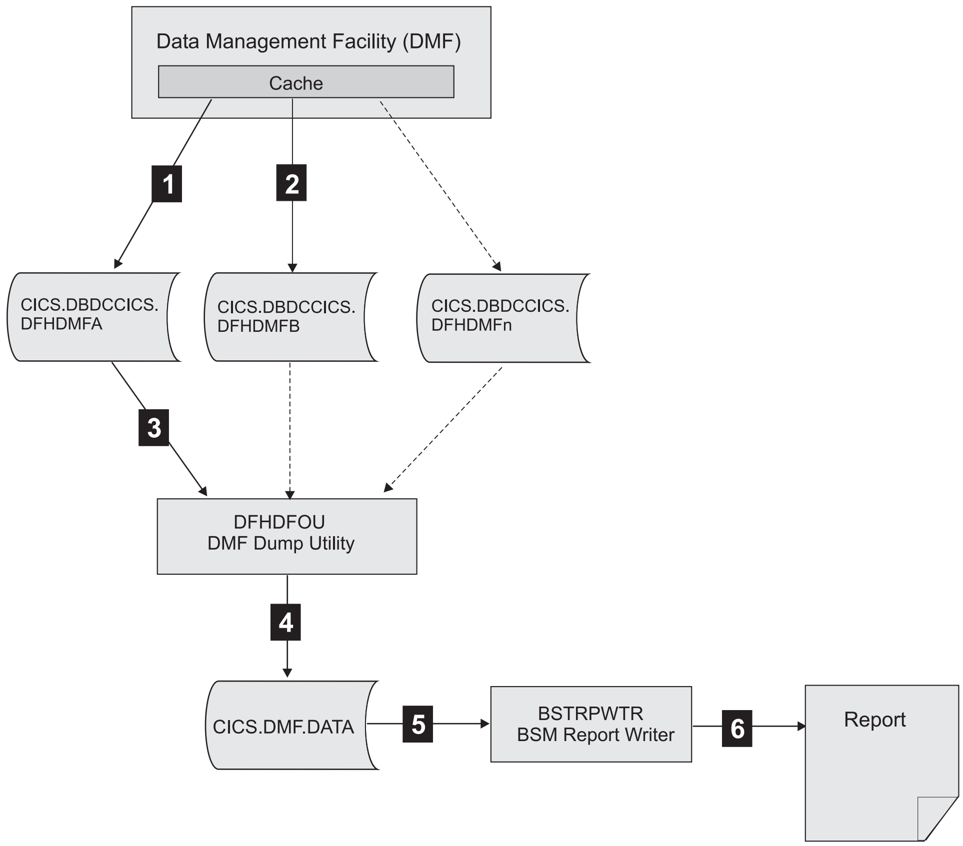


Figure 99. How DMF Is Used to Log Records and Create Reports

1 The `SETDMF FLUSH` command is used to write the current contents of the cache to the file `CICS.DBDCCICS.DFHDMFA`.

2 Before the dump utility can read `CICS.DBDCCICS.DFHDMFA` records, this file must be deactivated. Therefore, `CICS.DBDCCICS.DFHDMFA` is deactivated, and `CICS.DBDCCICS.DFHDMFB` (the next file in the group of defined data sets) is activated. To do so, the `SETDMF SWITCH` command is used.

3 The DMF dump utility DFHDFOU is started, which reads all SMF records in `CICS.DBDCCICS.DFHDMFA`.

4 The utility DFHDFOU dumps the security records (SMF80 records) to the file `CICS.DMF.DATA`.

5 The BSM Report Writer (BSTPRWTR) is started, which reads the SMF80 records from file `CICS.DMF.DATA`.

6 The BSM Report Writer creates the report.

Activating the Logging of SMF Records

To specify the logging of SMF records for a specific BSM profile, use the AUDIT parameter of the BSTADMIN ADD and CHANGE commands. You can also use the Interactive Interface dialog instead of the BSTADMIN command (see page 375).

For the logging of access attempts, you can choose between four *audit-levels*: ALL, FAILURES, NONE, or SUCCESS. For details, see “ADD | AD Command” on page 361 or “CHANGE | CH Command” on page 362.

Note: Use the auditing function with care! It will increase the BSM / DMF processing and might degrade z/VSE system performance!

Using the BSM Report Writer to Process DFHDFOU Output

The SMF records that the DMF dump utility DFHDFOU creates are stored in a data set created by yourself. You can then use the *BSM Report Writer* to create your own reports from these SMF records.

Here is an example of a DFHDFOU dump utility job:

```
* $$ JOB JNM=DFHDFOU,CLASS=0,DISP=D
// JOB OFFLOAD DMF, DATA SET
* Enter ID statement
// PAUSE ID stmt or enter
// DLBL VSESPUC,'VSESP.USER.CATALOG',,VSAM
// DLBL INFILE,'CICS.DBDCCICS.DFHDMFB',,VSAM,CAT=VSESPUC
// DLBL OUTFILE,'CICS.DMF.DATA',0,VSAM,CAT=VSESPUC, *
DISP=(NEW,KEEP),RECORDS=2000,RECSIZE=5750
// LIBDEF *,SEARCH=PRD1.BASE
* Option ALL dumps and clears DMF data set
// EXEC DFHDFOU,SIZE=DFHDFOU
INDD (INFILE, OPTIONS (ALL))
OUTDD (OUTFILE, TYPE(80))
/*
/&
* $$ EOJ
```

Figure 100. Example of DFHDFOU Dump Utility

Here is an example of a job to start the BSM Report Writer:

```
* $$ JOB JNM=BSTRPWTR,CLASS=0,DISP=D
// JOB BSTRPWTR
// PAUSE ID statement or enter
// DLBL VSESPUC,'VSESP.USER.CATALOG',,VSAM
// DLBL INPUT,'CICS.DMF.DATA',,VSAM,DISP=(OLD,KEEP),CAT=VSESPUC
// EXEC BSTRPWTR,SIZE=BSTRPWTR
/*
/&
* $$ EOJ
```

Figure 101. Example of Job Used to Start the BSM Report Writer

The BSM Report Writer reads the output of the DMF dump utility, and builds a report which contains:
- A detailed listing of the processed records.
- A summary of the user entries.
- A summary of the resource entries.
- A summary of security commands.

- A general summary.

Here is an example of the formatted output created by the BSM Report Writer:

```
07.282 12:59:18                            BSM Report - Listing of Process Records
                                       E
                                       v  Q
                                       e  u
                *Job/User              n  a
Date   Time      Name                  t  l
07.282 12:57:04  SYSA                  1  0 Job=(BSTADMIN) - User verification: Successful initiation / logon
                 AUGUST WONG                  Auth=(None),Reason=(None)
07.282 12:57:07  HUGO                  1  1 Job=(CICSICCF) - User verification: Invalid password
                 HUGO MAYER                   Auth=(None),Reason=(User verification failure)
07.282 12:57:16  HUGO                  1  0 Job=(CICSICCF) - User verification: Successful initiation / logon
                 HUGO MAYER                   Auth=(None),Reason=(None)
07.282 12:57:17  SYSA                 24  0 Job=(BSTADMIN) - PERFORM command: No violation detected
                 AUGUST WONG                  Auth=(Administrator),Reason=(Command)
                                                PERFORM PASSWORD LENGTH(8)
07.282 12:57:20  SYSA                 19  0 Job=(BSTADMIN) - PERMIT command: No violation detected
                 AUGUST WONG                  Auth=(Administrator),Reason=(Class)
                                                PERMIT APPL MYAPPL2 ID(HUGO) DELETE
07.282 12:57:28  SYSA                  1  8 Job=(BSTADMIN) - User verification: Successful termination
                 AUGUST WONG                  Auth=(None),Reason=(None)
07.282 12:57:30  HUGO                  2  1 Job=(CICSICCF) - Resource access: Insufficient authority
                 HUGO MAYER                   Auth=(Normal),Reason=(Audit options)
                                              Resource=CESN,Intent=Read,Allowed=None,Resource class=TCICSTRN,GenProf=CES
07.282 12:57:34  SYSA                  1  0 Job=(PAUSEBG ) - User verification: Successful initiation / logon
                 AUGUST WONG                  Auth=(None),Reason=(None)
07.282 12:57:37  SYSA                  2  0 Job=(PAUSEBG ) - Resource access: Successful access
                 AUGUST WONG                  Auth=(Administrator),Reason=(Administrator)
                                              Resource=MYAPPL.MYPRINT,Intent=Read,Allowed=Read,Resource class=FACILITY
07.282 12:57:38  HUGO                  1  8 Job=(CICSICCF) - User verification: Successful termination
                 HUGO MAYER                   Auth=(None),Reason=(None)
07.282 12:57:44  SYSA                  1  8 Job=(PAUSEBG ) - User verification: Successful termination
                 AUGUST WONG                  Auth=(None),Reason=(None)


07.282 12:59:18                            BSM Report - Listing of User Summary
                                                      ---------- R e s o u r c e   S t a t i s t i c s ----------
 User/    Name                ---- Job/Logon ----                    ------- I n t e n t s -------
*Job                             Success Violation      Success Violation     Alter    Update      Read     Total
 HUGO     HUGO MAYER                   1         1            0         1         0         0         1         1
 SYSA     AUGUST WONG                  2         0            1         0         0         0         1         1


07.282 12:59:18                            BSM Report - Listing of Resource Summary
                                                                     ------- I n t e n t s -------
  Resource Name                                         Success Violation     Alter    Update      Read     Total
Class = FACILITY
  MYAPPL.MYPRINT                                              1         0         0         0         1         1
Class = TCICSTRN
  CESN                                                        0         1         0         0         1         1


07.282 12:59:18                            BSM Report - Listing of Command Summary
                       Qualifier           Occurrences
Event = 19 - PERMIT command
             0 - No violation detected            1
Event = 24 - PERFORM command
             0 - No violation detected            1
        Accumulated totals -                      2


07.282 12:59:18                            BSM Report - General Summary

Process records:                                   11

                                       --- Job / Logon  Statistics ---
Total Job/Logon/Logoff                              7
Total Job/Logon successes                           3
Total Job/Logon violations                          1
Total Job/Logon attempts by undefined users         0
Total Job/Logon successful terminations             3

                                       --- Resource Statistics ---
Total resource accesses (all events)                2
Total resource access successes                     1
Total resource access violations                    1
```

Figure 102. Example of Output Created by the BSM Report Writer

Using VSE/ACLR to Log/Report Access Attempts to DTSECTAB

The z/VSE optional program *VSE/Access Control-Logging and Reporting* (VSE/ACLR) can be used if an installation plans to log the usage of certain protected resources *defined in DTSECTAB*. The program supports the logging of attempts to use protected resources without authorization and, optionally, of any **authorized** access to protected resources. It provides formatted reports of the information logged.

Note: Instead of using VSE/ACLR to log access attempts to resources protected via DTSECTAB, you can use the default BSM support that uses SMF records. For details, see "Using SMF/DMF to Log Access Attempts to DTSECTAB Resources" on page 433.

For a detailed description of this program, refer to the IBM manual *VSE/Access Control-Logging and Reporting: Program Reference and Operations Guide*.

The access control support can be used without the logging and reporting program being installed.

For each resource that is defined with access control class(es) in the access control table DTSECTAB, a logging option can be specified. Depending on the specification of that option, either **all accesses** to this resource, or only **access violations**, are logged. For further details refer to the description of the LOG parameter of the DTSECTAB macro on page 415.

The VSE/Access Control-Logging and Reporting program enables the security administrator to audit the access to system resources. The program logs access control related events and allows you to print them later for analysis.

Whenever a logging situation occurs the logging and reporting program writes a record on the log data set, a file on disk, to record the event. (When assembling and cataloging B-transients, it can happen that **two** records are written to the logging file. Take this into account when reading reports of such accesses.) The data sets used for logging need not be on dedicated disk volumes, but should be on separate volumes.

Note: You should protect the two log data sets (their file names are IJSYSL1 and IJSYSL2) in the access control table DTSECTAB. Define them without the ACC parameter. This ensures that only the VSE/ACLR program and a system administrator (SYSA) have access to those files.

Using VSE/ACLR to Log Access Attempts to Libraries

To produce accurate statistics on access to library entities (libraries, sublibraries and members), the following must be taken into account:

When a member is to be accessed, the access control checking is done at each level in the hierarchy (see Figure 103 on page 441). First of all, the access rights to library/sublibrary are valid that were established when the sublibrary was accessed (for example due to processing of a LIBDEF). When the user is attempting read access to the member and has the READ access right to the library in which the member is stored, access control does no checking at sublibrary or member level (see "Access Control for Libraries" on page 405).

The attempted access is not logged at sublibrary level, because the user was not attempting to read the sublibrary as a whole. It is not logged at member level either, because the Access Control Function stopped when it found the needed access right at library level. So this access is **not** logged, no matter what was specified in the LOG parameter for the member and sublibrary.

If, however, the user in this example had only the connect right to the library and sublibrary, access control checking would continue down to member level. The access would be logged (if the relevant access class is specified in the LOG parameter for the member; access violations are logged in any case).

In short, if all accesses are to be logged, the entity to be accessed must require a higher access right than those above it in the hierarchy, and the access rights to the entities above it must be insufficient to allow the attempted access. (It makes no difference whether the access rights are granted as individual rights in the user profile entry or as universal access rights in the resource profile entry of the access control table.)

Here are a few examples:

- For **all** accesses to a member to be logged, the access right to the library and sublibrary in which it is stored must be limited to **connect**. There will be an access check for each member access.

  As an example, consider the following generic member entry:

  ```
  IJSYSRS.SYSLIB.*,ACC=(1),LOG=(1)
  ```

  The user with ACC=(1,READ) can read all members, and each read is logged.

- For **update and alter** accesses to a member to be logged, the access right to the library and sublibrary must be **read** or lower. In this case, read accesses (to members) do not cause any access checking and hence no logging.

  For update/alter of a member, access checking will be done.

Due to the above checking mechanisms, an installation has to make a choice between satisfactory performance and detail of logging.

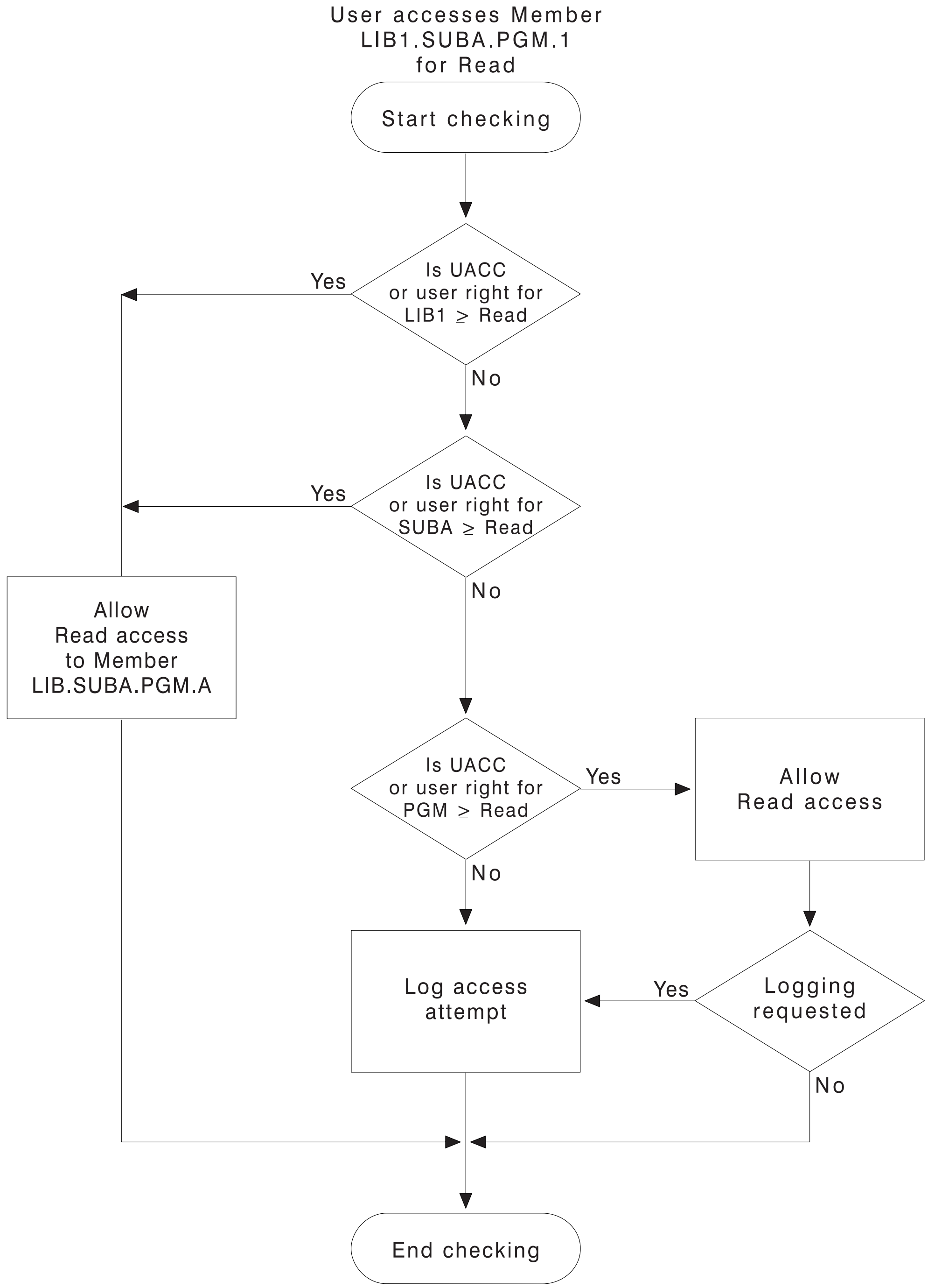
The same principle holds for the logging of accesses to sublibraries.

The security administrator should consider at which level each user should have access to the libraries, and grant only the **connect** access right to the level(s) above it. This will ensure accurate logging of access events.

The Reporting Module

The reporting module of the logging and reporting program creates printouts from the log data set according to your specifications. For more details on this topic, refer to “Using VSE/ACLR to Audit Access Attempts to Controlled Resources” on page 442.

The reporting module must run in a batch partition.



Note that all accesses to libraries and files performed by the system administrator (SYSA) are logged, regardless whether the LOG parameter in the resource access definition was specified or not.

Figure 103. Access Logging when Accessing a Library Member

Using VSE/ACLR to Audit Access Attempts to Controlled Resources

A record of unauthorized access attempts and attempts to access sensitive data can help the system administrator to identify data security problems. Auditing means analyzing these access attempts. The audit trail is the record of critical access attempts.

Careful analysis of the audit trail helps you to:
- Identify access violations and the user responsible for them;
- Find security weak points;
- Adapt access control measures to changing conditions;
- Recognize a need for corrective action by management;
- Use the system more efficiently.

VSE/Access Control-Logging and Reporting (VSE/ACLR), a z/VSE optional program, offers all of the above functions.

Using VSE/ACLR to Obtain an Audit Trail

The VSE/ACLR program enables the user, normally the security administrator, to get a printed audit trail from the information stored in the log data set. The reporting program (DSPRPM) produces printouts of the log data set, according to the selection criteria defined by the user in the ACCESS control statement of VSE/ACLR.

Unsuccessful identification and authentication attempts are not subject to logging by VSE/ACLR.

Hints for Auditing

Most access violations are the result of errors, and occur with a fairly constant frequency. Each system develops a characteristic pattern after a certain period of routine running. Any drastic change in this pattern in an otherwise unchanged system should be investigated.

For example, an unusually low number of violations may mean that someone has found a way to circumvent access control routines. The following examples give possible explanations for other symptoms:
- More people are authorized to work with the system and protected resources than are actually doing so: This may indicate that authorization is too easy to get without apparent necessity, or an anticipated necessity did not materialize.
- Many more authorized people work with the system and the protected resources than was anticipated: Again this may indicate that authorization is too easy to get, or that protection was defined for resources which actually need not be protected.
- Fewer people are authorized to use the system and protected resources than anticipated: This may indicate that authorization is too difficult to get, or the need to work with the system is not as high as estimated, or not all of the data that requires protected status has been defined.
- Some resources have an unexpectedly low access activity: It may indicate that the resource can possibly be removed from the system, or that an authorization is too difficult to get, or that a way has been found to circumvent protection.

- Some resources have an unexpectedly high access activity: In case of a file, it may indicate that a separation of sensitive and nonsensitive data should be considered to increase processing efficiency; for other resources, it may indicate that authorization is too easy to get.

These examples show that security measures must be reviewed frequently and be adjusted as the need arises.

The audit trail should be made available to the people and departments involved, and a list of violations should be distributed as often as possible to allow for quick reactions. A file owner, for example, should be notified if an unauthorized access to his file has been recorded.

Chapter 39. Protecting CICS Transactions with Access Control Table DTSECTXN

This chapter describes the method for protecting CICS transactions using table DTSECTXN.

The method described in this chapter for protecting CICS transactions using DTSECTXN *has been succeeded* by the method of protecting CICS resources using the *BSM control file* (VSE.BSTCNTL.FILE). For details, see Chapter 29, "Protecting Resources via BSTADMIN Commands," on page 357.

Furthermore, if your z/VSE 5.1 system was installed via an:

- **initial installation**, you *cannot use* the method of protecting CICS transactions described here.
- **FSU process** from a system that still used the old security concept, you *can use* the method of protecting CICS transactions described here.

This method of CICS transaction security is part of the basic security support (BSM) and protects CICS transactions from unauthorized access if they are specified in table DTSECTXN. This check is always active independent of the SEC setting in the IPL SYS command.

If you still wish to use the previous security concept, CICS will not allow access to transactions which are not defined in DTSECTXN.

The BSM compares the **security class** of a transaction with the transaction **security key(s)** defined for a particular user in the corresponding user profile. Refer also to the panel "Add or Change Resource Access Rights" that is shown in Figure 81 on page 296.

Attempts to invoke CICS transactions by unauthorized users are logged on the system console.

You can define entries in DTSECTXN through either the dialog or macro provided.

This chapter contains these main topics:

- "Using the Define Transaction Security Dialog"
- "Using the Macro DTSECTXN" on page 448

Using the Define Transaction Security Dialog

You can add, delete or alter security entries for transactions with the Transaction Security Dialog. These entries are contained in table DTRISEC.Z which is located in IJSYSRS.SYSLIB.

To access the dialog, start with the *z/VSE Function Selection* panel and select:

2 (Resource Definition)

8 (Security Maintenance)

5 (Define Transaction Security)

If you still use the old security concept, you are then asked whether you wish to migrate your CICS transaction entries to the BSM control file (by selecting Option 1) and therefore use the latest BSM security concept.

However, if you still wish to use the previous security concept that uses table DTSECTXN, you can select Option 2 “Proceed with the Define Transaction Security dialog”. The *Define Transaction Security* dialog then appears where you can press ENTER to list all available security entries, or specify the first characters of the CICS transaction names or the CICS region you want to be listed.

```
 TAS$SECF         DEFINE TRANSACTION SECURITY: SPECIFY FILTER
 Enter the required data and press ENTER.


Press ENTER to list all security entries.

Specify the prefix of the CICS transaction names or the CICS region you
want to be listed and press the ENTER key.


TRANSID.............. ____        Enter the full transaction name or
                                  1 - 3 prefix characters, e.g. AB for
                                  all transactions starting with AB

CICS REGION.......... ________    Enter the CICS region.




 PF1=HELP       2=REDISPLAY  3=END                        6=MERGE
```

Figure 104. Define Transaction Security Dialog

TRANSID
: Specifies the transaction name, or 1 to 3 prefix characters for a group of CICS transactions that are to be protected and which have the same prefix.

CICS REGION
: Specifies the CICS region for which the security entries are to be listed. The name of the CICS region is the user-ID under which CICS has been started; for example, DBDCCICS or PRODCICS. This user-ID is provided in the ID statement of the CICS startup job, independent of the settings of SYS SEC= in the IPL SYS command. The user-ID must be defined in the VSE Control File. In a predefined z/VSE system this user-ID is the same as the APPLID of this CICS system which is defined in the SIT (System Initialization Table). The list of entries will be sorted by transaction name in alphabetical order.

  Note: To activate the checking of the CICS region, you must specify SECPRFX=YES in the SIT.

Using **PF6=MERGE**, you can merge the new or changed system entries with the security table DTRISEC.Z in IJSYSRS.SYSLIB. This may be useful if you created your own member, or if you applied service, or if you did a Fast Service Upgrade (FSU) from a system still using DTSECTXN based security.

After pressing ENTER, the panel *Define Transaction Security* appears. It is a FULIST that displays selected security entries where you can add, alter, or delete entries.

You can include your own lower-case transaction security definitions in the IBM-supplied default member DTSECTXM.A (displayed in Figure 105 on page 447

447). These definitions are included in the assembly before the security definitions of the dialog, but they are not shown in the dialog.

```
TAS$SEC1                DEFINE TRANSACTION SECURITY

Enter the required data and press ENTER.

OPTIONS: 1 = ADD   2 = ALTER    5 = DELETE

 OPT      TRANSACTION NAME  CICS REGION   SECURITY CLASS  GENERIC
  _          AADD                            1
  _          ABRW                            1
  _          ACCT                            1
  _          ACEL                            1
  _          ACLG                            1
  _          AC01                            1
  _          AC02                            1
  _          AC03                            1
  _          AC05                            1
  _          AC06                            1

 LOCATE TRANSACTION NAME == >  ____
 INCLUDE MEMBER == >           IJSYSRS.SYSLIB.DTSECTXM.A
PF1=HELP      2=REDISPLAY  3=END                    5=PROCESS
              8=FORWARD
```

Figure 105. Define Transaction Security Dialog

Fill in the required data and press ENTER to check the values. Press **PF5=PROCESS** to process the changes and to submit the batch job to create table DTSECTXN including the entries defined. The source code is provided in DTSECTXS.A. Do not change this source code.

Generic Transaction Names

An **X** in column GENERIC indicates that this transaction name is interpreted as a generic name. In the example, there exists an explicit definition of security class 30 for transaction AB1, and a generic definition of security class 10 for AB. This means that security class 30 is valid for transaction AB1, and security class 10 for all those transactions that start with AB and do not have an explicit security definition like AB1.

Explanation of INCLUDE MEMBER Field

The job that is created contains an SLI statement for the member name. The member name is specified using its fully-qualified member name. This member contains lower-case transaction security definitions. These definitions are included in the assembly job in front of the security definitions of the dialog. However, these lower-case transaction security definitions are not displayed by the dialog.

The default member is IJSYSRS.SYSLIB.DTSECTXM.A.

Merging, Processing and Activating DTSECTXN

The MERGE function of the dialog can merge member DTRISEC.Z in IJSYSRS.SYSLIB with one or more of the following members:

- DTRISEC.U

  This member is created by the system in case of FSU or the application of service.
- A user-created member, for example, USERMEMB.Z in userlib.usersublib.

  The member must be of type Z and must have the special table format required by the dialog.

Example

You want to create a new DTSECTXN by merging macro member MYMAC.A in userlib.usersublib with DTRISEC.Z. The following steps are required:

1. Run a job stream for MYMAC.A that includes the following command to create the required table format:
   ```
   EXEC REXX=IPFTABLE,PARM='ULIB.USUBLIB.MYMAC.A ULIB.USUBLIB.MYTAB.Z'
   ```
2. Select the MERGE function of the dialog and enter your parameters (member name and sublibrary name) to merge your table with DTRISEC.Z in IJSYSRES.SYSLIB.
3. Select the PROCESS function of the dialog to catalog the new phase DTSECTXN into IJSYSRS.SYSLIB and PRD2.SAVE. In addition, macro DTSECTXS.A is cataloged into PRD2.CONFIG. Both, DTSECTXN and DTSECTS.A include the changes of MYTAB.Z.

   The PROCESS function:
   - Creates job CATSEC which you must submit for processing.
   - Updates table DTRISEC.Z which is used when the dialog is called the next time.
4. DTSECTXN is activated with the following command:
   ```
   CEMT PERFORM SECURITY REBUILD
   ```
   The command is issued by the job generated. You may issue this command directly if the update did not complete correctly. DTSECTXN is also activated by stopping/starting the CICS partition.

Using the Macro DTSECTXN

The DTSECTXN macro can be used to build the CICS transaction security table DTSECTXN. For each transaction to be defined, a security class must be assigned.

Syntax Diagram:

```
►►──DTSECTXN NAME=──┬─────────────────────┬──transid,TRANSEC=(class)──────────►
                    └─cics_region_prefix.─┘

►──┬───────────────┬──┬──────────────────────┬────────────────────────────────►◄
   └─,TYPE=─GENERIC─┘  └─,SUBTYPE=──┬─INITIAL─┬─┘
                                    └─FINAL───┘
```

Refer also to the definition of CICS REGION on the preceding page.

Parameter Description:

NAME=cics_region_prefix.transid
The cics_region_prefix is a 1 to 8-character identifier, and transid specifies a 1 to 4-character transaction name. Generic transaction names are allowed if TYPE=GENERIC has been specified. A transaction name may include alphabetic and numeric characters, and the following special characters:

`$ # @ / % ¢ ? ! : | " = ¬ ; < >`

Mixed case transaction names are supported.

The cics_region_prefix can be used to identify a particular CICS subsystem if two or more CICS subsystems run under z/VSE.

TRANSEC=(class)
Defines the security class (1 to 64) to control the invocation of a transaction. For example, if TRANSEC=(5) is specified, only users defined with security key 5 in their profile can start this transaction.

Only one security class can be specified for a transaction.

Security keys for a user profile can only be specified via the *Maintain User Profiles* dialog, which is described in Chapter Chapter 25, “Maintaining User Profiles via BSM Dialogs,” on page 289.

TYPE=GENERIC
This parameter is optional and causes a transaction name (transid) to be interpreted as a generic name. This means that an entry with TYPE=GENERIC covers all transactions whose names start with the characters defined as *transid*, except such a name is defined explicitly (refer also to “Generic Transaction Names” on page 447). Any abbreviation is accepted: TYPE=G, TYPE=GE, and so on.

SUBTYPE=INITIAL or FINAL
The parameter is required in the first and the last call of the DTSECTXN macro, and is not allowed in the calls between. The first macro call must include SUBTYPE=INITIAL, and the last one SUBTYPE=FINAL.

Examples:

```
DTSECTXN NAME=DBDCCICS.AADD,TRANSEC=(1)
DTSECTXN NAME=TR,TRANSEC=(1),TYPE=GENERIC
```

Example of the CICS Transaction Security Table DTSECTXM

As shipped, z/VSE provides a predefined DTSECTXM. DTSECTXM includes entries such as the following (it does not include generic entries):

```
DTSECTXN NAME=emai,TRANSEC=(1),SUBTYPE=INITIAL
DTSECTXN NAME=ftp,TRANSEC=(1)
DTSECTXN NAME=iccf,TRANSEC=(1)
DTSECTXN NAME=lpr,TRANSEC=(1)
DTSECTXN NAME=newc,TRANSEC=(1)
DTSECTXN NAME=ping,TRANSEC=(1)
DTSECTXN NAME=ropc,TRANSEC=(1)
DTSECTXN NAME=teln,TRANSEC=(1)
```

Figure 106. Extract of Table DTSECTXM

The above table contains transaction security definitions written in *lower case*. You can enter your own transaction security definitions in this table. These definitions are included in the assembly job in front of the security definitions of the dialog. However, these lower-case transaction security definitions are not displayed by the dialog.

Example of the CICS Transaction Security Table DTSECTXN

As shipped, z/VSE provides a predefined DTSECTXN. DTSECTXN includes entries such as the following (it does not include generic entries).

```
DTSECTXN NAME=AADD,TRANSEC=(1)
DTSECTXN NAME=ABRW,TRANSEC=(1)
DTSECTXN NAME=ACCT,TRANSEC=(1)
DTSECTXN NAME=ACEL,TRANSEC=(1)
DTSECTXN NAME=ACLG,TRANSEC=(1)
DTSECTXN NAME=AC01,TRANSEC=(1)
DTSECTXN NAME=AC02,TRANSEC=(1)
DTSECTXN NAME=AC03,TRANSEC=(1)
DTSECTXN NAME=AC05,TRANSEC=(1)
DTSECTXN NAME=AC06,TRANSEC=(1)
DTSECTXN NAME=AC2A,TRANSEC=(1)
DTSECTXN NAME=AC2C,TRANSEC=(1)
DTSECTXN NAME=AC2D,TRANSEC=(1)
DTSECTXN NAME=AC2E,TRANSEC=(1)
DTSECTXN NAME=AC2F,TRANSEC=(1)
DTSECTXN NAME=AC20,TRANSEC=(1)
DTSECTXN NAME=AC21,TRANSEC=(1)
DTSECTXN NAME=AC22,TRANSEC=(1)
DTSECTXN NAME=AC23,TRANSEC=(1)
DTSECTXN NAME=AC24,TRANSEC=(1)
DTSECTXN NAME=AC25,TRANSEC=(1)
```

⋮

```
DTSECTXN NAME=XPLA,
DTSECTXN NAME=2RPS,
DTSECTXN NAME=8888,
DTSECTXN NAME=9999,TRANSEC=(1),SUBTYPE=FINAL
END
```

Figure 107. Extract of Table DTSECTXN (Source Format DTSECTXS)

Chapter 40. Migrating CICS/VSE Security Information to the CICS TS

This chapter describes how you can migrate CICS/VSE 2.3 (or older) security-related information to the CICS TS.

Up to z/VSE V3R1.1, CICS transaction security was provided for the CICS TS (not CICS/VSE) via access control table DTSECTXN, which was used to define CICS *transactions* and their security class. From z/VSE V3R1.1 onwards, this method has been succeeded by the use of the BSM concept that includes the use of the BSM control file.

This chapter contains these main topics:

Overview of Migration Tasks

To migrate the your CICS/VSE 2.3 (or older) security-related information to the CICS TS, you must:

1. Migrate your **user** definitions stored in IESCNTL using utility IESBLDUP.
2. Migrate your **user** definitions stored in DFHSNT using utility IESBLDUP.
3. Use the CICS/VSE Security Migration Aid and then skeleton SKSECTX2 of ICCF library 59 to migrate your TRANSEC definitions to a *CICS TS system* that uses either table DTSECTXN or the BSM control file.
4. Use skeleton SKSECTXS of ICCF library 59 to migrate your TRANSEC definitions stored in macro DFHPCT.A to a *CICS TS system* that uses either table DTSECTXN or the BSM control file.
5. Use skeleton SKSECTX3 of ICCF library 59 to migrate your TRANSEC definitions that are stored in a library member containing CICS/VSE DFHCSDUP definitions (with format `DEFINE TRANSACTION(xxxx) ... TRANSEC(yy) ...`) to a *CICS TS system* that uses either table DTSECTXN or the BSM control file.

REXX procedures SKSECTX2, SKSECTXS, and SKSECTX3 create as output a library member whose name is to be defined in the OUTFILE parameter. Depending on the FORMAT parameter, the procedures create one of the following:

- A table in IPF format.

  Select this format if you wish to produce input for the dialog. Afterwards, the MERGE function of the *Define Transaction Security* dialog can be used. With this function, existing transaction security definitions in DTSECTXN can be merged with the migrated security definitions from CICS/VSE. Note that in this case a migrated security definition for a particular transaction overwrites an already existing definition for that transaction in DTSECTXN.
- A file consisting of DTSECTXN macro definitions.

Select this format if you wish to produce macro input for the DTSECTXN assembly run.

- A file consisting of `BSTADMIN ADD` and `BSTADMIN PERMIT` commands.

  Select this format if you wish to use the BSM security that is available from z/VSE V3R1.1 onwards (that uses the BSM control file).

These tasks are described in the topics that follow.

Migrating USER Definitions Stored in IESCNTL

For details of how to use the IESBLDUP utility to migrate your IESCNTL user definitions, refer to the chapter describing the IESBLDUP utility in the manual *z/VSE System Utilities*, SC34-2675.

Also ensure that the control options when using IESBLDUP are set correctly:

```
* CONTROL STATEMENT FOR MIGRATION FROM A VSE/SP OR VSE/ESA SYSTEM
CF=YES,DTSRSTR=NO,UPDATE=YES
```

Migrating USER Definitions Stored in DFHSNT and DTSFILE

For details of how to use the IESBLDUP utility to migrate your DFHSNT and DTSFILE user definitions, refer to the chapter describing the IESBLDUP utility in the manual *z/VSE System Utilities*, SC34-2675.

Also ensure that the control options when using IESBLDUP are set correctly:

```
CF=NO,DTSRSTR=NO,SNT=YES,ALT=YES,UPDATE=YES
ADMN=USRA,PROG=USRB,GENL=USRC
```

Migrating TRANSEC Definitions Using the Migration Aid

Step 1: Prepare Input Using the CICS Security Migration Aid

For details of how to use the CICS security migration aid, refer to the manual *CICS/VSE Security Migration Guide*, SC33-1406. You can find this manual on the *CICS for VSE/ESA V2R3* bookshelf.

You should check that you have APAR PQ50857 installed on your CICS/VSE system. If not, some transactions might be missing in the security definition.

If you wish to migrate to VSE/ESA 2.6, APAR PQ61103 will insure that the migration utility generates a compatible IPF table.

1. Run job DFHCSDUP (LIBDEF PRD2.CICSOLDP and corresponding CSD-File) with the statement `UPGRADE USING(DFHCU22X)` to create the RDO-group DFHXSM:

   ```
   * $$ JOB JNM=DFHCSDSM,CLASS=0,DISP=D
   // JOB DFHCSDSM  UPGRADE DFHCU22X
   // LIBDEF *,SEARCH=(PRD2.CONFIG,PRD2.CICSOLDP,PRD2.SCEEBASE,PRD1.BASE)
   // DLBL DFHCSD,'CICSO.CSD',0,VSAM,                                  X
                  CAT=VSESPUC
   // EXEC DFHCSDUP,SIZE=600K       INIT AND LOAD CICS CSD VSAM FILE
        UPGRADE USING(DFHCU22X)
   /*
   /&
   * $$ EOJ
   ```

2. Use transaction CEDA INSTALL group or transaction CEDA ADD group to add group DFHXSM to your GRPLIST of CICS/VSE 2.3.

3. Add the DFHFCT entry to your FCT, recompile the FCT, and COLD-Start your CICS/VSE after defining the DFHXSMA dataset using these statements:

```
DFHXSMA DFHFCT TYPE=DATASET,DATASET=DFHXSMA,ACCMETH=VSAM,  X
               SERVREQ=(UPDATE,ADD,BROWSE,DELETE),LOG=YES,
               RECFORM=(VARIABLE,BLOCKED),STRNO=2
```

   Define dataset DFHXSMA as the output file for security migration:

```
* $$ JOB JNM=DEFXSMA,CLASS=0,DISP=D,NTFY=YES
// JOB  EBER  DEFINE FILE
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER ( -
        NAME (VSE15.DFHXSMA                              ) -
        CYLINDERS(2       2         ) -
        SHAREOPTIONS (2) -
        RECORDSIZE (19     80    ) -
        VOLUMES (ESA010 ) -
        NOREUSE -
        INDEXED -
        FREESPACE (15 7) -
        KEYS (18  0     ) -
        NOCOMPRESSED -
        TO (99366  )) -
        DATA (NAME (VSE15.DFHXSMA.¹D¹                            ) -
        CONTROLINTERVALSIZE (4096  )) -
        INDEX (NAME (VSE15.DFHXSMA.¹I¹                          )) -
        CATALOG (ESCAT10.USER.CATALOG                        )
  IF LASTCC NE 0 THEN CANCEL JOB
/*
// OPTION STDLABEL=ADD
// DLBL DFHXSMA,'VSE15.DFHXSMA',,VSAM,                                 X
               CAT=ESCAT10
/*
// EXEC IESVCLUP,SIZE=AUTO
A VSE15.DFHXSMA                                DFHXSMA ESCAT10
/*
/&
* $$ EOJ
```

4. COLD-start your CICS/VSE 2.3 system after adding a DLBL statement to the CICS-startup job. For example:

```
// DLBL DFHXSMA,'VSE15.DFHXSMA',,VSAM,CAT=ESCAT10
```

5. Use the CESM transaction to migrate all wanted resources. Please note that CESM has defined RSCL=YES. If you do not want to define all applicable resources with RSL key values, do the following:
   a. Use transaction CEDA COPY to copy transaction CESM to a private group
   b. Use transaction CEDA ALTER to enter transaction CESM in this new group to RSLC=NO
   c. Use transaction CEDA ADD to add this new group to GRPLIST
   d. Use transaction CEDA INSTALL to install this new group

   After entering transaction CESM, the following screen is displayed:

```
XSM01          CICS/VSE Version 2.3 Security Migration Aid    Applid: A0006CI3


         _ Perform selective Update/Replace for this CICS system

         _ Replace all data for this CICS system

         _ Update all data for this CICS system



         _ Selectively display the contents of DFHXSMA



PF1=Help PF12=Quit ENTER=Continue
```

Figure 108. CICS/VSE Security Migration Aid Screen (XSM01)

6. Enter an 'X' next to the option "Perform selective ....". The following screen is then displayed:

```
XSM02          CICS/VSE Version 2.3 Security Migration Aid    Applid: A0006CI3


         _ Transient Data           (DCT)       _ Replace
         _ Files                    (FCT)       _ Update
         _ Journals                 (JCT)
         _ Transactions             (PCT)
         _ Programs                 (PPT)
         _ USER Profiles            (IUI/SNT/TCT)
         _ RCF authorized printers  (TCT)
         _ Temporary Storage        (TST)



PF1=Help PF4=Main Menu PF12=Quit ENTER=Continue
```

Figure 109. CICS/VSE Security Migration Aid Screen (XSM02)

Entering an 'X' in front of Transactions and in front of REPLACE will

a. Collect all of the transaction-related security information of the running CICS/VSE 2.3 system.
b. Store this security information in DFHXSMA.

Step 2: Migrate TRANSEC Definitions to a CICS TS System

This step can be followed for VSE/ESA V2R4.0 onwards and a CICS TS 1.1.0 or 1.1.1 system.

Skeleton SKSECTX2 executes procedure **DTSECTX2**, which uses as input the VSE/VSAM output file DFHXSMA created by the *Security Migration Aid (SMA)* of CICS/VSE (Step 1 above). The migration aid extracts security information from CICS/VSE and the VSE.CONTROL.FILE and stores it in file DFHXSMA. DTSECTX2 then reads the security information from DFHXSMA.

DTSECTX2 generates output for table DTSECTXN of a CICS TS system, or for the BSM control file of a CICS TS system.

To migrate your TRANSEC definitions, you should copy SKSECTX2 from ICCF library 59 to your own library, and modify the lines at the end of the skeleton according to your installation:

```
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE)
// EXEC REXX=DTSECTX2,PARM='INFILE=VSEESA.DFHXSMA          X
               OUTFILE=PRD2.CONFIG.DTSECTX2.A'
REGION=MYREGION
FORMAT=MACRO
CATALOG=VSESPUC
/*
```

Parameters can be specified in the PARM operand, via SYSIPT, or in a combination of both. Refer to "DTSECTX2 Parameters" for a description of the parameters and examples of how to use them. A description of the parameters required can also be found within this REXX procedure.

You then submit the job (skeleton) for processing. The job catalogs first procedure DTSECTX2 into PRD2.CONFIG and invokes it from there. You may change the name of the sublibrary used for cataloging.

Note: Be sure that you have closed the DFHXSMA file in the CICS/VSE 2.3 partition before running this REXX procedure.

DTSECTX2 Parameters

DTSECTX2 uses the following parameters:

Parameter
: **Description**

INFILE
: Defines the VSE/VSAM file created by the CICS/VSE Security Migration Aid (SMA).

FORMAT
: Defines the format of the output file; either IPF, MACRO, or BSTADMIN.

  IPF is the format required when using the MERGE function of the *Define Transaction Security* dialog to merge existing security entries in DTSECTXN with migrated entries. This is the default.

  MACRO
  is the format consisting of DTSECTXN macro definitions. This output must be assembled to create a DTSECTXN.PHASE and activated via the `CEMT PERF SECURITY REBUILD` command.

  BSTADMIN
  is the format consisting of `BSTADMIN ADD` and `BSTADMIN PERMIT` commands. You can use this format to migrate your CICS/VSE 2.3 security data to the BSM control file (VSE.BSTCNTL.FILE) that is available from z/VSE V3R1.1 onwards. For details, see "Performing the Migration" on page 285.

OUTFILE
: Defines the library member name of the file to be created containing the definitions for table DTSECTXN or for input to the BSM control file (BSTADMIN format). The default is:
  - `PRD2.CONFIG.DTRISEC.M` for the IPF format.
  - `PRD2.CONFIG.DTSECTXS.A` for the MACRO format.
  - `PRD2.CONFIG.DTSECTVTX.Z` for the BSTADMIN format.

CATALOG
: Defines the name of the VSE/VSAM catalog which includes the input file (INFILE). The default is the master catalog IJSYSCT.

REGION
Defines the name of the CICS-Region to be included in the DTSECTXN definitions. Default is the region name found in the input file (INFILE). If region names should not be included in the DTSECTXN definitions, specify REGION= followed by a blank.

Invoking DTSECTX2

The following example invokes DTSECTX2 and uses the PARM operand to provide the parameters INFILE and OUTFILE. The parameters REGION and FORMAT are provided via SYSIPT.

```
// EXEC REXX=DTSECTX2,PARM='INFILE=CICS.DFHXSMA           X
OUTFILE=LIB.SLIB.DTSECTX2.A REGION=DBDCCICS'
```

The following example invokes DTSECTX2 and uses SYSIPT to provide the parameters.

```
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE)
// EXEC REXX=DTSECTX2
INFILE=VSEESA.DFHXSMA
OUTFILE=PRD2.CONFIG.DTSECTX2.A
REGION=MYREGION
FORMAT=MACRO
CATALOG=VSESPUC
/*
```

DTSECTX2 Return Codes

DTSECTX2 may issue the following return codes:

0 Processing successful

2 Unsupported character in transaction name: <*tr-name*>

4 Syntax error: Invalid keyword <*keyword*>

8 Member <*mem.type*> already exists in sublibrary <*lib.slib*>

12 IDCAMS PRINT of VSE/VSAM file failed with RC=*n*

16 Transaction name longer than 4 characters: <*tr_name*>

24 No transaction specifications found

28 Accessing sublibrary <*lib.slib*> failed

32 Writing member <*lib.slib.mem.type*> failed with RC=*n*

After the job has run, you should verify your output file and the printout in the LST queue to ensure correct processing.

Migrating DFHPCT.A TRANSEC Definitions

Skeleton SKSECTXS executes procedure **DTSECTXS**, which uses as input existing library members that include Program Control Table (PCT) definitions. DTSECTXS generates output for table DTSECTXN of a CICS TS system, or the BSM control file of a CICS TS system. Examples of library members with PCT definitions are the CICS/VSE tables IESZPCT and DFHPCTxx.

To migrate your PCT TRANSEC definitions, you should copy skeleton SKSECTXS from ICCF library 59 to your own library and modify the following lines at the end of the skeleton according to your installation. An A-book of your current DFHPCT needs to be placed into the named INFILE library (in this sample PRD2.CONFIG).

```
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE)
// EXEC REXX=DTSECTXS,PARM='INFILE=IJSYSRS.SYSLIB.IESZPCT.A       X
              OUTFILE=PRD2.CONFIG.DTSECTXS.A'
REGION=DBDCCICS
FORMAT=MACRO
/*
```

Parameters can be specified in the PARM operand, via SYSIPT, or in a combination of both. Refer to "DTSECTXS Parameters" for a description of the parameters and examples of how to use them. A description of the parameters required can also be found within this REXX procedure.

You then submit the job (skeleton) for processing. The job catalogs first procedure DTSECTXS into PRD2.CONFIG and invokes it from there. You may change the name of the sublibrary used for cataloging.

DTSECTXS Parameters

DTSECTXS uses the following parameters:

Parameter
: **Description**

INFILE
: Defines the library member name of the file containing PCT definitions with TRANSEC values.

FORMAT
: Defines the format of the output file; either IPF, MACRO, or BSTADMIN.

  IPF is the format required when using the MERGE function of the *Define Transaction Security* dialog to merge existing security entries in DTSECTXN with migrated entries. This is the default.

  MACRO
  : is the format consisting of DTSECTXN macro definitions. This output must be assembled to create a DTSECTXN.PHASE and activated via the command `CEMT PERF SECURITY REBUILD`.

  BSTADMIN
  : is the format consisting of `BSTADMIN ADD` and `BSTADMIN PERMIT` commands. You can use this format to migrate your CICS/VSE 2.3 security data to the BSM control file (VSE.BSTCNTL.FILE) that is available from z/VSE V3R1.1 onwards. For details, see "Performing the Migration" on page 285.

OUTFILE
: Defines the library member name of the file to be created containing the definitions for table DTSECTXN or for input to the BSM control file (BSTADMIN format). The default is:
  - `PRD2.CONFIG.DTRISEC.M` for the IPF format.
  - `PRD2.CONFIG.DTSECTXS.A` for the MACRO format.
  - `PRD2.CONFIG.DTSECTVTX.Z` for the BSTADMIN format.

REGION
: Defines the name of the CICS-Region to be included in the DTSECTXN definitions. If region names should not be included in the DTSECTXN definitions, omit this parameter or specify REGION= followed by a blank.

Invoking DTSECTXS

The following example invokes DTSECTXS and uses the PARM operand to provide parameters:

```
// EXEC REXX=DTSECTXS,PARM='INFILE=IJSYSRS.SYSLIB.IESZPCT.A'
```

For the FORMAT, OUTFILE, and REGION parameters the defaults are effective.

The following example invokes DTSECTXS and uses SYSIPT to provide parameters:

```
// EXEC REXX=DTSECTXS
INFILE=IJSYSRS.SYSLIB.IESZPCT.A
FORMAT=MACRO
OUTFILE=LIB.SLIB.DTSECTXS.A
REGION=DBDCCICS
```

DTSECTXS Return Codes:

DTSECTXS may issue the following return codes:

0 Processing successful

2 Unsupported character in transaction name: <*tr-name*>

4 Syntax error: Invalid keyword <*keyword*>

8 Member <*mem.type*> already exists in sublibrary <*lib.slib*>

12 Reading member <*mem.type*> failed with RC=*n*

16 Transaction name longer than 4 characters: <*tr_name*>

20 TRANSEC definition is not a number: <*tran_sec*>

24 No transaction specifications found

28 Accessing sublibrary <*lib.slib*> failed

32 Writing member <*lib.slib.mem.type*> failed with RC=*n*

After the job has run, you should verify your output file and the printout in the LST queue to ensure correct processing.

Migrating DFHCSDUP TRANSEC Definitions

Skeleton SKSECTX3 executes procedure **DTSECTX3**, which uses as input the CICS/VSE DFHCSDUP definitions of format `DEFINE TRANSACTION(xxxx)... TRANSEC(yy)` stored in a library member. DTSECTX3 generates output for table DTSECTXN of a CICS TS system, or the BSM control file of a CICS TS system.

To migrate your DFHCSDUP TRANSEC definitions, you should copy skeleton SKSECTX3 from ICCF library 59 to your own library and modify the following lines at the end of the skeleton according to your installation. An Z-book of your current DFHCSDUP needs to be placed into the named INFILE library (in this sample PRD2.PROD).

```
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE)
// EXEC REXX=DTSECTX3,PARM='INFILE=PRD2.PROD.MGJCSD.Z       X
               OUTFILE=PRD2.CONFIG.DTSECTX3.A'
REGION=MYREGION
FORMAT=MACRO
/*
```

Parameters can be specified in the PARM operand, via SYSIPT, or in a combination of both. Refer to “DTSECTX3 Parameters” for a description of the parameters and examples of how to use them. A description of the parameters required can also be found within this REXX procedure.

You then submit the job (skeleton) for processing. The job catalogs first procedure DTSECTX3 into PRD2.CONFIG and invokes it from there. You may change the name of the sublibrary used for cataloging.

DTSECTX3 Parameters

DTSECTX3 uses the following parameters:

Parameter
: **Description**

INFILE
: Defines the name of the file (name used by Librarian) containing the DFHCSDUP definitions.

FORMAT
: Defines the format of the output file; either IPF, MACRO, or BSTADMIN.

  IPF is the format required when using the MERGE function of the *Define Transaction Security* dialog to merge existing security entries in DTSECTXN with migrated entries. This is the default.

  MACRO
  : is the format consisting of DTSECTXN macro definitions. This output must be assembled to create a DTSECTXN.PHASE and activated via the command `CEMT PERF SECURITY REBUILD`.

  BSTADMIN
  : is the format consisting of `BSTADMIN ADD` and `BSTADMIN PERMIT` commands. You can use this format to migrate your CICS/VSE 2.3 security data to the BSM control file (VSE.BSTCNTL.FILE) that is available from z/VSE V3R1.1 onwards. For details, see “Performing the Migration” on page 285.

OUTFILE
: Defines the library member name of the file to be created containing the definitions for table DTSECTXN or for input to the BSM control file (BSTADMIN format). The default is:
  - `PRD2.CONFIG.DTRISEC.M` for the IPF format.
  - `PRD2.CONFIG.DTSECTXS.A` for the MACRO format.
  - `PRD2.CONFIG.DTSECTVTX.Z` for the BSTADMIN format.

REGION
: Defines the name of the CICS-Region to be included in the DTSECTXN definitions. Default is the region name found in the input file (INFILE). If region names should not be included in the DTSECTXN definitions, omit this parameter or specify REGION= followed by a blank.

Invoking DTSECTX3

The following example invokes DTSECTX3 and uses the PARM operand to provide parameters:

```
// EXEC REXX=DTSECTX3,PARM='INFILE=LIB.SLIB.MYTRANS.DEFS            X
OUTFILE=LIB.SLIB.DTSECTX3.A REGION=DBDCCICS'
```

The following example invokes DTSECTX3 and uses SYSIPT to provide parameters:

```
// EXEC REXX=DTSECTX3
INFILE=LIB.SLIB.MYTRANS.DEFS
CATALOG=IJSYSCT
FORMAT=MACRO
OUTFILE=LIB.SLIB.DTSECTX3.A
REGION=DBDCCICS
```

DTSECTX3 Return Codes

DTSECTX3 may issue the following return codes:

0 Processing successful

2 Unsupported character in transaction name: <*tr-name*>

4 Syntax error: Invalid keyword <*keyword*>

8 Member <*mem.type*> already exists in sublibrary <*lib.slib*>

12 Reading member <*mem.type*> failed with RC=*n*

16 Transaction name longer than 4 characters: <*tr_name*>

20 TRANSEC definition is not a number: <*transec*>

21 Security number greater than 24

24 No transaction specifications found

28 Accessing sublibrary <*lib.slib*> failed

32 Writing member <*lib.slib.mem.type*> failed with RC=*n*

After the job has run, you should verify your output file and the printout in the LST queue to ensure correct processing.

Part 4. ENCRYPTION

Chapter 41. Implementing Hardware Cryptographic Support

This chapter describes how you can implement hardware cryptographic support in these main topics:

Related Topic:

For details of how to ...	Refer to the ...
set up cryptographic hardware using the *Hardware Management Console* and the *Support Element*	technical article "How to set up cryptographic hardware for VSE", which you can find in the "Documentation" section of the *z/VSE Homepage* (see "Where to Find More Information" on page xxiii).

Background

Crypto cards provide RSA encryption-assist support and can help to increase the throughput in a TCP/IP network using *SSL (Secure Sockets Layer)*:

- SSL uses cryptography both for authentication of clients and servers, and for data confidentiality. It is a public key cryptography-based extension to TCP/IP networking.
- The *CP Assist for Cryptographic Function* (CPACF) provides hardware support for symmetric cryptographic algorithms, like AES, DES, Triple-DES, and SHA-1.
- z/VSE supports the IBM Crypto Express2, Crypto Express3, and PCI Cryptographic Accelerator (PCICA) cards which provide encryption assist support for increased SSL throughput. The support is based on functions provided by *TCP/IP for VSE/ESA 1.5*.
- SSL transparently uses Crypto Express2, PCICA, and PCIXCC cards (if available).
- There is no need to change any applications already using SSL. Existing applications that use SSL *automatically* benefit from this transparent use of Crypto cards. For example, applications such as:
  - CICS Web Support (CWS),
  - VSE/POWER PNET,
  - z/VSE e-business connectors,
  - Secure FTP,
  - Secure Telnet.
  - WebSphere MQ with SSL
- If Hardware Crypto support is not available, *TCP/IP for VSE/ESA 1.5F* transparently provides software encryption.

Assigning Crypto Cards to a Specific LPAR

Note: This topic provides you with an *extract only* of the IBM Service Element program. For *all further information/inquiries*, please refer to the documentation supplied with your IBM server.

Installed Crypto cards are assigned to a specific LPAR using the System z platform's *Service Element*. To display the *current* status or add a Crypto device, you should:

1. Open the **Primary Support Element Workplace**.
2. Click on **Cryptos** using a window similar to Figure 110.

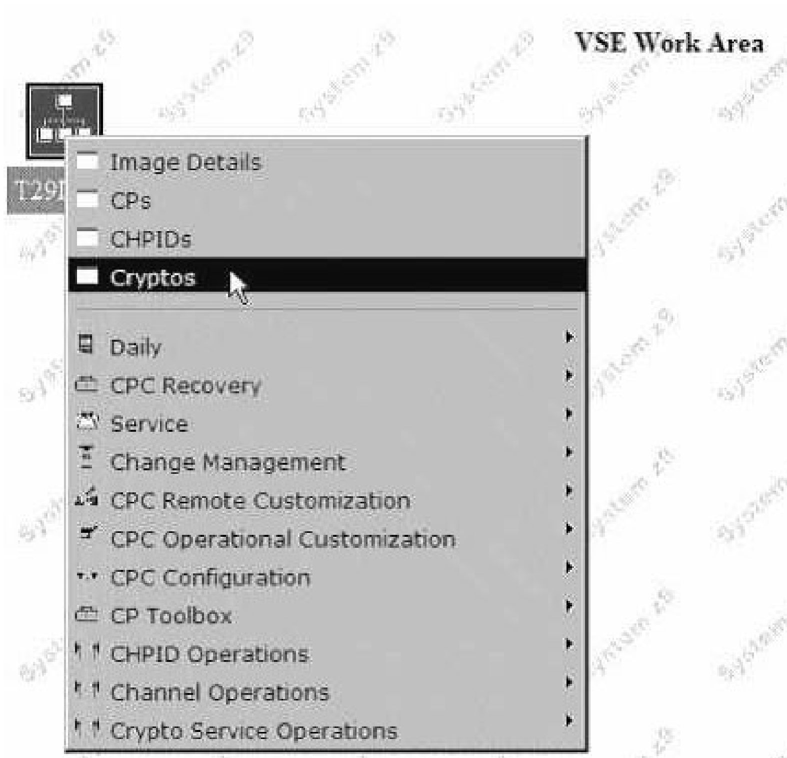


Figure 110. Using the Cryptos Option of the Service Element Program

3. The available Crypto devices and their assignment to this specific LPAR are then displayed.
4. Use the menu choice **Crypto Details** to display or change the properties of a Crypto device. Use the option **Crypto Service Operations** to add a Crypto device to this LPAR, as shown in Figure 111 on page 467.

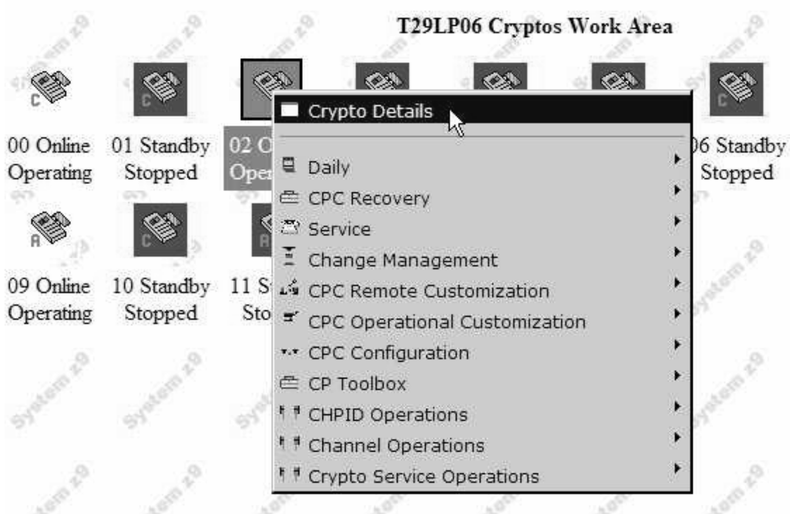


Figure 111. Example of the Crypto Service Operations Window

How Crypto Cards Are Used

z/VSE uses Crypto cards for RSA acceleration only:

- The PCICA card supports **RSA1024** and **RSA2048**. However, RSA2048 is *not* exploited by TCP/IP for VSE/ESA.
- A key length of 512 is not *directly* supported by the hardware. Instead, this key length is internally processed by the hardware using 1024-bit requests.
- TCP/IP for VSE/ESA exploits RSA1024 **and** RSA2048 on PCIXCC and Crypto Express2 cards.
- TCP/IP for VSE/ESA does **not** provide a software implementation for RSA keys greater than 1024 bits.

The Crypto card is plugged into an Adjunct Processor (AP), which is seen as an extension to the CPU. Once plugged into the system, the Crypto card is identified as follows:

- An eight-character serial number.
- A two-digit Adjunct Processor (AP) number.
- A CHPID number.

Since Crypto cards are seen as an extension to the CPU rather than as a new channel-attached device, the Crypto card requires no configuration, and as a result
- no device type
- no ADD statement
- no IOCDS definition.

During system initialization (IPL), z/VSE senses the hardware and recognizes the Crypto support, 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 AP 0
```

```
FB 0094 1J023I FOUND A CRYPTO EXPRESS2 CARD AT AP 2
FB 0094 1J022I CPU CRYPTOGRAPHIC ASSIST FEATURE AVAILABLE.
FB 0094 1J005I HARDWARE CRYPTO ENVIRONMENT INITIALIZED SUCCESSFULLY.
FB 0094 1J006I USING AP QUEUE 6
```

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 Crypto Support with a z/VSE Guest under z/VM

When z/VSE runs as a guest under z/VM, the Crypto support must be defined to the system in the *VM Directory Entry* using 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 address 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, cryptographic 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 APs in z/VM. If there are multiple APs available, z/VM will automatically balance the workloads and display *one* AP only to the z/VM guest. In the example below, there are two Crypto Express2 cards in use, where each card has two APs. However, z/VSE will access one “virtual” AP 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 numbers randomly, so it is normal for the guest to see a different AP number each time the guest is started. This is independent of the AP queue number (Cryptographic domain). 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 display the general cryptographic configuration, you use the following device driver's 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 CRYPTO DEVICE DRIVER STATUS:
FB 0011   AP CRYPTO SUBTASK STARTED .......... : YES
FB 0011   MAX REQUEST QUEUE SIZE ............. : 1
FB 0011   MAX PENDING QUEUE SIZE ............. : 1
FB 0011   TOTAL NO. OF AP REQUESTS ........... : 1234
FB 0011   NO. OF POSTED CALLERS .............. : 1234
FB 0011   AP-QUEUE INTERRUPTS AVAILABLE ...... : YES
FB 0011   AP-QUEUE INTERRUPTS STATUS ......... : DISABLED
FB 0011   AP CRYPTO POLLING TIME (1/300 SEC).. : 1
FB 0011   AP CRYPTO WAIT ON BUSY (1/300 SEC).. : 75
FB 0011   AP CRYPTO RETRY COUNT .............. : 5
FB 0011   AP CRYPTO TRACE LEVEL .............. : 3
FB 0011   TOTAL NO. OF WAITS ON BUSY ......... : 0
FB 0011   CURRENT REQUEST QUEUE SIZE ......... : 0
FB 0011   CURRENT PENDING QUEUE SIZE ......... : 0
FB 0011   ASSIGNED APS : PCICC / PCICA ....... : 0 / 0
FB 0011                  CEX2C / CEX2A ....... : 1 / 1
FB 0011                  CEX3C / CEX3A ....... : 0 / 0
FB 0011                  PCIXCC .............. : 0
FB 0011     AP  0 : CEX2C   - ONLINE
FB 0011     AP  1 : CEX2A   - ONLINE
FB 0011   ASSIGNED AP QUEUE (CRYPTO DOMAIN)... : 5
FB 0011 END OF CRYPTO DEVICE DRIVER STATUS
```

The above output shows the availability of:
- Two Crypto Express2 APs, configured in coprocessor mode.
- Three Crypto Express2 APs, configured in accelerator mode.
- The CPACF feature.
- The assigned AP queue (cryptographic domain) is 5.

To display the CPACF status, you use the following device driver's command:

```
msg fb,data=status=cpacf
```

The output from this command looks like following:

```
AR 0015 1I40I  READY
FB 0011 BST223I CURRENT STATUS OF THE SECURITY TRANSACTION SERVER:
FB 0011 CPU CRYPTOGRAPHIC ASSIST FEATURE:
FB 0011   CPACF AVAILABLE .................... : YES
FB 0011   INSTALLED CPACF FUNCTIONS:
FB 0011     DES, TDES-128, TDES-192
```

```
FB 0011     AES-128, AES-192, AES-256, PRNG
FB 0011     SHA-1, SHA-256, SHA-512
FB 0011     KMAC_DES, KMAC_TDES128, KMAC_TDES192
FB 0011   PROTECTED KEY CPACF FUNCTIONS:
FB 0011     ENCR_DES, ENCR_TDES128, ENCR_TDES192
FB 0011     ENCR_AES128, ENCR_AES192, ENCR_AES256
FB 0011     KMAC_ENCR_DES, KMAC_ENCR_TDES128, KMAC_ENCR_TDES192
FB 0011   ENCRYPTION MODES:
FB 0011     ECB, CBC
FB 0011 END OF CPACF STATUS
```

The above output (taken from a zEnterprise 196) shows the availability of the CPACF, which provides symmetric Crypto functions (such as DES, triple-DES, AES, and SHA-1 hashing).

CPACF is not available on all zSeries servers. Where it is available, it is used *transparently* in SSL sessions.

By displaying the command HELP of the SECSERV job, you can obtain a list of the available hardware Crypto commands. The listing below shows:
- The general Security Server commands that you can use.
- A list of the hardware Crypto commands.

```
msg fb,data=?
AR 0015 1I40I  READY
FB 0011 BST221I POSSIBLE SECURITY SERVER COMMANDS ARE:
FB 0011   DBSTARTCACHE ........: STARTS DATABASE CACHING
FB 0011   DBSTOPCACHE .........: STOPS DATABASE CACHING
FB 0011   STATUS ..............: SHOWS TOTAL SERVER STATUS
FB 0011   STATUS=ALL ..........: SHOWS TOTAL SERVER STATUS
FB 0011   STATUS=MAIN|PS|DB    : SHOWS SELECTED STATUS
FB 0011   STATUS=CR|CPACF .....: SHOWS SELECTED CRYPTO STATUS
FB 0011   LOGTIME=N ...........: SETS LOGTIME TO N MINUTES (1..9)
FB 0011   RESET ...............: CLEANUP EVERYTHING
FB 0011   STOP | SHUTDOWN .....: STOPS THE SERVER (USE WITH CAUTION)
FB 0011   SHUTDOWN NOPROMPT ...: STOPS THE SERVER WITHOUT CONFIRM.
FB 0011   OPENCNTL ............: OPENS THE II CONTROL FILE
FB 0011   CLOSECNTL ...........: CLOSES THE II CONTROL FILE
FB 0011   OPENBST .............: OPENS THE BSM CONTROL FILE
FB 0011   CLOSEBST ............: CLOSES THE BSM CONTROL FILE
FB 0011 HARDWARE CRYPTO COMMANDS:
FB 0011   APBUSY=NN ...........: SET AP CRYPTO WAIT ON BUSY (0..99)
FB 0011   APRETRY=NN ..........: SET AP CRYPTO RETRY COUNT (0..99)
FB 0011   APREM AP=NN .........: REMOVE (DISABLE) A CRYPTO DEVICE
FB 0011   APADD AP=NN .........: ADD (ENABLE) A DISABLED DEVICE
FB 0011   APQUE ...............: SHOW STATUS OF ASSIGNED AP QUEUE
FB 0011   APHIST ..............: SHOW HISTORY OF PROCESSED REQUESTS
FB 0011   APWAIT=NN ...........: SET AP CRYPTO POLLING TIME (0..99)
FB 0011   APSENSE .............: START SENSING OF CRYPTO HARDWARE
FB 0011   APTRACE=N ...........: SET AP CRYPTO TRACE LEVEL (0..3)
FB 0011   APEAI ...............: ENABLE AP-QUEUE INTERRUPTS
FB 0011   APDAI ...............: DISABLE AP-QUEUE INTERRUPTS
FB 0011   APSTAT AP=NN ........: DISPLAY ADAPTER STATUS
```

Using Hardware Crypto Commands

You can use the commands described below to manage your hardware Crypto configuration.

Using the APADD Command to Dynamically Add/Enable a Crypto Card

On IBM System z10 (z10) and zEnterprise 196 (z196) platforms, you can use the APADD command to add/enable *an AP (Crypto card)* that has been removed/disabled from use with z/VSE via the APREM command. The AP (Crypto card) is then flagged as being available for processing Crypto requests in z/VSE.

The AP (Crypto card) is specified using a number between 1 and 63.

You should be aware that:
- No "physical" change is made to the Crypto device.
- The APADD command is used to control the use of an *assigned* Crypto device by z/VSE.

Here is an example of the APADD command:

```
msg fb,data=apadd ap=1
AR 0015 1I40I  READY
FB 0011 1J025I AP 1 ENABLED SUCCESSFULLY.
```

If you are not using the BSM, for guidance you should refer to "Using Crypto Support and an External Security Manager" on page 476.

Using the APBUSY Command to Set the Wait-On-Busy Time Interval

You can use the APBUSY command to set the *wait-on-busy time interval*. This is the wait-time between attempts to re-queue a request after a:
- device-busy condition,
- reset-in-progress condition,
- queue-full condition.

The default value is 75/300th seconds. Valid values are 0 to 99.

Here is an example of the APBUSY command to set the AP wait-on-busy time interval to 50/300th seconds:

```
msg fb,data=apbusy=50
AR 0015 1I40I  READY
FB 0011 1J040I WAIT ON BUSY TIME SET TO 50 * 1/300 SEC.
```

Using the APEAI Command to Enable AP-Queue Interrupts

You can use the APEAI command to enable *AP-queue interrupts*. The AP-queue adapter-interruption facility is a function of z10 and z196 platforms. When performing cryptographic operations on Crypto Express2 and Crypto Express3 cards, the calling program is notified via a hardware interrupt when a response is ready for de-queueing from a card. Previously, the calling program had to use a polling mechanism for this function. In certain situations, using AP-queue interrupts might result in enhanced performance.

Note: the AP-queue adapter-interruption facility is *not* available when running under z/VM.

Here is an example of the APEAI command to check if AP-queue interrupts are available:

```
msg fb,data=status=cr
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-QUEUE INTERRUPTS AVAILABLE ...... : YES
FB 0011   AP-QUEUE INTERRUPTS STATUS ......... : DISABLED
⋮
```

When the APEAI command is executed, AP-queue interrupts will be enabled for *all APs* assigned to the current LPAR, as shown in this example:

```
msg fb,data=apeai
AR 0015 1I40I  READY
FB 0011 1J048I AP QUEUE ADAPTER INTERRUPTS ENABLED.
```

Using the APDAI command to disable AP-queue interrupts

You can use the APDAI command to disable *AP-queue interrupts* for all APs assigned to the current LPAR.

Here is an example of the APEAI command:

```
msg fb,data=apdai
AR 0015 1I40I  READY
FB 0011 1J049I AP QUEUE ADAPTER INTERRUPTS DISABLED.
```

Using the APHIST Command to Obtain an Overview of Processed Crypto Requests

You can use the APHIST command to obtain an overview of the *Crypto requests* that have been processed since the last IPL or last restart of the Security Server.

The output from APHIST provides statistics that:
- show all APs assigned to this LPAR or VM guest.
- list the Crypto requests that have been processed.

Here is an example of the APHIST command:

```
msg fb,data=aphist
AR 0015 1I40I  READY
FB 0011 1J046I HISTORY OF AP QUEUE 5:
FB 0011 AP       : 0  (CEX2C)   1  (CEX2A)
FB 0011 ---------- ------------ ------------
```

```
FB 0011 RSA1024E : 123          456
FB 0011 RSA1024D : 78           95
FB 0011 RSA2048E : 12           34
FB 0011 RSA2048D : 5            6
FB 0011 ---------- ------------ ------------
```

Using the APQUE Command to Display Current Requests

You can use the APQUE command to display:
- The number of requests that are currently being processed.
- A list of APs (Crypto cards) that are assigned to this LPAR or VM guest.

The number of currently-processed requests must be *zero* before you can toggle an AP (Crypto card) to “OFF” using the Support Element. Typically, an administrator will use:
1. APREM command to disable an AP.
2. APQUE command to check if the AP can be safely toggled to “OFF” using the Support Element.

An AP (Crypto card) can be specified using a number between 1 and 63.

Here is an example of the APQUE command that shows that *both* APs can be safely removed from the LPAR or VM guest:

```
msg fb,data=apque
AR 0015 1I40I  READY
FB 0011 1J045I NUMBER OF REQUESTS BEING PROCESSED BY AP QUEUE 5:
FB 0011   AP 0 : 0
FB 0011   AP 1 : 0
```

Using the APREM Command to Dynamically Remove/Disable a Crypto Card

You can use the APREM command to remove/disable an *AP (Crypto card)* from z/VSE without the need to restart the LPAR or VM guest. The AP (Crypto card) is then flagged as being unavailable for processing Crypto requests in z/VSE.

On z10 and z196 platforms, APREM can be used to:
- set an AP (Crypto card) to offline.
- dynamically remove an AP (Crypto card) from a z/VSE LPAR via the server's Support Element.

Typically, you would:
1. Use APREM to prevent this AP (Crypto card) from being used to process further Crypto requests.
2. Repeatedly use the APQUE command to check if there are replies pending in this AP.
3. When no replies are pending, use the server's Support Element to set the AP to offline or remove the AP (Crypto card) from the LPAR .

The AP (Crypto card) is specified using a number between 1 and 63.

You should be aware that:
- The APREM command causes an AP to be no longer available for processing Crypto requests.
- The status of the Crypto device is not changed.
- The APADD command can be used to re-enable the AP for processing Crypto requests.

Here is an example of the APREM command:

```
msg fb,data=aprem ap=1
AR 0015 1I40I  READY
FB 0011 1J026I AP 1 DISABLED SUCCESSFULLY.
```

Using the APRETRY Command to Set the Number of Retry Attempts

You can use the APRETRY command to set the number of *retry attempts* on a:
- device-busy condition,
- reset-in-progress condition,
- queue-full condition.

The default value is *five* retry attempts. Valid values are 0 to 99.

Here is an example of the APRETRY command to set the number of retry attempts to 10:

```
msg fb,data=apretry=10
AR 0015 1I40I  READY
FB 0011 1J036I RETRY COUNT SET TO 10.
```

Using the APSENSE Command to Refresh Your Hardware Crypto Configuration

You can configure an AP online or offline without the need to restart the LPAR or VM guest. In addition, you can dynamically change the configuration of a Crypto Express2 and later adapter from co-processor mode into accelerator mode or vice versa.

You can use the APSENSE command to *dynamically reflect such configuration changes in your z/VSE system*. This command updates the hardware Crypto environment *without the need to re-IPL your z/VSE system*.

Here is an example of the APSENSE command:

```
msg fb,data=apsense
AR 0015 1I40I  READY
FB 0095 1J022I CPU CRYPTOGRAPHIC ASSIST FEATURE AVAILABLE.
FB 0095 1J031I HARDWARE CRYPTO ENVIRONMENT REFRESHED.
```

If you are not using the BSM, for guidance you should refer to "Using Crypto Support and an External Security Manager" on page 476.

Using the APTERM Command to Terminate Crypto Subtask IJBCRYPT

You can use the APTERM command to terminate the *z/VSE Crypto device* (subtask IJBCRYPT). This subtask is part of the SECSERV (Security Server) phase **BSTPSTS**.

Here is an example of the APTERM command:

```
msg fb,data=apterm
AR 0015 1I40I  READY
FB 0095 1J032I HARDWARE CRYPTO DEVICE DRIVER TERMINATED.
```

To restart the device driver, you can use the APSENSE command (as described in "Using the APSENSE Command to Refresh Your Hardware Crypto Configuration").

Using the APTRACE Command to Enable the Hardware Crypto Trace

You can use the APTRACE command to turn on/off the *internal trace facility* contained in z/VSE hardware Crypto support. The trace output is sent to the Operator Console. You can use these four trace settings:

0 Full trace including informational messages.
1 Show warning and error messages.
2 Show error messages only.
3 Trace off (**the default**).

Here is an example of the APTRACE command to activate *full trace* (by setting the trace level to zero):

```
msg fb,data=aptrace=0
AR 0015 1I40I  READY
FB 0011 1J034I CRYPTO TRACE LEVEL SET TO 0.
```

Using the APWAIT Command to Set the AP Polling Time Interval

Crypto cards process RSA encrypt and decrypt operations via an *asynchronous interface*. This means that:

1. Requests are enqueued to an Adjunct Processor (AP).
2. After a certain time interval, these requests will be dequeued.

You can use the APWAIT command to *specify the time interval* (in 1/300th seconds) from the time between:

1. Enqueuing a request into the internal processing queue of an AP.
2. The first attempt to dequeue a response.

Higher values will increase elapsed job time, but also decrease CPU time. Lower values will minimize elapsed job time, but might also increase CPU time significantly. The default value is 1/300th seconds. Valid values are 0 to 99.

Here is an example of the APWAIT command to set the AP wait-time interval to 2/300th seconds:

```
msg fb,data=apwait=2
AR 0015 1I40I  READY
FB 0011 1J038I POLLING TIME SET TO 2 * 1/300 SEC.
```

Using the APSTAT Command to Obtain Details about an AP

You can use the APSTAT command to display details about an Adjunct Processor (AP). The output differs depending upon whether the AP is configured in accelerator mode or coprocessor mode.

For coprocessors, the output values are described in the topic "ICSF Query Facility" of the manual *z/OS ICSF Application Programmer's Guide*, SA22-7522.

Here is an example of using the APSTAT command:

```
msg fb,data=apstat ap=8
FB 0114 Adapter Status of AP 8 (Coprocessor)
FB 0114 General:
FB 0114   AP type ................................ : CEX3C
FB 0114   Device status .......................... : ONLINE
FB 0114   Disabled by operator ................... : NO
FB 0114   Queue depth ............................ : 8
FB 0114   Crypto domain index .................... : 7
```

```
...
FB 0114 Crypto Facility information (Coprocessors only):
FB 0114   Active coprocessors on this card ....... : 1
FB 0114   DES hardware level ..................... : 0
FB 0114   RSA hardware level ..................... : 0
FB 0114   Power-On self-test firmware version .... : 011b057c
FB 0114   Coprocessor operating system name ...... : Linux
FB 0114   Coprocessor operating system version ... : 2.6
FB 0114   Coprocessor part number ................ : 45D5117
FB 0114   Coprocessor EC level ................... : 0G43192
FB 0114   Miniboot version ....................... : 000c0118
FB 0114   CPU speed in MHz ....................... : 400
FB 0114   EPROM mem size in 64kb increments ...... : 64
FB 0114   DRAM memory size in KB ................. : 131072
FB 0114   Battery backed memory in KB ............ : 4096
FB 0114   Unique serial number ................... : 99000422
FB 0114 End of Adapter status information.
```

Using 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.

Note: Console Interface Not Available! If you are using an ESM, you should be aware that you **cannot** use the Console Interface that is provided by the SECSERV job. As a result, you cannot use BSM commands together with your environment. For example, you **cannot** issue the APSENSE command to dynamically refresh your Crypto configuration.

Chapter 42. Preparing Your System to Use SSL

This chapter describes the steps you should follow to set up your z/VSE system so that you can implement Secure Sockets Layer (SSL) support.

After completing the steps in this chapter, you can then proceed to implement:

- Server authentication (described in Chapter 43, "Configuring for Server Authentication," on page 489).
- Client authentication (described in Chapter 44, "Configuring for Client Authentication," on page 505).

Note: Check Home Page for Latest Information! Before starting to configure your z/VSE system for SSL, you are advised to check the *z/VSE e-business Connectors and Utilities* home page for any new information relating to SSL. The URL is:

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

z/VSE provides *hardware crypto support* which requires a PCI Cryptographic Accelerator (PCICA) card or equivalent.

This chapter contains these main topics:

Related Topics:

For details of how to ...	Refer to the ...
set up SSL with: • Secure FTP • Secure Telnet • The VSE LDAP client • The Java-based connector • CICS Web Support (CWS) • WebSphere MQ for z/VSE • The VSE Script connector	• IBM Redbook *Security on IBM z/VSE*, SG24-7691. • technical articles located in the "Documentation" section of the z/VSE Homepage: `http://www.ibm.com/servers/eserver/zseries/zvse/documentation/`

After being implemented, server authentication can be used by:

- CICS Web Support,
- VSE/POWER networking,
- Secure FTP,
- Secure Telnet,

- Any other installed TCP/IP application that runs on z/VSE.

Step 1: Activate TCP/IP for VSE/ESA

When you activate *TCP/IP for VSE/ESA* you have access to all TCP/IP functions, including SSL support. Please note that for SSL support, you require the TCP/IP for VSE/ESA Application Pak.

For details of how to activate TCP/IP for VSE/ESA, refer to the *z/VSE TCP/IP Support*, SC34-2640.

The instructions for using SSL with z/VSE are provided in the *TCP/IP for VSE 1.5, Optional Features*, which you can obtain from either:

- The z/VSE Homepage (described in "Where to Find More Information" on page xxiii).
- Disk 3 (PDFs) of the *VSE Collection* Online Library, SK2T-0060.
- The *z/VSE Collection on DVD* Online Library, SK3T-8348, which contains all manuals (both PDFs and BOOK files).

Step 2: Create a Client Keyring File (KeyRing.pfx)

On the workstation from which you wish to implement server authentication, you must have installed a copy of the IBM-supplied client keyring file **KeyRing.pfx**. This file will be used for storing keys and certificates used by server authentication.

To obtain a copy of **KeyRing.pfx**, you must install the VSE Connector Client on your workstation, as described in the chapter "Installing and Operating the Java-Based Connector" in the *z/VSE e-business Connectors User's Guide*, SC34-2629.

During the installation of the VSE Connector Client, **KeyRing.pfx** is automatically stored in the directory:

```
\vsecon\samples
```

After being created during the installation, **KeyRing.pfx** contains a copy of the IBM-supplied sample root certificate, and a sample client certificate which has been signed by the root certificate. The file is initially protected by the password 'ssltest'.

To:

- change the password,
- manage the certificates stored in the client keyring file,

you use the IBM-supplied *Keyman/VSE* tool (described in "Step 3: Download and Customize the Keyman/VSE Tool" on page 479).

Step 3: Download and Customize the Keyman/VSE Tool

Note: JDK 1.5 or higher is required! Before you begin, you must already have installed the Java Development Kit (JDK) 1.5 or higher on the development platform where you plan to install the Keyman/VSE. If you do not have JDK 1.5 or higher installed, refer to the chapter “Installing the Common Prerequisite Programs” in the *z/VSE e-business Connectors User's Guide*, SC34-2629 for details of how to install it.

You can use the Keyman/VSE tool for most of the activities concerning SSL keys and certificates. You install it on a Java-enabled platform.

Obtaining a Copy of Keyman/VSE

To obtain a copy of Keyman/VSE, start your Web browser and go to URL:

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

From within the Keyman/VSE section, select **Details and Download**, then download the keyman zip file to the directory where you wish to install Keyman/VSE. The default is directory **c:\vkeyman** (for Windows) or **/vkeyman** (for Linux).

Performing the Installation of Keyman/VSE

To perform the installation of Keyman/VSE, you must:

1. Unzip the keyman zip file which contains these files:
   - setup.jar (contains the Keyman/VSE code)
   - setup.bat (an install batch file for Windows 2000/XP/Vista)
   - setup.cmd (an install batch file for Windows NT)
   - setup.sh (an install script for Linux/Unix)
2. Start the batch file (by double-clicking the file) that is applicable to your operating-system platform.
3. The installation process now begins, and you are guided through various installation menus.
4. To access the HTML-based documentation, you can now use your Web browser to open the file **vkeyman.html** in the **\vkeyman\help** subdirectory.

Customize the Keyman/VSE Settings

During the operation of Keyman/VSE, access is required from the workstation where you have installed Keyman/VSE, to the VSE Keyring Library stored on the z/VSE Host. You must customize Keyman/VSE so it can communicate with the z/VSE host, as described below.

Specify the Properties of Your z/VSE Host

Before you can start to create keys and certificates, you must specify the properties of your z/VSE host. To begin this action, click **VSE Host properties** on the toolbar, as shown in Figure 112 on page 480.

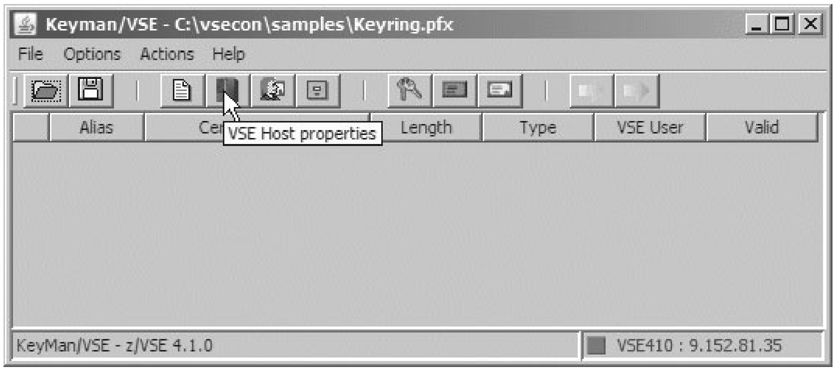


Figure 112. VSE Host Properties icon on Keyman/VSE Toolbar

The *VSE Host – Properties* window is then displayed. Click **New** and you can enter the details of your z/VSE host, as shown in Figure 113:

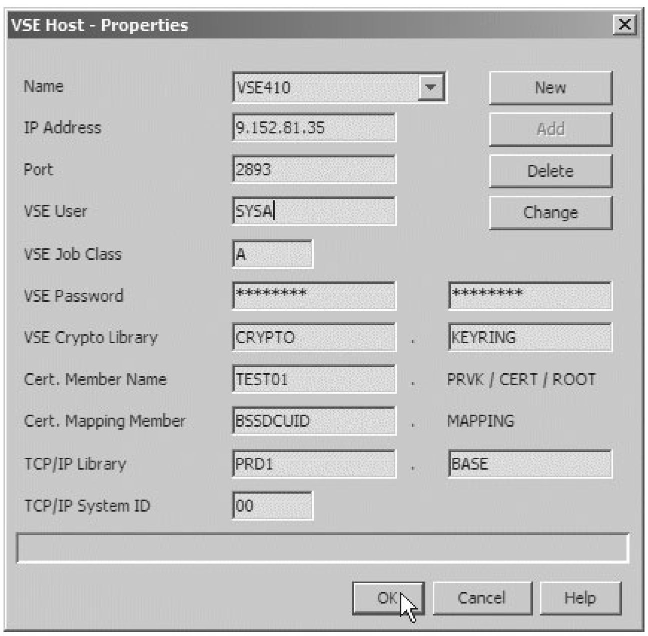


Figure 113. Using Keyman/VSE to Specify the Properties of the z/VSE Host

These are the fields you must complete in Figure 113:

Name The name you wish to use for the z/VSE host system that is specified by **IP Address** (below).

IP Address
The IP address of the z/VSE host.

Port The port number of the VSE Connector Server running on your z/VSE host (the default is 2893).

VSE User
A valid VSE User ID that has been defined using the Interactive Interface *Maintain User Profiles* dialog.

VSE Job Class
The VSE/POWER Job class which denotes the partition in which Jobs, submitted via Keyman/VSE, will run. These jobs catalog keyring members in CRYPTO.KEYRING.

VSE Password
The password used by **VSE User** to logon to the z/VSE host. You must enter this password twice.

VSE Crypto Library
The name of the library on the z/VSE host used for storing keyring members. Library CRYPTO.KEYRING is shown as default, since this library is automatically defined on IJSYSCT (the VSAM master catalog) during the installation of z/VSE. However, you can use your own library names.

Cert. Member Name
The name you wish to assign to each member of the keyring to be created. For example, if you enter MYNAME here, a keyring consisting of these members will be created in the VSE Crypto Library: MYNAME.PRVK (key pair), MYNAME.CERT (server certificate) and MYNAME.ROOT (root certificate).

Cert. Mapping Member
The name of the VSE library member containing the list of client certificates and their VSE User-ID mappings. The default is BSSDCUID. For details, see Chapter 45, "Implementing Client Authentication with VSE User-ID Mapping," on page 515.

TCP/IP Library
The library where TCP/IP is installed. The default is PRD1.BASE.

TCP/IP System ID
The System ID of the TCP/IP partition you wish to use together with Keyman/VSE. The default is 00.

When you have finished entering the details of your z/VSE host, click either:
- **OK** to store the details and complete this activity.
- **Add** to store the details and then start to add the details for another z/VSE host.

Specify the Properties of Your Local Keyring File

To customize Keyman/VSE so that it contains the properties of the IBM-supplied keyring file **Keyring.pfx** that is stored on the client, click **Local file properties** as shown in Figure 114.

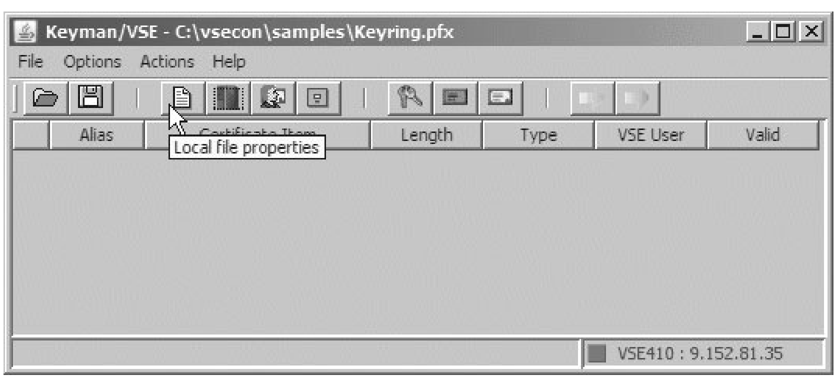


Figure 114. Local File Properties icon on Keyman/VSE Toolbar

The *Local Keyring File – Properties* window is then displayed as shown in Figure 115 on page 482.

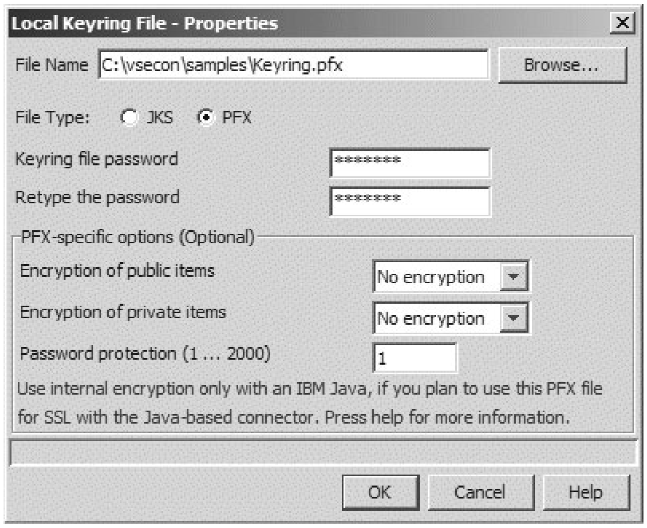


Figure 115. Entering the Properties of the Client Keyring File

These are the fields you must complete in Figure 115:

File Name
: The name of the client keyring file. The default is **Keyring.pfx**, which is the name of the IBM-supplied keyring file. However, you can enter your own file name which will then be used for creating a new client keyring file.

Keyring File Password
: The password you wish to assign to your client keyring file. Later, you will be prompted to enter this password when you:
  - Open the client keyring file.
  - Import the client keyring file into a Web browser.

Encryption of public items
: The algorithm you wish to use for the *internal* encryption of the public Certificate items (items which do not contain a private key).

Encryption of private items
: The algorithm you wish to use for the *internal* encryption of the private Certificate items (which do contain a private key).

Password protection
: The strength of security with which the password should be encrypted, where **1** is the weakest level and **2000** (the recommended setting) the strongest level of encryption. **Note:** If you have problems importing the keyring file into your Web browser, reduce the password encryption strength to **1**.

Step 4: Ensure That Your VSE Keyring Library Members Are Secure

You must ensure that the members that will be copied into the VSE Keyring Library (private keys, server certificates, and root certificates) are secure. To do so, use the:

1. z/VSE *Access Control Function*, which is part of the Basic Security Manager (BSM).
2. Access control table **DTSECTAB**. The default VSE Keyring Library (CRYPTO.KEYRING) is secured with all parts.

For details, refer to Chapter 31, “Overview of DTSECTAB-Based VSE Security,” on page 385.

In addition, you must:

- Ensure that your system is started with `SEC=YES` in the IPL SYS parameter.
- Ensure that your private key and server certificate have full read/write access protection.
- Include an ID statement in the startup job of each application that is to use certificates for SSL. The application might be a CICS application or the VSE Connector Server, for example.
- Submit the startup job for an application:
  – when security is active (that is, `SEC=YES` in the IPL SYS parameter).
  – using an *authorized* user (for example, the Administrator).
- Ensure that your VSE Keyring Library (the IBM-default is CRYPTO.KEYRING) cannot be accessed via FTP. For example, you should *not* specify the name of your VSE Keyring Library in any `DEFINE FILE` statement.

Getting Started Using the IBM-Supplied Keyring Set

To get started using SSL, you can use the IBM-supplied keyring set for testing and learning purposes. To do so, you simply have to run Job SKSSLKEY to catalog the IBM-supplied keyring set into the VSE Keyring Library on the z/VSE host. This keyring set consists of:

- A private key SAMPLE.PRVK
- A server certificate (which includes the public key) SAMPLE.CERT
- A root certificate SAMPLE.ROOT

Note: You can use this keyring set together with CICS Web Support and the z/VSE Java-based connector.

The VSE Keyring Library (CRYPTO.KEYRING) is *automatically* defined on IJSYSCT (the VSAM Master Catalog) during the installation of z/VSE.

After you have run Job SKSSLKEY, you can start to use this keyring set immediately. However, the length of the key pair in the IBM-supplied keyring set is 512 bits, which is *not* secure enough for applications that require high security.

```
* $$ JOB JNM=SETUPSSL,DISP=D,CLASS=0
// JOB SETUPSSL  DEFINE SSL SAMPLE ENVIRONMENT
* *******************************************************************
*
* STEP 1: CREATE RSA PRIVATE KEY 'SAMPLE.PRVK'
*
* *******************************************************************
// OPTION SYSPARM='00'             SYSID OF MAIN TCP/IP PARTITION
// LIBDEF PHASE,SEARCH=(PRD1.BASE)
// EXEC CIALPRVK,SIZE=CIALPRVK,PARM='CRYPTO.KEYRING.SAMPLE'
-----BEGIN RSA Private Key-----
hXNnvtgWEHuF4rhLWODrmJhG7yNyDYhXjTN1sALJEn2wCYsuaqhnmco5WbJOKdPe
g+oFi0o1MrsPQABoDtes/tNfMtTVzS6Vz/5EmpdrO0MlpNdK/QLdzyS5SgSSA0ZN
gWhVe3eY4+2FQb3x8D5pnjhGuMc3NzxZynBa2j+dz5ae8+nAH8qfQRsPfcXU715Y

:
-----END RSA Private Key-----
/*
* *******************************************************************
*
* STEP 2: CREATE VSE SERVER CERTIFICATE 'SAMPLE.CERT'
*
* *******************************************************************
// OPTION SYSPARM='00'             SysId of main TCP/IP partition
// LIBDEF PHASE,SEARCH=(PRD1.BASE)
// EXEC CIALCERT,SIZE=CIALCERT,PARM='CRYPTO.KEYRING.SAMPLE'
-----BEGIN CERTIFICATE-----
MIICJTCCAc8CBHiMye4wDQYJKoZIhvcNAQEFBQAwgZYxIDAeBgkqhkiG9w0BCQEW
EXZzZWVzYUBkZS5pYm0uY29tMQswCQYDVQQGEwJERTETMBEGA1UEBxMKQm9lYmxp
bmdlbjEUMBIGA1UEChMLSUJNIEdlcm1hbnkxGDAWBgNVBAsTD1ZTRSBEZXZlbG9w

:
-----END CERTIFICATE-----
/*
* *******************************************************************
*
* STEP 3: CREATE ROOT CERTIFICATE 'SAMPLE.ROOT'
*
* *******************************************************************
// OPTION SYSPARM='00'             SysId of main TCP/IP partition
// LIBDEF PHASE,SEARCH=(PRD1.BASE)
// EXEC CIALROOT,SIZE=CIALROOT,PARM='CRYPTO.KEYRING.SAMPLE'
-----BEGIN CERTIFICATE-----
MIICGzCCAcUCBHiLtz0wDQYJKoZIhvcNAQEFBQAwgZYxIDAeBgkqhkiG9w0BCQEW
EXZzZWVzYUBkZS5pYm0uY29tMQswCQYDVQQGEwJERTETMBEGA1UEBxMKQm9lYmxp
bmdlbjEUMBIGA1UEChMLSUJNIEdlcm1hbnkxGDAWBgNVBAsTD1ZTRSBEZXZlbG9w

:
-----END CERTIFICATE-----
/*
/&
* $$ EOJ
```

Figure 116. Job SKSSLKEY to Catalog a Sample Keyring Set into the VSE Keyring Library

For production purposes, you should use the Keyman/VSE tool to create:
- one keyring set to be used by the CICS Transaction Server for VSE/ESA.
- one keyring set to be used by the z/VSE Java-based connector. It is recommended that you use a different keyring set to the keyring set used by the CICS Transaction Server for VSE/ESA.

For details of how to do so, see "Configuring for Server Authentication Using CA-Signed Certificates" on page 494.

Currently-Supported SSL Cipher Suites

You define the SSL Version using SSLVERSION, which is contained in the:

- Java properties object of the VSE Connector Client (see Figure 122 on page 501).
- Java properties file of the VSE Connector Client (see Figure 123 on page 502).
- SSL configuration file of the VSE Connector Server (see Figure 120 on page 499).

Table 15 shows the SSL cipher suites that are currently supported. It represents the format you use when defining these cipher suites for the *VSE Connector Client*:

- For the format you use when defining these cipher suites for the *VSE Connector Server*, see Figure 120 on page 499.

Table 15. Currently Supported SSL Cipher Suites

Hex Code	Cipher Suite	Encryption	See Notes...
01	SSL_RSA_WITH_NULL_MD5	No	1, 2, 3
02	SSL_RSA_WITH_NULL_SHA	No	1, 2, 3
08	SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	40-bit	1, 2, 4
09	SSL_RSA_WITH_DES_CBC_SHA	56-bit	1, 4
0A	SSL_RSA_WITH_3DES_EDE_CBC_SHA	168-bit	1, 4
62	RSA1024_EXPORT_DESCBC_SHA	56-bit	1, 3, 5
2F	TLS_RSA_WITH_AES_128_CBC_SHA	128-bit	1, 6
35	TLS_RSA_WITH_AES_256_CBC_SHA	256-bit	1, 6, 7

Note:

1. The cipher suites NULL_MD5 (X'01'), NULL_SHA (X'02'), and RSA1024_EXPORT_DESCBC_SHA (X'62') require the *SSL 3.0 handshaking*. They cannot be used with TLS 1.0 handshaking.
2. The cipher suites SSL_RSA_EXPORT_WITH_DES40_CBC_SHA (X'08'), SSL_RSA_WITH_DES_CBC_SHA (X'09'), and SSL_RSA_WITH_3DES_EDE_CBC_SHA (X'0A') can be used with *both* SSL 3.0 and TLS 1.0, handshaking.
3. The cipher suite RSA1024_EXPORT_DESCBC_SHA (X'62') is *not* supported by the VSE Connector Client. For a detailed description of the SSL V3 protocol and supported cipher suites, refer to the *SSL Protocol Version 3.0*.
4. The 2048-bit SSL handshaking additionally requires a Crypto Express2 and later card and the latest version of TCP/IP for VSE/ESA. On a System z9® server or equivalent, AES-128 can be transparently performed using the CPACF hardware crypto feature. For details, refer to the chapter “Hardware Support” of the *z/VSE Planning*, SC34-2635.
5. See also “Obtaining Unlimited-Strength Jurisdiction Policy Files” on page 486.

Obtaining Unlimited-Strength Jurisdiction Policy Files

By default, your Java installation does *not* support AES with key sizes that are *greater than 128 bits*. However, to use *AES-256* you need *unlimited strength* cryptography.

Due to import-control restrictions imposed by some countries, the jurisdiction policy files shipped with Java only permit strong cryptography to be used. An unlimited strength version of these files (that is, with no restrictions on cryptographic strength) is available for download on this Web page:

`http://www.oracle.com/technetwork/java/javase/tech/`

To activate unlimited strength cryptography in Java:

1. Replace the files `local_policy.jar` and `US_export_policy.jar` in the directory `...\lib\security` of your Java installation.
2. Restart your Java application.

The same files can also be used to activate unlimited strength cryptography for an *IBM Java*.

Differences in SSL Support Provided by IBM/Oracle JDKs

Java provides its SSL functionality through the JSSE ("Java Secure Socket Extension") package.

The Java Development Kit (JDK) 1.3 provided *separate* class libraries for the JSSE package. The VSE Connector Client could therefore include a class library **ibmjsse.jar** which implemented SSL connectivity to a VSE system.

However, *from JDK 1.4 onwards* the JSSE functionality is *fully integrated* into Java itself. As a result, the SSL implementations from IBM and Oracle are in some cases different:

- The current Oracle JDK 1.4.1 and 1.4.2 implementations *cannot* read a PKCS12-encoded SSL keyring file. This keyring file is used in some of this manual's scenarios that use SSL.
- The IBM JDK 1.4.n implementation *can* read a PKCS12-encoded SSL keyring file (providing the prerequisites have been fulfilled).

As a result:

- If you are using JDK 1.3, you **must** use the IBM-provided ibmjsse.jar file (that is, it must be in your classpath).
- If you are using JDK 1.4 or higher, you are **not required** to have ibmjsse.jar in your Java classpath. However, if you do then you should check the tables below for possible problems that might occur.
- However, the restrictions described in this topic might not apply to future releases of the JDK from IBM and/or Oracle.

The following tables show the relationships between different Java (JDK) versions and SSL functionality.

Table 16 on page 487 shows the SSL certificate-support provided by the JSSE packages supplied with IBM's JDK 1.3 and 1.4.

Table 16. SSL Certificate-Support Provided by IBM's JSSE Packages

Java Version (IBM)	Root Certificate and Server Certificate in Keyring File?	Only Root Certificate in Keyring File?
JDK 1.3	Yes	No (A GUI dialog, in which users can accept/reject a server certificate during SSL handshake, **cannot** be displayed. This is required when a local copy of the server certificate is not present in the Keyring file.)
JDK 1.4	Yes	Yes

Table 17 shows the SSL certificate-support provided by the JSSE packages supplied with Oracle' JDK 1.3, 1.4, and 1.5 Beta 2.

Table 17. SSL Certificate-Support Provided by Oracle's JSSE Packages

Java Version (Oracle)	Root Certificate and Server Certificate in Keyring File?	Only Root Certificate in Keyring File?
JDK 1.3	Yes	Yes
JDK 1.4	Yes	Yes
JDK 1.5	Yes	Yes
JDK 1.6	Yes	Yes

You usually use a key management tool (such as Keyman/VSE or keytool.exe) to define the PKCS12 properties of your local keyring file, including the:
- Encryption of public and private items (certificates) in the keyring file.
- Strength of the file's password encryption.

Using the Keyman/VSE tool, you define the above properties in the window "Local Keyring File – Properties".

Table 18 shows the SSL encryption-support provided by the PKCS12 files supplied with IBM's JDK 1.3 and 1.4 .

Table 18. PKCS12 Encryption-Support Provided by IBM's JDKs

	Encryption of Public and/or Private Items (Certificates)			
Java Version (IBM)	**None**	**DES (168-bit)**	**RC4 (128-bit or 40-bit)**	**RC2 (128-bit or 40-bit)**
JDK 1.3	Yes	Yes	Yes	Yes
JDK 1.4	Yes	No	Yes	Yes

Table 19 shows the SSL encryption-support provided by the PKCS12 files supplied with Oracle's JDK 1.3, 1.4, and 1.5 Beta 2.

Table 19. PKCS12 Encryption-Support Provided by Oracle's JDKs

	Encryption of Public and/or Private Items (Certificates)			
Java Version (Oracle)	**None**	**DES (168-bit)**	**RC4 (128-bit or 40-bit)**	**RC2 (128-bit or 40-bit)**
JDK 1.3	Yes	Yes	Yes	Yes
JDK 1.4	Yes	No	No	No
JDK 1.5	Yes	No	No	No

Table 19. PKCS12 Encryption-Support Provided by Oracle's JDKs (continued)

	Encryption of Public and/or Private Items (Certificates)			
Java Version (Oracle)	**None**	**DES (168-bit)**	**RC4 (128-bit or 40-bit)**	**RC2 (128-bit or 40-bit)**
JDK 1.6	Yes	No	No	No

SSL Examples Provided With the Online Documentation

You might also refer to the SSL examples provided with the VSE Connector Client's online documentation:

- **SSLApiExample** shows how to code a Java application to connect to the VSE Connector Server via SSL. **Note:** You can also find a step-by-step description of this example in the online documentation (refer to “Using the Online Documentation Option” in the *z/VSE e-business Connectors User’s Guide*, SC34-2629).
- **SSLConsoleExample** shows how to connect via SSL, to submit a console command, and then obtain the resulting console messages.

Both of these examples are ready-to-run, and use the IBM-provided Client Keyring File (**Keyring.pfx**). In addition, you must also have either:

- Submitted job SKSSLKEY to catalog the corresponding entries in the VSE Keyring Library (CRYPTO.KEYRING).
- Created your own keyrings.

For details, see “Getting Started Using the IBM-Supplied Keyring Set” on page 483.

You can find the complete Java source code for the above SSL examples in the `Samples` sub-directory of the directory where you installed the VSE Connector Client. For details of how to use the online documentation, refer to “Using the Online Documentation Option” in the *z/VSE e-business Connectors User’s Guide*, SC34-2629.

Chapter 43. Configuring for Server Authentication

This chapter describes the steps you take to configure your z/VSE system for Secure Sockets Layer (SSL) *server authentication*. For most applications, server authentication provides a sufficient level of SSL security, and means that a server certificate is provided by the server (in this case, the z/VSE host) to authenticate the server to clients. However, if you require client authentication in addition to server authentication, also refer to the instructions provided in Chapter 44, "Configuring for Client Authentication," on page 505.

This chapter contains these main topics:

Note: Before starting this chapter, you should understand the differences between the SSL support offered by IBM's and Oracle's Java Development Kits. For details, see "Differences in SSL Support Provided by IBM/Oracle JDKs" on page 486.

Note: Support for 2048-Bit and 4096-Bit RSA Key Pairs! The example in this chapter is based upon the use of *1024-bit* RSA key pairs. However, you can use:

- *2048-bit* RSA key pairs if you are using:
  - A Crypto Express2 card or later.
  - The latest version of TCP/IP for VSE/ESA.
  - An IBM System z9 or later platform.
- *4096-bit* RSA key pairs if you are using:
  - A Crypto Express3 card or later.
  - The latest version of TCP/IP for VSE/ESA.
  - An IBM System z10 or later platform.

Configuring for Server Authentication Using Self-Signed Certificates

To configure your z/VSE system for server authentication that uses your self-signed *server certificates*, you should follow the steps given below.

Note: Certificates that you sign using your own root certificate are not usable outside of your own company's test or intranet environment.

Prerequisites:

- You must have completed all the steps described in Chapter 42, "Preparing Your System to Use SSL," on page 477.
- The VSE Connector Server must be running on the z/VSE host in **non-SSL mode** (for details, refer to the chapter "Installing and Operating the Java-Based Connector" in the *z/VSE e-business Connectors User's Guide*, SC34-2629).

- You must have installed the Java Development Kit (JDK) 1.4 or higher on the platform where you will use the Keyman/VSE (for details, refer to the chapter "Installing the Common Prerequisite Programs" in the *z/VSE e-business Connectors User's Guide*, SC34-2629).

1. **Create an RSA Key Pair**
   a. On the Keyman/VSE toolbar click **Generate new RSA key pair** as shown in Figure 117.

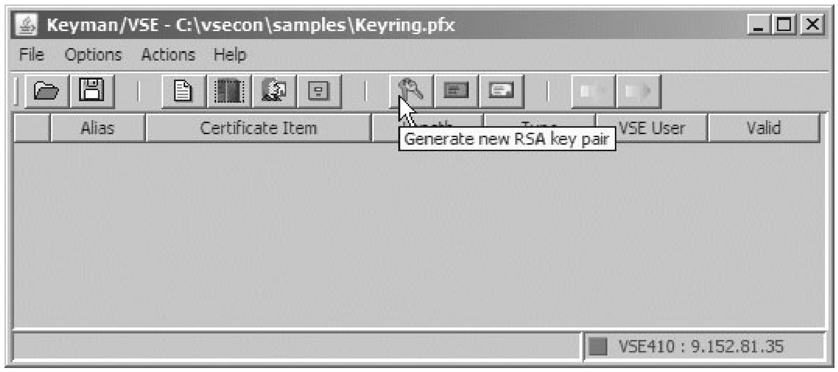


Figure 117. Generate RSA Key Pair icon on Keyman/VSE Toolbar

   b. Select a key length of **1024**, and click **Generate key** (key lengths of 512 bits are not secure). The entry is then created and displayed. You can examine the properties of this RSA Key Pair entry by double-clicking it.

2. **Create a Self-Signed Root Certificate**
   a. On the Keyman/VSE toolbar click **Generate ROOT certificate** as shown in Figure 118. The *Enter Personal Information for ROOT Cert* window is then displayed.

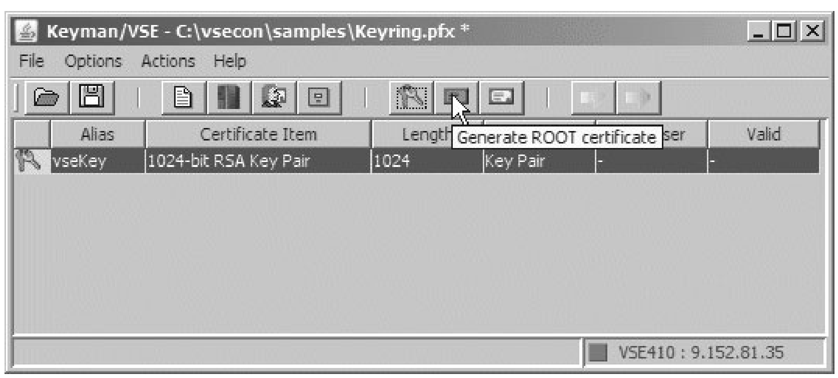


Figure 118. Generate ROOT Certificate icon on Keyman/VSE Toolbar

   b. Complete the fields in this window (make sure you select a key length of 1024).
   c. Click **Generate cert** to create your root certificate. The self-signed root certificate is then created and added to the list of displayed Certificate items. You can examine the properties of this root certificate by double-clicking it.

3. **Create a Request for a VSE Server Certificate**In this step, a request for a server certificate is created using the RSA key pair that was previously created in Step 1.
   a. Select the entry **1024-bit RSA Key Pair**, and then display the pop-up menu by clicking the entry using the right mouse-button. Select **Create VSE server cert request**.
   b. Complete the fields in this window and click **Generate** to create your certificate request. The **1024-bit Certificate Request** is then created and added to the list of displayed Certificate items.

4. **Sign the Request for a VSE Server Certificate**In this step, the request for a server certificate is signed using the previously-created root certificate (using the Clipboard).
   a. Select the entry **1024-bit Certificate Request**, and then display the pop-up menu by clicking the entry using the right mouse-button. Select **Copy to clipboard**.
   b. Select your root certificate from the displayed list. Next, display the pop-up menu by clicking the root certificate using the right mouse-button. Select **Sign certificate request**. The *Sign Certificate Request* window is then displayed.
   c. Paste the certificate request from the Clipboard into this window, using the usual keyboard combination (for example, **Shift + Insert**).
   d. Click **Generate**, and the generated VSE server certificate that has now been signed by your root certificate, is added to the list of displayed Certificate items.
   e. The **1024-bit Certificate Request** entry is no longer required, and you should delete it by selecting this 1024-bit certificate request and then pressing the **Del** key.

5. **Upload the 1024-Bit RSA Key Pair to the z/VSE Host**In this step, the RSA key pair is uploaded to the z/VSE host and cataloged into the VSE Keyring Library (CRYPTO.KEYRING), where it will have the suffix *.PRVK*.
   a. Select the **1024-bit RSA Key Pair** from the displayed list. Next, display the pop-up menu by clicking the **1024-bit RSA Key Pair** entry using the right mouse-button. Select **Upload to VSE**. The *Send Certificate Item to VSE* window is then displayed.
   b. The values displayed in this window are taken from your Host settings (described in "Specify the Properties of Your z/VSE Host" on page 479). You can change them here if you wish. When you are ready, click **Upload**. The CIALSRVR utility is started on the z/VSE host, which catalogs the RSA key pair in the VSE Keyring Library.
   c. On the *Send Certificate Item to VSE* window you will now see status messages. On the z/VSE console you will see messages similar to those below:

```
R1 0045 IESC1023I CLIENT CONNECTED FROM IP:   9.164.183.168
R1 0045 IESC1028I CLIENT SESSION ESTABLISHED FOR USER: JSCH
BG 0001 1Q47I   BG CIALSRVR 01148 FROM (JSCH) , TIME=11:35:12
BG 0000 // JOB CIALSRVR
        DATE 03/04/2006, CLOCK 11/35/12
BG 0000 CIALSRVR 01.05 A  12/23/02 12.55
BG 0000 Default password phrase will be used
BG 0000 SETPORT 6045
BG 0000 Waiting for PC to send rsa private key.
```

```
BG 0000 1024-bit RSA key written into CRYPTO .KEYRING .JSCH01  .PRVK
BG 0000 EOJ CIALSRVR  MAX.RETURN CODE=0000
        DATE 03/04/2006, CLOCK 11/35/18, DURATION   00/00/06
BG 0001 1Q34I   BG WAITING FOR WORK
```

6. **Upload the Root Certificate to the z/VSE Host**In this step, the self-signed root certificate is uploaded to the z/VSE host and cataloged into the VSE Keyring Library (CRYPTO.KEYRING), where it will have the suffix *.ROOT*.
   a. Select your root certificate from the displayed list. Next, display the pop-up menu by clicking the root certificate entry using the right mouse-button. Select **Upload to VSE**. The *Send Certificate Item to VSE* window is then displayed.
   b. The values displayed in this window are taken from your Host settings (described in "Specify the Properties of Your z/VSE Host" on page 479). You can change them here if you wish. When you are ready, click **Upload**. The CIALROOT utility is started on the z/VSE host, which catalogs the root certificate in the VSE Keyring Library.
   c. On the *Send Certificate Item to VSE* window you will now see status messages. On the z/VSE console you will see messages similar to those below:

```
BG 0001 1Q47I   BG CIALROOT 01149 FROM (JSCH) , TIME=12:16:32
BG 0000 // JOB CIALROOT
        DATE 03/04/2006, CLOCK 12/16/32
BG 0000 CIALROOT 01.05 A  12/23/02 12.55
BG 0000 EOJ CIALROOT  MAX.RETURN CODE=0000
```

7. **Upload the Server Certificate to the z/VSE Host**In this step, the self-signed server certificate is uploaded to the z/VSE host and cataloged into the VSE Keyring Library (CRYPTO.KEYRING), where it will have the suffix *.CERT*.
   a. Select your server certificate from the displayed list. Next, display the pop-up menu by clicking the server certificate entry using the right mouse-button. Select **Upload to VSE**. The *Send Certificate Item to VSE* window is then displayed.
   b. The values displayed in this window are taken from your Host settings (described in "Specify the Properties of Your z/VSE Host" on page 479). You can change them here if you wish. When you are ready, click **Upload**. The CIALCERT utility is started on the z/VSE host, which catalogs the server certificate in the VSE Keyring Library.
   c. On the *Send Certificate Item to VSE* window you will now see status messages. On the z/VSE console you will see messages similar to those below:

```
BG 0000 // JOB CIALCERT
        DATE 03/04/2006, CLOCK 12/18/22
BG 0000 CIALCERT 01.05 A  12/23/02 12.54
BG 0000 EOJ CIALCERT  MAX.RETURN CODE=0000
        DATE 03/04/2006, CLOCK 12/18/22, DURATION   00/00/00
BG 0001 1Q34I   BG WAITING FOR WORK
```

8. **Save the Client Keyring File**Since the RSA key pair, root certificate, and server certificate have been uploaded to the z/VSE host and cataloged into the VSE Keyring Library (CRYPTO.KEYRING), you can now delete certificates items that are no longer required from your client keyring file **KeyRing.pfx**, and then save this file.

- You *must* keep the self-signed root certificate in the client keyring file, so that the client can verify server certificates sent to it from the z/VSE host during the SSL handshake.
- You can keep the self-signed server certificate in the client keyring file, so that this server certificate can be compared with the server certificate sent to it from the z/VSE host during the SSL handshake. This avoids having to store the server certificate sent from the z/VSE host at the start of each communications session.
- You can delete the other Certificate items (such as the 1024-bit RSA key pair, or certificate requests).

After you have deleted the unwanted Certificate items, select **File** and then **Save keyring file** from the Menu bar. Enter your password and click **OK**.

9. **Import the Client Keyring File into a Web Browser (CICS Web Support Only)**This step is only required when configuring for server authentication in *CICS Web Support*. In this step you import into *each* of your CICS clients (Web browsers) the client keyring file that you have saved in Step 8.
   - For *Microsoft Internet Explorer* Web browsers, you double-click **KeyRing.pfx** and the Internet Explorer is started. The *Certificate Manager Import Wizard* window is then displayed, with general information. Now follow the instructions provided for adding the client keyring file to your certificate store. During the procedure, you will be requested to enter the:
     a. File you wish to import (for example **c:\vsecon\samples\KeyRing.pfx**).
     b. Password that you specified when you saved the client keyring file in Step 8.
     c. Certificate store (the system area where certificates are stored). You can either define your own store, or let the system automatically select a store. This is the recommended option.
   - For *Netscape* and *Mozilla* Web browsers, you start your Web browser, and then select **Edit — Preferences — Certificates**. Now follow the instructions for importing your client keyring file. During the procedure, you will be requested to enter a password with which you can protect the client keyring file.
   - If you receive the message “The data cannot be decrypted because it was encrypted using an algorithm or key size which is not allowed by this configuration”, then the options set in the client keyring file prevent this file from being imported into your Web browser. Check and amend the PKCS#12 options using the Keyman/VSE tool.

10. **Verify Your Certificates in the VSE Keyring Library**To verify that your root certificate and server certificate are valid and correct, you can submit the job CIALSIGV.JCL on the z/VSE host. In this job, you specify the name of the root and server certificates.

Configuring for Server Authentication Using CA-Signed Certificates

To configure your z/VSE system for server authentication that uses certificates signed by a Certificate Authority (CA), you should follow the steps given below.

Note: Certificates that are signed by a CA can be used throughout the internet for use with customers and business partners.

Prerequisites:
- You must have completed all the steps described in Chapter 42, "Preparing Your System to Use SSL," on page 477.
- The VSE Connector Server must be running on the z/VSE host in **non-SSL mode** (for details, refer to the chapter "Installing and Operating the Java-Based Connector" in the *z/VSE e-business Connectors User's Guide*, SC34-2629).
- You must have installed the Java Development Kit (JDK) 1.4 or higher on the platform where you will use the Keyman/VSE (for details, refer to the chapter "Installing the Common Prerequisite Programs" in the *z/VSE e-business Connectors User's Guide*, SC34-2629).

1. **Create an RSA Key Pair**
   a. On the Keyman/VSE toolbar click **Generate new RSA key pair** as shown in Figure 117 on page 490.
   b. Select a key length of **1024**, and click **Generate key** (key lengths of 512 bits are not secure). The entry is then created and displayed. You can examine the properties of this RSA Key Pair entry by double-clicking it.

2. **Create a Request for a VSE Server Certificate**In this step, a request for a server certificate is created using the RSA key pair that was previously created in Step 1.
   a. Select the entry **1024-bit RSA Key Pair**, and then display the pop-up menu by clicking the entry using the right mouse-button. Select **Create VSE server cert request**. The *Enter Your Personal Information* window is then displayed.
   b. Complete the fields in this window and click **Generate** to create your certificate request. The **1024-bit Certificate Request** is then created and added to the list of displayed Certificate items.

3. **Request a VSE Server Certificate**In this step, the request for a server certificate is copied to the Clipboard so that it can be signed by a CA. Select the entry **1024-bit Certificate Request**, and then display the pop-up menu by clicking the entry using the right mouse-button. Select **Copy to clipboard**.

4. **Proceed To a CA Web Page for Obtaining Free Certificates**You must now submit your certificate request to any Certificate Authority to be signed. This example shows you how to obtain a free signed server certificate from the Thawte Corporation.
   a. Start your Web browser and proceed to the Thawte Corporation Web site:
      ```
      https://www.thawte.com/cgi/server/test.exe
      ```
      You must firstly register with Thawte. Then, you given instructions on how to proceed to the Web page for obtaining test server certificates, which are free.

Note:
1) The server certificate is only valid for a limited number of days.
2) You also need a root certificate from Thawte that can be used with the server certificate. You must manually insert this root certificate into your Web browser, as described in Step 6.

5. **Obtain a Signed Server Certificate and Copy into Your Client Keyring File**
   a. Paste your server certificate request from the Clipboard into the appropriate area of the *Test Thawte Certificates* window. Complete the other details, such as:
      - For *Type of test certificate*, select **Test X509v3 SSL Cert**.
      - For *Format for Chained CA Certs*, select **"Use the standard format"** (the BASE64 encoding of an X.509v3 certificate).
   b. Click **Generate Test Certificate**. The output from this request is a signed server certificate that is displayed under the heading **Here is your certificate:**. An example is shown in Figure 119.

```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmgAwIBAgIDBYAlMA0GCSqGSIb3DQEBBAUAMIGHMQswCQYDVQQGEwJa
QTEiMCAGA1UECBMZRk9SIFRFU1RJTkcgUFVSUE9TRVMgT05MWTEdMBsGA1UEChMU
VGhhd3RlIENlcnRpZmljYXRpb24xFzAVBgNVBAsTDlRFU1QgVEVTVCBURVNUMRww
:
-----END CERTIFICATE-----
```

Figure 119. A Thawte Signed Server Certificate

   c. Using your mouse, select the server certificate. Include the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.
   d. Copy the server certificate to your Clipboard, using the usual keyboard combination (for example, **Control + Insert**).
   e. Paste the server certificate from your Clipboard into the Keyman/VSE window, by selecting **File** and then **Read clipboard** from the Keyman/VSE Menu bar. The server certificate is then added to the list of displayed Certificate items.

6. **Obtain the CA's Public Root Certificate and Copy into Your Client Keyring File**You must now obtain a root certificate from the Certificate Authority. This root certificate is a “public” root certificate. This means, you cannot use it for signing other certificates (as you can for self-signed certificates). Instead, this “public” root certificate is used during the SSL handshake to verify certificates that you receive (to check the identity of the sender).
   a. Proceed again to the Thawte Corporation Web site:
      ```
      https://www.thawte.com/cgi/server/test.exe
      ```
      and follow the instructions on how to proceed to the Web page for obtaining the Thawte public root certificate, which is free.
   b. Using your mouse, select the root certificate. Include the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.
   c. Copy the root certificate to your Clipboard, using the usual keyboard combination (for example, **Control + Insert**).

d. Paste the root certificate from your Clipboard into the Keyman/VSE window, by selecting **File** and then **Read clipboard** from the Keyman/VSE Menu bar. The root certificate is then added to the list of displayed Certificate items.

7. **Upload the 1024-Bit RSA Key Pair to the z/VSE Host**In this step, the RSA key pair is uploaded to the z/VSE host and cataloged into the VSE Keyring Library (CRYPTO.KEYRING), where it will have the suffix *.PRVK*.
   a. Select the **1024-bit RSA Key Pair** from the displayed list. Next, display the pop-up menu by clicking the **1024-bit RSA Key Pair** entry using the right mouse-button. Select **Upload to VSE**. The *Send Certificate Item to VSE* window is then displayed.
   b. The values displayed in this window are taken from your Host settings (described in "Specify the Properties of Your z/VSE Host" on page 479). You can change them here if you wish. When you are ready, click **Upload**. The CIALSRVR utility is started on the z/VSE host, which catalogs the RSA key pair in the VSE Keyring Library.
   c. On the *Send Certificate Item to VSE* window you will now see status messages. On the z/VSE console you will see messages similar to those below:

```
R1 0045 IESC1023I CLIENT CONNECTED FROM IP:   9.164.183.168
R1 0045 IESC1028I CLIENT SESSION ESTABLISHED FOR USER: JSCH
BG 0001 1Q47I   BG CIALSRVR 01148 FROM (JSCH) , TIME=11:35:12
BG 0000 // JOB CIALSRVR
        DATE 03/04/2006, CLOCK 11/35/12
BG 0000 CIALSRVR 01.05 A  12/23/02 12.55
BG 0000 Default password phrase will be used
BG 0000 SETPORT 6045
BG 0000 Waiting for PC to send rsa private key.
BG 0000 1024-bit RSA key written into CRYPTO .KEYRING .JSCH01  .PRVK
BG 0000 EOJ CIALSRVR  MAX.RETURN CODE=0000
        DATE 03/04/2006, CLOCK 11/35/18, DURATION   00/00/06
BG 0001 1Q34I   BG WAITING FOR WORK
```

8. **Upload the CA's Public Root Certificate to the z/VSE Host**In this step, the public root certificate is uploaded to the z/VSE host and cataloged into the VSE Keyring Library (CRYPTO.KEYRING), where it will have the suffix *.ROOT*.
   a. Select the CA's root certificate from the displayed list. Next, display the pop-up menu by clicking the root certificate entry using the right mouse-button. Select **Upload to VSE**. The *Send Certificate Item to VSE* window is then displayed.
   b. The values displayed in this window are taken from your Host settings (described in "Specify the Properties of Your z/VSE Host" on page 479). You can change them here if you wish. When you are ready, click **Upload**. The CIALROOT utility is started on the z/VSE host, which catalogs the root certificate in the VSE Keyring Library.
   c. On the *Send Certificate Item to VSE* window you will now see status messages. On the z/VSE console you will see messages similar to those below:

```
BG 0001 1Q47I   BG CIALROOT 01149 FROM (JSCH) , TIME=12:16:32
BG 0000 // JOB CIALROOT
        DATE 03/04/2006, CLOCK 12/16/32
BG 0000 CIALROOT 01.05 A  12/23/02 12.55
BG 0000 EOJ CIALROOT  MAX.RETURN CODE=0000
```

9. **Upload the CA-Signed Server Certificate to the z/VSE Host**In this step, the CA-signed server certificate is uploaded to the z/VSE host and cataloged into the VSE Keyring Library (CRYPTO.KEYRING), where it will have the suffix *.CERT*.
   a. Select the CA-signed server certificate from the displayed list. Next, display the pop-up menu by clicking the server certificate entry using the right mouse-button. Select **Upload to VSE**. The *Send Certificate Item to VSE* window is then displayed.
   b. The values displayed in this window are taken from your Host settings (described in "Specify the Properties of Your z/VSE Host" on page 479). You can change them here if you wish. When you are ready, click **Upload**. The CIALCERT utility is started on the z/VSE host, which catalogs the server certificate in the VSE Keyring Library.
   c. On the *Send Certificate Item to VSE* window you will now see status messages. On the z/VSE console you will see messages similar to those below:

```
BG 0000 // JOB CIALCERT
        DATE 03/04/2006, CLOCK 12/18/22
BG 0000 CIALCERT 01.05 A  12/23/02 12.54
BG 0000 EOJ CIALCERT  MAX.RETURN CODE=0000
        DATE 03/04/2006, CLOCK 12/18/22, DURATION   00/00/00
BG 0001 1Q34I   BG WAITING FOR WORK
```

10. **Save the Client Keyring File**Since the RSA key pair, root certificate, and server certificate have been uploaded to the z/VSE host and cataloged in the VSE Keyring Library, you can now delete certificates that are no longer required from your client keyring file **KeyRing.pfx**, and then save this file.
    - You *must* keep the CA's public root certificate in the client keyring file, so that the client can verify certificates sent to it from the z/VSE host during the SSL handshake.
    - You can keep the signed server certificate in the client keyring file, so that this server certificate can be compared with the server certificate sent to it from the z/VSE host during the SSL handshake. This avoids having to store the server certificate sent from the z/VSE host at the start of each communications session.
    - You can delete the other Certificate items (such as the 1024-bit RSA key pair, or certificate requests).

    After you have deleted the unwanted Certificate items, select **File** and then **Save keyring file** from the Menu bar. Enter your password and click **OK**.

11. **Import the Client Keyring File into a Web Browser (CICS Web Support Only)**This step is only required when configuring for server authentication in *CICS Web Support*. In this step you import into *each* of your CICS clients (Web browsers) the client keyring file that you have saved in Step 10.
    - For *Microsoft Internet Explorer* Web browsers, you double-click **KeyRing.pfx** and the Internet Explorer is started. The *Certificate Manager Import Wizard* window is then displayed, with general information. Now follow the instructions provided for adding the client keyring file to your certificate store. During the procedure, you will be requested to enter the:
      a. File you wish to import (for example **c:\vsecon\samples\KeyRing.pfx**).
      b. Password that you specified when you saved the client keyring file in Step 10.

c. Certificate store (the system area where certificates are stored). You can either define your own store, or let the system automatically select a store.

- For *Netscape* and *Mozilla* Web browsers, you start your Web browser, and then select **Edit — Preferences — Certificates**. Now follow the instructions for importing your client keyring file. During the procedure, you will be requested to enter a password with which you can protect the client keyring file.
- If you receive the message “The data cannot be decrypted because it was encrypted using an algorithm or key size which is not allowed by this configuration”, then the options set in the client keyring file prevent this file from being imported into your Web browser. Check and amend the PKCS#12 options using the Keyman/VSE tool.

12. **Verify Your Certificates in the VSE Keyring Library**To verify that your root certificate and server certificate are valid and correct, you can submit the job CIALSIGV.JCL on the z/VSE host. In this job, you specify the name of the root and server certificates.

Configuring the VSE Connector Server for Server Authentication

Note: Not Relevant for CICS Web Support! The information in this topic is not relevant for implementing server authentication in CICS Web Support.

This topic describes the steps you must follow for *each* VSE Connector Server installation, in order to implement server authentication.

Prerequisites: You must have completed the steps described in either “Configuring for Server Authentication Using Self-Signed Certificates” on page 489 or “Configuring for Server Authentication Using CA-Signed Certificates” on page 494.

You can run the VSE Connector Server either in SSL-mode, or in non-SSL-mode.

Note: You cannot have both SSL-mode and non-SSL-mode connections using the *same* VSE Connector Server. However you can, of course, run multiple VSE Connector Servers in different partitions on the z/VSE host, and with or without SSL.

This topic contains these sub-topics:
- “Step 1: Configure and Catalog the VSE Connector Server's SSL Profile”
- “Step 2: Activate SSL Profile in Main Configuration File” on page 499

Step 1: Configure and Catalog the VSE Connector Server's SSL Profile

Use job skeleton SKVCSSSL (in ICCF library 59) to configure and catalog the VSE Connector Server's SSL profile on the z/VSE host. Here is an example of skeleton SKVCSSSL:

```
; ******************************************************************
;      SSL CONFIGURATION MEMBER FOR VSE CONNECTOR SERVER
; ******************************************************************
; SSLVERSION   SPECIFIES THE MINIMUM VERSION THAT IS TO BE USED
;              POSSIBLE VALUES ARE:   SSL30 AND TLS31
; KEYRING      SPECIFIES THE SUBLIBRARY WHERE THE KEY FILES ARE
;              STORED.
; CERTNAME     NAME OF THE CERTIFICATE THAT IS USED BY THE SERVER
; SESSIONTIMEOUT NUMBER OF SECONDS THAT THE SERVER WILL USE TO
;              ALLOW A CLIENT TO RECONNECT WITHOUT PERFORMING A
;              FULL HANDSHAKE. (86440 SEC = 24 HOURS)
; AUTHENTICATION TYPE OF AUTHENTICATION. POSSIBLE VALUES ARE:
;              SERVER - SERVER AUTHENTICATION ONLY
;              CLIENT - SERVER AND CLIENT AUTHENTICATION
;              LOGON  - SERVER AND CLIENT AUTENTICATION WITH LOGON.
;                       THE CLIENT CERTIFICATE IS USED FOR THE LOGON.
; ******************************************************************
SSLVERSION     = SSL30
KEYRING        = CRYPTO.KEYRING
CERTNAME       = SAMPLE
SESSIONTIMEOUT = 86440
AUTHENTICATION = SERVER

; *****************************************************************
; CIPHERSUITES SPECIFIES A LIST OF CIPHER SUITES THAT ARE ALLOWED
; *****************************************************************
CIPHERSUITES =   ; COMMA SEPARATED LIST OF NUMERIC VALUES
               01,  ; RSA512_NULL_MD5
               02,  ; RSA512_NULL_SHA
               08,  ; RSA512_DES40CBC_SHA
               09,  ; RSA1024_DESCBC_SHA
               0A,  ; RSA1024_3DESCBC_SHA
               62,  ; RSA1024_EXPORT_DESCBC_SHA
               2F   ; TLS_RSA_WITH_AES_128_CBC_SHA
               35   ; TLS_RSA_WITH_AES_256_CBC_SHA
```

Figure 120. Skeleton SKVCSSSL (Configure SSL for the VSE Connector Server)

Note:

1. You specify cipher suites as a list of hex numbers.
2. For a complete list of supported cipher suites refer to Table 15 on page 485.
3. The cipher suite with hex number 62 (shown above) is currently not supported by the VSE Connector Client.

Step 2: Activate SSL Profile in Main Configuration File

Use job skeleton SKVCSCFG (in ICCF library 59) to enable SSL and catalog the SSL profile in the VSE Connector Server's main configuration file. Here is an example of skeleton SKVCSCFG:

```
⋮
   SERVERPORT   = 2893
   MAXCLIENTS   = 256
   SSLENABLE    = YES
⋮
   LIBRCFGFILE   = DD:PRIMARY.TEST(IESLIBDF.Z)
   USERSCFGFILE  = DD:PRIMARY.TEST(IESUSERS.Z)
   PLUGINCFGFILE = DD:PRIMARY.TEST(IESPLGIN.Z)
   SSLCFGFILE    = DD:PRIMARY.TEST(IESSSLCF.Z)
⋮
```

Figure 121. Skeleton SKVCSCFG (Activate SSL Profile for the VSE Connector Server)

Configuring Self-Written Clients for Server Authentication

Note: Not Relevant for CICS Web Support! The information in this topic is not relevant for implementing server authentication in CICS Web Support.

This topic describes the changes for SSL server-authentication support that you must make to:

- Self-written clients installed on either the physical/logical middle-tier of a 3-tier environment, or on workstations in a 2–tier environment.
- The SSL examples supplied with the VSE Connector Client (for details, see "SSL Examples Provided With the Online Documentation" on page 488).

Prerequisites: You must have completed the steps described in either "Configuring for Server Authentication Using Self-Signed Certificates" on page 489 or "Configuring for Server Authentication Using CA-Signed Certificates" on page 494.

This topic contains these main sub-topics:

- "Step 1: Set SSL Flag in Class VSEConnectionSpec"
- "Step 2: Configure SSL Profile" on page 501
- "Step 3: Copy a Server Certificate Into the Client Keyring File" on page 502

Step 1: Set SSL Flag in Class VSEConnectionSpec

From VSE/ESA 2.6 onwards, class *VSEConnectionSpec* of the VSE Connector Client's VSE Java Beans supports an SSL flag, which can be set for your user applications. When this flag is set, you must also specify additional SSL-related parameters. To do so, you can either:

- Define a Java properties object.
- Create a Java properties file to contain the required SSL parameters.

In the example shown in Figure 122 on page 501, the SSL parameters are set using a Java *properties object*.

```
⋮
try {
   spec = new VSEConnectionSpec(
                 InetAddress.getByName(ipAddr),
                 2893,userID,password);
}
catch (UnknownHostException e) { ... }

/* Specify secure SSL connection */
spec.setSSL(true);

/* Specify SSL properties */
sslProps = new Properties();
sslProps.put("SSLVERSION", "SSL");
sslProps.put("CIPHERSUITES",
     "SSL_RSA_WITH_NULL_MD5," +
     "SSL_RSA_WITH_NULL_SHA," +
     "SSL_RSA_EXPORT_WITH_DES40_CBC_SHA," +
     "SSL_RSA_WITH_DES_CBC_SHA," +
     "SSL_RSA_WITH_3DES_EDE_CBC_SHA");
sslProps.put("KEYRINGFILE", "c:\\vsecon\\KeyRing.pfx");
sslProps.put("KEYRINGPWD", "ssltest");
spec.setSSLProperties(sslProps);

/* Create VSE system instance with this connection */
system = new VSESystem(spec);

/* Connect to host */
system.connect();
⋮
```

Figure 122. Set SSL Parameters Using a Properties Object

Step 2: Configure SSL Profile

To configure the SSL profile, you must create a *Java properties file* containing these pairs of keys / values (an example of which is shown in Figure 123 on page 502:

Key **Value**

SSLVERSION
: SSL or TLS

CIPHERSUITES
: Specifies a list of symmetric encryption/decryption algorithms which the client can use to negotiate the cipher used in the related connection. For an SSL-connection to be established, the client and the VSE Connector Server must support the same cipher with the same encryption strength (40-bit, 56-bit, 128-bit, and so on). For a list of the currently-supported cipher suites, see Table 15 on page 485.

KEYRINGFILE
: Pathname of a *client keyring file* which is stored on either each Web client of a 2-tier environment or on the physical/logical middle-tier of a 3-tier environment, and is protected by a password. See “Specify the Properties of Your Local Keyring File” on page 481 for a description of the keyring file.

Note: The name that you define here must also be used when you create your client keyring file for the first time.

KEYRINGPWD
Password to protect the client keyring file. For the client keyring file automatically created during installation, "ssltest" is the pre-set password.

LOGONWITHCERT
YES or NO. An *implicit logon* is possible providing:
1. The Client Keyring File contains a client certificate which is *also* stored as a .CCERT member in the VSE Keyring Library on the z/VSE host.
2. The .CCERT member in the VSE Keyring Library has been mapped to a z/VSE User ID.

An implicit logon means that the client does not have to provide logon information (that is, a User ID and password) explicitly. If LOGONWITHCERT is set to YES, you must also specify AUTHENTICATION=LOGON in the VSE Connector Server's configuration file (see Figure 120 on page 499 for details).

You can use the hash character '**#**' as a comment delimiter.

Here is an example of a Java properties file:

```
SSLVERSION=SSL    # SSL or TLS
CIPHERSUITES=SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
KEYRINGFILE=c:\\vsecon\\keyring.pfx
LOGONWITHCERT=NO
```

Figure 123. Example of Java Properties File for the VSE Connector Client and VSE Navigator

Note:
1. You are not required to define *all* SSL parameters in your Java properties file.
2. If you do not define an SSL parameter in your Java properties file, you must ensure that your user application requests any missing information from the user, before connecting to the server.
3. In Figure 123, parameter KEYRINGPWD has not been specified.

Step 3: Copy a Server Certificate Into the Client Keyring File

A *server certificate* includes a public key, and is provided by the server (the z/VSE host) to authenticate the server to clients (VSE Connector Clients). If you copy a server certificate into each client keyring file, this will avoid the user having to store this certificate on the first occasion when it is sent from the server.

Decide if you wish to use the IBM-supplied sample server certificate with each VSE Connector Client, or if instead you wish to use your *own* server certificate. Depending upon your decision, you must copy either:
- the IBM-supplied sample server certificate (`SAMPLE.CERT`),
- your own server certificate,

from the VSE Keyring Library on the z/VSE host, into the client keyring file (**KeyRing.pfx**) located on either:
- Each Web client of a 2-tier environment.
- The physical/logical middle-tier of a 3-tier environment.

For 2-tier environments, if a Web client does not contain a copy of the server certificate and communicates with the z/VSE host, the user will be prompted to decide if a copy of the server certificate should be stored on the Web client. You can therefore either:
- Provide each Web client with a copy of the server certificate before starting.
- Let each Web client take a copy during the first processing that takes place.

Summary of Server Authentication Tasks for the Java-Based Connector

Note: Not Relevant for CICS Web Support! The information in this topic is not relevant for implementing server authentication in CICS Web Support.

Table 20 provides a summary of the tasks you must carry out to implement server authentication for the Java-based connector (that is, for VSE Connector Clients and the VSE Connector Server).

Table 20. Tasks Involved in Configuring the Java-Based Connector for Server Authentication

	Server Authentication
On the z/VSE Host Side	Set AUTHENTICATION = SERVER in skeleton SKVCSSSL.
On the Client Side	Set LOGONWITHCERT = NO in member **ssl.prop**.
Certificates	Do not use Job SKSSLKEY to catalog the 1024–bit sample private key and certificates. Instead, use Keyman/VSE to create your own 1024-bit or 2048-bit keys and certificates. **Note:** 512-bit keys are not secure.
Using Keyman/VSE	Use the wizard dialog **Actions –> Create CA-signed Keyring** to create the complete keyring. All certificates are therefore signed by an external Certificate Authority (CA) such as Thawte. However, you will be charged by Thawte for these actions.
Client Keyring File	Your client keyring file must contain the CA's public root certificate. A client certificate is not required.

Chapter 44. Configuring for Client Authentication

If client authentication is required (in addition to server authentication), a *client certificate* is provided by clients to authenticate the client to the server.

To implement client authentication in your Java-based connector, you must configure for client authentication:

- *Each* VSE Connector Client installed on Web clients of a 2-tier environment.
- The VSE Connector Client installed on the physical/logical middle-tier of a 3-tier environment.
- The VSE Connector Server.

To implement client authentication in CICS Web Support, you must configure *each* CICS client (Web browser) for client authentication. The procedure for configuring the VSE Connector Server does not apply to CICS Web Support.

Note: Before starting this chapter, you should understand the differences between the SSL support offered by IBM's and Oracle's Java Development Kits. For details, see “Differences in SSL Support Provided by IBM/Oracle JDKs” on page 486.

Note: Support for 2048-Bit and 4096-Bit RSA Key Pairs! The example in this chapter is based upon the use of *1024-bit* RSA key pairs. However, you can use:

- *2048-bit* RSA key pairs if you are using:
  - A Crypto Express2 card or later.
  - The latest version of TCP/IP for VSE/ESA.
  - An IBM System z9 platform or later.
- *4096-bit* RSA key pairs if you are using:
  - A Crypto Express3 card or later.
  - The latest version of TCP/IP for VSE/ESA.
  - An IBM System z10 platform or later.

This chapter contains these main topics:

Configuring for Client Authentication Using Self-Signed Certificates

To configure your z/VSE system for client authentication using your self-signed certificates, you should follow the steps given below.

Note: Certificates that you sign using your own root certificate are not usable outside of your own company's test or intranet environment.

Prerequisites:

- You must have completed all the steps described in Chapter 42, “Preparing Your System to Use SSL,” on page 477.

- The VSE Connector Server must be running on the z/VSE host in **non-SSL mode** (for details, refer to the chapter "Installing and Operating the Java-Based Connector" in the *z/VSE e-business Connectors User's Guide*, SC34-2629).
- You must have installed the Java Development Kit (JDK) 1.4 or higher on the platform where you will use Keyman/VSE (for details, refer to the chapter "Installing the Common Prerequisite Programs" in the *z/VSE e-business Connectors User's Guide*, SC34-2629).

1. **Complete Steps for "Server Authentication Using Self-Signed Certificates"**Complete all steps described in "Configuring for Server Authentication Using Self-Signed Certificates" on page 489.
2. **Open the Keyring File**
   a. On the Keyman/VSE toolbar click **Open new input file** as shown in Figure 124. The *Enter Keyring File Password* window is then displayed.

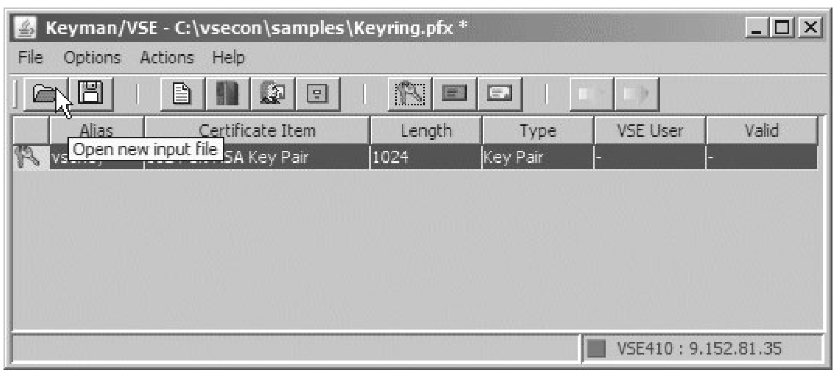


Figure 124. Open New Input File icon on Keyman/VSE Toolbar

   b. Enter your password for this keyring file and click **OK**.
3. **Create a Client Certificate That is Signed by the Root Certificate**
   a. Select your root certificate (that was created during "Configuring for Server Authentication Using Self-Signed Certificates" on page 489) from the displayed Certificate items, and then display the pop-up menu by clicking it using the right mouse-button. Select **Create user certificate**. The *Enter Your Personal Information* window is then displayed.
   b. Complete the fields in this window. If you wish the client certificate to be mapped to a VSE User ID (for details of this process, see Chapter 45, "Implementing Client Authentication with VSE User-ID Mapping," on page 515), you must enter a valid z/VSE User ID in the field **Map to VSE User**. When you have completed the fields in this window, click **Generate user cert** to create your client (user) certificate. The client certificate is then created and added to the list of displayed Certificate items.
4. **Upload the Self-Signed Client Certificate to the z/VSE Host**
   a. Select the client certificate from the displayed list. Next, display the pop-up menu by clicking the client certificate using the right mouse-button. Select **Upload to VSE**. The *Send Certificate Item to VSE* window is then displayed. You do not need to change the values in this window, since you have

already set them in "Configuring for Server Authentication Using Self-Signed Certificates" on page 489. Therefore, click **Upload**. The BSSDCERT utility is started on the z/VSE host, which catalogs the client certificate in the VSE Keyring Library where it will have the suffix *.CCERT*. BSSDCERT also performs the mapping of the client certificate to a VSE User ID.

b. On the *Send Certificate Item to VSE* window you will now see status messages. The progress of job BSSDCERT is displayed on both the *Send Certificate Item to VSE* window and the z/VSE console.

5. **Save the Client Keyring File**Since the properties of the client keyring file are unchanged, you can now click **Save** on the toolbar to save the client keyring file **KeyRing.pfx**.

6. **Import the Client Keyring File into a Web Browser (CICS Web Support Only)**This step is only required when configuring for client authentication in *CICS Web Support*. In this step you import into your CICS client (Web browser) the client keyring file that you have saved in Step 5.
   - For *Microsoft Internet Explorer* Web browsers, you double-click **KeyRing.pfx** and the Internet Explorer is started. The *Certificate Manager Import Wizard* window is then displayed, with general information. Now follow the instructions provided for adding the client keyring file to your certificate store. During the procedure, you will be requested to enter the:
     a. File you wish to import (for example **c:\vsecon\samples\KeyRing.pfx**).
     b. Password that you specified when you saved the client keyring file in Step 5.
     c. Certificate store (the system area where certificates are stored). You can either define your own store, or let the system automatically select a store.
   - For *Netscape* and *Mozilla* Web browsers, you start your Web browser, and then select **Edit — Preferences — Certificates**. Now follow the instructions for importing your client keyring file. During the procedure, you will be requested to enter a password with which you can protect the client keyring file.
   - If you receive the message "The data cannot be decrypted because it was encrypted using an algorithm or key size which is not allowed by this configuration", then the options set in the client keyring file prevent this file from being imported into your Web browser. Check and amend the PKCS#12 options using the Keyman/VSE tool.

7. **Verify Your Certificates in the VSE Keyring Library**To verify that your root certificate and server certificate are valid and correct, you can submit the job CIALSIGV.JCL on the z/VSE host. In this job, you specify the name of the root and server certificates.

Configuring for Client Authentication Using CA-Signed Certificates

To configure your z/VSE system for client authentication using certificates that have been signed by an external Certificate Authority (CA), you should follow the steps given below in which a wizard is used to simplify and shorten the process.

Note: Certificates that are signed by a CA can be used throughout the internet.

Prerequisites:
- You must have completed all the steps described in Chapter 42, "Preparing Your System to Use SSL," on page 477.
- The VSE Connector Server must be running on the z/VSE host in **non-SSL mode** (for details, refer to the chapter "Installing and Operating the Java-Based Connector" in the *z/VSE e-business Connectors User's Guide*, SC34-2629).
- You must have installed the Java Development Kit (JDK) 1.4 or higher on the platform where you will use Keyman/VSE (for details, refer to the chapter "Installing the Common Prerequisite Programs" in the *z/VSE e-business Connectors User's Guide*, SC34-2629).

1. **Get Started and Check Properties**
   a. On the Keyman/VSE toolbar click **Create CA-signed keyring** as shown in Figure 125.

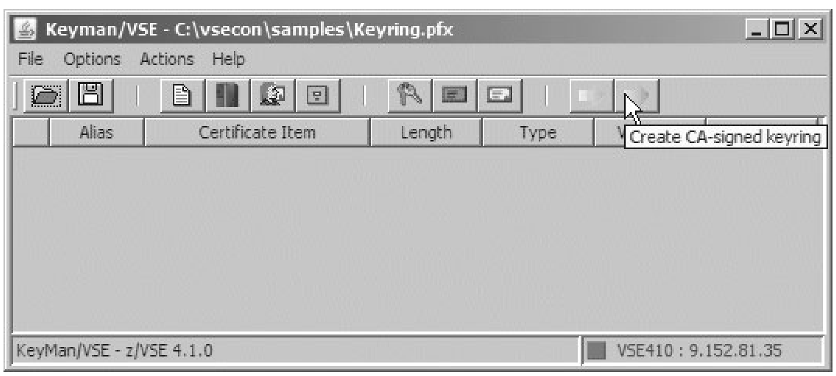


Figure 125. Create CA-Signed Keyring icon on Keyman/VSE Toolbar

   b. The *VSE Host Properties* window is then displayed. Check that the values shown are correct (for details of the fields in this window, see "Specify the Properties of Your z/VSE Host" on page 479). Then click **Next**.
   c. The *Local Keyring File Properties* window is then displayed. Check that the values shown are correct (for details of the fields in this window, see "Specify the Properties of Your Local Keyring File" on page 481). Then click **Next**. The *Generate RSA Key Pair* window is then displayed.

2. **Create an RSA Key Pair and Server Certificate Request**
   a. Select a key length of **1024**, and click **Next** (key lengths of 512 bits are not secure). The key pair entry is then created, and the *Personal Information for VSE Server Cert* window is displayed. Complete the fields in this window, and click **Next** to create your server certificate request.

3. **Request a VSE Server Certificate**The certificate request is displayed in the *Request VSE Server Certificate from a CA* window. Click **Next**, and:
   a. The request for a server certificate is copied to the Clipboard (so that it can be later signed by a CA, such as Thawte Inc.).
   b. The *Get VSE Server Certificate from CA* window is displayed containing an area into which you paste the signed certificate, in Step 5.

4. **Proceed To a CA Web Page for Obtaining Free Certificates**You must now submit your certificate request to any CA to be signed. This example shows you how to obtain a free signed server certificate from the Thawte Corporation. Start your Web browser and proceed to the Thawte Corporation Web site:

   ```
   https://www.thawte.com/cgi/server/test.exe
   ```

   You must firstly register with Thawte. Then, you given instructions on how to proceed to the Web page for obtaining test server certificates, which are free.

   Note: The server certificate is only valid for a limited time period.

5. **Obtain a Signed Server Certificate**
   a. Paste your server certificate request from the Clipboard into the appropriate area of the *Test Thawte Certificates* window. Complete the other details, such as:
      - For *Type of test certificate*, select **Test X509v3 SSL Cert**.
      - For *Format for Chained CA Certs*, select **"Use the standard format"** (the BASE64 encoding of an X.509v3 certificate).
   b. Click **Generate Test Certificate**. The output from this request is a signed server certificate that is displayed under the heading **Here is your certificate:**.
   c. Using your mouse, select the server certificate. Include the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.
   d. Copy the server certificate to your Clipboard, using the usual keyboard combination (for example, **Control + Insert**).
   e. Paste the server certificate from your Clipboard into the *Get VSE Server Certificate from CA* window, using the usual keyboard combination (for example, **Shift + Insert**). Click **Next**, and:
      1) The signed server certificate is added to the list of Certificate items.
      2) The *Personal Information for VSE Client Cert* window is displayed.

6. **Request a VSE Client Certificate**Complete the fields in the *Personal Information for VSE Client Cert* window. If you wish the client certificate to be mapped to a VSE User ID (for details of this process, see Chapter 45, "Implementing Client Authentication with VSE User-ID Mapping," on page 515), you must enter a valid z/VSE User ID in the field **Map to VSE User**. When you have completed the fields in this window, click **Next** to create your request for a client (user) certificate. The client certificate request is displayed in the *Request VSE Client Certificate from CA* window. Click **Next**, and:
   a. The request for a client certificate is copied to the Clipboard (so that it can be later signed by a CA, such as Thawte Inc.).
   b. The *Get VSE Client Certificate from CA* window is displayed containing an area into which you can paste the signed certificate, in Step 8.

7. **Proceed To a CA Web Page for Obtaining Free Certificates**You must now submit your certificate request to any CA to be signed. This example shows you how to obtain a free signed client certificate from the Thawte Corporation. Start your Web browser and proceed to the Thawte Corporation Web site:

   ```
   https://www.thawte.com/cgi/server/test.exe
   ```

   You must firstly register with Thawte. Then, you given instructions on how to proceed to the Web page for obtaining test client certificates, which are free.

   Note: The client certificate is only valid for a limited time period.

8. **Obtain a Signed Client Certificate**
   a. Paste your client certificate request from the Clipboard into the appropriate area of the *Test Thawte Certificates* window. Complete the other details, such as:
      - For *Type of test certificate*, select **Test X509v3 SSL Cert**.
      - For *Format for Chained CA Certs*, select **"Use the standard format"** (the BASE64 encoding of an X.509v3 certificate).
   b. Click **Generate Test Certificate**. The output from this request is a signed client certificate that is displayed under the heading **Here is your certificate:**. An example is shown in Step (5.).
   c. Using your mouse, select the client certificate. Include the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.
   d. Copy the client certificate to your Clipboard, using the usual keyboard combination (for example, **Control + Insert**).
   e. Paste the client certificate from your Clipboard into the *Get VSE client certificate from CA* window, using the usual keyboard combination (for example, **Shift + Insert**). Click **Next**, and:
      1) The signed client certificate is added to the list of Certificate items.
      2) The *Get Root Certificate from CA* window is displayed.

9. **Obtain the CA's Public Root Certificate**You must now obtain a root certificate from the Certificate Authority. This root certificate is a "public" root certificate. This means, you cannot use it for signing other certificates (as you can for self-signed certificates). Instead, this "public" root certificate is used during the SSL handshake to verify certificates that you receive (to check the identity of the sender).
   a. Proceed again to the Thawte Corporation Web site:

      ```
      https://www.thawte.com/cgi/server/test.exe
      ```

      and follow the instructions on how to proceed to the Web page for obtaining the Thawte public root certificate, which is free.
   b. Using your mouse, select the root certificate. Include the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.
   c. Copy the root certificate to your Clipboard, using the usual keyboard combination (for example, **Control + Insert**).
   d. Paste the root certificate from your Clipboard into the *Get Root Certificate from CA* window, using the usual keyboard combination (for example, **Shift + Insert**). Click **Next**, and:
      1) The public root certificate is then added to the list of Certificate items.
      2) The *Create Client/Server Keyring* window is displayed.

10. **Upload the Keyring Members to the z/VSE Host**When you click **Finish** in the *Create Client/Server Keyring* window, these members are uploaded to the z/VSE host and cataloged in the VSE Keyring Library (CRYPTO.KEYRING):
    - The 1024-bit RSA key pair (which will have the suffix *.PRVK*). Job CIALSRVR is used to catalog this member in the VSE Keyring Library.
    - The public root certificate (which will have the suffix *.ROOT*). Job CIALROOT is used to catalog this member in the VSE Keyring Library.
    - The signed server certificate (which will have the suffix *.CERT*). Job CIALCERT is used to catalog this member in the VSE Keyring Library.
    - The client certificate (which will have the suffix *.CCERT*). Job BSSDCERT is used to catalog this member in the VSE Keyring Library and perform the mapping to a VSE User ID.

    The progress of these jobs is displayed on both the *Create Client/Server Keyring* window and the z/VSE console.

11. **Save the Client Keyring File**Since the RSA key pair, root certificate, server certificate, and client certificate have been uploaded to the VSE Keyring Library on the z/VSE host, you can now delete certificates that are no longer required from your client keyring file **KeyRing.pfx**, and then save this file.
    - You *must* keep the CA's public root certificate in the client keyring file, so that the client can verify certificates sent to it from the z/VSE host during the SSL handshake.
    - You *must* keep the signed client certificate in the client keyring file, so that the client send this certificate to the z/VSE host during the SSL handshake.
    - You can keep the signed server certificate in the client keyring file, so that this server certificate can be compared with the server certificate sent to it from the z/VSE host during the SSL handshake. This avoids having to store the server certificate sent from the z/VSE host at the start of each communications session.
    - You can delete the other Certificate items (such as the 1024-bit RSA key pair, or certificate requests).

    After you have deleted the unwanted Certificate items, select **File** and then **Save keyring file** from the Menu bar. Enter your password and click **OK**.

12. **Import the Client Keyring File into a Web Browser (CICS Web Support Only)**This step is only required when configuring for server authentication in *CICS Web Support*. In this step you import into your CICS client (Web browser) the client keyring file that you have saved in Step 10.
    - For *Microsoft Internet Explorer* Web browsers, you double-click **KeyRing.pfx** and the Internet Explorer is started. The *Certificate Manager Import Wizard* window is then displayed, with general information. Now follow the instructions provided for adding the client keyring file to your certificate store. During the procedure, you will be requested to enter the:
        a. File you wish to import (for example **c:\vsecon\samples\KeyRing.pfx**).
        b. Password that you specified when you saved the client keyring file in Step 10.
        c. Certificate store (the system area where certificates are stored). You can either define your own store, or let the system automatically select a store.
    - For *Netscape* and *Mozilla* Web browsers, you start your Web browser, and then select **Edit — Preferences — Certificates**. Now follow the instructions

for importing your client keyring file. During the procedure, you will be requested to enter a password with which you can protect the client keyring file.

- If you receive the message "The data cannot be decrypted because it was encrypted using an algorithm or key size which is not allowed by this configuration", then the options set in the client keyring file prevent this file from being imported into your Web browser. Check and amend the PKCS#12 options using the Keyman/VSE tool.

13. **Verify Your Certificates in the VSE Keyring Library**To verify that your root certificate and server certificate are valid and correct, you can submit the job CIALSIGV.JCL on the z/VSE host. In this job, you specify the name of the root and server certificates.

Configuring the VSE Connector Server for Client Authentication

Note: Not Relevant for CICS Web Support! The information in this topic is not relevant for implementing client authentication in CICS Web Support.

To configure the VSE Connector Server for client authentication, you use SSL configuration member SKVCSSSL contained in ICCF Library 59. An example of the required settings is shown in Figure 126.

```
; ******************************************************************
;      SSL CONFIGURATION MEMBER FOR VSE CONNECTOR SERVER
; ******************************************************************

; ******************************************************************
; SSLVERSION   SPECIFIES THE MINIMUM VERSION THAT IS TO BE USED
;              POSSIBLE VALUES ARE:   SSL30 AND TLS31
; KEYRING      SPECIFIES THE SUBLIBRARY WHERE THE KEY FILES ARE
;              STORED.
; CERTNAME     NAME OF THE CERTIFICATE THAT IS USED BY THE SERVER
; SESSIONTIMEOUT NUMBER OF SECONDS THAT THE SERVER WILL USE TO
;              ALLOW A CLIENT TO RECONNECT WITHOUT PERFORMING A
;              FULL HANDSHAKE. (86440 SEC = 24 HOURS)
; AUTHENTICATION TYPE OF AUTHENTICATION. POSSIBLE VALUES ARE:
;              SERVER - SERVER AUTHENTICATION ONLY
;              CLIENT - SERVER AND CLIENT AUTENTICATION
;              LOGON  - SERVER AND CLIENT AUTENTICATION WITH LOGON
;                       THE CLIENT CERTIFICATE IS USED TO LOGON.
; ******************************************************************
SSLVERSION     = SSL30
KEYRING        = CRYPTO.KEYRING
CERTNAME       = SAMPLE
SESSIONTIMEOUT = 86440
AUTHENTICATION = CLIENT or LOGON
  .
  .
  .
```

Figure 126. Job to Configure the VSE Connector Server for Client Authentication

You can set the parameter AUTHENTICATION to specify:

SERVER
Server authentication only.

CLIENT
Server authentication *and* client authentication.

LOGON
Server authentication *and* client authentication that uses a client certificate to logon.

Note: If you specify that you require client authentication in Figure 126 on page 512, then *each* client (where a VSE Connector Client is installed) must:
- have a valid client certificate stored in the *client keyring file*.
- provide a client certificate when connecting to the VSE Connector Server (it is not possible to have client authentication for *some* clients only).

Summary of Client Authentication Tasks for the Java-Based Connector

Table 21 provides a summary of the tasks you must complete to configure the Java-based connector for client authentication, including mapping client certificates to VSE User IDs. The table includes the information contained in Chapter 45, "Implementing Client Authentication with VSE User-ID Mapping," on page 515.

Table 21. Tasks Involved in Configuring the Java-Based Connector for Client Authentication

	Client Authentication	**Client Authentication with VSE User ID Mapping**
On the z/VSE Host Side	Set AUTHENTICATION = CLIENT in skeleton SKVCSSSL.	Set AUTHENTICATION = LOGON in skeleton SKVCSSSL.
On the Client Side	Set LOGONWITHCERT = NO in member **ssl.prop**.	Set LOGONWITHCERT = YES in member **ssl.prop**.
Certificates	Do not use Job SKSSLKEY to catalog the 1024–bit sample private key and certificates. Instead, use Keyman/VSE to create your own 1024-bit or 2048-bit keys and certificates. **Note:** 512-bit keys are not secure.	Do not use Jobs SKSSLKEY and SKCCERT to catalog the 1024–bit sample private key and certificates. Instead, use Keyman/VSE to create your own 1024-bit or 2048-bit keys and certificates. **Note:** 512-bit keys are not secure.Client certificates must be mapped to VSE User IDs using either (a) the VSE Interactive Interface (Fast Path **2.9**) or (b) Keyman/VSE
Using Keyman/VSE	Use the wizard dialog **Actions –> Create CA-signed Keyring** to create the complete keyring. All certificates are therefore signed by an external Certificate Authority (CA) such as Thawte. However, you will be charged by Thawte for these actions.	Use the wizard dialog **Actions –> Create CA-signed Keyring** to create the complete keyring. All certificates are therefore signed by an external Certificate Authority (CA) such as Thawte. However, you will be charged by Thawte for these actions.
Client Keyring File	Your client keyring file must contain the CA's public root certificate, and a client certificate which has been signed by this CA.	Your client keyring file must contain the CA's public root certificate, and the client certificate which you use to signon to the Host without an explicit logon. This certificate must have been signed by this CA and mapped to a valid VSE User ID.

Chapter 45. Implementing Client Authentication with VSE User-ID Mapping

If client authentication is required in addition to server authentication, a client certificate is provided by clients to authenticate the client to the server.

Using the service functions for client authentication described in this chapter, you can introduce access checking on client certificates via z/VSE User IDs that have been assigned to these client certificates.

The client certificates belong to either CICS clients or VSE Connector Clients. Therefore using client certificates, you can control the access rights from:
- CICS clients to z/VSE host resources.
- VSE Connector Clients to z/VSE host resources.

A client-certificate/User-ID mapping list can be built and maintained using either the batch function BSSDCERT, or using the Client-Certificates/User-IDs dialog.

This chapter contains these main topics:
- "Prerequisites For Client Authentication with VSE User-ID Mapping"
- "Using the Batch Service Function BSSDCERT"
- "Changing the Defaults (Optional)" on page 517
- "Using the Client-Certificates/User-IDs Dialog" on page 517

Related Topic:
- "Summary of Client Authentication Tasks for the Java-Based Connector" on page 513

Prerequisites For Client Authentication with VSE User-ID Mapping

Before you can use the service functions described here, you must have obtained and stored in the VSE Keyring Library on the z/VSE host (default CRYPTO.KEYRING) *at least one* client certificate. This is a client certificate in Base64 format, to which you wish to assign a z/VSE User ID.

For details of how to obtain and store client certificates , refer to Chapter 44, "Configuring for Client Authentication," on page 505.

Using the Batch Service Function BSSDCERT

This topic describes how you can use the BSSDCERT service *in batch* to build and to maintain the mapping list of client-certificate/User-ID pairs.

```
EXEC BSSDCERT,PARM='options'

options:
       ADD,<CertMemName>,<Uid>,TRUST|NOTRUST
       CHG,<CertMemName>,<Uid>,TRUST|NOTRUST
       DEL,<CertMemName>
       LST[,<ListMemName>[,I]]
       ACT
       CML,<CmdListMemName>
       STA,<ALL>
```

where:

ADD Adds a new certificate to the client-certificate/User-ID mapping list.

CHG Changes the details of an entry in the client-certificate/User-ID mapping list.

DEL Deletes an entry for a given certificate in the client-certificate/User-ID mapping list.

LST Extracts a readable list from the contents of the client-certificate/User-ID mapping list. The list contains the CertMemName, the assigned userid, the trusted indication, and from the certificate information about the subject. This can be the subject's common name (for example, the certificate owner), and the subject's organization. However, the information will be truncated to fit on the line. The created mapping list can be either displayed on the console, or written to a librarian member in plain text format or IPF format. IPF format is required so that the mapping list can be displayed in the Client-Certificates/User-IDs Dialog (Figure 127 on page 518).

ACT Builds an incore version of the client-certificate/User-ID mapping list. It also activates the list so that it can be used by CICS Web Support (CWS).

CML Specifies a librarian member name that contains a list of BSSDCERT function calls.

STA Shows the status of the client-certificate/User-ID mapping list. If this list is active, the storage size of this list and the number of records, will be displayed. If you specify parameter ALL, the current name settings of the related z/VSE library members are also displayed. For further details about name settings, see "Changing the Defaults (Optional)" on page 517.

CertMemName
The name of the library member that contains the client certificate. The member suffix is CCERT. The client certificate is a Base64 encoded X.509 certificate as returned from a PKCS #10 certificate request. The data must include the string `'-----BEGIN CERTIFICATE-----'` immediately before the Base64 encoding, and the string `'-----END CERTIFICATE-----'` immediately following it.

Uid The z/VSE User ID defined using, for example, the Basic Security Manager (BSM). This User ID will be associated with a client certificate. The client (a CICS client or a VSE Connector Client) will then have the access rights defined for this User ID, during a connection to the z/VSE host.

TRUST
The specified client certificate is trusted.

NOTRUST
The specified client certificate is not trusted.

ListMemName
The name of librarian member to which the output of the LST function should be written.

I Specifies that the output of the LST function should be written to in IPF format to the librarian member specified using ListMemName. (IPF format is required so that the mapping list can be displayed in the Client-Certificates/User-IDs Dialog).

CmdListMemName
The name of a librarian member that contains a list BSSDCERT function calls.

Changing the Defaults (Optional)

You might wish to change the library and member-names defaults that are used by BSSDCERT (for example, the name of the VSE Keyring Library). However normally, you should not need to change these defaults.

The SETPARM defaults are shown below. To change any of the defaults, you use the **SETPARM SYSTEM,BSSDC..='value'** statement.

```
Certificate/userid mapping file:
      BSSDCUI='l.s[.mn[.mt]]'
            default: CRYPTO.KEYRING.BSSDCUID.MAPPING

   Client certificates:
      BSSDCCL='l.s'
            default: CRYPTO.KEYRING

   List output in files:
      BSSDCLT='l.s'
            default: IJSYSRS.SYSLIB

   Command list input file:
      BSSDCCS='l.s'
            default: IJSYSRS.SYSLIB
```

For example, to change the *Certificate/Userid mapping file* default to MYCRYPTO.KEYRING, you would enter at the system console:

```
SETPARM SYSTEM,BSSDCUI='MYCRYPTO.KEYRING'
```

Note: If you change the above defaults for *List output in files* or *Command list input file*, you will no longer be able to use the Client-Certificates/User-IDs Dialog.

For details of how to use the SETPARM command, refer to *z/VSE System Control Statements*.

Using the Client-Certificates/User-IDs Dialog

The Client-Certificates/User-IDs dialog is provided so that you can more easily manage your client-certificate/User-ID mapping list. It calls the BSSDCERT function, which was described in "Using the Batch Service Function BSSDCERT" on page 515.

Step 1: Starting the Dialog

To start the Client-Certificates/User-IDs dialog, you select *29 Maintain Certificate - User ID List* on the administrator's *Resource Definition Selection* panel (IESEDEF). The panel shown in Figure 127 on page 518 is then displayed, which shows the list of all client-certificate/User-ID pairs defined for your z/VSE system.

```
TAS$CERS                 CLIENT CERTIFICATES - USER IDS
Enter the required data and press ENTER

OPTIONS: 1 = ADD         2 = CHANGE      5 = DELETE

 OPT         CERTIFICATE                              USERID  TRUSTED
     COMMON NAME                            MEMBER NAME
 _   John Haeberle, IBM                     KRL00010     JOHNHB    X
 _   Paul Gallagher, IBM                    KRL00012     PAULGL    X
 _   Helmut Hellner, IBM                    KRL00015     HELHELL
 _   Alexander Schoettle, IBM               KRL00017     ASCHOETT  X
 _   TCPIP4VSE, Connectivity Systems        KRL00018     473337    X
 _   TCPIP5VSE, Connectivity Systems        KRL00019     460341    X
 _   TCPIP6VSE, Connectivity Systems        KRL00022     155287    X
 _   Herbert Nass, IBM                      KRL00024     NASSHER
 _   Anita Stark, IBM                       KRL00026     NETTANI
 _   Elke Schaefer, IBM                     KRL00027     ELKESCHA  X

 LOCATE MEMBER NAME == >        ________
PF1=HELP       2=REDISPLAY    3=END     5=PROCESS    6=ACTIVATE
               8=FORWARD
```

Figure 127. Listing All Client-Certificate/User-ID Pairs

Here is an explanation of the fields displayed in Figure 127:

OPT
In this column you can enter either a 1 (to add a certificate to the mapping list), 2 (to change a certificate in the mapping list), or 5 (to delete a certificate from the mapping list). Explained in "Step 2: Selecting an Option."

CERTIFICATE COMMON NAME
Common name contained on the client certificate, and the organization of the certificate owner.

CERTIFICATE MEMBER NAME
Name of the member that contains the client certificate.

USERID
z/VSE User ID that the administrator has assigned to the client certificate.

TRUSTED
If an X is displayed next to a client certificate, then the administrator has decided that this client's User ID can be used for access checking. Otherwise, this client's User ID will not be used for access checking. You can use this field to temporarily deactivate the assignment of a client certificate to a User ID.

The use of the PF5 (Process) and PF6 (Activate) keys are described in "Step 3: Creating the Output Job" on page 519.

Step 2: Selecting an Option

From the panel shown in Figure 127, you can enter in the OPT column one of these options:

1 (ADD)
If you enter a **1** (in Figure 127) to add a new client-certificate/User-ID pair, you get the panel shown in Figure 128 on page 519. The system displays default values if you select an empty line. If you selected an already-defined client-certificate/User-ID pair, the dialog uses this client-certificate/User-ID pair and its parameters (for `USER ID` and `TRUSTED`) as a model for the new client-certificate/User-ID pair.

```
 TAS$CER             CLIENT CERTIFICATES - USER IDS: ADD

 Enter the required data and press ENTER.




MEMBER NAME............ ________    Unique name of the library member
                                    that contains the client certificate

USER ID................ UAX         VSE User Id associated to
                                    the certificate.

TRUSTED................ 1           The certificate is trusted.
                                    Enter 1 for yes and 2 for no.





 PF1=HELP       2=REDISPLAY  3=END
```

Figure 128. Adding a Client-Certificate/User-ID Pair

You must then enter in the field:

Member Name
The name of the member that contains the client certificate. By default, this member will be stored in the VSE Keyring Library.

User ID
The z/VSE User ID of the client-certificate/User-ID pair to be added.

Trusted
Either the value:

1 The client certificate should be trusted.

2 The client certificate should not be trusted.

After completing the fields above, you press Enter to save your changes and return to the Client-Certificates/User-IDs dialog (Figure 127 on page 518).

2 (CHANGE)
After entering a **2** next to a client-certificate/User-ID pair, you can then overtype either the User ID or the Trusted parameter of the pair, Press Enter to save your changes.

5 (DELETE)
After entering a **5** next to a client-certificate/User-ID pair, you can then press Enter to carry out the deletion.

Step 3: Creating the Output Job

After all your changes have been entered, you can either press PF5 (Process) or PF6 (Activate).

PF5 (Process only)
A job is created (shown in "Step 4: Submitting or Storing the Output Job" on page 520) in which the mapping list of Client-Certificates/User-IDs is to be updated with a list of your changes. These changes might include:

- new pairs whose details you have defined using Option 1.
- changed pairs whose details you have defined using Option 2.
- pairs to be deleted, which you have identified using Option 5.

You can activate these changes at a later time using the PF6 function.

PF6 (Process and Activate)
As for PF5, a job is created in which the mapping list of Client-Certificates/User-IDs is to be updated with a list of your changes. In addition however:

- an incore version of the new mapping list will be built and activated.
- the automatic activation of the new mapping list during all subsequent system-startups, is prepared.

Step 4: Submitting or Storing the Output Job

After completing "Step 3: Creating the Output Job" on page 519, the dialog creates a job with the default name CATCERT. On the *Job Disposition* panel, you can submit the job to batch, file it in your default primary library, or both.

Chapter 46. Implementing Hardware-Based Tape Encryption

This chapter describes how you can encrypt tapes using the hardware-based encryption facilities provided by an *encryption-capable* tape drive. Examples of encryption-capable tape drives are the IBM System Storage TS1140, TS1130, and TS1120[3].

This chapter contains these main topics:

Related Topics:

For details of how to ...	Refer to ...
use the *Interactive Interface* to add a tape drive	Chapter 7, "Configuring Disk, Tape, and Printer Devices," on page 87
configure tape libraries containing encryption-capable tape drives	Chapter 21, "Implementing Tape Library Support," on page 255
• display the *key encrypted key labels* of the tape medium using the Query Tape (QT) command • specify the *mode* for an encryption-capable tape drive, using a JCL ASSGN statement • use a JCL KEKL statement to force encryption when writing data to a tape	"Job Control and Attention Routine" in the *z/VSE System Control Statements*.
backup/export to encrypted tapes using the *Interactive Interface*	"Backing Up and Restoring Data" in the *z/VSE Operation*.
create encrypted tapes using the POFFLOAD command	"VSE/POWER Operator Commands" in the *VSE/POWER Administration and Operation*.

For details of the currently-supported tape drives, tape controllers, and tape recording formats, refer to "Tape Device Support" in the *z/VSE Planning*.

3. The TS1140 is also referred to as the 3592 Model E07. The TS1130 is also referred to as the 3592 Model E06. The TS1120 is also referred to as the 3592 Model E05.

Overview of Hardware-Based Tape Encryption

Figure 129 provides a simplified description of hardware-based tape encryption.

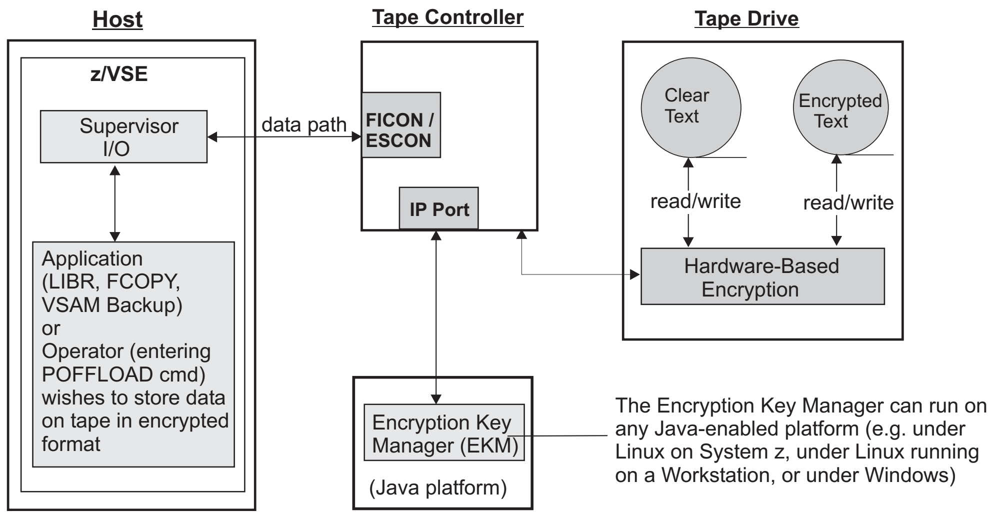


Figure 129. Overview of Hardware-Based Tape Encryption

Here is an explanation of Figure 129:

1. A request is made for data to be backed-up to tape in *encrypted format*. This request can originate from a:
   - LIBR, FCOPY, or VSAM backup job running on the z/VSE host.
   - VSE/POWER **POFFLOAD** command entered at the z/VSE console by an operator.

   An ASSGN statement (contained in the job or entered at the console) sets the Mode for the **TS1120**, **TS1130**, or **TS1140** tape drives to *encryption*.

   A KEKL statement (contained in the job or entered at the console), which contains one or two key-encryption-key labels, informs z/VSE to associate a tape unit with one or two key-encryption-key labels.
2. The key-encryption-key labels are passed via the z/VSE Supervisor to the Encryption Key Manager (EKM).
3. "Key negotiation" takes place between the tape drive and the EKM, during which the EKM validates/supplies encryption keys with the tape drive. The tape drive and EKM communicate via the TCP/IP protocol.
4. If the key-verification process is successful, the data on the tape cartridge will be encrypted. If not, an error message is returned.

Prerequisites for Using Hardware-Based Tape Encryption

These are the prerequisites for using hardware-based tape encryption:

- The Encryption Key Manager must be running on a Java platform. For details, see "Obtaining and Installing the Encryption Key Manager" on page 524.
- You must have installed and configured an *encryption-capable tape drive*, such as the IBM System Storage TS1120, TS1130, or TS1140 and a tape controller. For a list of currently-supported tape drives, refer to the chapter "Hardware Support" in the *z/VSE Planning*, SC34-2635
- You must record your data using one of the following:

– *Encrypted Enterprise Format 2* (EEFMT2), which is the encrypted form of EFMT2.
– *Encrypted Enterprise Format 3* (EEFMT3), which is the encrypted form of EFMT3.
– *Encrypted Enterprise Format 4* (EEFMT4), which is the encrypted form of EFMT4.

For details, refer to the chapter “Hardware Support” in the manual *z/VSE Planning*.

- If you wish to use the REKEY parameter (explained in “Specifying KEKL Statements” on page 525), you must have either:
  – TS1120 C06 tape controller, machine code level `1.21.3.xx` or later, or
  – TS1120 J70 tape controller, machine code level `1.19.7.xx` or later.

  Both machine code levels were generally available on August 31st, 2007.

Restrictions When Using Hardware-Based Tape Encryption

The following restrictions apply to the use of encrypted tapes:

- A tape cartridge cannot contain both encrypted and non-encrypted data.
- If the first file written to a tape is encrypted, all subsequent files written to that same tape cartridge will be encrypted using the *same key* (except for the volume label structure for the first file sequence).
- The DOSVSDMP utility is *not supported for encryption*. Therefore, a request to create an encrypted stand-alone dump tape using the DOSVSDMP utility will be rejected!

Tape Encryption When Running z/VSE as a Guest Under z/VM

The z/VSE and z/VM operating systems *both* support hardware-based tape encryption.

You are therefore recommended to use hardware-based tape encryption on *either* z/VSE or z/VM (*not* on both operating systems). Otherwise, there could be errors caused by different key encryption key labels being used. Below is a summary of using encryption with z/VM and z/VSE.

1. z/VM-Based Encryption IS Active (via an ATTACH or SET RDEV Command):

- If encryption *has not* been enabled in z/VSE via the *mode* setting of an ASSGN statement, encryption will be processed according to the encryption-settings *in z/VM*. However, although z/VSE is not “aware of” encryption, the QT (“Query Tape”) command *will* display the encryption-indication of the volume.
- If encryption *has* been enabled in z/VSE via the *mode* setting of an ASSGN statement, encryption will be handled according to the encryption-settings *in z/VSE*.

2. z/VM-Based Encryption IS NOT Active:

- If encryption *has* been enabled in z/VSE via the *mode* setting of an ASSGN statement, encryption will be handled according to the encryption-settings *in z/VSE*.

For further details about implementing hardware-based tape encryption under z/VM, refer to *z/VM CP Planning and Administration*, SC24-6083-04 or later.

Obtaining and Installing the Encryption Key Manager

The Encryption Key Manager (EKM) is a common-platform Java application that is used to generate and protect AES (Advanced Encryption Standard) keys. Upon request, the EKM generates AES keys to be used for encryption, and protects these keys using RSA key pairs.

To obtain a copy of the EKM from the Internet, you should:

1. Enter the following URL:
   `http://www.ibm.com/support/us/`
2. Search for "Encryption Key Manager" and locate the zipped file containing the EKM.
3. Download the zipped file containing the EKM to the directory where you want to install it.
4. Install and customize the EKM by following the instructions provided in the *IBM System Storage Tape Encryption Key Manager, Introduction, Planning and User Guide*, GA76-0418.

Using a Job to Backup Data With Encryption

Jobs for backing-up to tape with encryption (LIBR, FCOPY, VSAM Backup) *must* contain an ASSGN statement and *might* contain a KEKL statement. An example for a LIBR job is given below.

Example of a LIBR Job to Backup/Encrypt the Contents of a Library

The LIBR job below will backup to tape and encrypt the contents of library PRD2. It specifies *two* key-encryption-key labels (KEKL1 and KEKL2). The tape drive has a unit address (cuu) of 480.

```
// JOB ENCRYPT
// ID USER=user-ID,PWD=password
// ASSGN SYS005,480,03
// KEKL UNIT=480,KEKL1='HUSKEKL1',KEM1=L,KEKL2='HUSKEKL2',KEM2=L
// EXEC LIBR
BACKUP LIB=PRD2 TAPE=SYS005
/*
/&
```

Using a POFFLOAD Command to Backup Data With Encryption

When using the POFFLOAD command to backup encrypted data, the operator enters ASSGN and KEKL statement *at the console*. However, the syntax of the ASSGN and KEKL statements are the same as when they are contained within a job running a LIBR, FCOPY, or VSAM backup. Here is an example:

```
POFFLOAD BACKUP,RDR,480,KEKL=
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1Q7GA  SPECIFY POFFLOAD BACKUP KEY ENCRYPTION LABEL KEKL1= AND KEM1= OR
        "CANCEL" FOR 480
G 480,KEKL1='MYKEK LABEL1',KEM1=L,
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1Q7HA  SPECIFY POFFLOAD BACKUP KEY ENCRYPTION LABEL KEKL2= AND KEM2= OR
        "CANCEL" FOR 480
G 480,KEKL2='MYKEK LABEL2',KEM2=L
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
AR 0015 1Q7JI POFFLOAD BACKUP TAPE(S) ON 480 WILL BE ENCRYPTED
F1 0001 1Q2AI  OFFLOADING BACKUP SUCCESSFULLY COMPLETED ON 480, JOURNAL LST
        ENTRY $OFJ0154 CREATED (10/27/06 19:11:31)
```

Operator commands for reading encrypted tapes are *not affected* by VSE/POWER's tape-encryption support. Using an `// ASSGN PERM` mode statement and by defining default KEKL-values in the Encryption Key Manager (EKM), the system administrator can remove the need for any special encryption-command operands when creating POFFLOAD BACKUP|SAVE|PICKUP tapes.

For further details, refer to the chapter "VSE/POWER Operator Commands" in the *VSE/POWER Administration and Operation*, SC34-2625.

Specifying KEKL Statements

Jobs for backing-up to tape with encryption (LIBR, FCOPY, VSAM backup) *might include* a KEKL statement.

If your job does *not contain* a KEKL statement, the EKM will *use the defaults* that you previously generated and stored in the EKM.

The KEKL statement has the following syntax:

```
// KEKL UNIT={cuu|SYSnnn},KEKL1='KEKL1',KEM1={L|H},KEKL2='KEKL2',KEM2={L|H}[,REKEY]
// KEKL UNIT={cuu|SYSnnn},KEKL1='KEKL1',KEM1={L|H}[,REKEY]
// KEKL UNIT={cuu|SYSnnn},CLEAR
```

where:

cuu Specifies the tape unit for which the key-encryption-key labels are to be used.

SYS*nnn*
Specifies the logical unit of the tape unit for which the key-encryption-key labels are to be used. *This logical unit must have been previously assigned*. The value of *nnn* can be:
- between 000 and 255
- LST
- PUN

KEKL1 Is the label for the first key-encryption-key to be used by the EKM to encrypt the data to be stored on the tape. Must be enclosed in single quotation marks. If you do *not* specify a KEKL1, z/VSE uses the default KEKL1 *and* KEKL2 that are stored by the EKM.

KEKL2 Is the label for the second key-encryption-key to be used by the EKM to encrypt the data to be stored on the tape. Must be enclosed in single quotation marks.
- You cannot specify a KEKL2 without having specified a KEKL1.
- If you specify a KEKL1 but do *not* specify a KEKL2 , z/VSE uses the value of KEKL1 for KEKL2.
- If you do *not* specify a KEKL1 *and* a KEKL2, z/VSE uses the default KEKL1 and KEKL2 that are stored by the EKM.

KEM1 Specifies how the label for the first key-encryption-key (KEKL1) is encoded by the EKM and stored on the tape cartridge. The values can be either:
L Encoded as the specified label.
H Encoded as a hash of the public key.

KEM2 Specifies how the label for the second key-encryption-key (KEKL2) is encoded by the EKM and stored on the tape cartridge. The values can be either:
L Encoded as the specified label.
H Encoded as a hash of the public key.

REKEY
: Enables a tape cartridge that has already been encrypted to have its *data key* re-encrypted using one or two *new* key-encryption-keys that are specified by new KEKL(s): KEKL1/KEM1 and possibly KEKL2/KEM2. This enables a tape cartridge to be "re-keyed" without having to copy the data to another volume (that is, the same data key will be encrypted using new key-encryption-keys) .
  - The rules when specifying new key-encryption-keys are the same as when specifying key-encryption-keys *without* the REKEY parameter.
  - If a REKEY request is submitted against a volume which is *not* positioned at Load Point (LP), z/VSE will force a rewind of the tape *before* the REKEY is processed.

  The prerequisites for using REKEY are given in "Prerequisites for Using Hardware-Based Tape Encryption" on page 522.

CLEAR
: Indicates that the information previously established by a KEKL statement is cleared.

  Note: You might need to reset the KEKL (the default KEKL, or the KEKL from a previous KEKL statement) on a previously-encrypted volume. To do so, you must issue a WRITE command (for example, writing a tape mark) from the Beginning-Of-the-Tape (BOT) with *encryption mode not active*.

For further details about using the KEKL statement, refer to the chapter "Job Control and Attention Routine" in the *z/VSE System Control Statements*, SC34-2637.

Specifying ASSGN Statements

Jobs (LIBR, FCOPY, VSAM backup) that require hardware-based tape encryption must include an ASSGN statement as follows:

Method 1: Specify the Device Mode of the Encryption-Capable Tape Drive.

The syntax of the ASSGN statement is as follows:

```
// ASSGN SYSnnn,cuu,mode
```

where:
- *cuu* is the device address of the encryption-capable tape drive.
- *mode* is a 1-byte field that determines how the data on the tape should be written.

These are the encryption-related modes you can use:

X'03' Encryption Write Mode for the TS1120 (3592 Model E05)
X'04' Encryption Write Mode for the TS1130 (3592 Model E06) or TS1140 (3592 Model E07)
X'0B' Encryption and IDRC (compression) Write Mode for the TS1120 (3592 Model E05)
X'0C' Encryption and IDRC (compression) Write Mode for the TS1130 (3592 Model E06) or TS1140 (3592 Model E07)
X'23' Encryption with unbuffered Write Mode for the TS1120 (3592 Model E05)
X'24' Encryption with unbuffered Write Mode for the TS1130 (3592 Model E06) or TS1140 (3592 Model E07)
X'2B' Encryption and IDRC (compression) and unbuffered Write Mode for the TS1120 (3592 Model E05)

X'2C' Encryption and IDRC (compression) and unbuffered Write Mode for the TS1130 (3592 Model E06) or TS1140 (3592 Model E07)

Note: IDRC is an abbreviation for Improved Data Recording Capability.

Method 2: Let z/VSE Find a Suitable Tape Drive

Here, you:
1. Specify that you require an encrypted write-format.
2. Let z/VSE locate a suitable tape drive.

The syntax of the ASSGN statement is as follows:

```
// ASSGN SYSnnn,device_class,mode
```

If the tape *device_class* (recording format) is set to EEFMT2, EEFMT3, or EEFMT4 z/VSE will search your system for an encryption-capable tape drive. The *mode* specifies any encryption mode.

For further details about using the ASSGN statement, refer to the chapter "Job Control and Attention Routine" in the *z/VSE System Control Statements*, SC34-2637.

Using the Query Tape (QT) Command to Display Tape Information

You can use the Attention Routine (AR) **QT** command to display the mode of a tape drive that is attached to your z/VSE system.

In the following example:
- CODE 5603 indicates:
  - A tape drive that uses the TPA is attached to z/VSE (the **56** part of 5603).
  - This tape drive is assigned to encryption mode (the **03** part of 5603).
- 3592-E05 is the device type for the IBM System Storage TS1120 (IBM 3592 E05) tape drive.
- KEY_LABEL_001 is the label for the first key-encryption-key to be used by the EKM to encrypt the data encryption key.
- KEY_LABEL_002 is the label for the second key-encryption-key to be used by the EKM to encrypt the data encryption key.

```
QT A83
AR 0015 CUU  CODE DEV.-TYP   VOLID  USAGE   MED-TYP   STATUS    POSITION
AR 0015 A83  5603 3592-E05   PAUL01 BG      CST5 /E   RESERVED  8 BLK
AR 0015      CU   3592-C06          LIB     3494-L10  (GALL88)
AR 0015                             FAST-ACC.SEG.=     0 MB    FILES = 2
AR 0015 KEKL1:KEY_LABEL_001
AR 0015 KEKL2:KEY_LABEL_002
AR 0015 1I40I  READY
```

Figure 130. Using the QT Command to Display the Details of an Encrypted Tape

For further details, refer to the chapter "Job Control and Attention Routine" in the *z/VSE System Control Statements*, SC34-2637.

Reading the Contents of an Encrypted Tape

Figure 131 is an example of how to read the contents of an encrypted tape *using the LIBR utility*:

- This job does *not* require KEKL statements to specify the encryption keys to be used.
- If KEKL statements are included in the job, *they will be ignored*.
- To read the encrypted data, the job uses the keys that are already stored on the tape.

However, these are the prerequisites for running this job:

- The z/VSE host where the job is to run must be connected to an EKM.
- The tape must have been previously encrypted using keys that are known by the currently-connected EKM.
- The encryption keys must not have been deleted from the currently-connected EKM.

```
* $$ JOB JNM=LIBSCAN,DISP=D,PRI=3,                                         C
* $$ NTFY=YES,                                                             C
* $$ LDEST=*,                                                              C
* $$ CLASS=0
// JOB LIBSCAN  SCAN VSE LIBRARY BACKUP TAPE
* THIS FUNCTION USES A TAPE FOR INPUT
* MOUNT TAPE BACKUP ON DEVICE 480
* THEN CONTINUE. IF NOT POSSIBLE CANCEL THIS JOB.
// PAUSE
// MTC REW,480
// ASSGN SYS004,480
// EXEC LIBR,PARM='MSHP'
   RESTORE *                          /* LIBRARY IDENTIFICATION */ -
          SCAN  = YES                 /* SCAN SPECIFICATION     */ -
          TAPE = SYS004               /* TAPEADDRESS            */
/*
// MTC RUN,480
/&
* $$ EOJ
```

Figure 131. Using a LIBR Job to Read the Contents of an Encrypted Tape

Understanding Message 0P68I KEYXCHG ER

The EKM generates this message explanation:

```
0P68I Encryption key negotiation with the EKM failed
```

However, the *sense data* of the message contains additional useful information. For example:

```
804C08C022402751 0001FF0000000000 0005EE3100000092 2004E82061BA2111

CU=00 DRIVE=000000 EKM=EE31
```

In the above example (which starts at byte 0), bytes 4 and 5 might contains **2240**. This is the return code and reason-qualifier code (RC-RQC). It means that the required encryption-key exchange *has failed*. Furthermore:

1. Byte 8 contains the *CU reason code* (in the above example, **00**).
2. Bytes 13 ,14 and 15 contain the *sense key from the device* (in the above example, **000000**).

3. Bytes 18 and 19 contain the *sense key from the EKM*:
   - A value **0000** would mean that a failure has occurred whilst connecting to the EKM.
   - The value **EE31** (shown in the above example) means that an encryption configuration problem has occurred in which the error has something to do with the *key store*.

For further details of this and other EKM error messages, refer to the *IBM System Storage Tape Encryption Key Manager, Introduction Planning and User Guide*, GA76-0418.

Hints and Tips

This topic contains various hints and tips that you might find useful.

Assigning System Logical Units

This problem-situation might arise:

1. The assignment of system logical units results in OPEN processing (during VOL1 and header-label checking). After OPEN processing:
   a. The tape (for labeled tapes) is positioned *behind* the VOL1 label.
   b. Previously-used KEKLs are still active.
2. A subsequent KEKL statement that is set *after* the ASSGN statement causes the job *to be cancelled*.

To overcome this problem, you can create the following job:

```
// ASSGN SYSnnn,cuu,mode
// MTC REW,cuu
// KEKL UNIT=cuu,KEKL1='TEST',KEM1=L
```

The *mode* must be one of the encryption modes (for example, **03**).

Positioning of the Tape When Using the ASSGN Statement

The syntax of the ASSGN statement is as follows:

```
ASSGN SYSnnn,cuu,mode
```

If the tape specified in the *cuu* device address is *at load point*, the new *mode* setting is *immediately effective*.

If the tape specified in the *cuu* device address is *not at load point*, the new *mode* setting will be effective *the next time a write occurs at load point*.

For *mode* set to *encryption mode* (for example, X'03'):

- If the tape was *at load point*, the tape will be written *as encrypted*.
- If the tape was *not at load point*, the tape will continue writing *in the current mode*.

If the first file written to a tape is encrypted, all subsequent files written to that *same tape cartridge* will be encrypted using *the same data key*.

Handling Situations Where the EKM is not Available

If a tape contains encrypted data and is rewritten *without* encryption activated, the job might fail with a key-exchange error (described in "Understanding Message 0P68I KEYXCHG ER" on page 528).

This is because certain stand-alone utilities read the VOL1 label *before* starting the I/O process.

To overcome this problem, before you resubmit the job you should:
1. write a tape mark,
2. rewind the tape.

Running Stand-Alone Utilities (FCOPY, ICKDSF, DITTO, LIBR)

The stand-alone utilities FCOPY, ICKDSF, DITTO, and LIBR can be called from an encrypted stand-alone backup tape.

Backups performed from any of these utilities will be in *unencrypted format* only.

Additional Considerations When Using LIBR Utility

A LIBR BACKUP job with RESTORE=STANDALONE can be written in encrypted format.

- If an IPL is made from an *unlabeled* tape, there might be a delay when the key-exchange occurs. You might be required to re-IPL the tape drive using the `IPL cuu` command.
- If an IPL is made from a *labeled* tape, you might have to enter the `IPL cuu` command approximately *four times*, until the tape marks at the beginning of the tape have been skipped. This problem can also occur without encryption.

Overwriting Encrypted Volumes

If an encrypted volume is processed but the key is unknown to the EKM, access might fail with the message "0P68I Key Exchange Error" (described in "Understanding Message 0P68I KEYXCHG ER" on page 528).

To overcome this error, you can write a tape mark at the Beginning-Of-the-Tape (BOT).

Multivolume File Processing

To process a multivolume file on an *alternate* volume, you must specify the *same KEKL* as was specified for the *original volume*.

Here is an example of how to process a multivolume file. In this example, the specified alternate tape must also be assigned to encryption mode. The entry '**3**' represents encryption mode X'03'.

```
// ASSGN SYS005,cuu1,3
// ASSGN SYS006,cuu2,3
// ASSGN SYS005,cuu2,ALT
// KEKL UNIT=cuu1,KEKL1='TEST',KEM1=L
// KEKL UNIT=cuu2,KEKL1='TEST',KEM1=L
```

Chapter 47. Implementing the Encryption Facility for z/VSE

This chapter describes how you can use the *Encryption Facility for z/VSE* to implement the *software-based encryption* of:

- SAM files,
- VSAM files,
- VSE Library members,
- tapes and virtual tapes
- complete backups that were made using any z/VSE backup utility (such as IDCAMS, LIBR, POFFLOAD) or a vendor product.

Note: Throughout this chapter, the term:

- *EF for z/VSE* is used as the short-form for the *Encryption Facility for z/VSE*.
- *z/OS Java Client V1.2* is used as the short-form for the *Encryption Facility for z/OS Client Version 1 Release 2*.
- *EF for z/OS V1.1* is used as the short-form for the *Encryption Facility for z/OS Version 1 Release 1*.

Note: You must have a partition size of at least **8 MB** for the partition in which the EF for z/VSE is to run.

You can also use the *EF for z/VSE OpenPGP* to implement the software-based encryption of *OpenPGP* data formats, where the term *PGP* is an abbreviation for "Pretty Good Privacy". For details, see Chapter 48, "Implementing the Encryption Facility for z/VSE OpenPGP," on page 569.

This chapter contains these main topics:

Related Topics:

For details of how to ...	Refer to ...
display your current hardware crypto environment	"Using Crypto Support with a z/VSE Guest under z/VM" on page 468 and "Displaying Hardware Crypto Status Information Under z/VSE" on page 469.
activate crypto support if you are using an External Security Manager (ESM)	"Using Crypto Support and an External Security Manager" on page 476.
encrypt *tapes* using an *encryption-capable tape device*	Chapter 46, "Implementing Hardware-Based Tape Encryption," on page 521.
install and use the EF for z/OS (the *Encryption Facility for z/OS*)	*IBM Encryption Facility for z/OS: User's Guide*, SA23-1349
encrypt using PGP ("Pretty Good Privacy") tools and features	Chapter 48, "Implementing the Encryption Facility for z/VSE OpenPGP," on page 569.

Overview of the EF for z/VSE

The EF for z/VSE (Encryption Facility for z/VSE) provides data protection by offering the encryption of data for:
- data exchange,
- archiving, and
- backing up.

It is a *software-based* encryption facility. Depending on the IBM server and the type of cryptographic hardware that you have installed, the EF for z/VSE uses hardware-accelerated crypto support for encryption and decryption.

Encrypted data can be exchanged between different operating systems:
- The EF for z/VSE can read encrypted files that were created using:
  - the EF for z/OS (Encryption Facility for z/OS),
  - the z/OS Java Client.
- Encrypted files that were created using the EF for z/VSE can be read by:
  - the EF for z/OS,
  - the z/OS Java Client,
  - z/OS Decryption Client.

The EF for z/VSE provides encryption for:
- SAM files,
- VSAM files,
- VSE Library members,
- complete backups made with any z/VSE backup utility (such as IDCAMS, LIBR, POFFLOAD) or a vendor product.

For single VSAM files or VSE Library members, the filenames are specified directly in the JCL that is used to invoke the IJBEFVSE utility.

For full backups, you must:
1. backup your data to a physical tape or Virtual Tape,
2. encrypt this backup tape to an encrypted dataset,
3. write the encrypted dataset to a second tape or DASD.

You can choose between *two* methods for encrypting data:

- *Passphrase-based encryption* (PBE), which is the simplest method since it does not require any key-generation or key-handling. On systems without TCP/IP for VSE/ESA or with other TCP/IP stacks, you can *only* use passphrase-based encryption. It is described in Figure 132.
- *Public-key encryption* (PKE), which requires the use of a *key management tool* (for example, Keyman/VSE) to either generate private and public keys, or to import public keys. It is described in Figure 133 on page 534.

Figure 132 provides a simplified description of *passphrase-based encryption*.

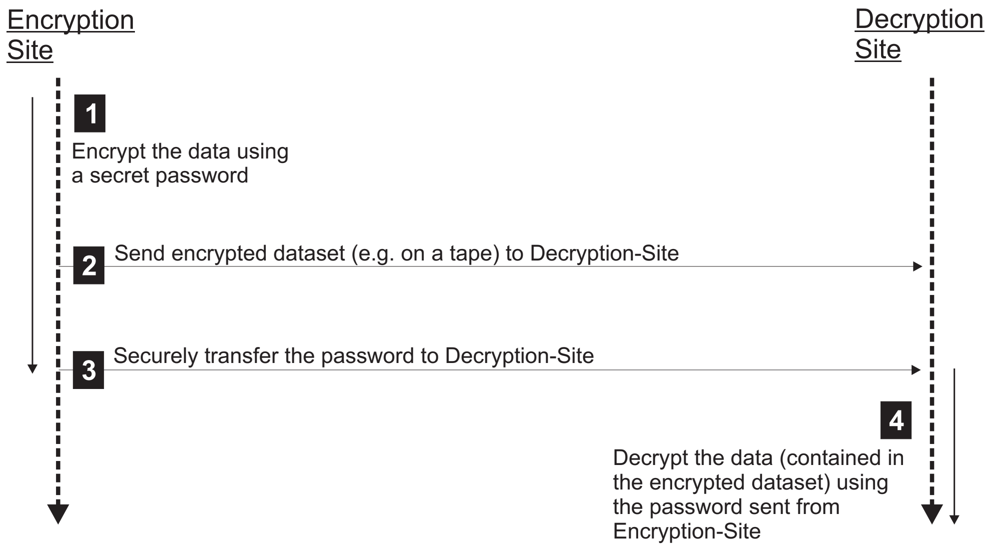


Figure 132. Overview of How Passphrase-Based Encryption (PBE) is Used

The number of each list item below describes a step shown in Figure 132:

1 The Encryption-Site encrypts the data on the dataset using a secret password. To do so, it runs a job that executes the IJBEFVSE utility and includes a secret password together with the dataset. The IJBEFVSE utility uses the secret password to generate a *data key* (a symmetric key), which it then uses to encrypt the data on the dataset. The secret password would then typically be stored using a password-management program.

2 The Encryption-Site sends the encrypted dataset to the Decryption-Site. For example, an encrypted dataset containing sales information is sent to the head office of the company.

3 In addition to sending the encrypted dataset, Encryption-Site also sends the secret password to Decryption-Site. This might be done by phone, by registered letter, and so on.

4 Decryption-Site decrypts the dataset using the secret password. To do so, it runs a job that executes the IJBEFVSE utility and includes the secret password together with the encrypted dataset. This job uses:
1. the secret password to generate the *data key*,
2. the data key to decrypt the data on the dataset.

The dataset is therefore made available for use.

Examples of the jobs described here are provided in “Examples of Using the IJBEFVSE Utility” on page 554.

Figure 133 provides a simplified description of *public-key encryption*.

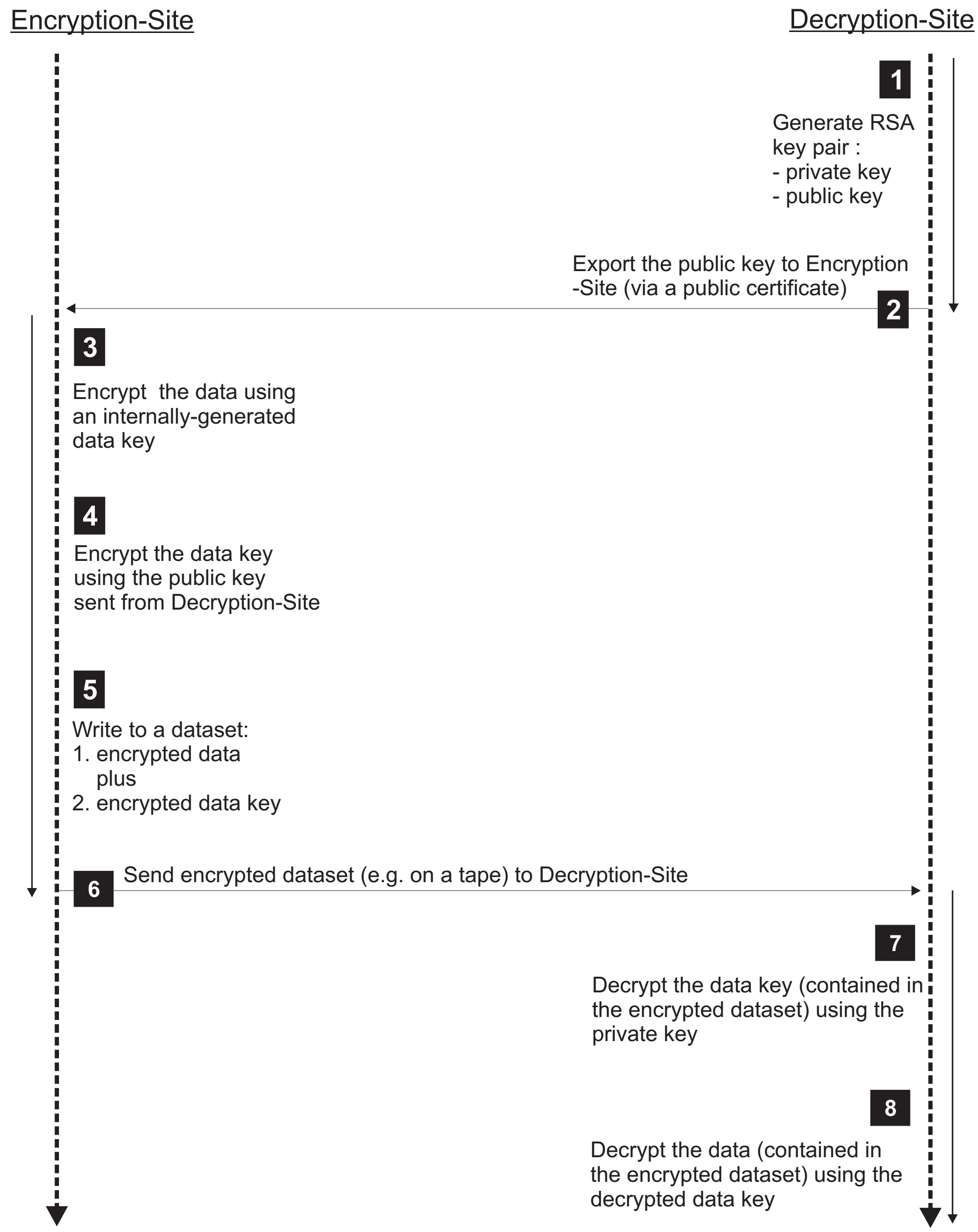


Figure 133. Overview of How Public-Key Encryption (PKE) is Used

The number of each list item below describes a step shown in Figure 133:

1 The Decryption-Site uses a key management tool (for example, Keyman/VSE) to generate an RSA *key pair* (a public key and a private key).

The public key is contained in a public certificate.

2 The Decryption-Site exports the public key to Encryption-Site using the public certificate.

3 The Encryption-Site generates its own data key and encrypts the data on the dataset. To do so, it runs a job that executes the IJBEFVSE utility and includes the data key together with the dataset.

4 The Encryption-Site encrypts the data key (that was used in **3**) using the public key. To do so, it runs a job that executes the IJBEFVSE utility and specifies the name of the VSE library member that contains the public key (as shown in "Example: Encrypt a Library Member Using Public-Key Encryption" on page 559).

5 The encrypted data, together with the encrypted data key, are written to a dataset.

6 The Encryption-Site sends the dataset to the Decryption-Site. For example, an encrypted dataset containing sales information is sent to the head office of the company.

7 The Decryption-Site uses its private key to decrypt the data key (Encryption-Site used the corresponding public key to encrypt the data key). To do so, it runs a job that executes the IJBEFVSE utility and specifies the name of the VSE library member that contains the corresponding *private key* (as shown in "Example: Decrypt a Tape That was Encrypted Using Public-Key Encryption" on page 559).

8 Using the same job that was started in **7**, the Decryption-Site uses the data key to decrypt the data contained on the dataset. The job includes the data key together with the encrypted dataset.

Examples of the jobs described here are provided in "Examples of Using the IJBEFVSE Utility" on page 554.

Prerequisites for Using the IJBEFVSE (or IJBEFPGP) Utility

To use *public-key encryption* you must have:

- defined a partition size of at least 8 MB for the partition in which the IJBEFVSE utility is to run.
- installed the Java Development Kit (JDK) 1.5 or higher.
- activated TCP/IP for VSE/ESA, as described in "Step 1: Activate TCP/IP for VSE/ESA" on page 478.
- set up an IBM Crypto Express2 card or equivalent that will be used to process 2048-bit RSA keys (see Chapter 41, "Implementing Hardware Cryptographic Support," on page 465).
- installed the TCP/IP for VSE/ESA system-fixes `ZP15E214`, `ZP15F246`, and `ZP15F230` that are supplied by *Connectivity Systems Inc*. You can obtain these fixes from `http://www.e-vse.com/csi-support/zaps15e.htm`.
- installed either the:
  - *Keyman/VSE*, as described in "Step 3: Download and Customize the Keyman/VSE Tool" on page 479, or
  - *CIAL client*, that is provided by *Connectivity Systems Inc* and can be downloaded free-of-charge. Using the CIAL client, you can generate an RSA key pair and upload this key pair to the z/VSE host. You can obtain the CIAL client from `www.e-vse.com`.
- activated the IBM *CPU Assist for Cryptographic Function* (CPACF) feature. To activate the CPACF feature, you must install the latest relevant microcode updates. To check if the CPACF feature has been activated, enter this command at the z/VSE console:

  ```
  MSG FB,DATA=STATUS=CR
  ```

  The resulting display shows if algorithms such as DES and TDES are available. In the following example, the CPACF feature *has* been activated:

```
FB 0011 CPU CRYPTOGRAPHIC ASSIST FEATURE:
FB 0011   CPACF AVAILABLE .................... : YES
FB 0011   INSTALLED CPACF FUNCTIONS:
FB 0011     DES, TDES-128, TDES-192
FB 0011     SHA-1
```

However, if the display shows only SHA-1 being available, this indicates that the CPACF feature *has not* been activated. For details of the algorithms that available on IBM servers, see Table 29 on page 592.

The only prerequisite for using *passphrase-based encryption* is that you must have activated the CPACF feature, as described for public-key encryption.

Restrictions When Using the IJBEFVSE (or IJBEFPGP) Utility

- The IJBEFVSE utility is *only* compatible with V1.1 of the EF for z/OS (Encryption Facility for z/OS).
- You can exchange encrypted data between z/VSE and non-z/VSE platforms *providing* the encrypted data was created using:
  - the IJBEFVSE utility,
  - the EF for z/OS V1.2,
  - the z/OS Java Client V1.2.
- The IJBEFVSE utility does *not* support secure key operations (ENCTDES).
- Depending upon the record format, some z/OS clear-data files (such as PDS members) will not be supported by z/VSE. A z/OS PDS cannot be restored into a z/VSE LIBR member or VSAM cluster.
- As stated previously, the IJBEFVSE utility supports encryption of *complete* tapes (CLRTAPE). If this tape is decrypted using the EF for z/OS V1.1 or the z/OS Java Client V1.2, this produces a tape in AWSTAPE format.
- When *decrypting* data, the CLRFILE file-attributes (LRECL, RECFM, BLKSIZE, and so on) must match the specifications in the header of the encrypted dataset. You can use the INFO option to display these attributes.
- When *encrypting binary files*, the z/OS Java Client V1.2 uses by default a fixed record length of *32760 bytes*. Therefore, if you transfer a file *to z/VSE* which has been encrypted using the z/OS Java Client V1.2, the destination file *must* be defined with a maximum record length of *at least 32760*.
- The z/OS Java Client V1.2 *cannot* decrypt data that was compressed using either the IJBEFVSE utility or the EF for z/OS V1.1.

Installing the EF for z/VSE

Installation Steps

1. Ensure you have met the requirements outlined in "Prerequisites for Using the IJBEFVSE (or IJBEFPGP) Utility" on page 535.
2. When you have ordered EF for z/VSE (Encryption Facility for z/VSE), you will receive a tape with the product which is included with the other "Optional Products" that are available with z/VSE. The procedure how to install an optional product within z/VSE is described in "Optional Installation Tasks" of the *z/VSE Installation*, SC34-2631.
3. You must install the product in a VSE sublibrary ('*lib.sublib*'). IBM recommends that you use sublibrary PRD2.PROD.

Fast Service Upgrade (FSU) Considerations

After refreshing your z/VSE system via an FSU, the EF for z/VSE members are still contained in *lib.sublib* providing *lib.sublib* is **not** one of the following:
- IJSYSRS.SYSLIB
- PRD1.BASE
- PRD2.SCEEBASE

The above libraries are *automatically* replaced during an FSU. Therefore, *the EF for z/VSE members will be deleted if installed there!*

Note that IBM always recommends upgrading optional products when upgrading the base operating system. Therefore the EF for z/VSE members should also be upgraded to the most current service level when performing an FSU.

Installing the z/OS Java Client

Note: IBM Support! The Encryption Facility for z/OS Client (whose short-form is the *z/OS Java Client*) is separately licensed *IBM Java Client* and *IBM Decryption Client for z/OS* code that is provided "AS IS" (without warranty, without product support) and is available as a no-charge, web download. While IBM has no obligation under this license to provide support, IBM does intend to review and analyze problems reported. At IBM's discretion, an updated z/OS Java Client web download (which includes the IBM Java Client and the IBM Decryption Client for z/OS) may be provided. To report problems with the Java Client or Decryption Client when used to exchange data with the Encryption Facility for z/VSE (5686-CF9), please contact: **`zvse@de.ibm.com`**.

To install the z/OS Java Client, you must:
1. Download the z/OS Java Client from:

   `http://www.ibm.com/servers/eserver/zseries/zos/downloads/#efclient`
2. Follow the instructions for installing the z/OS Java Client that are provided on the Web site.

Performance considerations For Using the IJBEFVSE Utility

This topic is also relevant for using the IJBEFPGP utility (described in Chapter 48, “Implementing the Encryption Facility for z/VSE OpenPGP,” on page 569).

The overall performance of an encryption or decryption process depends on these issues:
- **Compression.** Compressing data prior to encryption usually speeds up the process, because less data has to be encrypted. The compression speed can be defined using the parameter COMPRESSION (described in Table 23 on page 544). Note that ZIP compression uses *host CPU cycles*, whereas the System z-based compression used with the IJBEFVSE utility does *not* use host CPU cycles.
- **Encryption algorithm**. There are significant differences in terms of speed between the supported encryption algorithms. AES performs much faster than TDES. Larger key lengths will slow down performance. When using public-key encryption (PKE), there is no significant difference between the different public key sizes, because only the *data key* is encrypted using a public key.
- **Hardware support.** There are various hardware dependencies:
  - On a z9 platform, *only AES-128* is provided by CPACF, whereas AES-256 (for example) would have to be performed in software.

  - On z10, z114, and z196 platforms, *all AES key lengths* are supported by CPACF.
- **Physical I/O.** When encrypting a file with many small records (for example a KSDS file), the encrypted dataset usually has a much larger record length, using the maximum possible of the underlying ESDS file. Therefore, writing *encrypted* data to disk requires much less I/O than writing the *decrypted* dataset (with its many small records) during the decryption process.

Setting Up to Use Passphrase-Based Encryption (IJBEFVSE)

Note:

1. If you plan to delete your original unencrypted data after encryption, ***you are strongly recommended*** to:
   - Verify that your data can be decrypted successfully *before* destroying any original data.
   - Keep a copy of the IJBEFVSE utility version you used to encrypt the data, to ensure you can perform the decryption at any time in future.
2. In this description, the term "password" is used instead of "passphrase".

When you use a password to encrypt the data contained in a dataset, the data key is *never* explicitly accessed by the user. You must only ensure that the password you use is stored in a secure location.

Usually, you will store the password together with the associated tape label or file name. At some time in the future, you can then restore the encrypted tape or dataset using the password and tape label or file name.

Using passphrase-based encryption, the *data key* that is used for encryption is generated from:
- the password you enter,
- the iteration count, and
- an 8-byte random number (the "salt") which varies according to the encryption process.

In the *header* of the encrypted dataset are stored:
- the iteration count,
- the "salt" value.

If the same dataset is encrypted twice using the *same* password and iteration count, the resulting encrypted datasets *will be completely different!* This is because a *randomly-generated* salt value is included in the creation of the data key. For details about the `ICOUNT=` iteration-count parameter, see Table 23 on page 544.

To manage your passwords, you can use any available tool from vendors, freeware, or shareware. You can find suitable tools by searching the Web for the term "Password Manager".

Setting Up to Use Public-Key Encryption (PKE)

Note: If you plan to delete your original unencrypted data after encryption, ***you are strongly recommended*** to:
- Verify that your data can be decrypted successfully *before* destroying any original data.
- Keep a copy of the IJBEFVSE utility version you used to encrypt the data, to ensure you can perform the decryption at any time in future.

Overview of How Keys/Certificates are Used

As described in Figure 133 on page 534, the randomly-generated *data key* is encrypted using a *public key* sent from the decryption site. This *encrypted data key* is stored in the *encrypted dataset header*. When the decryption site wishes to decrypt the dataset, it will use its corresponding *private key*. The encryption and decryption sites can, of course, be the same site.

This process guarantees that although multiple sites can *encrypt* data, only *one single site* is able to *decrypt* the data. This is because private keys, which are used for decryption, are never (intentionally) distributed between systems. Therefore, public-key encryption *is more secure* (but also more restrictive) than passphrase-based encryption.

However, if you wish to create local backups which *do not leave* the encryption site, public-key encryption is relatively easy to implement.

The *VSE Keyring Library* on the z/VSE host has the default name **CRYPTO.KEYRING**. It contains these members:
- *RSA key pairs*, which have the suffix **.PRVK**. Each .PRVK member contains a key pair (a private key *and* a public key) which are *not* stored as certificates. **Note:** the name “PRVK” might be confusing, since it “sounds” as if it *only* contains a private key.
- *Public key certificates*, which contain public keys and have the suffix **.CERT**.
- *Root certificates* which have the suffix **.ROOT**. These certificates can be either self-signed, or signed by a Certificate Authority (CA).

Using *Keyman/VSE* running on the Java platform of a PC or workstation, you can:
1. generate an RSA key pair on a PC or workstation (**Note**: you *cannot* generate an RSA key pair on your z/VSE host).
2. store the RSA key pair in a password-protected *local* key store, which can be a:
   - z/VSE-supplied client keyring file (**KeyRing.pfx**, described in “Step 2: Create a Client Keyring File (KeyRing.pfx)” on page 478, or
   - z/OS-supplied client keyring file **Keyring.jks**.
3. upload the RSA key pair to a z/VSE system, where the key pair is catalogued as a .PRVK library member in the *VSE Keyring Library* (CRYPTO.KEYRING).

When *encrypting* data, you can specify up to *16 RSA control statements*:
- Each control statement identifies the label of *one* RSA public key member (PRVK or CERT member).
- Depending on the number of RSA control statements, you can send an encrypted file to up to *16 different recipients*.

When *decrypting* data, you specify *one RSA control statement only*. This identifies the .PRVK member which contains the corresponding *private key* to one of the 16 possible *public keys* used that was used to encrypt the data.

Table 22 shows the possibilities for using public-key encryption to encrypt and decrypt data. The actions listed are described in the topics that follow the table.

Table 22. Encryption/Decryption Possibilities When Using Public-Key Encryption (PKE)

	Decrypt on z/VSE Using IJBEFVSE utility	**Decrypt on z/OS Using EF for z/OS V1.1**	**Decrypt on Java Platform Using z/OS Java Client**
Encrypt on z/VSE Using IJBEFVSE utility	Generate RSA key-pair with Keyman/VSE and upload this key pair to z/VSE.	Generate RSA key-pair with Keyman/VSE and export the public key "enclosed in a certificate" for further use on z/OS.	Generate RSA key-pair with Keyman/VSE and export the public key "enclosed in a certificate" for further use on a Java workstation.
Encrypt on z/OS Using EF for z/OS V1.1	Import the certificate containing the z/OS public key into Keyman/VSE and upload it to z/VSE.	Not applicable to z/VSE	Not applicable to z/VSE
Encrypt on Java Platform Using z/OS Java Client	Import the certificate containing the public key from the JKS keystore into Keyman/VSE and upload it to z/VSE.	Not applicable to z/VSE	Not applicable to z/VSE

Define Properties of Host and Generate/Upload a Key Pair to the Host

This topic involves the use of the Keyman/VSE tool. To use it, you must have installed the *VSE Connector Client*. For details, refer to the chapter "Installing and Operating the Java-Based Connector" in the *z/VSE e-business Connectors User's Guide*, SC34-2629.

These are the steps you should follow:

1. Define the properties of your z/VSE system in the *VSE Host Properties* dialog of the Keyman/VSE tool (providing you have not already done so). For details, see "Specify the Properties of Your z/VSE Host" on page 479. If you will decrypt on a z/VSE system that is different to the one used for encryption, repeat this step for the second z/VSE system.
2. Create an RSA key pair and select your required key length. See "Step 1. Create an RSA Key Pair" of "Configuring for Server Authentication Using Self-Signed Certificates" on page 489.
3. Upload this key pair to your z/VSE system. See "Step 5. Upload the 1024-Bit RSA Key Pair to the z/VSE Host" of "Configuring for Server Authentication Using Self-Signed Certificates" on page 489. If you are decrypting on a different z/VSE system, also upload the key pair to the second z/VSE system.
4. If you now wish to follow the procedure described in "Export a Public Key for Use with the z/OS Java Client" on page 541, you should *save the key pair*. Otherwise, you can delete the key pair from Keyman/VSE (or simply quit Keyman/VSE without saving).

If the *same public key* is used on *different z/VSE systems*, you are strongly recommended to use the *same z/VSE library-member name*.

Export a Public Key for Use with the z/OS Java Client

This topic describes how to export the *public key*, that was generated in "Define Properties of Host and Generate/Upload a Key Pair to the Host" on page 540, into a JKS Keystore. This public key is contained in a *signed server certificate*.

A z/OS Java Client can then use the public key (the signed server certificate) to encrypt data. This encrypted data can then be later *decrypted* using a *private key* stored on the z/VSE host.

These are the steps you should follow:

1. Complete these four steps using Keyman/VSE:
   a. Create a Self-Signed Root Certificate
   b. Create a Request for a VSE Server Certificate
   c. Sign the Request for a VSE Server Certificate
   d. Upload the RSA Key Pair to the z/VSE Host

   The above four steps are described in Steps 2 to 5 of "Configuring for Server Authentication Using Self-Signed Certificates" on page 489.
2. The signed VSE server certificate is now stored in Keyman/VSE. It contains the *public key* belonging to the RSA key pair. You can now delete the other items from Keyman/VSE, since they are no longer required.
3. On the Keyman/VSE toolbar click **Local file properties**, as shown in Figure 134.

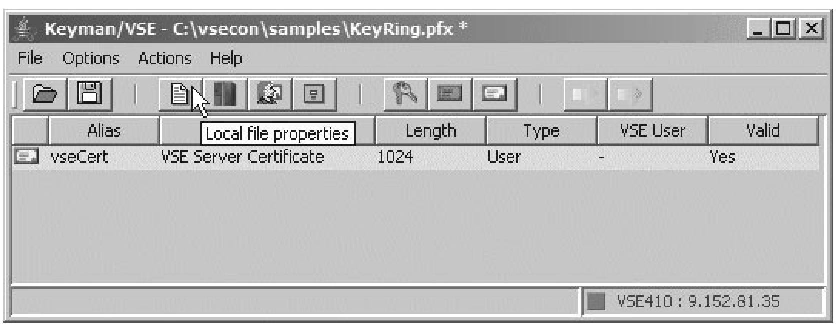


Figure 134. Local File Properties icon on Keyman/VSE Toolbar

4. Select the JKS option, as shown in Figure 135 on page 542. To protect the JKS Keystore, enter a password. Then click **OK**.

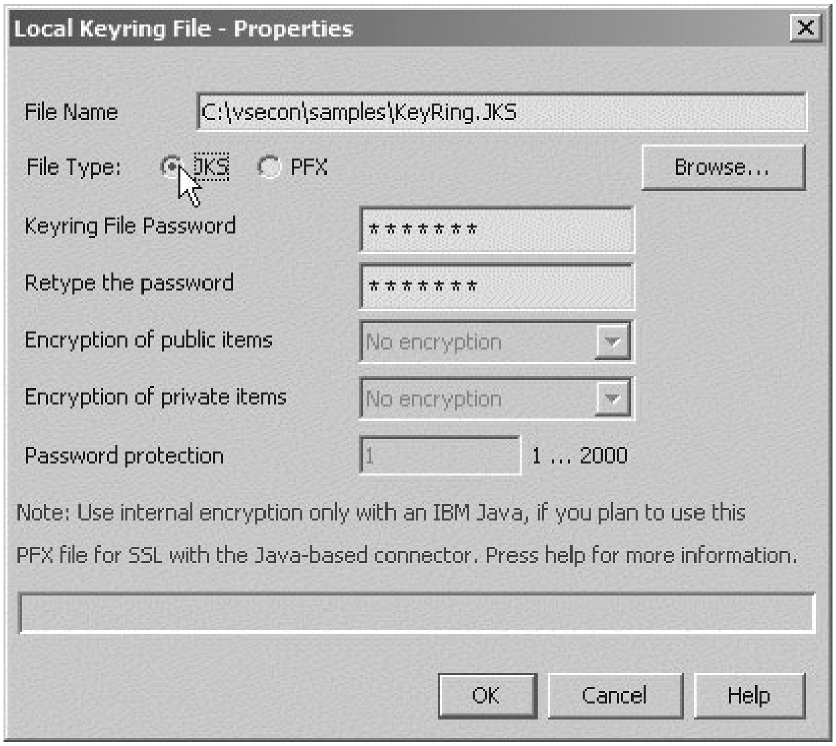


Figure 135. Selecting JKS Option in Keyman/VSE

5. Click the **Save** icon to "export" the JKS Keystore file for later use, as shown in Figure 136. The JKS Keystore now contains the *signed server certificate with its public key*. Please not that you *cannot* save RSA key pairs in a JKS Keystore, since keys must *always* be stored on certificates. However, you *can* save keys in the z/VSE-supplied **KeyRing.pfx**.

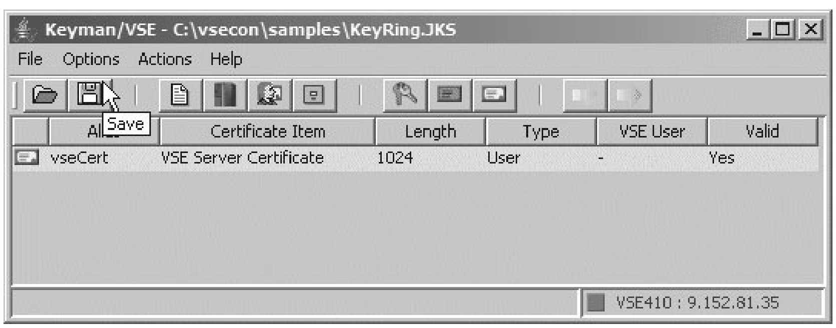


Figure 136. Save icon on Keyman/VSE Toolbar

Export a Public Key for Use on z/OS or a Java Platform

This topic describes how to *export* the *public key*, that was generated in "Define Properties of Host and Generate/Upload a Key Pair to the Host" on page 540, into a JKS Keystore that can then be used by:

- the EF for z/OS (Encryption Facility for z/OS) V1.1 on z/OS, or
- a z/OS Java Client on a Java platform.

As explained in "Overview of How Keys/Certificates are Used" on page 539, the VSE Keyring Library on the z/VSE host contains *Keyrings*. In *each* Keyring, the *public key* is stored in both the:

- **.PRVK** member (together with the corresponding private key),
- **.CERT** member (a server certificate).

To export the public key for use on either a z/OS or Java platform, you must:

1. Either download the server certificate using Keyman/VSE, or copy the text-form of the server certificate onto the Clipboard.
2. Use the Export function of Keyman/VSE to store the server certificate in a local file.
3. Import the local file into a JKS Keystore, where it can be used by the EF for z/OS V1.1 or z/OS Java Client.

Import a Public Key into z/VSE from z/OS or a Java Platform

This topic describes how to *import* a *public key into z/VSE* that was generated on z/OS or a Java platform. The public key will be contained in a *user certificate* that was sent from z/OS or a Java platform, to the z/VSE system.

After the user certificate has been exported on z/OS or a Java platform and transferred to the *Java platform where Keyman/VSE is installed*, you can:

1. *Import* the user certificate into Keyman/VSE.
2. *Upload* the user certificate to z/VSE and store it in a **.CERT** member.

The `RSA=` parameter of the IJBEFVSE utility identifies the *public key member*, for example:

```
RSA=CRYPTO.KEYRING(ZOSPUBKY)
```

Please Note: It is important to ensure that the name of the VSE library **.CERT** member *is not the same as any other* ***.PRVK*** *member*. Otherwise, *the public key might be taken from a public/private key pair!* This is because when processing the `RSA=` parameter, the IJBEFVSE utility:

1. attempts to find a matching **.PRVK** member (containing a public/private key pair), and if it *does not exist*
2. attempts to find a matching **.CERT** member (containing a public key).

If the VSE Keyring Library on the z/VSE host contains a **.CERT** member that was imported from z/OS or a Java platform, there *will not be* a corresponding **.PRVK** member. Therefore, the public key that is stored in the **.CERT** member *will always be used.*

To *export* a *user certificate* from a JKS Keystore, you must:

1. Use the keytool `list` command to display the contents of the keystore:

   ```
   keytool -list -keystore keystore.jks -storepass password
   ```
2. The `list` command produces output similar to the following:

```
Keystore type: jks
Keystore provider: ORACLE

Your keystore contains 2 entries

zos_key, Jun 15, 2005, keyEntry,
Certificate fingerprint (MD5): FC:83:9A:30:F8:EC:CE:31:AB:C7:21:DE:17:CF:1C:E0
zos_cert, Jun 15, 2005, trustedCertEntry,
Certificate fingerprint (MD5): 7A:69:0E:41:CA:8E:34:FF:AE:05:BB:18:CA:7E:AA:B0
```

3. Export the required user certificate into a plain text file:

```
keytool -export -alias zos_cert -file zos_pubkey.txt
-keystore keystore.jks -storepass password
```

4. Import the user certificate into Keyman/VSE, and upload it to your z/VSE system.

Invoking the IJBEFVSE Utility

This topic describes the:
- syntax of the IJBEFVSE utility,
- control statements that can be used with the IJBEFVSE utility,
- rules for specifying file names when using the CLRFILE and ENCFILE control statements,
- rules for specifying record formats.
- jobs must include these statements:

```
// EXEC IJBEFVSE [,PARM='[SYSLST=DD:SYSnnn] [DEBUG]']
control statements
 :
/*
```

where:

SYSLST=DD:SYS*nnn*
: Optional. Specifies a logical unit to-which the listing should be written (for example SYSLST=DD:SYS004)

DEBUG
: Optional. Enables debugging (messages will be sent to SYSLST and to the console).

control statements
: Details are given in Table 23. Control statements are passed via SYSIPT.

Table 23 lists the control statements you can use with the IJBEFVSE utility. If you specify the same control statement more than once, only the *last* specification will be used. All control statements are passed via SYSIPT.

Table 23. Control Statements Used With the IJBEFVSE Utility

Statement	Explanation
DESC='*text*'	Optional, and only used for *encryption* processing. Specifies 1 to 64 EBCDIC character bytes of descriptive text to be placed in the header record. IJBEFVSE places the text in the header record. The text is used to assist in identifying the source of the encrypted data in the output. • You must enclose the text in single quotation marks. • Imbedded blanks are allowed. • All text must be included on *one* control statement line.

Table 23. Control Statements Used With the IJBEFVSE Utility (continued)

Statement	Explanation
CLRTDES	Only used for *encryption* processing. For decrypting, this information is contained in the header of the encrypted file. Specifies that the input file is to be encrypted with a clear TDES triple-length key.
CLRAES128	Only used for *encryption* processing. For decrypting, this information is contained in the header of the encrypted file. Specifies that the input file is to be encrypted with a clear 128-bit AES key.
COMPRESSION=NO|YES	Specifies whether you want compression of the clear input before encryption of the data occurs: • COMPRESSION=NO indicates that compression does not occur. • COMPRESSION=YES causes compression to be performed before encryption. • If you do not specify the COMPRESSION keyword, the default is NO.
ENCTDES	*Not supported by z/VSE.* In z/OS, ENCTDES specifies that the input file is to be encrypted with a secure TDES triple-length key. Only used for encryption processing. For decrypting, this information is contained in the header of the encrypted file.
RSA=*label*	Specifies the 64-byte label of an existing RSA private/public key pair that is present in the VSE Keyring library (for example, CRYPTO.KEYRING). The program uses this key to encrypt the data encryption key. When you decrypt the data, the RSA key must be present at the recovery site. In z/VSE, this parameter specifies the 8 character name of the private key member in the keyring library. The member name is specified as follows: • lib.slib(*member*) – including name of keyring library, where • *member* – the default keyring library is used (KEYRING.LIBRARY) To encrypt data, you can use up to 16 `RSA=` keywords to specify up to 16 public-key labels. Depending on the number of multiple RSA labels, you can send the encrypted file to up to 16 individual recipients. The RSA keyword is optional when: • you decrypt data that is encrypted with a single `RSA=` control statement, or • the RSA private key label is the same as the RSA public key label used to protect the data-encrypting key. This is because the RSA key label is stored in the header record. For data encrypted with multiple `RSA=` control statements, you must specify one single RSA keyword for decryption. Decryption does *not* allow multiple RSA keywords. For details, see "Example: Use Multiple RSA Control Statements for Multiple Remote Systems" on page 560.
PASSWORD=*pwd*	Specifies the 8- to 32- EBCDIC character password to be used to generate: • a clear TDES triple-length key, or • a clear 128-bit AES key. Leading and trailing blanks and tab characters are removed; imbedded blanks and tab characters are allowed. Passwords are case sensitive. In order to minimize problems caused by code-page differences at the encrypting and decrypting sites, IBM suggests you use only the upper and lower-case letters A through Z, numerals 0 – 9, and the underscore character (_).

Table 23. Control Statements Used With the IJBEFVSE Utility (continued)

Statement	Explanation
ICOUNT=*nnnnn*	Only used for *encryption* processing. For decrypting, this information is contained in the header of the encrypted file. When you specify a password, ICOUNT specifies the number of iterations that the SHA-1 hash algorithm is to be performed in the generation of the data key and the initial chaining vector (ICV) for encryption. *nnnnn* is an integer between 1 and 10,000. If you do not specify ICOUNT, the default is 16. ICOUNT allows you to strengthen security when you use PASSWORD. If you specify a robust password (specifying 32 random characters), the default is sufficiently secure.
COMPRESSION=NO|YES	*Compression is not supported by z/VSE*. Therefore, only COMPRESSION=NO is accepted. This parameter can only be used for encryption processing on a *z/OS system*. *z/VSE cannot decrypt datasets that were compressed prior to encryption!*
INFO	Specifies that file decryption is *not* to be performed, but information about the defaults that IJBEFVSE establishes *is to be recovered* and written to SYSLST. When the information is written, IJBEFVSE processing ends. This option is useful when you want to: • determine the original clear-text file format information, or • ensure that a specified RSA key is present in the current keyring library.
ENCRYPT	Specifies that encryption is to be performed. IJBEFVSE will read the input file, encrypt the data, and write it to the output file. The output file will contain a header with enough information to later decrypt the file.
DECRYPT	Specifies that decryption is to be performed. IJBEFVSE will read the input file, decrypt the data, and write it to the output file.
CLRFILE=*name*	Specifies the name of the clear-data file or dataset: • For encryption, the clear data is the input. • For decryption, the clear data is the output. CLRFILE can specify a: • LIBR member, • VSAM cluster (ESDS, RRDS, KSDS), or • SAM dataset, on tape or DASD. See also "Specifying File Names for CLRFILE and ENCFILE" on page 548.
CLRTAPE=SYS*nnn*	Specifies the logical unit of the clear-data tape. • For encryption, the clear data is the input. • For decryption, the clear data is the output. The whole content of the specified tape is used, including all tape marks, tape labels and so on.
ENCFILE=*name*	Specifies the name of the encrypted file or dataset. • For encryption, the file or dataset is the input. • For decryption, the file or dataset is the output. ENCFILE can specify a: • LIBR member - Note that when specifying a LIBR member, you must use a member type that consist of more than one character. You *cannot* use member types with only one character (A-Z, 0-9, $, #, @) for ENCFILE. • VSAM ESDS, KSDS and RRDS clusters, • SAM dataset, on tape or DASD. See also "Specifying File Names for CLRFILE and ENCFILE" on page 548.

Table 23. Control Statements Used With the IJBEFVSE Utility (continued)

Statement	Explanation
RECFM=*n*	Specifies the record format of the CLRFILE. For VSAM clusters and LIBR members this parameter is *not* required, since the format information is read from the VSAM catalog or Library. For SAM datasets on tape or DASD this parameter *is* required. Valid values are: • F – fixed • V – variable • B – blocked • S – spanned • U – undefined • FB – fixed blocked • FBS – fixed blocked spanned • VB – variable blocked • VBS – variable blocked spanned In addition to the above formats, you can add an: • A (ASA print-control characters), or • M (machine print-control characters). When decrypting, the RECFM information is contained in the header of the encrypted file. If used with the IGNOREHEADER option, the settings in the header can be overruled. For further details about supported records formats and defaults, refer to: • "Record, Block, and Control Interval" in the *z/VSE System Macros User's Guide*, SC33-8407, and • "Record Formats" in the *LE/VSE C Run-Time Programming Guide*, SC33-6688.
LRECL=*nnn*	Specifies the logical record length of the CLRFILE. For VSAM clusters and LIBR members this parameter is *not* required, since the record length information is read from the VSAM catalog or Library. For SAM datasets on tape or DASD, this parameter *is* required. When specifying LRECL, you must also specify the RECFM control statement. When decrypting, this information is contained in the header of the encrypted file. If used with the IGNOREHEADER option, the settings in the header can be overruled. For further details about supported records formats and defaults, refer to: • "Record, Block, and Control Interval" in the *z/VSE System Macros User's Guide*, SC33-8407, and • "Record Formats" in the *LE/VSE C Run-Time Programming Guide*, SC33-6688.
BLKSIZE=*nnn*	Specifies the bloc size of the CLRFILE. For VSAM clusters and LIBR members this parameter is *not* required, since the record length information is read from the VSAM catalog or Library. For SAM datasets on tape or DASD this parameter *is* required. When specifying BLKSIZE, you must also specify the RECFM control statement. When decrypting, this information is contained in the header of the encrypted file. If used with the IGNOREHEADER option, the settings in the header can be overruled. For further details about supported records formats and defaults, refer to: • "Record, Block, and Control Interval" in the *z/VSE System Macros User's Guide*, SC33-8407, and • "Record Formats" in the *LE/VSE C Run-Time Programming Guide*, SC33-6688.
IGNOREHEADER	Only used for *decryption* processing. Specifies that the record format, logical record length, and block size specifications in the header of ENCFILE should *not* be used. Instead, the specifications from the RECFM, LRECL and BLKSIZE control statements are used. This option is useful when you want to restore an encrypted file into a file or dataset with different record formats and lengths.

Table 23. Control Statements Used With the IJBEFVSE Utility (continued)

Statement	Explanation
APPEND	Only used for *decryption* processing. Specifies that the CLRFILE is to be opened for append. Useful for existing VSAM clusters that have not been defined with REUSE=YES.
CODEPAGE=*nnnn*	Specifies the EBCDIC code page used for SYSIPT input. This code page is important when specifying the password. Special characters (such as '@') are on different code points in different code pages. Default is IBM-1047. The code page statement must appear before the PASSWORD=*nnn* control statement.

Deciding Whether or Not to Use Data Compression

If you have to archive large amounts of encrypted data, you might wish to *compress* the data. For example, compressing the data might help to reduce the number of tape volumes you require.

Because encrypted data is *not* very compressible, you might use the COMPRESSION option (described in Table 23 on page 544) to compress data *before* encryption is performed.

If you plan to use the z/OS Java Client to decrypt data, please note that the z/OS Java Client cannot decrypt data that was compressed using the IJBEFVSE program.

Specifying File Names for CLRFILE and ENCFILE

Note:

1. The CLRFILE parameter is used when decrypting a *complete tape*.
2. The ENCFILE parameter is used when encrypting a *complete tape*.
3. For details of CLRFILE and ENCFILE, see Table 23 on page 544.

The following rules apply for specifying file names with the CLRFILE and ENCFILE control statements.

File names for LIBR members

`DD:<library>.<sub-library>(<member>.<type>)` For example:

```
DD:PRIMARY.SYSA(MYFILE.TXT)
```

Note: When specifying a value for ENCFILE, you must use a member type that consist of more than one character. You *cannot* use member types with only one character (A-Z, 0-9, $, #, @) for ENCFILE.

File names for VSAM clusters

`DD:<dlbl>` The specified DLBL must be present in the system standard, class or partition label area and must point to an existing VSAM cluster. Please note that VRDS clusters are *not* supported. ENCFILE only accepts VSAM ESDS clusters. CLRFILE accepts ESDS, KSDS and RRDS clusters. For example

```
DD:VSAMCLU
```

File names for SAM datasets on tape

`DD:<lu>-<tlbl>` or `DD:<lu>` The LU specifies a logical unit in the form SYSnnn. A tape must be assigned to this LU. If the TLBL (tape label) is specified (concatenated by a dash), a labeled tape file will be opened. If TLBL is not specified, an unlabeled tape file is opened. The specified TLBL must be present in the system standard, class or partition label area. Multivolume files and multiple files on a volume are supported. For example

```
DD:SYS004-TAPEFIL
```

File names for SAM datasets on DASD
The specified DLBL must be present in the system standard, class or partition label area. EXTENT information must be also present via a DLBL. For example

```
DD:<dlbl>
```

Specifying File Attributes and Record Formats

When encrypting data on z/VSE, you must pass the *file attributes of the clear data* to the IJBEFVSE utility. If you are encrypting a VSAM file or a library member, the file attributes will be known by the VSAM Catalog or the Library.

If you are encrypting datasets on DASD or tape, the file attributes are *not known*. Therefore, you must specify the file attributes when you invoke the IJBEFVSE utility. To do so, you use the control statements RECFM, LRECL and BLKSIZE (described in Table 23 on page 544 and Table 27 on page 574):

- The statements RECFM, LRECL and BLKSIZE might be dependent on each other: for example, if you are using a blocked format then you must specify BLKSIZE.
- The IJBEFVSE utility stores the file attributes in the header record of the encrypted dataset.
- If you specify file attributes for VSAM files or library members that *do not match* the attributes in the catalog or Library, a warning message will be written to SYSLST and the file attributes from the catalog or library will be used.
- If you specify *incorrect record formats*, z/VSE's *Basic Access Method* might cancel the IJBEFVSE utility. For further details about record formats, refer to:
  - "Record, Block, and Control Interval" in the *z/VSE System Macros User's Guide*, SC33-8407, and
  - "Record Formats" in the *LE/VSE C Run-Time Programming Guide*, SC33-6688.

When decrypting data on z/VSE, the file attributes of the clear data are contained in the *header record* (described in "Layout of Header-Record of Encrypted Dataset" on page 551) of the encrypted dataset:

- You can use the INFO option to display the file attributes stored in the header.
- When decrypting into a dataset on DASD or tape, the information from the header is used to create the dataset.
- When decrypting into a VSAM file or library member, the file attributes may conflict with the file attributes that are stored in the catalog or Library. In this case, the attributes form the catalog or Library will be used and an appropriate warning message will be written to SYSLST.
- You can override the attributes stored in the header by using the IGNOREHEADER control statement (described in Table 23 on page 544). During this process, the file attributes from the header will be ignored. You must therefore specify these attributes using the control statements RECFM, LRECL and BLKSIZE (described in Table 23 on page 544).
- The statements RECFM, LRECL and BLKSIZE may be dependent on each other: for example, you must specify BLKSIZE if you are using a blocked format.
- When decrypting into a VSAM file or library member, no specifications are required, since they are taken from the catalog or library.

Encrypting and Exchanging Record-Based Data

This topic describes how you can:

- encrypt/decrypt *record-based data*.
- exchange encrypted record-based data with other operating-system platforms.

You are more likely to need to encrypt record-based data than *JPEG data* (which was described in previous topics, and which does *not* have a record structure).

See also "Known Problems When Encrypting and Exchanging Data" on page 567.

Types of Data That Might Need to be Encrypted

These are the types of data that you might need to encrypt, and which affect the encryption actions you must take:

Stream-based data
Binary data without any internal structure. Its representation on different platforms is exactly the same. However, there might be some difference in the underlying file systems:

- On *workstation platforms* we have flat files without any record structure.
- On *z/VSE platforms*, we have VSAM files in which a binary stream might be split into multiple records according to internal VSAM logic.

Line-based data
The simplest example of line-based data is a plain-text file with line-break characters. On ASCII platforms, there is a CRLF sequence (X'0D0A') at the end of each line. On EBCDIC platforms, lines are split by a new-line character (X'15').

Note: All lines are explicitly separated by *line breaks*.

Record-based data
Record-based data is stored in files that contain logical records *without* line-break characters. For example:

- A *VSAM file* consists of record-based data in which each logical data record is stored in one physical record of the VSAM file.
- A *CMS file* in z/VM contains data records whose lengths are given by the record length of the CMS file.

You should also be aware that if you use the z/OS Java Client to encrypt an EBCDIC encoded data file on z/OS, when you decrypt on z/VSE the clear-file contents will be restored in EBCDIC-readable form. However:

- If you decrypt the file on an ASCII platform (such as on a Windows PC), the decrypted data will *still* be encoded in EBCDIC.
- There is no option provided in which you could specify a character set that might be used when decrypting textual data.

Layout of Header-Record of Encrypted Dataset

The encrypted data that is produced by the IJBEFVSE utility (as well as by the *EF for z/OS*) includes a *header record*. This record contains the information you require in order to *decrypt* the data on a:
- mainframe, or
- workstation running the *z/OS Java Client*.

Table 24 shows the layout of the header record.

Table 24. Layout of Header Record That is Included in Encrypted Data

Offset (Decimal)	Name of Header Field	Type of Data	Description
0	HEADER_EYE	Character	An eye-catcher: “HEADER” in EBCDIC
6	HDR_VERSION	Character	Version of the header record for Encryption Facility.
8	HDR_DESC	Character	EBCDIC description (DESC keyword) of encrypted data.
72	HDRLEN	Integer	Length of entire header record (integer format).
76	HDRSALT	Character	8-byte field (salt value) used with password.
84	HDRICNT	Integer	Iteration count (ICOUNT keyword), integer format from 1 to 10,000 to be used with password.
88	HDRKEYLN	Character	Modulus length (hexadecimal format from 512 ... 2048) in bits of the RSA public/private key taken from the RSA keyword information.
90	HDRRSA	Character	Pathname of the PRVK or CERT member containing the RSA public/private key in the format *lib.sublib*(*membername*)
154	HDRICV	Character	Initialization chaining vector to be used with encryption/decryption.
170			Reserved for IBM use.
174	HDAESDES	Bit	Type of key to be used to encrypt/decrypt data: • X'01' use a clear TDES triple.length key. • X'02' use a clear 128-bit AES key. • X'03' use a secure TDES triple-length key.
175	HDRFLAGS	Bit	Bit string that indicates type of output, compression options, and format of encrypted data: • Bit 0 = 1: unused. • Bit 1 = 1: indicates output data compressed . • Bit 2 = 1: indicates compression dictionary is present in the encrypted data. • Bit 3 = 1: indicates clear data is binary. • Bit 3 = 0: indicates clear data is text (not used by z/VSE).
176	HDR_COMPVER	Character	Version of Encryption Facility compression used.
178	HDRIRECF	Bit	Input file record format: • Bit 0 = 1, Bit 1 = 0: Fixed. • Bit 0 = 0, Bit 1 = 1: Variable. • Bit 0 = 1, Bit 1 = 1: Undefined. • Bit 3 = 1: Blocked records. • Bit 5 = 1: ASA control character. • Bit 6 = 1: Machine control character.
179	HDRIRECL	Integer	Input file logical record length.
181	HDRIBLKS	Integer	Input file block size.

Table 24. Layout of Header Record That is Included in Encrypted Data (continued)

Offset (Decimal)	Name of Header Field	Type of Data	Description
185	HDRORECF	Bit	Output file record format: • Bit 0 = 1, Bit 1 = 0: Fixed. • Bit 0 = 0, Bit 1 = 1: Variable. • Bit 0 = 1, Bit 1 = 1: Undefined. • Bit 3 = 1: Blocked records. • Bit 5 = 1: ASA control character. • Bit 6 = 1: Machine control character.
186	HDRORECL	Integer	Length of output file logical record.
188	HDROBLKS	Integer	Output file block size.
192	HDR_KEYVAL	Integer	Encrypted data-encryption key.
448			Reserved for IBM use.
464	HDR_RSA_CNT	Integer	**Applies only when the “HEADER” is version X'0002' or greater**: Number of RSA= control statements.
468	HDR_RSA	Character	**Applies only when the “HEADER” is version X'0002' or greater**: An array consisting of information for multiple RSA= control statements. The length is variable based on the number of RSA= control statements, with each entry 344 bytes in length.
	HDR_RSA_LAB	Character	An element of HDR_RSA consisting of a 64-byte label of one of the RSA public/private keys, that is, the pathname of the PRVK or CERT member containing the RSA public/private key in the format *lib.sublib*(*membername*).
532	HDR_KEY_LN	Character	An element of HDR_RSA consisting of Modulus length (hexadecimal format from 512 - 2048) in bits of the RSA public/private key in this entry.
534		Character	Two-byte placeholder of HDR_RSA.
536	HDR_KEY_VAL	Character	An element of HDR_RSA consisting of the hexadecimal encrypted data-encrypting key. This value is encrypted by the RSA key in this entry.
792	HDR_RSA_TAG	Character	An element of HDR_RSA consisting of a hexadecimal value used for validation.
812			End of Header record or begin of another HDR_RSA element.

Tape Format Used by the IJBEFVSE Utility

If you use the IDCAMS utility to backup a dataset *to tape*, the backup creates:
- two or more tape files,
- tape marks.

However, a *dataset stored on a tape* consists of:
- *one* tape file, followed by
- *one* tape mark.

In both of the above cases, a tape label (TLBL) might precede the data on tape.

The IJBEFVSE utility *always* creates encrypted datasets that consist of:
- a header record,
- the encrypted data.

These encrypted datasets are BAM (Basic Access Method) datasets that can be stored on either on disk *or* tape.

The clear data to be encrypted can be either:
- a dataset (VSE library member, a VSAM file, a SAM file on DASD or tape), or
- a complete tape that was created using any proprietary backup process.

Before the encryption is performed, the IJBEFVSE utility *internally* converts tape data into AWSTAPE format. This means that an encrypted tape can be:
- decrypted into a dataset.
- further processed as a standard AWSTAPE on all platforms.
- decrypted into the original tape format on a physical tape or Virtual Tape.

Situations Where an Encrypted Dataset Does Not Fit on a Tape

In general, an encrypted dataset can reside on disk or tape. If using *tapes*, in some cases an encrypted dataset (for example, an IDCAMS backup) will not fit on one single tape. Furthermore, an encrypted dataset is *always* larger than the original clear dataset because of the space required for the header structure and some internal control information.

Although tape devices usually compress data internally, this data-compression will not work as well for encrypted data as it does for clear data.

If the non-encrypted data *does not fit on one tape*, you will probably have created multiple volumes of non-encrypted data. For example, the IDCAMS backup process might have requested a second tape to be mounted because the first tape is full. As a result, you now have two or more non-encrypted backup tapes:
- The IJBEFVSE utility will now encrypt each clear tape in a *separate* step.
- Because each clear tape is full and because of the increase in size, you will probably need *two* encrypted tapes per clear tape.

If the non-encrypted data *does* fit on one tape but the encrypted data exceeds one tape *because of its overhead*, the encrypted dataset is probably a SAM file:
- SAM will ask for a second tape when IJBEFVSE utility reaches the end-of-tape.
- This is normal behavior when writing multi-volume datasets.

Using Virtual Tapes as Intermediate Storage

You can use virtual tapes to store the *intermediate clear data*, before the final encrypted data is written to physical tapes.

However, when using virtual tapes in VSAM the maximum tape-image size is limited to **4 GB** (because of the size limit of a VSAM ESDS cluster).

Alternatively, you can use a VTAPE on a *remote* platform, such as *Linux on System z*. In this case, the remote AWSTAPE file:
- has *no size limitation*,
- can be easily deleted after creating the encrypted backup.

Messages Generated by the IJBEFVSE Utility

The IJBEFVSE utility writes to *SYSLST*:

- Informational messages
- Warning messages
- Error messages
- Statistics

During the encryption of a clear tape and after reading past the END_OF_VOLUME of the input tape, this message is sent to the *console*:

```
F8 0014 0P36I P  NO REC FND SYS005=480
       CCSW=0200700CA80200FFFF CCB=700C80
       SNS= 08402031 00002420 00000000 00000000 00000088 04020202
            020200F0 000000FF
```

- This occurs because the IJBEFVSE utility has no other method of determining the END_OF_VOLUME.
- This situation is internally handled and you can ignore it.
- It is not relevant whether:
  - the tape has been used previously, or
  - data from a previous write-process is still stored on the tape after the current data.

Examples of Using the IJBEFVSE Utility

This topic provides you with practical examples of how to perform the encryption of tapes, disks, files, and volumes using the *IJBEFVSE utility*.

For examples of how to perform the encryption of tapes, disks, files, and volumes using the *IJBEFPGP utility*, see "Examples of Using the IJBEFPGP Utility" on page 593.

Example: Encrypt a VSE Library Member into a VSAM File

This example uses *passphrase-based encryption*. The job below *encrypts* the specified VSE library member and creates an encrypted VSAM dataset. You can use ESDS, RRDS, and KSDS clusters for clear datasets. Only ESDS clusters can be used for encrypted datasets. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=ENCMEM,CLASS=0,DISP=D
// JOB ENCMEM
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
ENCRYPT
DESC='ENCRYPTION TEST'
CLRTDES
PASSWORD=MYPASSWD
ICOUNT=233
CLRFILE=DD:PRD2.CONFIG(PICTURE.JPG)
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

The job below *decrypts* the encrypted VSAM dataset. Note that algorithm and iteration count need not to be specified, because they are contained in the header of the encrypted dataset. Also DESC is applicable only for encryption. The specified descriptive text is contained in the header of the encrypted dataset.

```
* $$ JOB JNM=DECMEM,CLASS=0,DISP=D
// JOB DECMEM
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
DECRYPT
PASSWORD=MYPASSWD
CLRFILE=DD:PRD2.CONFIG(PICTURE.JPG)
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

Example: Create an Encrypted VSAM File

This example uses *passphrase-based encryption*. The job below *encrypts* a VSAM file. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=ENCVSAM,CLASS=0,DISP=D
// JOB ENCVSAM
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
ENCRYPT
DESC='ENCRYPTION TEST'
CLRAES128
PASSWORD=MYPASSWD
ICOUNT=233
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

The job below *decrypts* the encrypted VSAM file:

```
* $$ JOB JNM=DECVSAM,CLASS=4,DISP=D
// JOB DECVSAM
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
DECRYPT
PASSWORD=MYPASSWD
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

Example: Encrypt a VSE Library Member and Store on Virtual Tape

This example uses *passphrase-based encryption*. The job below *encrypts* a VSE library member, and then stores the encrypted file on a virtual tape. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=ENCRYPT,CLASS=0,DISP=D
// JOB ENCRYPT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
MTC REW,480
// ASSGN SYS006,480
// TLBL OUTFILE,'ENCRYPTED.DATA'
// EXEC IJBEFVSE,PARM='DEBUG'
ENCRYPT
DESC='ENCRYPTION TEST'
CLRTDES
PASSWORD=MYPASSWD
ICOUNT=100
CLRFILE=DD:PRD2.CONFIG(TEST01.TXT)
```

```
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

The job below uses a VTAPE which is located on a remote workstation. This requires the Virtual Tape Server running on the workstation. The Virtual Tape Server is a Java application that can be downloaded from the *z/VSE Homepage* (whose URL is given in "Where to Find More Information" on page xxiii).

```
* $$ JOB JNM=ENCRMV,DISP=D,CLASS=0,USER=FOERY
// JOB ENCRMV
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=9.152.216.57,FILE='ENCRMV'
MTC REW,480
// ASSGN SYS006,480
// TLBL OUTFILE,'ENCRMV.DATA'
// EXEC IJBEFVSE
ENCRYPT
DESC='ENCRMV TEST'
CLRTDES
PASSWORD=MYPASSWD
ICOUNT=123
CLRFILE=DD:PRD2.CONFIG(MYMEMB.Z)
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

The job below reads the remote VTAPE and writes the *decrypted* data back to the original VSE library member.

```
* $$ JOB JNM=DECRMV,DISP=D,CLASS=0,USER=FOERY
// JOB DECRMV
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=9.152.216.57,FILE='ENCRMV',READ
MTC REW,480
// ASSGN SYS006,480
// TLBL INFILE,'ENCRMV.DATA'
// EXEC IJBEFVSE
DECRYPT
PASSWORD=MYPASSWD
CLRFILE=DD:PRD2.CONFIG(MYMEMB.Z)
ENCFILE=DD:SYS006-INFILE
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

Example: Create an Encrypted IDCAMS Backup on Tape

This example uses *passphrase-based encryption*. For explanations of the control statements used here, see Table 23 on page 544. The creation of an encrypted backup from an IDCAMS backup tape requires three steps:

1. Write the IDCAMS backup to a physical tape or Virtual Tape.
2. Run the IJBEFVSE utility against the backup tape, and create an *encrypted* tape or dataset.
3. Erase the clear IDCAMS backup tape, delete the related VSAM file, or overwrite it with zeros or random data.

The following JCL shows steps 1 and 2.

```
* $$ JOB JNM=BKUPVTAP,CLASS=0,DISP=D
* $$ LST DISP=D,CLASS=A,RBS=100
* ********************************************************************
* STEP 1: CREATE IDCAMS BACKUP TO VTAPE
* ********************************************************************
// JOB BKUPVTAP BACKUP VSAM TO VTAPE
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
MTC REW,480
// ASSGN SYS005,480
// TLBL INFILE,'BACKUP.FILE'
// DLBL IJSYSCT,'VSAM.MASTER.CATALOG',,VSAM
// EXEC IDCAMS,SIZE=AUTO
     BACKUP (CALC.KSDS CALC.ESDS -
             PAYROLL.CONTROL.FILE1/MPWD1 -
             PAYROLL.FILE.BRANCH01/MPWD2 -
     ) -
    BUFFERS(4) -
    BLOCKSIZE(32758) -
    STDLABEL(INFILE) –
    REWIND
/*
* ********************************************************************
* STEP 2: CREATE ENCRYPTED BACKUP TAPE
* ********************************************************************
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=481,LOC=VSAM,FILE='VTAPE2'
MTC REW,480
// ASSGN SYS006,481
// TLBL OUTFILE,'ENCRYPTED.BACKUP'
// EXEC IJBEFVSE,PARM='DEBUG'
ENCRYPT
DESC='ENCRYPTION TEST'
CLRTDES
PASSWORD=MYPASSWD
ICOUNT=100
CLRTAPE=SYS005
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS005,UA
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
VTAPE STOP,UNIT=481
/&
* $$ EOJ
```

Note: if you are using a VSAM Virtual Tape, as shown in this example, its size is limited to **4 GB**. This is due to the limitation of the underlying VSAM ESDS file.

Example: Restore/Decrypt an Encrypted IDCAMS Backup from Tape

This example uses *passphrase-based encryption*. The *restore/decrypt* of an encrypted backup from tape (for example IDCAMS, but also any other proprietary backup) results in the creation of the original tape. This can be used with the related proprietary restore function. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=RESTVSAM,CLASS=4,DISP=D
// JOB RESTVSAM RESTORE ENCRYPTED BACKUP FROM VTAPE
* *******************************************************************
* STEP 1: DECRYPT BACKUP VTAPE
* *******************************************************************
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
VTAPE START,UNIT=481,LOC=VSAM,FILE='VTAPE2'
// ASSGN SYS005,480
// ASSGN SYS006,481
// TLBL OUTFILE,'ENCRYPTED.BACKUP'
// EXEC IJBEFVSE
DECRYPT
PASSWORD=MYPASSWD
CLRTAPE=SYS005
ENCFILE=DD:SYS006-OUTFILE
/*
* *******************************************************************
* STEP 2: RESTORE CLEAR IDCAMS BACKUP FROM VTAPE
* *******************************************************************
// TLBL INFILE,'BACKUP.FILE'
// DLBL IJSYSCT,'VSAM.MASTER.CATALOG',,VSAM
MTC REW,480
// EXEC IDCAMS,SIZE=AUTO
    RESTORE OBJECTS((AIX.CALC.KSDS) -
                    (PAYROLL.CONTROL.FILE1/MPWD1)) -
    ) -
    BUFFERS(4) -
    STDLABEL(INFILE)
/*
// ASSGN SYS005,UA
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
VTAPE STOP,UNIT=481
/&
* $$ EOJ
```

Example: Restore/Decrypt an Encrypted IDCAMS Backup to a Dataset

This example uses *passphrase-based encryption*. If an encrypted backup tape is *restored/decrypted* to a clear dataset, the resulting clear dataset will be in AWSTAPE format. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=DECRYPT,CLASS=4,DISP=D
// JOB DECRYPT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
MTC REW,480
// ASSGN SYS006,480
// TLBL OUTFILE,'ENCRYPTED.DATA'
// EXEC IJBEFVSE,PARM='DEBUG'
DECRYPT
PASSWORD=MYPASSWD
CLRFILE=DD:CLRDATA
ENCFILE=DD:SYS006-OUTFILE
```

```
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

Example: Encrypt a Library Member Using Public-Key Encryption

This example uses *public-key encryption*. In this job, the RSA parameter points to a VSE library member CEX2TEST.PRVK which contains the RSA key pair used to wrap the data key. The data key is randomly created, and then used to encrypt the data. For further information about creating RSA key pairs, see "Configuring for Server Authentication Using Self-Signed Certificates" on page 489. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=ENCMEMB,CLASS=4,DISP=D
// JOB ENCMEMB ENCRYPT LIB MEMBER
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
MTC REW,480
// ASSGN SYS006,480
// TLBL OUTFILE,'ENCRYPTED.DATA'
// EXEC IJBEFVSE,PARM='DEBUG'
ENCRYPT
DESC='ENCRYPTION TEST'
CLRTDES
RSA=CRYPTO.KEYRING(CEX2TEST)
CLRFILE=DD:PRIMARY.JSCH(IESMODEL.Z)
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

Example: Decrypt a Tape That was Encrypted Using Public-Key Encryption

This example uses *public-key encryption*. If you are *decrypting* a tape or dataset on the same z/VSE system, you do not need to specify the location of the RSA key again, since this information is stored in the header of the encrypted dataset. However, the matching public RSA key *must* be available on the z/VSE system in the specified PRVK member. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=DECRSA,CLASS=4,DISP=D
// JOB DECRSA
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
MTC REW,480
// ASSGN SYS006,480
// TLBL OUTFILE,'ENCRYPTED.DATA'
// EXEC IJBEFVSE,PARM='DEBUG'
DECRYPT
RSA=CRYPTO.KEYRING(CEX2TEST)
CLRFILE=DD:PRIMARY.JSCH(IESMODEL.CLEAR)
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

Example: Use Multiple RSA Control Statements for Multiple Remote Systems

This example uses *public-key encryption*. The following job uses multiple RSA control statements to *encrypt* a VSE library member. Each RSA label references either a CERT or a PRVK member. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=ENCMULT,CLASS=4,DISP=D
// JOB ENCMULT ENCRYPT WITH MULTIPLE RSA STMTS
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
ENCRYPT
DESC='ENCRYPTION TEST'
CLRAES128
RSA=CRYPTO.KEYRING(PUBKEY1)
RSA=CRYPTO.KEYRING(PUBKEY2)
RSA=CRYPTO.KEYRING(PUBKEY3)
RSA=CRYPTO.KEYRING(PRIVKEY1)
CLRFILE=DD:PRD2.CONFIG(IPINIT00.L)
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

In this example, the data key is encrypted with *three* RSA public keys and *one* RSA private key. This allows decrypting the output dataset on *three* remote systems where the corresponding private RSA keys are present. The remote systems can be z/OS or z/VSE systems, but also any Java workstations. In addition to this, the encrypted dataset can be decrypted on the same system using the public key that belongs to the private key specified in the last RSA control statement.

Each remote system will then use its private key to *decrypt* the data that might be transferred via an unsecure network.

```
* $$ JOB JNM=DECRSA,CLASS=4,DISP=D
// JOB DECRSA DECRYPT USING A PRVK
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
DECRYPT
RSA=CRYPTO.KEYRING(PRVKEY2)
CLRFILE=DD:PRIMARY.JSCH(IPINIT00.DECRYPTD)
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

In this example, remote system number 2 uses its corresponding private key to public key PUBKEY2 to decrypt the data. The public keys had to be transferred to the encrypting site prior to the encryption process. For further information about transferring public keys, see:

- “Export a Public Key for Use with the z/OS Java Client” on page 541
- “Import a Public Key into z/VSE from z/OS or a Java Platform” on page 543
- “Export a Public Key for Use on z/OS or a Java Platform” on page 543

Example: Encrypt a VSE/POWER POFFLOAD Tape

This example uses *passphrase-based encryption*. Although the POFFLOAD command is an attention routine (AR) command, we could use DTRIATTN to build a VSE/POWER job for an encrypted backup of POWER queues. However, the second job step which *encrypts* the POFFLOAD tape would have to wait until the OFFLOAD has completed. Inserting a PAUSE into the job or writing a REXX procedure could meet this requirement. However, this example simply performs two manual steps. For explanations of the control statements used here, see Table 23 on page 544.

Step 1: Create a POFFLOAD tape

```
r rdr,pausebg
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R88I  OK : 1 ENTRY PROCESSED BY R RDR,PAUSEBG
BG 0001 1Q47I   BG PAUSEBG 05218 FROM (SYSA) , TIME=17:53:40
BG 0000 // JOB PAUSEBG
        DATE 07/12/2006, CLOCK 17/53/40
BG-0000 // PAUSE
0 VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
BG 0000 1YM3I  TAPE DATA HANDLER INITIALIZATION IN PROGRESS
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R88I  OK : 1 ENTRY PROCESSED BY PRELEASE RDR,TAPESRVR
R1 0001 1Q47I   R1 TAPESRVR 05219 FROM (SYSA) , TIME=17:53:57, LOG=NO
BG 0000 1YM4I  TAPE DATA HANDLER INITIALIZATION COMPLETED
BG-0000 1YM6I  TAPE DATA HANDLER ACCESSED SPECIFIED FILE SUCCESSFULLY
o backup,lst,480
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1QB9A  480, HEADER: HDR1BACKUP.FILE      006193006193 ,L-OFF (REPLY:
        PGO 480...)
pgo 480,ignore
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1Q2AI  OFFLOADING BACKUP SUCCESSFULLY COMPLETED ON 480, JOURNAL LST
        ENTRY $OFJ5220 CREATED (07/12/06 17:57:01)
```

Step 2: Encrypt the POFFLOAD tape

```
* $$ JOB JNM=ENCOFFL,CLASS=0,DISP=D
// JOB ENCOFFL ENCRYPT POFFLOAD TAPE
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
VTAPE START,UNIT=481,LOC=VSAM,FILE='VTAPE2'
// ASSGN SYS005,480
// ASSGN SYS006,481
// TLBL OUTFILE,'ENCRYPTED.DATA'
// EXEC IJBEFVSE
ENCRYPT
DESC='ENCRYPTED POFFLOAD TAPE'
CLRTDES
PASSWORD=MYPASSWD
ICOUNT=100
CLRTAPE=SYS005
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS005,UA
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
VTAPE STOP,UNIT=481
/&
* $$ EOJ
```

For Step 3, you would erase the clear tape.

Example: Restore/Decrypt an Encrypted POFFLOAD Tape

This example uses *passphrase-based encryption*. As in "Example: Encrypt a VSE/POWER POFFLOAD Tape" on page 561, the *restore/decrypt* of an encrypted POFFLOAD tape consists of two manual steps: firstly restoring the clear POFFLOAD tape, and secondly issuing a POFFLOAD LOAD command. The job below *decrypts* the encrypted POFFLOAD tape. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=DECOFFL,CLASS=0,DISP=D
// JOB DECOFFL DECRYPT POFFLOAD TAPE
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
VTAPE START,UNIT=481,LOC=VSAM,FILE='VTAPE2'
// ASSGN SYS005,480
// ASSGN SYS006,481
// TLBL OUTFILE,'ENCRYPTED.DATA'
// EXEC IJBEFVSE
DECRYPT
PASSWORD=MYPASSWD
CLRTAPE=SYS005
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS005,UA
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
VTAPE STOP,UNIT=481
/&
* $$ EOJ
```

After running this job, tape unit 480 contains the clear offload data and can be restored.

Example: Encrypt a LIBR Backup Tape

This example uses *passphrase-based encryption*. This job shows how to create an *encrypted* LIBR backup. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=BKUPLIBR,CLASS=4,DISP=D
// JOB BKUPLIBR LIBR BACKUP TO VTAPE
* *******************************************************************
* STEP 1: CREATE LIBR BACKUP TO VTAPE
* *******************************************************************
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
// ASSGN SYS005,480
// TLBL INFILE,'BACKUP.FILE'
// EXEC LIBR,SIZE=256K
    BACKUP LIB=PRD2  -
    TAPE=SYS005 -
    TAPELABEL=INFILE -
    LIST=YES
/*
* *******************************************************************
* STEP 2: CREATE ENCRYPTED BACKUP TAPE
* *******************************************************************
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=481,LOC=VSAM,FILE='VTAPE2'
MTC REW,480
// ASSGN SYS006,481
// TLBL OUTFILE,'ENCRYPTED.BACKUP'
// EXEC IJBEFVSE
ENCRYPT
DESC='ENCRYPTION TEST'
CLRTDES
PASSWORD=MYPASSWD
```

```
ICOUNT=100
CLRTAPE=SYS005
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS005,UA
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
VTAPE STOP,UNIT=481
/&
* $$ EOJ
```

Example: Restore/Decrypt an Encrypted LIBR Backup

This example uses *passphrase-based encryption*. This job *restores/decrypts* the VSE library that was backed up in "Example: Encrypt a LIBR Backup Tape" on page 562. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=RESTLIBR,CLASS=8,DISP=D
// JOB RESTLIBR RESTORE LIBR BACKUP FROM VTAPE
* ********************************************************************
* STEP 1: DECRYPT ENCRYPTED BACKUP TAPE
* ********************************************************************
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
VTAPE START,UNIT=481,LOC=VSAM,FILE='VTAPE2'
// ASSGN SYS005,480
// ASSGN SYS006,481
// TLBL OUTFILE,'ENCRYPTED.BACKUP'
// EXEC IJBEFVSE
DECRYPT
PASSWORD=MYPASSWD
CLRTAPE=SYS005
ENCFILE=DD:SYS006-OUTFILE
/*
* ********************************************************************
* STEP 2: RESTORE CLEAR BACKUP FROM TAPE
* ********************************************************************
// TLBL INFILE,'BACKUP.FILE'
MTC REW,480
// EXEC LIBR,SIZE=256K
    RESTORE LIB=PRD2  -
    TAPE=SYS005 -
    REPLACE=NO -
    TAPELABEL=INFILE -
    LIST=YES
/*
// ASSGN SYS005,UA
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
VTAPE STOP,UNIT=481
/&
* $$ EOJ
```

Example: Write an Encrypted SAM Dataset to VTAPE

This example uses *passphrase-based encryption*. The job below *encrypts* a SAM file on disk and writes the encrypted dataset to VTAPE. Note that the RECFM and LRECL parameters must be specified and must match the characteristics of the related SAM file. When encrypting VSE library members or VSAM files, the IJBEFVSE utility can get this information from the z/VSE library or VSAM catalog. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=ENCSAM,CLASS=0,DISP=D
// JOB ENCSAM
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
MTC REW,480
```

```
// ASSGN SYS006,480
// TLBL ENCDATA,'ENCRYPTED.DATA'
// DLBL CLRDATA,'SOME.CLEAR.SAM.DATA',99/366,SD
// EXTENT SYS005,DOSRES,,,19125,10000
// ASSGN SYS005,ANYDISK,TEMP,VOL=DOSRES,SHR
// EXEC IJBEFVSE,PARM='DEBUG'
ENCRYPT
RECFM=F
LRECL=80
DESC='ENCRYPTION TEST'
CLRAES128
PASSWORD=MYPASSWD
ICOUNT=233
CLRFILE=DD:CLRDATA
ENCFILE=DD:SYS006-ENCDATA
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

Example: Restore/Decrypt an Encrypted SAM Dataset From VTAPE

This example uses *passphrase-based encryption*. The job below *restores/decrypts* the encrypted dataset from VTAPE. Note that the LRECL and RECFM parameters must *not* be specified since they are contained in the header of the encrypted dataset. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=DECSAM,CLASS=0,DISP=D
// JOB DECSAM
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE1'
MTC REW,480
// ASSGN SYS006,480
// TLBL ENCDATA,'ENCRYPTED.DATA'
// DLBL CLRDATA,'SOME.CLEAR.SAM.DATA',99/366,SD
// EXTENT SYS005,DOSRES,,,19125,10000
// ASSGN SYS005,ANYDISK,TEMP,VOL=DOSRES,SHR
// EXEC IJBEFVSE
DECRYPT
PASSWORD=MYPASSWD
CLRFILE=DD:CLRDATA
ENCFILE=DD:SYS006-ENCDATA
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

Example: Write an Encrypted SAM Dataset to Disk

This example uses *passphrase-based encryption*. The job below *encrypts* a SAM dataset on disk and writes the encrypted data into another SAM dataset on disk. For explanations of the control statements used here, see Table 23 on page 544.

```
* $$ JOB JNM=ENCSAMD,CLASS=0,DISP=D
// JOB ENCSAMD ENCRYPT SAM DATASET ON DISK
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// DLBL CLRDATA,'SOME.CLEAR.SAM.DATA',99/366,SD
// EXTENT SYS005,DOSRES,,,19125,10000
// ASSGN SYS005,ANYDISK,TEMP,VOL=DOSRES,SHR
// DLBL ENCDATA,'SOME.ENCRYPTED.SAM.DATA',99/366,SD
// EXTENT SYS006,DOSRES,,,29125,10000
// ASSGN SYS006,ANYDISK,TEMP,VOL=DOSRES,SH
```

```
// EXEC IJBEFVSE
ENCRYPT
RECFM=F
LRECL=80
DESC='ENCRYPTION TEST'
CLRAES128
PASSWORD=MYPASSWD
ICOUNT=2345
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

Example: Encrypt a Tape or VTAPE Using the DynamT Utility

This example uses *public-key encryption*. The job below *encrypts* a physical tape and outputs the result to a virtual tape. The tape management is performed using the *DynamT* tape management product.

```
* $$ JOB JNM=ENCRYPT,CLASS=R,DISP=D
// JOB ENCRYPT - ENCRYPT FCOPY DUMP VOLUME TO TAPE
// TLBL CLRDATA,'RAID23,U,P'
// TLBL ENCDATA,'EFVSE003,U'
// EXEC TDYNASN
OPEN CLRDATA,SYS005,INPUT
/*
// EXEC IJBEFVSE
ENCRYPT
DESC='ENCRYPTED DATA'
RSA=CRYPTO.KEYRING(KEY01)
COMPRESSION=YES
RECFM=F
LRECL=32758
CLRAES128
ICOUNT=1234
CLRTAPE=SYS005
ENCFILE=DD:ENCDATA
/*
// EXEC TDYNASN
VTSTOP SYS005
/*
/&
* $$ EOJ
```

Example: Decrypt a Tape or VTAPE Using the DynamT Utility

This example uses *public-key encryption*. The job below *decrypts* a virtual tape and outputs the result to a physical tape. The tape management is performed using the *DynamT* tape management product.

```
* $$ JOB JNM=DECRYPT,CLASS=R,DISP=D
// JOB DECRYPT - DECRYPT FCOPY DUMP VOLUME FROM ENCRYPTED TAPE
// TLBL CLRDATA,'RAID23,U,P'
// TLBL ENCDATA,'EFVSE003,U'
// EXEC TDYNASN
OPEN CLRDATA,SYS006,OUTPUT
/*
// EXEC IJBEFVSE
DECRYPT
RSA=CRYPTO.KEYRING(KEY01)
RECFM=F
LRECL=32758
CLRTAPE=SYS006
ENCFILE=DD:ENCDATA
/*
// EXEC TDYNASN
```

```
CLOSE CLRDATA,SYS006,OUTPUT
VTSTOP SYS006
/*
/&
* $$ EOJ
```

Example: Encrypt a Binary File Using the z/OS Java Client

This example uses *passphrase-based encryption*. You can use a script similar to that below in order to *encrypt* a *workstation file*. Here it is assumed that a Java runtime environment or SDK is installed on the workstation. The following example *encrypts* a *JPEG image*. For explanations of the control statements used here, refer to the *documentation supplied with the z/OS Java Client*.

```
java com.ibm.encryptionfacility.EncryptionFacility
-mode encrypt
-underlyingKey PBEWithSHA1And3DES
-password mypasswd
-inputFile mypic.jpg
-outputFile mypic.jpg.enc
-iterations 233
```

Next, the encrypted file mypic.jpg.enc could be uploaded to z/VSE (for example, into a VSAM file with a LRECL of at least 32760 bytes) because this value is internally used by the Java Client. Using a VSAM file with a shorter max record length would result in truncated records and thus in an unusable file.

When uploading the encrypted file to z/VSE via FTP, you must specify these FTP parameters:

```
bin
quote site recfm v
quote site lrecl 32760
```

Decrypting the file on z/VSE returns an exact copy of the original file, which can be downloaded to the same or another workstation for further processing.

```
* $$ JOB JNM=DECRJPG,CLASS=4,DISP=D
// JOB DECRJPG
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
DECRYPT
PASSWORD=MYPASSWD
CLRFILE=DD:PRIMARY.JSCH(MYPIC.JPG)
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

You could now use the *VSE Navigator* to view the decrypted image in the z/VSE library. You can download the VSE Navigator from the *z/VSE Homepage* (whose URL is given in “Where to Find More Information” on page xxiii).

Example: Use z/OS Java Client with Public-Key Encryption

This example uses *public-key encryption*. You can use this example providing you have created a *JKS keystore* (as described in "Export a Public Key for Use with the z/OS Java Client" on page 541).

The example below shows how to *encrypt* a binary file **mypic.jpg** using the public RSA key from a previously-created JKS keystore.

Note: The `–keyStoreCertificateAlias` parameter must be set to "**0**", because Keyman/VSE enumerates the keystore items beginning with zero. You must also specify the keystore password that was entered when the JKS file was saved in Keyman/VSE. For explanations of the control statements used here, refer to the documentation *supplied with the z/OS Java Client*.

```
java com.ibm.encryptionfacility.EncryptionFacility
-mode encrypt
-underlyingKey AES16
-keyStoreName Keyring.jks
-keyStoreType JKS
-keyStoreCertificateAlias 0
-password mypasswd
-inputFile mypic.jpg
-outputFile mypic.jpg.enc
```

Using these FTP commands, the encrypted file is uploaded into a VSE/VSAM file on z/VSE:

```
bin
Quote site lrecl 32760
Quote site recfm v
```

Finally, the IJBEFVSE utility is invoked in order to *decrypt* the VSAM file into a clear VSAM file with the same maximum record length.

The job below references the PRVK member that you created when uploading the RSA key pair to z/VSE. For details, see "Define Properties of Host and Generate/Upload a Key Pair to the Host" on page 540.

```
* $$ JOB JNM=DECRJPG,CLASS=4,DISP=D
// JOB DECRJPG
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
DECRYPT
RSA=CRYPTO.KEYRING(EFVSE01)
CLRFILE=DD:PRIMARY.JSCH(MYPIC.JPG)
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

Known Problems When Encrypting and Exchanging Data

Looping When Using CA DynamT to Open a Clear Tape or Virtual Tape

Symptom:

- When using CA *DynamT* to open a clear tape or virtual tape, the EF for z/VSE loops endlessly. The TAPESRVR job starts (to load the Virtual Tape Data Handler), but there is no apparent I/O against the VTAPE file after the file-open sequence has completed.
- You use JCL similar to this:

```
// EXEC TDYNASN
OPEN CLRDATA,SYS005,INPUT
/*
```

Possible Reason / Response:

- Add the 'P' option to the TLBL for CLRDATA, which should solve the problem:

```
// TLBL CLRDATA,'RAID23,U,P'
```

- Dynam/T then opens the VTAPE successfully.
- The cause of the problem was:
  - Dynam/T performed a dynamic *logical unit block* (LUB) allocation and did not assign the VTAPE to SYS005.
  - As shown in the JCL example above, the EF for z/VSE expected CLRDATA to be assigned to SYS005

Chapter 48. Implementing the Encryption Facility for z/VSE OpenPGP

This chapter describes how you can use the *Encryption Facility for z/VSE OpenPGP* to implement the *software-based encryption* of:
- SAM files,
- VSAM files,
- VSE Library members,
- tapes and virtual tapes
- complete backups that were made using any z/VSE backup utility (such as IDCAMS, LIBR, POFFLOAD) or a vendor product.

Note: Throughout this chapter, the term:
- *EF for z/VSE OpenPGP* is used as the short-form for the *Encryption Facility for z/VSE OpenPGP*
- *EF for z/VSE* is used as the short-form for the *Encryption Facility for z/VSE*.
- *EF for z/OS V1.1* is used as the short-form for the *Encryption Facility for z/OS Version 1 Release 1*.

This chapter contains these main topics:

Related Topics:

For details of ...	Refer to ...
OpenPGP	• `http://en.wikipedia.org/wiki/Openpgp` • `http://tools.ietf.org/html/rfc4880` (RFC4880)
GnuPG	• `http://www.gnupg.org/` • `http://www.gnupg.org/docs.html` (documentation) • `http://www.spywarewarrior.com/uiuc/gpg/gpg-com-0.htm` (commands)
the issues that affect overall performance of an encryption or decryption process	"Performance considerations For Using the IJBEFVSE Utility" on page 537
how to compress encrypted data	"Deciding Whether or Not to Use Data Compression" on page 548

For details of ...	Refer to ...
how to pass the *file attributes* of clear data to the IJBEFPGP utility	"Specifying File Attributes and Record Formats" on page 549
tape formats used by the IJBEFPGP utility	"Tape Format Used by the IJBEFVSE Utility" on page 552
how to handle situations where an encrypted dataset (for example, an IDCAMS backup) will not fit on one single tape	"Situations Where an Encrypted Dataset Does Not Fit on a Tape" on page 553
how to encrypt and decrypt record-based data, and exchange this data with other operating systems	"Encrypting and Exchanging Record-Based Data" on page 550
known problems when encrypting and decrypting data, and exchanging this data with other operating systems	"Known Problems When Encrypting and Exchanging Data" on page 567
how to use virtual tapes to store the *intermediate clear data*, before the final encrypted data is written to physical tapes	"Using Virtual Tapes as Intermediate Storage" on page 553
messages that might be generated by the IJBEFPGP utility	"Messages Generated by the IJBEFVSE Utility" on page 554
how to display your current hardware crypto environment	"Using Crypto Support with a z/VSE Guest under z/VM" on page 468 and "Displaying Hardware Crypto Status Information Under z/VSE" on page 469.
how to activate crypto support if you are using an External Security Manager (ESM)	"Using Crypto Support and an External Security Manager" on page 476.
how to encrypt *tapes* using an *encryption-capable tape device*	Chapter 46, "Implementing Hardware-Based Tape Encryption," on page 521.
how to install and use the Encryption Facility for z/OS	*IBM Encryption Facility for z/OS: User's Guide*, SA23-1349

Overview of PGP and the EF for z/VSE OpenPGP

OpenPGP is a standard protocol for ensuring the integrity of data that can be exchanged between trusted partners. It is designed to help provide data integrity through:

- Data encryption using either a randomly-generated symmetric session key or a password.
- OpenPGP certificates for the exchange of key information that can provide the data integrity service.

For compatibility with workstation-based PGP tools, IBM has tested the EF for z/VSE OpenPGP together with the *OpenPGP implementation GnuPG*.

OpenPGP support is designed to offer even more choice and flexibility for exchanging data with business partners. For example, it does not require that your business partners purchase new storage hardware, have a mainframe, or run z/VSE. PGP ("Pretty Good Privacy") uses many different encryption and hash algorithms, so that you can take full advantage of System z hardware features.

The EF for z/VSE OpenPGP offers the same functionality as the EF for z/VSE (described on "Overview of the EF for z/VSE" on page 532), but also offers support for OpenPGP. It therefore provides the:

- IJBEFVSE utility, which is the same as the utility used by the EF for z/VSE.
- IJBEFPGP utility for *OpenPGP* encryption processing.

Note:

1. The IJBEFVSE utility uses the *System z* data format.
2. The IJBEFPGP utility supports the *OpenPGP* data format.
3. The support for OpenPGP under z/VSE *is compatible* with the support for OpenPGP *under z/OS*.
4. The functions and services supported by the OpenPGP format are not compatible with the functions and services of the *System z format*.

As is the case for the IJBEFVSE utility, when using the IJBEFPGP utility you can choose between *two* methods for encrypting data:

- *Passphrase-based encryption*, which is described in Figure 132 on page 533.
- *Public-key encryption*, which is described in Figure 133 on page 534.

Differences to the IJBEFVSE utility

The OpenPGP standard permits *all combinations* of symmetric algorithms, key lengths, hash algorithms, and compression algorithms. However, the System z format only supports a *restricted number* of algorithms and key lengths.

Table 25 summarizes the differences between the IJBEFVSE and IJBEFPGP utilities.

Table 25. Differences Between the IJBEFVSE and IJBEFPGP Utilities

	IJBEFVSE	IJBEFPGP
Encrypted data format	• System z format	• OpenPGP format
Compatibility with	• EF for z/OS V.1.1 • EF for z/OS Java client • Decryption Client for z/OS	• OpenPGP implementations, such as GnuPG or EF for z/OS V1.2 OpenPGP
Symmetric Algorithms	• TDES • AES-128	• DES • TDES • AES-128 • AES-192 • AES-256
Hash algorithms	• SHA1	• MD5 • SHA1 • SHA224 • SHA256 • SHA384 • SHA512
Compression	• System z-provided compression	• ZIP-based compression • ZLIB-based compression
RSA key lengths	• 512 • 1024 • 2048	• 512 • 1024 • 2048 • 4096
Public key format	• x.509 certificates	• PGP certificates

Although z/VSE datasets are encrypted according to the OpenPGP standard and are therefore exchangeable between z/VSE and workstation platforms, not all types of z/VSE datasets make sense in a workstation environment. The following section discusses the different file and record formats.

Differences to GnuPG and the EF for z/OS

This section describes the differences between the IJBEFPGP utility and the:
- IJBEFVSE utility
- GnuPG (if used)
- the EF for z/OS (Encryption Facility for z/OS)

The IJBEFPGP utility:
- does *not* maintain keyrings in an OpenPGP environment. Therefore, it does *not* provide any commands or options to import or export *PGP* public keys. In a z/VSE environment, this is done using the *Keyman/VSE* tool.
- allows the specification of *single-key DES* (DES_SK) as encryption algorithm (GnuPG and z/OS do *not* allow this). However, these algorithms (described in RFCs 2440 and 4880) cannot be used with z/VSE:
  - cast5
  - idea
  - blowfish
  - twofish
- always uses *an MDC* for data integrity when encrypting, but accepts encrypted datasets that were encrypted *without* using an MDC.

Prerequisites for Using the IJBEFPGP Utility

The prerequisites for using the IJBEFPGP utility are the same as those for using the IJBEFVSE utility. Therefore, for a list of the prerequisites refer to "Prerequisites for Using the IJBEFVSE (or IJBEFPGP) Utility" on page 535.

Restrictions When Using the IJBEFPGP Utility

- For enhanced data integrity, the OpenPGP standard supports *DSA signatures* and *RSA signatures*. However, the IJBEFPGP utility *does not support signatures*. Furthermore:
  - RSA signatures *will probably* be supported in a future release of z/VSE.
  - DSA signatures are *permanently restricted* for use with z/VSE.
- The OpenPGP standard defines the use of "key-IDs" to identify an RSA public key in a keystore. This key-ID is given by 8 bytes derived from the public key bytes. When encrypting using public-key encryption, the key-ID is calculated and added to the encrypted dataset. The recipient of the encrypted dataset may use the key-ID to identify the corresponding private key in order to decrypt the session key. However:
  - When decrypting, the IJBEFPGP utility *ignores* key-IDs and instead uses a private key that was specified using the `RECIPIENT_ALIAS` parameter.
  - When encrypting, the IJBEFPGP utility uses a *null key-ID*, called the "speculative key-ID". This causes other PGP implementations to search their key database for a matching *private key*. For details, refer to the RFC4880.
- The IJBEFPGP utility supports record-based VSAM data by storing extra information about record lengths and record formats. However, you cannot decrypt an *encrypted* VSAM file using *another PGP tool.* Only the *IJBEFPGP utility* will process the relevant meta information correctly.
- The following *symmetric algorithms* are *not* supported:
  - cast5
  - blowfish
  - twofish
  - idea
- The following *hash algorithm* is *not* supported:

– RIPEMD-160

- The following *public-key algorithm* is *not* supported:
  – DSA

Installing the Prerequisite and Optional Programs

1. (**Required**) You require the Keyman/VSE program in order to:
   - Create RSA key pairs and certificates,
   - Upload RSA key pairs and certificates to z/VSE,
   - Import and export PGP certificates.
2. (**Optional**) To exchange data with *non-System z* platforms, you can optionally use any PGP implementations. "Examples of Using the IJBEFPGP Utility" on page 593 provides examples of how to exchange data with non-System z platforms.
3. (**Optional**) Download and install the *Gnu Privacy Guard* from `http://www.gnupg.org`. The install package is contained in one file *gnupg-1.4.7.tar.gz*. Simply unpack the contents of this file into a new folder.
4. (**Optional**) Download and install the *GPG4Win* Windows GUI which includes the Windows Explorer extension *GPGee*. Download GPG4Win from `http://www.gpg4win.de/`. The install file is a Windows file *gpg4win-1.1.3.exe*. Double-click this file and follow the instructions provided in the install dialogs.

   Note: The documentation that accompanies GPG4Win is currently only available in the German language!

Summary of Commands Available With the IJBEFPGP Utility

You can use the following commands with the IJBEFPGP utility:

Table 26. Commands Available With the IJBEFPGP Utility

Command	Description
PB_ENCRYPT	Password-based encryption.
PK_ENCRYPT	Public-key encryption.
DECRYPT	Decryption
LIST_ALGO	Print available algorithms on SYSLST. Available encryption algorithms are dependent on the machine. An IBM z9 server also provides AES-128, z10 and z196 servers additionally provide AES-192 and AES-256. Available RSA key lengths depend on installed crypto cards.
INFO	Specifies that file decryption is not to be performed, but information about the encrypted input file is to be recovered and written to SYSLST. When the information is written, processing ends. This option is useful when you want to: • determine the original clear-text file format information, or • ensure that a specified RSA key is present in the current keyring library.
HELP	Print command help on SYSLST

Invoking the IJBEFPGP Utility

This topic describes the:
- syntax of the IJBEFPGP utility,
- control statements that can be used with the IJBEFPGP utility,
- rules for specifying file names when using the CLRFILE and ENCFILE control statements,
- rules for specifying record formats.
- jobs must include these statements:

```
// EXEC IJBEFPGP [,PARM='[SYSLST=DD:SYSnnn] [DEBUG]']
control statements
:
/*
```

where:

SYSLST=DD:SYS*nnn*
: Optional. Specifies a logical unit to-which the listing should be written (for example SYSLST=DD:SYS004)

DEBUG
: Optional. Enables debugging (messages will be sent to SYSLST and to the console).

control statements
: Details are given in Table 27. Control statements are passed via SYSIPT.

Table 27 lists the control statements you can use with the IJBEFPGP utility. If you specify the same control statement more than once, only the *last* specification will be used. The availability of some algorithms and key lengths depends upon the System z platform, as described in Table 29 on page 592. All control statements are passed via SYSIPT.

Note: The table uses the abbreviations **PBE** for *passphrase-based encryption*, and **PKE** for *public-key encryption*.

Table 27. Control Statements Used With the IJBEFPGP Utility

Option	Description	Applicable for	Required
CLRFILE=*name*	Dataset containing *clear data*. For encryption, the clear data is the input, for decryption it is the output. Valid file types are: • Librarian members • VSAM ESDS, KSDS and RRDS clusters • SAM dataset on tape or DASD See also "Specifying File Names for CLRFILE and ENCFILE" on page 548.	All	Yes [1]

Table 27. Control Statements Used With the IJBEFPGP Utility (continued)

Option	Description	Applicable for	Required
ENCFILE=*name*	Dataset containing encrypted data. For encryption, this file is the output, for decryption it is the input. Valid file types are: • Librarian members - Note that when specifying a LIBR member, you must use a member type that consist of more than one character. You *cannot* use member types with only one character (A-Z, 0-9, $, #, @) for ENCFILE. • VSAM ESDS clusters • SAM dataset on tape or DASD See also “Specifying File Names for CLRFILE and ENCFILE” on page 548.	All	Yes
CLRTAPE=SYS*nnn*	Specifies the logical unit of the clear data tape. For encryption, the clear data is the input. For decryption, the clear data is the output. The whole content of the specified tape is used, including all tape marks, tape labels and so on.	All	Yes [2]
S2K_PASSPHRASE=*pwd*	8 to 32 char password. For compatibility with PGP, EBCDIC passwords are internally translated to ASCII before generating the encryption key. Refer to parameters ASCII_CODEPAGE and EBCDIC_CODEPAGE.	PBE, Decrypt	Yes for PBE
S2K_CIPHER_NAME=*name*	Name of encryption algorithm: • DES_SK (single-key DES) • TRIPLE_DES (default) • AES_128 • AES_192 • AES_256 Use the LIST_ALGO command to list the available encryption algorithms.	PBE, PKE	No
COMPRESSION=*n*	Set compression level (0 ... 9): 0 : Do not use compression 1 : Use best speed for compression 6 : default 9 : Use best compression	PBE, PKE	No
COMPRESS_NAME=*name*	Name of compression algorithm: • UNCOMPRESSED • ZIP (default) • ZLIB	PBE, PKE	No
DIGEST_NAME=*name*	Name of hash algorithm: • MD5 • SHA_1 (default) • SHA224 • SHA256 • SHA384 • SHA512 Use the LIST_ALGO command to list the available encryption algorithms.	PBE, PKE	No
RECIPIENT_ALIAS=*name*	Member name of a PRVK or CERT member containing an RSA private or public key. Up to 16 RECIPIENT_ALIAS statements can be specified.	PKE, Decrypt	Yes for PKE

Table 27. Control Statements Used With the IJBEFPGP Utility (continued)

Option	Description	Applicable for	Required
CONFIDENTIAL	Indicate in the PGP message that input data is very sensitive. PGP implementations usually do not save this type of data to disk, but instead only display data on the console.	PBE, PKE	No
USE_EMBEDDED_FILENAME	Store received data with the original file name as specified in the PGP message. You can also specify the CLRFILE parameter as a fallback, in case the embedded file name is not usable on z/VSE. The IJBEFPGP utility will first try to use the embedded file name. If not usable, it will take the dataset name as specified by CLRFILE. **Note**: while on workstation platforms a new file with the embedded filename is created during the decryption process, this is often not possible on z/VSE. For example, when using VSAM data, the related cluster must exist before it can be used by the IJBEFPGP utility.	Decrypt	No
USE_RECORDINFO	Information about LRECL, BLKSIZE, and RECFM is stored in the encrypted dataset. In addition to that, the length of each plain input record is maintained in order to be able to restore the original record structure when decrypting later. This information is z/VSE-specific and is silently skipped by other PGP implementations. On z/VSE, this information is used when decrypting a dataset to restore clear data with its original record structure. This parameter should always be used when encrypting record based datasets, such as SAM, or VSAM files.	PBE, PKE	No
LRECL=*nnn*	Specifies the logical record length of the CLRFILE. For VSAM clusters and LIBR members this parameter is not required, since the record length information is read from the VSAM catalog or Library. For SAM datasets on tape or DASD it is required. You must also specify the RECFM control statement when specifying LRECL. For further details about supported records formats and defaults, refer to: • "Record, Block, and Control Interval" in the *z/VSE System Macros User's Guide*, SC33-8407, and • "Record Formats" in the *LE/VSE C Run-Time Programming Guide*, SC33-6688.	PBE, PKE, Decrypt	Yes for SAM datasets

Table 27. Control Statements Used With the IJBEFPGP Utility (continued)

Option	Description	Applicable for	Required
RECFM=*n*	Specifies the record format of the CLRFILE. For VSAM clusters and LIBR members this parameter is not required, since the format information is read from the VSAM catalog or Library. For SAM datasets on tape or DASD it is required. Valid values are: • F – fixed • V – variable • B – blocked • S – spanned • U – undefined • FB – fixed blocked • FBS – fixed blocked spanned • VB – variable blocked • VBS – variable blocked spanned In addition an A (ASA print-control characters) or M (machine print-control characters) can be added to above formats. For further details about supported records formats and defaults, refer to: • "Record, Block, and Control Interval" in the *z/VSE System Macros User's Guide*, SC33-8407, and • "Record Formats" in the *LE/VSE C Run-Time Programming Guide*, SC33-6688.	PBE, PKE, Decrypt	Yes for SAM datasets
BLKSIZE=*nnn*	Specifies the bloc size of the CLRFILE. For VSAM clusters and LIBR members this parameter is not required, since the record length information is read from the VSAM catalog or Library. For SAM datasets on tape or DASD it is required. You must also specify the RECFM control statement when specifying BLKSIZE. For further details about supported records formats and defaults, refer to: • "Record, Block, and Control Interval" in the *z/VSE System Macros User's Guide*, SC33-8407, and • "Record Formats" in the *LE/VSE C Run-Time Programming Guide*, SC33-6688.	PBE, PKE, Decrypt	Yes for SAM datasets
ASCII_CODEPAGE=*nnnn*	Specifies the ASCII code page used for SYSIPT input. This code page is important when specifying the password. Special characters (such as '@') are on different code points in different code pages. Default is IBM-850. The code page statement must appear before the PASSWORD control statement.	PBE	No
EBCDIC_CODEPAGE=*nnnn*	Specifies the EBCDIC code page used for SYSIPT input. This code page is important when specifying the password. Special characters (such as '@') are on different code points in different code pages. Default is IBM-1047. The code page statement must appear before the PASSWORD control statement.	PBE	No

Note:

1. Parameter CLRFILE is not required when USE_EMBEDDED_FILENAME is specified.

2. Either parameter CLRFILE or CLRTAPE is required. For tapes, embedded file names cannot be used.

Setting Up to Use Passphrase-Based Encryption (IJBEFPGP)

Note:

1. If you plan to delete your original unencrypted data after encryption, ***you are strongly recommended*** to:
   - Verify that your data can be decrypted successfully *before* destroying any original data.
   - Keep a copy of the IJBEFPGP utility version you used to encrypt the data, to ensure you can perform the decryption at any time in future.
2. In this description, the term "password" is used instead of "passphrase".
3. For a general overview of passphrase-based encryption (PBE), see Figure 132 on page 533.

The major advantage of passphrase-based encryption is that you do *not* require *keystores*. The encryption key is directly derived from the password. The IJBEFPGP utility converts the EBCDIC password specified in the JCL to ASCII.

Entering a Password (Passphrase): When entering a password (passphrase), you should be aware that:

- Passwords are *case-sensitive*, so make sure you specify your password correctly both in JCL and on any other related platform.
- Do *not* use any multicultural-support specific characters (for example, a German umlaut) in a password. This could cause problems when translating the password to ASCII depending on the used code page.

To manage your passwords, you can use any available tool from vendors, freeware, or shareware. You can find suitable tools by searching the Web for the term "Password Manager".

Specifying a Code page: The IJBEFPGP utility uses the following code pages *by default*:

- ASCII code page : **IBM-850**
- EBCDIC code page : **IBM-1047**

You can change the code page using parameters `ASCII_CODEPAGE` and `EBCDIC_CODEPAGE` (described in Table 27 on page 574). To ensure that your code page is active when translating the password, specify the code-page parameters **before** the `S2K_PASSWORD` parameter.

OpenPGP PBE With the Encryption Done on z/VSE

This example describes how a VSAM dataset (ENCDATA) is encrypted (passphrase-based encryption) *on the z/VSE host*, downloaded to a Windows workstation, and finally decrypted using *GPGee* (which is part of the GPG4Win installation). It uses the example file `encdata.gpg`.

These are the steps we follow:

1. Submit a Job similar to this one.

```
* $$ JOB JNM=EFPGP,CLASS=0,DISP=D
// JOB EFPGP TEST OPENPGP SUPPORT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
PB_ENCRYPT
```

```
S2K_PASSPHRASE=BLAHBLAH
S2K_CIPHER_NAME=AES_128
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

2. Use FTP to download the encrypted dataset (the example uses `encdata.gpg`) from the z/VSE host to a Windows workstation.

```
ftp> get encdata encdata.gpg
200 Command okay
150-About to open data connection
File:EFVSE.ENCDATA
Type:Binary Recfm:FB Lrecl: 80 Blksize: 80
CC=ON UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON NAT=NO CONT=OFF
MODE=Stream STRU=File
150 File status okay; about to open data connection
226-Bytes sent: 83,736
Transfer Seconds: .20 ( 409K per second)
File I/O Seconds: .01 ( 8177K per second)
226 Closing data connection
ftp: 83736 bytes received in 0.81Seconds 103.00Kbytes/sec.
ftp>
```

3. Decrypt the file stored on the Windows workstation using GPGee.

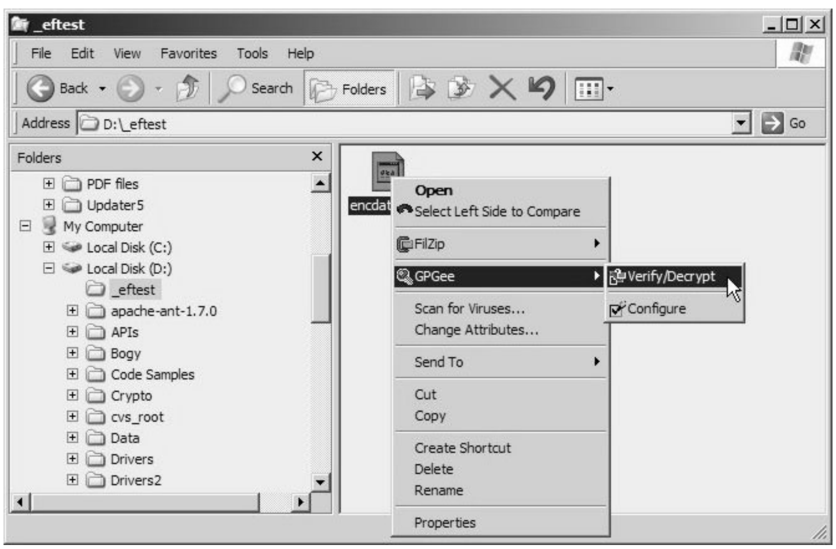


Figure 137. Decrypt a File on Windows Using GPGee

We are now prompted to enter a passphrase.

4. Key the passphrase and click **OK**.

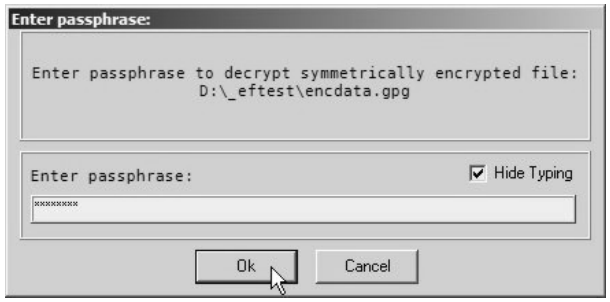


Figure 138. Entering a Decryption Passphrase in GPGee

A confirmation message is then displayed.

OpenPGP PBE With the Decryption Done on z/VSE

This example describes how the file `picture.jpg` is encrypted (passphrase-based encryption) using GPGee on a Windows workstation, uploaded to z/VSE, and finally *decrypted on z/VSE*. There is no special setup necessary for password-based encryption.

These are the steps we follow:

1. Encrypt a file on a Windows workstation.

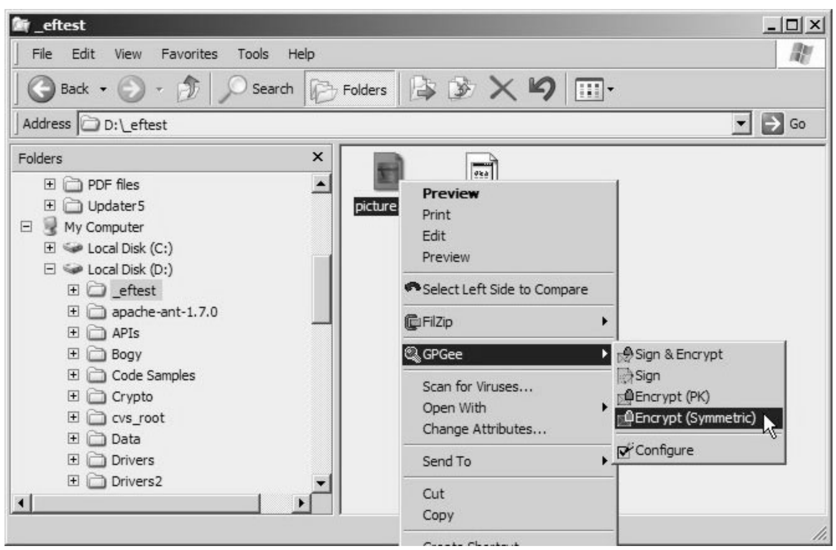


Figure 139. Encrypt a File on Windows Using GPGee

2. Enter a password. Here, uppercase is used because in the later z/VSE JCL the password is also specified in uppercase.

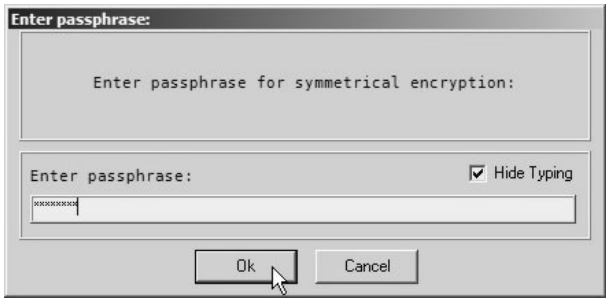


Figure 140. Entering an Encryption Passphrase in GPGee

GPG then creates an encrypted output file `picture.jpg.gpg`.

3. Rename the file to `picture.gpg`, and upload to the z/VSE host.

```
ftp> put encdata.gpg encdata
200 Command okay
150-About to open active data connection
    File:EFVSE.ENCDATA
    Type:Binary Recfm:FB Lrecl:    80 Blksize:      80
    CC=ON UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON NAT=NO CONT=OFF
    MODE=Stream  STRU=File
150 File status okay; about to open data connection
226-Bytes received: 83,556
    Records received: 1,045
    Transfer Seconds:          .06 (  1360K per second)
    File I/O Seconds:          .09 (   907K per second)
226 Closing data connection
ftp: 83556 bytes sent in 0.00Seconds 83556000.00Kbytes/sec.
ftp>
```

4. Decrypt the encrypted dataset.

```
* $$ JOB JNM=EFPGP,CLASS=0,DISP=D
// JOB EFPGP TEST OPENPGP SUPPORT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
DECRYPT
S2K_PASSPHRASE=BLAHBLAH
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

The resulting *clear data* is then written into the clear dataset indicated by the CLRFILE parameter.

Setting Up to Use OpenPGP Public-Key Encryption (PKE)

Note: If you plan to delete your original unencrypted data after encryption, ***you are strongly recommended*** to:

- Verify that your data can be decrypted successfully *before* destroying any original data.
- Keep a copy of the IJBEFPGP utility version you used to encrypt the data, to ensure you can perform the decryption at any time in future.

As described in Figure 133 on page 534, public-key encryption (PKE) requires the setup of *keystores* (such as the VSE Keyring Library or the GnuPG keystore) on both the:

- Encryption-Site's platform,

- Decryption-Site's platform.

Furthermore, Figure 133 on page 534 also shows how the Encryption-Site requires a *public key*, and the Decryption-Site uses the corresponding *private key*. Both these keys were originally generated at the *Decryption-Site*.

However, there is a problem that arises when setting up the keystores: PGP uses *DSA keys by default*, whereas z/VSE *only supports RSA keys*:

- The Windows GUI *GPG4Win* is back-level compared to the GPG command-line tool and does *not* allow the creation of RSA keys.
- You must therefore use the GPG command-line tool *directly* in order to set up a keystore with an *RSA* key pair.

OpenPGP PKE With the Encryption Done on z/VSE

These are the steps we follow:

1. Create an *RSA key pair* using the GnuPG command-line tool:

```
C:\Program Files\GNU\GnuPG\pub>gpg --gen-key
gpg (GnuPG) 1.4.7; Copyright (C) 2006 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Please select what kind of key you want:
   (1) DSA and Elgamal (default)
   (2) DSA (sign only)
   (5) RSA (sign only)
Your selection? 5
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 1y
Key expires at 10/22/08 14:05:18
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Joerg Schmidbauer
Email address: jschmidb@de.ibm.com
Comment: Blah
You selected this USER-ID:
    "Joerg Schmidbauer (Bin ich) <jschmidb@de.ibm.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
.....+++++
.....+++++
gpg: key E57429F5 marked as ultimately trusted
public and secret key created and signed.
```

```
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   3  signed:   3  trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: depth: 1  valid:   3  signed:   0  trust: 0-, 1q, 0n, 1m, 1f, 0u
gpg: next trustdb check due at 2008-10-22
pub   2048R/E57429F5 2007-10-23 [expires: 2008-10-22]
      Key fingerprint = 7B53 8429 007F BA9B C064  3227 EADE 6428 E574 29F5
uid                  Joerg Schmidbauer (Bin ich) <jschmidb@de.ibm.com>

Note that this key cannot be used for encryption.  You may want to use
the command "--edit-key" to generate a subkey for this purpose.

C:\Program Files\GNU\GnuPG\pub>
```

Once created, the GPG4Win tool can then display and process the generated RSA key.

2. Export the public key from the GnuPG keystore:
   a. Open the GUI and select **Keys – Export Keys**.

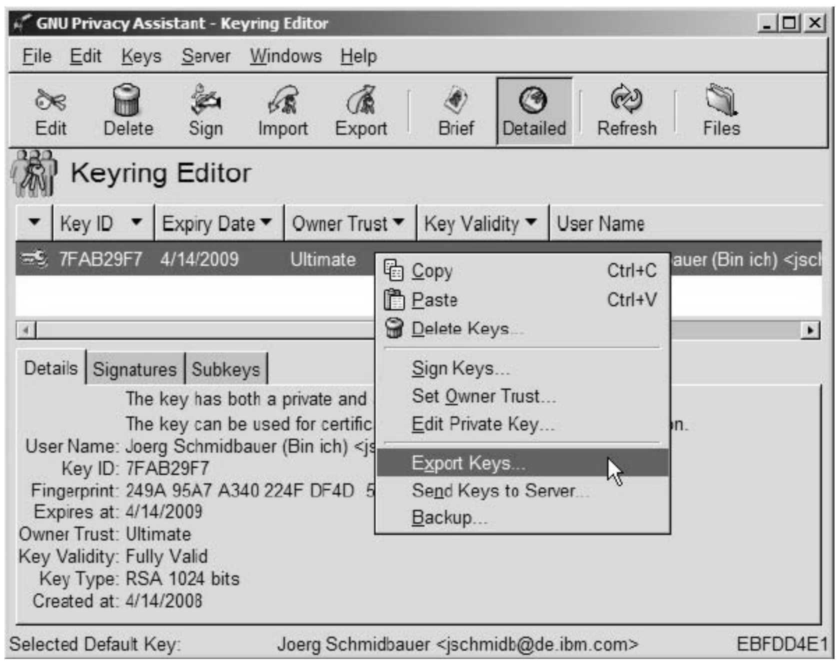


Figure 141. Export a Public Key From the GnuPG Keystore

   b. Specify an output filename and click **OK**.

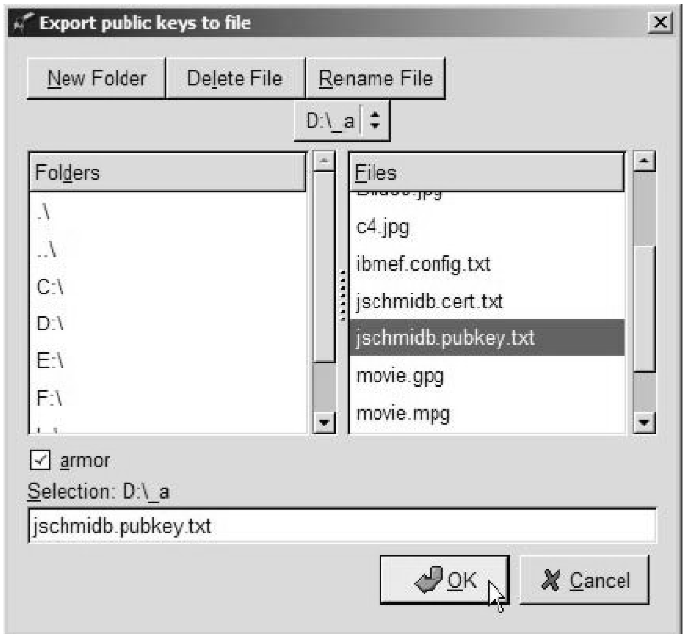


Figure 142. Specify a Filename for the File to Contain the Public Key

In the example used, the PGP public key is now contained in file `jschmidb.pubkey.txt`.

3. Import the PGP public key into Keyman/VSE:
   a. Start Keyman/VSE and click **File – Import PGP public key from file**.

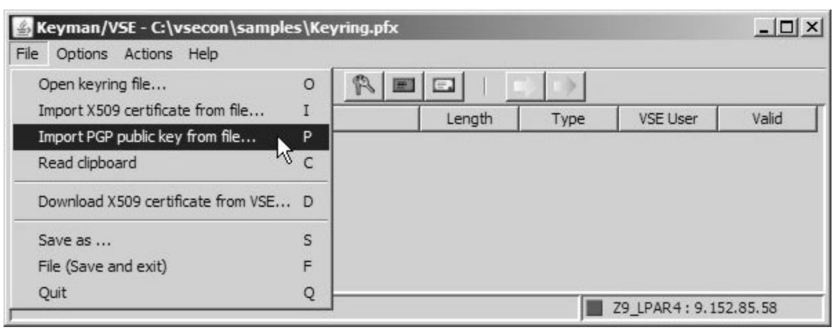


Figure 143. Import a PGP Public Key into Keyman/VSE

4. Select the file containing the PGP public key, which was exported in the previous step, and click **Open**. The “Select private key for signature” window is displayed, which asks whether or not you will use an existing root certificate to sign the certificate containing the public key.
5. Select **Create new private key for signature** and click **Continue**. The “Specify personal information” window is displayed into which you must key the certificate's details (name, organization, e-mail address, when the certificate should expire, and so on).
6. Enter the certificate's details and click **Continue**. The PGP public key will then be imported into Keyman/VSE.
7. Upload the PGP public key to the VSE Keyring Library on the z/VSE host:

a. In Keyman/VSE, right-click the imported PGP public key and select **Upload to VSE**.

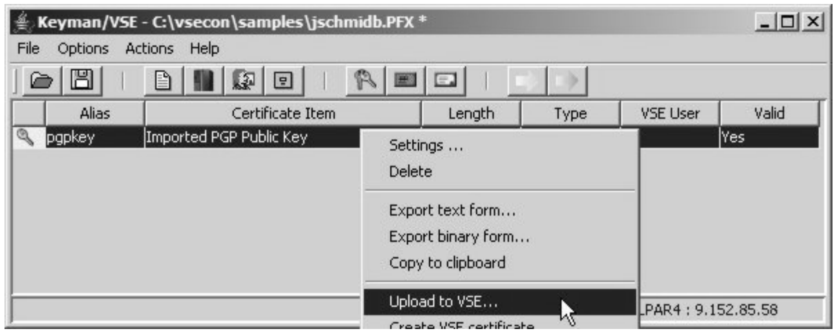


Figure 144. Upload PGP Public Key from Keyman/VSE to z/VSE host

b. The "Send Certificate Item to VSE" window is displayed. Select member type **CERT** and click **Upload**.

In this example, the PGP public key has now been stored in VSE Keyring Library member JSCHMIDB.CERT. Now we can start to encrypt a z/VSE dataset using OpenPGP public-key encryption.

8. Perform the encryption of a z/VSE dataset on z/VSE (where the *public key* to be used must be stored on z/VSE in a .CERT or .PRVK member, and the corresponding *private key* must be stored in the GnuPG keystore on the workstation-side). To do so, submit this job to encrypt the z/VSE dataset:

```
* $$ JOB JNM=EFPGP,CLASS=0,DISP=D
// JOB EFPGP TEST OPENPGP SUPPORT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
PK_ENCRYPT
RECIPIENT_ALIAS=CRYPTO.KEYRING(JSCHMIDB)
S2K_CIPHER_NAME=AES_128
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
$$ EOJ
```

To allow *multiple recipients* to decrypt the encrypted dataset, we could also specify *multiple* RECIPIENT_ALIAS parameters. This requires that the public key of *each recipient* is available on z/VSE in a .PRVK or .CERT member. Here is an example job:

```
* $$ JOB JNM=EFPGP,CLASS=S,DISP=D
// JOB EFPGP ENCRYPT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
PK_ENCRYPT
RECIPIENT_ALIAS=CRYPTO.KEYRING(BOBSKEY)
RECIPIENT_ALIAS=CRYPTO.KEYRING(JIMSKEY)
RECIPIENT_ALIAS=CRYPTO.KEYRING(RODSKEY)
RECIPIENT_ALIAS=CRYPTO.KEYRING(ALICEKEY)
RECIPIENT_ALIAS=CRYPTO.KEYRING(MYKEY)
S2K_CIPHER_NAME=AES_128
COMPRESSION=1
COMPRESS_NAME=ZIP
DIGEST_NAME=SHA_1
CLRFILE=DD:CLRDATA
```

```
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

Note: Currently, the maximum number of RECIPIENT_ALIAS parameters is **16**.

9. Use FTP to download the encrypted dataset (in *binary*) to a Windows workstation:

```
ftp> get encdata encdata.gpg
200 Command okay
150-About to open data connection
File:EFVSE.ENCDATA
Type:Binary Recfm:FB Lrecl: 80 Blksize: 80
CC=ON UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON NAT=NO CONT=OFF
MODE=Stream STRU=File
150 File status okay; about to open data connection
226-Bytes sent: 83,736
Transfer Seconds: .20 ( 409K per second)
File I/O Seconds: .01 ( 8177K per second)
226 Closing data connection
ftp: 83736 bytes received in 0.81Seconds 103.00Kbytes/sec.
ftp>
```

10. Decrypt the encrypted file on a Windows workstation.
    a. Select the encrypted file:

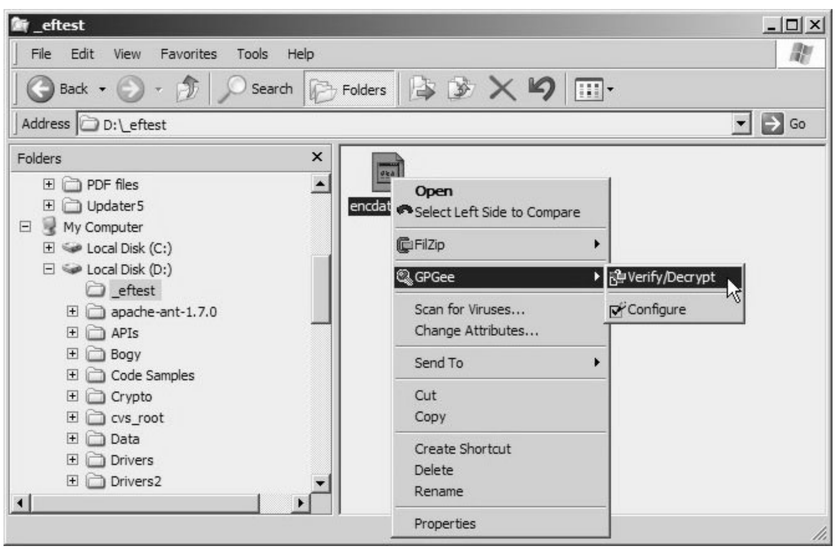


Figure 145. Select the Encrypted z/VSE Dataset Stored on a Workstation

The “Enter passphrase” window is displayed.

b. Key the passphrase of the private key and click **OK**. A confirmation message "Successfully decrypted and written to file *filename*" is displayed.

OpenPGP PKE With the Decryption Done on z/VSE

When decrypting on z/VSE, we need a .PRVK member on z/VSE that contains a *private key*. We start the process on the z/VSE-side, and then export the z/VSE public key to the *GnuPG* keystore on the Windows workstation.

These are the steps we follow:

1. Create an *RSA key pair* using the *Keyman/VSE* tool:

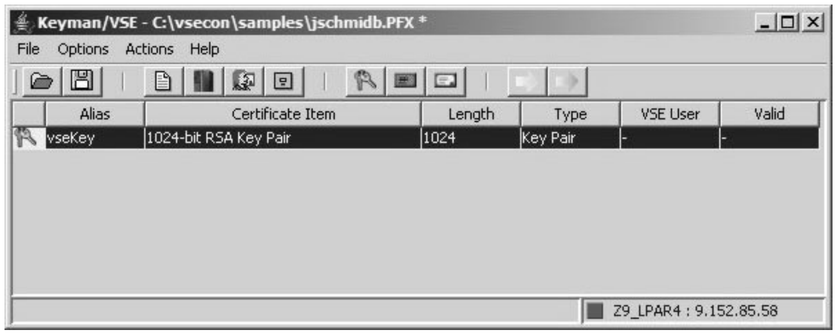


Figure 146. Create an RSA Key Pair Using Keyman/VSE

2. Upload the private key to z/VSE:
   a. Start the VSE Connector Server on z/VSE in *non-SSL mode* and upload the key pair to z/VSE.

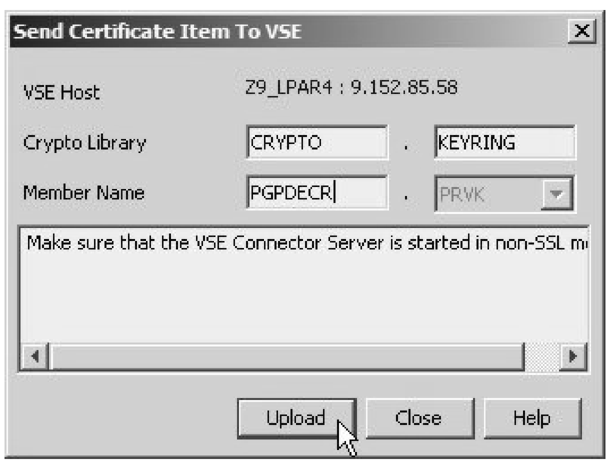


Figure 147. Upload RSA Key Pair to z/VSE

   b. As the *private key* will be used for PGP decryption, in this example the library member is given the name `PGPDECR.PRVK`.
3. Export the public key as a *PGP* public key file:
   a. In Keyman/VSE, right-click the RSA key pair and select **Export PGP public key**.

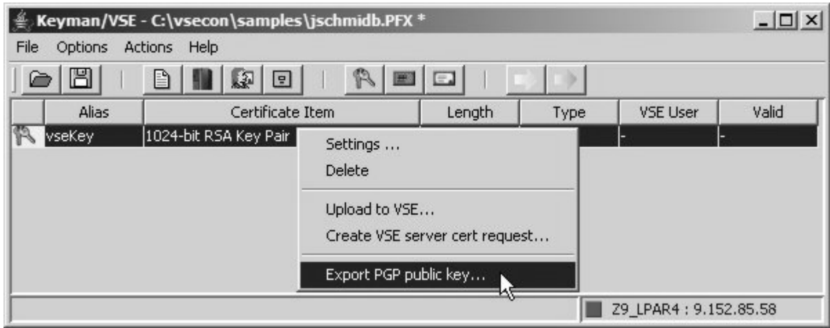


Figure 148. Export a PGP Public Key

   b. Specify the output file and click **Save**. In the example, the name `pgpdecr.pubkey.txt` is used. The “Enter Personal Information for PGP Public Key” window is displayed.

4. Key the personal information into the relevant fields (name, comment, e-mail address, when the certificate should expire, and so on). After entering these details, click **OK**. The file containing the PGP public key looks like this:


```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: IBM Encryption Facility for z/VSE V1.2
xo0ESAYBqwEEAIWsS1KT6aM0qdBrBDHat0wiAQljbtyw6GWxcpfO/mL4RYA/37lxxbVlO9BkMJzk
W5JNX4MYodUiCZ7B98Wda8kMs90xtyEb6bikVD8W228b1m8K5amg5NRTTztYoH3exwtItq31oIll
QHI2AQRCshy57lKGCTARInqf8/DQPtBpABEBAAHCuwQfAQIAJQUCSAYBqwIeAQIbDwULAgkIBwUV
AgoJCAMWAQICF4AFCQEDt4AACgkQj0AvFgM5ohcYnQP9GMWdqoRa6rKMI9C7wnKKVHaAE1uCY8dA
SWTALHrLufR+5Ua10nBE36YcGGxN/NNZu4CO2t551+Lro4Lh3dnU8TtP1kx2w0eMToobDZ2n1ivv
8G1T0AqdyW09b8qJ53pa7sZKa1ZVylfAESWUiXBfUPHEz4bJUMP78cmx/Gx8ssrNJVBHUCBEZWNj
cnlwdGlvbiA8anNjaG1pZGJAZGUuaWJtLmNvbT7CuwQTAQIAJQUCSAYBrAIeAQIbDwULAgkIBwUV
AgoJCAMWAQICF4AFCQEDt4AACgkQj0AvFgM5ohc7aAP9GMg1gDR3z5YNvHwAi3LXzyiOkae/wh1z
fE6OmyjzpmPNy2iJ+nVfQXCzuPrWYeA0sWVVLDrseVGJkJqKfaUDCsxoAoEpRUHaFc16JsFa2YCB
SIfBzrhMyROmFJwAygTnSuy7rYmr1VouO65mfkvidV1JBXTVHXIwD9bKlO93FOg=
=zugp
-----END PGP PUBLIC KEY BLOCK-----
```


5. A window is displayed which includes two confirmation lines: `1 public keys read` and `1 public keys imported`. Click **Close**. If the details of the new public key are displayed in the “GNU Privacy Assistant - Keyring Editor” window, the key only has a *public-key* (and no private-key) part.

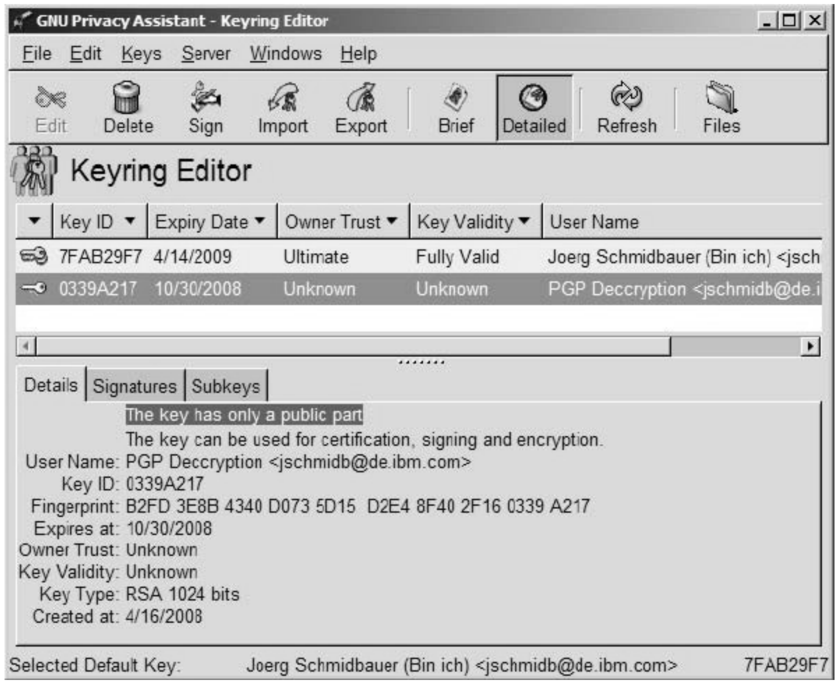


Figure 149. Display Public Key Part in the GNU Privacy Assistant window

6. Encrypt the example file `picture.jpg` "locally" (on the Windows workstation) using the *GPGee* tool.
   a. Use the Windows Explorer to select the file to be encrypted:

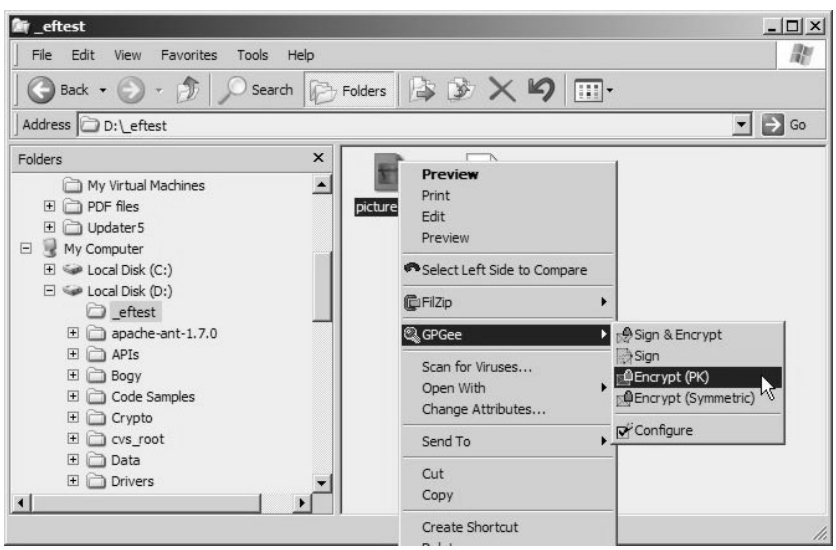


Figure 150. Select File to be Encrypted by the GPGee Tool

   b. Select the key to be used for the encryption:

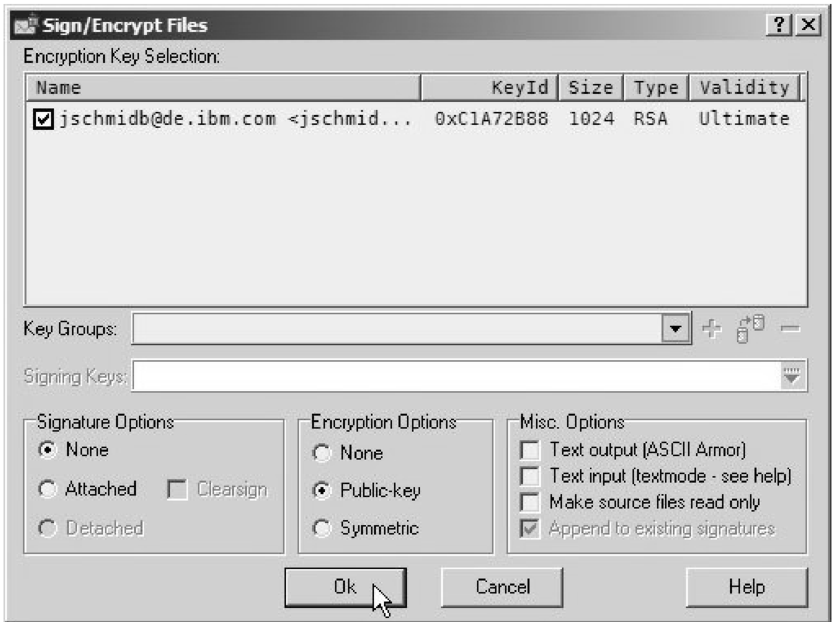


Figure 151. Select Public-Key to be Used for the Encryption

c. GnuPG creates an encrypted file `picture.jpg.gpg`. Rename the file to `picture.gpg` before uploading to z/VSE. Use FTP to perform the upload:

```
ftp> put encdata.gpg encdata
200 Command okay
150-About to open active data connection
    File:EFVSE.ENCDATA
    Type:Binary Recfm:FB Lrecl:    80 Blksize:      80
    CC=ON UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON NAT=NO CONT=OFF
    MODE=Stream  STRU=File
150 File status okay; about to open data connection
226-Bytes received: 83,556
    Records received: 1,045
    Transfer Seconds:          .06 (  1360K per second)
    File I/O Seconds:          .09 (   907K per second)
226 Closing data connection
ftp: 83556 bytes sent in 0.00Seconds 83556000.00Kbytes/sec.
ftp>
```

7. Perform the decryption of the file `encdata.gpg` on z/VSE (where the *private key* to be used must be stored on z/VSE in a .PRVK member, and the corresponding *public key* must be stored in the GnuPG keystore on the workstation-side). To do so, submit this job to decrypt the file:

```
* $$ JOB JNM=EFPGP,CLASS=0,DISP=D
// JOB EFPGP TEST OPENPGP SUPPORT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
DECRYPT
RECIPIENT_ALIAS=CRYPTO.KEYRING(PGPDECR)
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

Valid Record Formats

While workstation files are always *byte streams*, z/VSE datasets have a *record-based* structure (VSAM KSDS, RRDRS, and so on). If both encryption and decryption takes place on a *z/VSE system*, you can maintain the *original* record structure of a clear input dataset by specifying the parameter `USE_RECORDINFO`:

- The IJBEFPGP utility uses a private/experimental data packet (described in RFCs 2440 and 4880) with tag number 60 to store fields LRECL, BLKSIZE, and RECFM of the clear dataset.
- Each plaintext data record is prefixed with its length prior to encryption. This information is again used when the dataset is decrypted in order to restore the *original* record structure of the plaintext data.

OpenPGP implementations that ignore the z/VSE-specific `RECORDINFO` packet during decryption will *not* be able to distinguish between record length prefix and record data. For example, when decrypting a dataset with record information on a non-z/VSE platform, this prefix will be treated as *part of the data*. Therefore, the `USE_RECORDINFO` option should only be used when encrypting and decrypting on z/VSE.

When encrypted datasets are exchanged between different platforms (such as workstations using GnuPG) or between z/OS using the EF for z/OS, a limited number only of clear-data formats can be exchanged. Table 28 shows the combinations of clear and encrypted datasets that can be exchanged on various platforms.

Table 28. Valid Combinations When Exchanging Encrypted Datasets Between Platforms

z/VSE	Workstations	z/OS
Tape or VTAPE	N/A	Tape or VTAPE
SAM dataset	Encrypted data contains a 6-byte length field in front of each data record.	z/OS dataset. However, EF for z/OS does not retain information about the number of bytes for each record. The length information as provided by z/VSE cannot be recognized.
VSAM ESDS	Plain binary or text file	Not supported
VSAM KSDS (only clear dataset)	Encrypted data contains a 6-byte length field in front of each data record.	Not supported
VSAM RRDS (only clear dataset)	Encrypted data contains a 6-byte length field in front of each data record.	Not supported
VSAM VRDS	Not supported because of LE/VSE runtime.	Not supported
VSE Library Member	Plain binary or text file	Unix System Services file

Note: VRDS clusters are *not* supported because LE/VSE does not support VRDS.

Algorithms Supported by the IJBEFPGP Utility on System z

The OpenPGP standard supports many different algorithms. However, only some of these algorithms are supported by the IJBEFPGP utility. Other algorithms are available on specific System z platforms only. Table 29 shows a list of supported algorithms and their prerequisites.

Table 29. Algorithms Supported by the IJBEFPGP Utility

Algorithm	z9	z10	z196
MD5	yes [1]	yes [1]	yes [1]
SHA-1	yes	Yes	Yes
SHA-224	yes	Yes	Yes
SHA-256	yes	Yes	Yes
SHA-384	-	Yes	Yes
SHA-512	-	Yes	Yes
DES	yes	Yes	Yes
TDES	yes	Yes	Yes
AES-128	yes	Yes	Yes
AES-192	yes [1]	Yes	Yes
AES-256	yes [1]	Yes	Yes
RSA encrypt / decrypt	yes [2]	yes [2]	yes [2]
DSA	-	-	-

[1] algorithm available as software implementation in TCP/IP for VSE/ESA.
[2] requires latest version of TCP/IP for VSE/ESA 1.5E or higher. 2048 bit keys require a PCIXCC or Crypto Express2.

Note:
1. When using password-based encryption, parameter DIGEST_NAME only affects the creation of the data key from a given password.
2. When using public-key encryption, symmetric algorithms are used to encrypt data, whereas RSA is used to encrypt the session key.
3. Although it is possible to use any combination of symmetric, asymmetric and hash algorithm, you are recommended to use algorithms and key lengths of *similar strength*. For example, you should not encrypt an AES-256 key with a 512-bit RSA key.

Table 30 is taken from RFC4880 and shows the equivalent algorithm strength for different key sizes.

Table 30. Equivalent algorithm strength for various key sizes

Asymmetric key size (bits)	Hash size (bits)	Symmetric key size (bits)
1024	160	80
2048	224	112
3072	256	128
7680	384	192
15360	512	256

In a z/VSE environment, a good choice for public-key encryption would be to use:

- *AES-128* for data encryption.
- A *2048-bit RSA key* to encrypt the session key.

Examples of Using the IJBEFPGP Utility

This topic provides you with practical examples of how to perform the encryption of tapes, disks, files, and volumes using the *IJBEFPGP utility*.

For examples of how to perform the encryption of tapes, disks, files, and volumes using the *IJBEFVSE utility*, see "Examples of Using the IJBEFVSE Utility" on page 554.

OpenPGP Example: Obtain Help Information

This example shows how you can obtain a list of the available commands and options.

```
* $$ JOB JNM=EFPGP,CLASS=0,DISP=D
// JOB EFPGP TEST OPENPGP SUPPORT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
HELP
/*
/&
* $$ EOJ
```

OpenPGP Example: Obtain a List of Available Algorithms

This example prints a list of available algorithms to SYSLST. The list is dependent on the server being used and also the version of TCP/IP that is being used.

```
* $$ JOB JNM=EFPGP,CLASS=0,DISP=D
// JOB EFPGP TEST OPENPGP SUPPORT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
LIST_ALGO
/*
/&
* $$ EOJ
```

OpenPGP Example: Obtain Information About the Original Input File

The job below specifies that file decryption is *not* to be performed, but information about the original clear-text input file is to be recovered and written to SYSLST.

- If you specify a `RECIPIENT_ALIAS`, the IJBEFPGP utility checks if the data key can be decrypted using this private key.
- If you specify a password, the IJBEFPGP utility checks if the length is correct but does *not* attempt a decryption.

```
* $$ JOB JNM=EFPGP,CLASS=S,DISP=D
// JOB EFPGP SHOW INFO
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
INFO
RECIPIENT_ALIAS=CRYPTO.KEYRING(MYKEY)
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

OpenPGP Example: Encrypt a Library Member Using PBE

This example uses *passphrase-based encryption* (PBE). For explanations of the control statements used here, see Table 27 on page 574.

```
* $$ JOB JNM=EFPGP,CLASS=0,DISP=D
// JOB EFPGP TEST OPENPGP SUPPORT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
PB_ENCRYPT
S2K_PASSPHRASE=BLAHBLAH
S2K_CIPHER_NAME=AES_256
COMPRESSION=1
COMPRESS_NAME=ZIP
USE_RECORDINFO
DIGEST_NAME=SHA256
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

OpenPGP Example: Encrypt a Library Member Using PKE

This example uses *public-key encryption* (PKE). For explanations of the control statements used here, see Table 27 on page 574.

```
* $$ JOB JNM=EFPGP,CLASS=0,DISP=D
// JOB EFPGP TEST OPENPGP SUPPORT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
PK_ENCRYPT
RECIPIENT_ALIAS=CRYPTO.KEYRING(BOBSKEY)
S2K_CIPHER_NAME=AES_128
COMPRESSION=1
COMPRESS_NAME=ZIP
USE_RECORDINFO
DIGEST_NAME=SHA256
CLRFILE=DD:CLRDATA
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

OpenPGP Example: Decrypt a PGP Message

This example uses *public-key encryption*. The job below attempts to use *three* different private keys in the keyring library to decrypt a specified PGP message. You can specify up to 16 RECIPIENT_ALIAS statements. For explanations of the control statements used here, see Table 27 on page 574.

```
* $$ JOB JNM=EFPGP,CLASS=S,DISP=D
// JOB EFPGP DECRYPT WITH MULTIPLE PKS
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFPGP
DECRYPT
RECIPIENT_ALIAS=CRYPTO.KEYRING(BOBSKEY)
RECIPIENT_ALIAS=CRYPTO.KEYRING(JOHNSKEY)
RECIPIENT_ALIAS=CRYPTO.KEYRING(MYKEY)
CLRFILE=DD:VTAPE1
ENCFILE=DD:VTAPE2
/*
/&
* $$ EOJ
```

When specifying multiple RECIPIENT_ALIAS statements for *encryption*, the data key is encrypted multiple times by *each* specified public key. The resulting PGP message then contains multiple "Public-key encrypted session key packets".

If a recipient is able to decrypt *at least one* of these packets (that is, has at least one of the corresponding private keys) the PGP message can be decrypted.

OpenPGP Example: Encrypt a Library Member to Virtual Tape

This example uses *passphrase-based encryption*. For explanations of the control statements used here, see Table 27 on page 574.

```
* $$ JOB JNM=ENCTAPE,CLASS=S,DISP=D
// JOB ENCTAPE ENCRYPT TAPE
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
* INPUT FILE IS LIBR MEMBER
* OUTPUT FILE ON TAPE 480
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE2'
MTC REW,480
// ASSGN SYS006,480
// TLBL OUTFILE,'ENCRYPTED.DATA'
// EXEC IJBEFPGP
PB_ENCRYPT
S2K_PASSPHRASE=MYPASSWD
S2K_CIPHER_NAME=AES_128
COMPRESSION=1
COMPRESS_NAME=ZIP
DIGEST_NAME=SHA256
CLRFILE=DD:PRD2.CONFIG(IPINIT00.L)
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

OpenPGP Example: Decrypt a Library Member Contained on Virtual Tape

This example uses *passphrase-based encryption*. For explanations of the control statements used here, see Table 27 on page 574.

```
* $$ JOB JNM=DECTAPE,CLASS=S,DISP=D
// JOB DECTAPE DECRYPT TAPE
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
* INPUT FILE IS TAPE
* OUTPUT FILE IS LIBR MEMBER
VTAPE START,UNIT=480,LOC=VSAM,FILE='VTAPE2'
MTC REW,480
// ASSGN SYS006,480
// TLBL OUTFILE,'ENCRYPTED.DATA'
// EXEC IJBEFPGP
DECRYPT
S2K_PASSPHRASE=MYPASSWD
CLRFILE=DD:PRD2.CONFIG(IPINIT00.DECR)
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

OpenPGP Example: Encrypt a Library Member to a Remote Virtual Tape

This example uses *passphrase-based encryption*. For explanations of the control statements used here, see Table 27 on page 574.

```
* $$ JOB JNM=ENCRYPT,DISP=D,CLASS=0
// JOB ENCRYPT
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
VTAPE START,UNIT=480,LOC=9.152.216.57,FILE='MYTAPE'
MTC REW,480
// ASSGN SYS006,480
// TLBL OUTFILE,'MYTAPE.DATA'
// EXEC IJBEFPGP
PB_ENCRYPT
S2K_PASSPHRASE=MYPASSWD
S2K_CIPHER_NAME=AES_128
COMPRESSION=1
COMPRESS_NAME=ZIP
DIGEST_NAME=SHA256
CLRFILE=DD:PRD2.CONFIG(IPINIT00.L)
ENCFILE=DD:SYS006-OUTFILE
/*
// ASSGN SYS006,UA
VTAPE STOP,UNIT=480
/&
* $$ EOJ
```

Known Problems When Using the IJBEFPGP Utility

This topic describes some known problems when using the IJBEFPGP utility, and provides troubleshooting hints and tips.

Access to PRVK failed

Symptom:

```
T046: SSL303E IPDSCRFI failed RC=00000008(LIBROPIF) reason=00000418 00000008
T046: SSL113W IPDSCRFI get for CRYPTO KEYRING BOBSKEY PRVK failed
T046: SSL303E IPDSCRFI failed RC=000007E7(LIBRCALL) reason=000005D0
```

Possible Reason / Response:

- You are encrypting or decrypting by using one or more RECIPIENT_ALIAS statements. For example:

  ```
  // EXEC IJBEFPGP
  PK_ENCRYPT
  RECIPIENT_ALIAS=CRYPTO.KEYRING(BOBSKEY)
  RECIPIENT_ALIAS=CRYPTO.KEYRING(MYKEY)
  ```

- The IJBEFPGP utility attempts to read the specified VSE Keyring Library members with member type .PRVK and .CERT (if .PRVK not found). If a library member cannot be found in the specified VSE Keyring Library, The IJBEFPGP utility issues the messages shown above.
- Check the SYSLST output, which when
  - *decrypting*, will contain this type of output:

    ```
    SUCCESSFULLY DECRYPTED THE SESSION KEY USING CRYPTO.KEYRING(MYKEY)
    ```

  - *encrypting*, will contain this type of output:

    ```
    ENCRYPTING SESSION KEY WITH RSA PUBLIC KEY FROM PRVK:
    CRYPTO.KEYRING(MYKEY)
    ```

- In both the above cases, the output means that there was at least one member found which could be used to encrypt/decrypt the *session key*.

RSA decryption failed

Symptom:

```
SSL203E RSAD failed RC=0000002E(RSADLBD5) reason=00000144
SSL203E RSAD failed RC=0000002E(RSADNZFI) reason=00000444
SSL203E RSAD failed RC=0000002E(RSADNZFI) reason=00000444
SSL203E RSAD failed RC=0000002E(RSADLBAD) reason=00000300
```

Possible Reason / Response:

- You are decrypting using one or more private keys specified via the parameter RECIPIENT_ALIAS. Note that for *decryption* a *private key* is required. This means, the VSE Keyring Library members must be .PRVK members containing a private key.

```
// EXEC IJBEFPGP
DECRYPT
RECIPIENT_ALIAS=CRYPTO.KEYRING(MYKEY)
RECIPIENT_ALIAS=CRYPTO.KEYRING(MYKEY2)
RECIPIENT_ALIAS=CRYPTO.KEYRING(MYKEY3)
```

- The IJBEFPGP utility could read a member specified in the VSE Keyring Library, but the private key could not be used to decrypt the session key.
- Providing there is *at least one* line in the output job that is similar to the line below, one of the specified keys *could be used* to decrypt the session key.

```
SUCCESSFULLY DECRYPTED THE SESSION KEY USING CRYPTO.KEYRING(MYKEY2)
```

The text file cannot be decrypted on a workstation

Symptom:

After being encrypted on z/VSE, a plain-text file cannot be decrypted on a workstation.

Possible Reason / Response:

You probably uploaded the plain-text file to z/VSE with *character translation* (for example, using the ASCII option with FTP).

When downloading the encrypted file to your workstation, you did not change this option to BINARY.

The decrypted file contains garbage

Symptom:

When decrypting on a workstation, the decrypted file contains repeated garbage bytes located between clear-text data.

Possible Reason / Response:

You encrypted a file on z/VSE with the USE_RECORDINFO option.

Therefore, the encrypted file contains a RECORDINFO packet, which is ignored when decrypting the file on a workstation.

In addition, each data record is prefixed with a data structure that contains the length of the record that follows. This information cannot be interpreted by other PGP implementations, and is therefore treated as being part of the data.

To solve the problem, you should *not* use the USE_RECORDINFO option if the decryption is *not* to be performed on z/VSE.

The MDC cannot be found in the encrypted dataset

Symptom:

When decrypting on z/VSE, the MDC could not be found in the encrypted dataset.

Possible Reason / Response:

You encrypted the file on a workstation and then uploaded the file to z/VSE.

You probably did not specify a *variable record format* for the z/VSE target dataset (you did not specify `recfm v` when uploading via FTP). This resulted in additional padding bytes being inserted at the end of the encrypted dataset, up to the record length of the target file.

When decrypting, these padding bytes are treated as if they belonged to the file contents. This causes this error.

Duplicate key during decryption of a VSAM file

Symptom:

- When decrypting on z/VSE, the MDC could not be found in the encrypted dataset.
- You get the following message when decrypting an encrypted dataset, where the *original* input file was a VSAM KSDS file:

```
ERROR: FAILED TO WRITE DATA RECORD.
  BYTES TO WRITE : 80
  BYTES WRITTEN : 0
  REASON: DUPLICATE KEY.
```

Possible Reason / Response:

When encrypting the file, you did not use the `USE_RECORDINFO` option.

As a result, the original record structure *cannot be restored*.

Clear output records are simply written using the length of the target VSAM file, which can result in *duplicate keys*.

You must encrypt the clear file again, using the `USE_RECORDINFO` option.

Part 5. MISCELLANEOUS

Chapter 49. Supporting Application Development

This chapter describes two administrative tasks for application development: (a) tailoring compile skeletons, and (b) creating application job streams via dialog.

It contains these main topics:
- "Tailoring Compile Skeletons"
- "Creating an Application Job Stream" on page 609

Tailoring Compile Skeletons

"Handling VSE/ICCF Library Members" in the manual *VSE/ESA Programming and Workstation Guide* describes the *Program Development Library* dialog. You use this dialog to access and work with VSE/ICCF libraries. Various options allow you to create, maintain, and process library members. One of the options allows you to compile library members.

Before using the *Compile a Member* option, the related compile skeletons must be tailored as needed for your installation.

The compile skeletons are available in **VSE/ICCF library 2**. The skeleton names are:
- **C$$xxyyy**
- **C$Qxxyyy**
- **C$Dxxyyy**

where:
- **Q** identifies the skeletons provided for use with the DB2 Server.
- **D** identifies the skeletons provided for use with DL/I.

For a complete list of the skeletons available refer to the manual *z/VSE Planning* under "Tailoring Compile Skeletons".

xx can be:

CN (for C for z/VSE)
CV (for COBOL for z/VSE)
PV (for PL/I for z/VSE)
AS (for High Level Assembler for VSE)
RP (for RPG II)
FO (for VS FORTRAN)

From z/VSE V4R2.0 onwards, RPG II support is available for *CICS Transaction Server for VSE/ESA* online programs.

yyy can be:

ONL for online program.
BAT for batch program.
SUB for batch subroutine.

MAP for BMS map definition. **Note:** You can also use the TEMPLATE parameter of the *Compile Job Generation* panel (Fast Path 51) to generate an HTML map definition from a BMS MAP (for details, refer to the *VSE/ESA Programming and Workstation Guide*).

Before you tailor the skeletons, you should consider who will use them and how they will be used.

1. You can give the skeletons to all or some application programmers. Copy them from library 2 to a library to which the programmer has write access. The programmer can then tailor the skeletons.
2. You can tailor the skeletons for the entire system. In this way, you can establish certain standards for compile jobs and have every programmer use the same skeletons. For this method, tailor the skeleton files and leave them in library 2.

When a user selects the COMPILE option in the *Program Development Library* dialog, the system searches for the compile skeleton in the following order:

1. User's primary library
2. User's current secondary library (if any)
3. Common library (VSE/ICCF library 2)

Example: Skeletons C$$ASBAT and C$$ASONL

Figure 152 on page 603 shows skeleton C$$ASBAT for compiling a High Level Assembler **batch** program. Figure 153 on page 605 shows skeleton C$$ASONL for compiling a High Level Assembler **online** (CICS) program.

You should tailor the skeletons according to the needs of your program development environment. Use the information provided for the two examples when tailoring other skeletons.

Note: In the skeletons and compile examples on the following pages the statement

```
// EXEC ASMA90,SIZE=(ASMA90,64K)....
```

calls the High Level Assembler. Note that the 'C' at the end of the first line is the continuation character. Refer to the topic “Changing from DOS/VSE Assembler to High Level Assembler” in the manual *z/VSE Planning* for further details about the High Level Assembler.

Skeleton C$$ASBAT

```
* $$ JOB JNM=&JOBNAME,DISP=D,CLASS=A,NTFY=YES
* $$ LST DISP=D,CLASS=Q,PRI=3
// JOB &JOBNAME COMPILE PROGRAM &PROGNAME
// SETPARM CATALOG=&CATALOG
// IF CATALOG = 2 THEN
// GOTO NOCAT
// LIBDEF PHASE,CATALOG=lib.sublib
// OPTION ERRS,SXREF,SYM,NODECK,CATAL
   PHASE &PROGNAME,*
// GOTO ENDCAT
/. NOCAT
// OPTION ERRS,SXREF,SYM,LIST,NODECK
/. ENDCAT
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
* $$ SLI ICCF=(&PROGNAME,&PASSWORD),LIB=(&LIBNO)
/*
// IF CATALOG EQ 2 OR $MRC GT 4 THEN
// GOTO NOLNK
// EXEC LNKEDT,SIZE=256K
/. NOLNK
/&
* $$ EOJ
```

Figure 152. Compile Skeleton (C$$ASBAT) for Batch High Level Assembler Programs

Batch and Online Skeleton Information

The following information applies in general to any batch and online compile skeleton. Note that parameters beginning with an & are replaced with information the user enters when using the COMPILE option. You must change the LIBDEF statement and replace the variable **lib.sublib** with the library and sublibrary name you want to use. In addition, review and consider changing the following:

- Job class and disposition in the * $$ JOB statement.

  ```
  DISP=D
  CLASS=A
  ```

- Print class and disposition in the * $$ LST statement.

  ```
  DISP=D
  CLASS=Q
  ```

- Additional LIBDEF statements for your own libraries should be added to the supplied LIBDEF statements and inserted after each job statement.
- Check all C/VSE skeletons whether the DLBL for the C/370™ message file is the same you used during the installation of C/VSE.
- COBOL options in the OPTION statements.
- For Map Definitions (BMS map) it is possible to also generate HTML templates. These templates are stored in PRD2.DFHDOC and are used with the CICS Web Support (CWS).
- CICS preprocessor options (online program).
- SLI statement.

  VSE/POWER and VSE/ICCF features allow you to include a VSE/ICCF member at **execution** time. This reduces submit time and uses disk space more efficiently. The member does not have to be transferred from the VSE/ICCF library to the VSE/POWER reader queue. Consider the following about these features:

  1. You should **not** change the VSE/ICCF member until the job completes **or** if you need to change it, replace the SLI statement with a **/INCLUDE** statement. This will put the member in the reader queue at submit time.

2. If the compile job stream runs on another system, a correctly named VSE/ICCF member must be available at the remote system **or** if you want a member at your system compiled at another system, replace the SLI statement with a **/INCLUDE** statement. An incorrect VSE/ICCF member may cause unpredictable results during execution.

Note that names of CICS tables must start with **DFH**. When assembling CICS tables and this naming convention is not observed, the job stream does not work correctly.

Do **not** change the * $$ PUN and $ $$ PUN statements, otherwise program IESINSRT does not work correctly.

Program IESINSRT

Program IESINSRT supports any nesting level and copies all input from SYSIPT to SYSPCH up to the statement * $$ END. Statement * $$ END itself is not copied, but causes the program to exit. The punched output of IESINSRT is used to build a new entry in the VSE/POWER reader queue since the VSE/POWER JECL statement * $$ PUN DISP=I,PRI=9,CLASS=A is part of the JCL for IESINSRT. Statement * $$ END may be hidden like a VSE/POWER JECL statement by a $ instead of an *. On the highest nesting level:

- All statements starting with $ $$ are changed to * $$.
- Statements /* and /& may be hidden by a # instead of a /.
- All statements starting with # are changed to /. The # is used to avoid a premature EOF (end-of-file) condition.
- All // JOB and /& statements must be hidden by a #. The reason is that all JCL statements between a GOTO and the target label statement are ignored except for // JOB and /&. If these statements are not hidden by a #, they cause a job termination.
- If a complete job is generated by IESINSRT, make sure you also generate VSE/POWER JECL statements. Otherwise, VSE/POWER generates an AUTONAME job.

Skeleton C$$ASONL

```
------------------------------------------------------ JOB 1 (Part 1) -------------
* $$ JOB JNM=&JOBNAME,DISP=D,CLASS=A,NTFY=YES
* $$ LST DISP=D,CLASS=Q,PRI=3
* $$ PUN DISP=I,PRI=9,CLASS=A
// JOB &JOBNAME TRANSLATE PROGRAM &PROGNAME
// ASSGN SYSIPT,SYSRDR
// EXEC IESINSRT
------------------------------------------------------- JOB 1 (Data Part 1) ---------
$ $$ LST DISP=D,CLASS=Q,PRI=3
// JOB &JOBNAME COMPILE PROGRAM &PROGNAME
// SETPARM CATALOG=&CATALOG
// IF CATALOG = 1 THEN
// GOTO CAT
// OPTION ERRS,SXREF,SYM,LIST,NODECK
// GOTO ENDCAT
-------------------------------------------------------- JOB 2 (Part 1) --------------
/. CAT
// LIBDEF PHASE,CATALOG=lib.sublib
// OPTION ERRS,SXREF,SYM,CATAL,NODECK
   PHASE &PROGNAME,*
   INCLUDE DFHEAI
/. ENDCAT
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
* $$ END
-------------------------------------------------- JOB 1 (Part 2) -----------
// ON $CANCEL OR $ABEND GOTO ENDJ2              (this job generates:
// OPTION NOLIST,NODUMP,DECK                     Part 2 of JOB 2)
// EXEC DFHEAP1$,SIZE=512K
*ASM XOPTS(CICS)
* $$ SLI ICCF=(&PROGNAME,&PASSWORD),(LIB=&LIBNO)
/*
/. ENDJ2
// EXEC IESINSRT
/*
-------------------------------------------------- JOB 1 (Data Part 2) --------
// IF CATALOG NE 1 OR $MRC GT 4 THEN
// GOTO NOLNK
// EXEC LNKEDT,SIZE=256K
-------------------------------------------------- JOB 2 (Part 3) -------------
/. NOLNK
#&
$ $$ EOJ
-------------------------------------------------- JOB 1 (Part 3) --------------
* $$ END
/&
* $$ EOJ
```

Figure 153. Compile Skeleton (C$$ASONL) for Online High Level Assembler Programs

Note that JOB 1 (Part 2) contains the statement:

```
// EXEC DFHEAP1$,SIZE=512K
```

The CICS preprocessor for High Level Assembler programs punches the preprocessed source code as Part 2 of JOB 2. Data Part 2 of JOB 1 is punched as Part 3 of JOB 2 by IESINSRT.

Compile Example (Part 1)

Skeleton C$QASONL is used as an example for showing a compile skeleton (Figure 154 on page 607) and the jobs it generates (Figure 155 on page 608).

If the **DB2 Server for VSE** is not installed in **PRD2.DB2740**, you must change the LIBDEF statement accordingly.

```
-------------------------------------------- JOB 1 (Part 1)-----------------------
* $$ JOB JNM=&JOBNAME,DISP=D,CLASS=A,NTFY=YES
* $$ LST DISP=D,CLASS=Q,PRI=3
* $$ PUN DISP=I,DEST=*,PRI=9,CLASS=A
// JOB &JOBNAME DB2 PRE PROCESS &PROGNAME
// ASSGN SYSIPT,SYSRDR
// EXEC IESINSRT
-------------------------------------------- JOB 2 (Part 1) ----------------------
$ $$ LST DISP=D,CLASS=Q,PRI=3
$ $$ PUN DISP=I,DEST=*,PRI=9,CLASS=A
// JOB &JOBNAME CICS PRE PROCESS &PROGNAME
// ASSGN SYSIPT,SYSRDR
// EXEC IESINSRT
-------------------------------------------- JOB 3 (Part 1) ----------------------
$ $$ LST DISP=D,CLASS=Q,PRI=3
// JOB &JOBNAME COMPILE PROGRAM &PROGNAME
// LIBDEF *,SEARCH=PRD2.DB2740
// SETPARM CATALOG=&CATALOG
// IF CATALOG = 1 THEN
// GOTO CAT
// OPTION ERRS,SXREF,SYM,LIST,NODECK
// GOTO ENDCAT
/. CAT
// LIBDEF PHASE,CATALOG=lib.sublib
// OPTION ERRS,SXREF,SYM,CATAL,NODECK
   PHASE &PROGNAME,*
   INCLUDE DFHEAI
/. ENDCAT
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
$ $$ END
----------------------------------------------- JOB 2 (Part 2) -------------------
// ON $CANCEL OR $ABEND GOTO ENDJ3
// OPTION NOLIST,NODUMP,DECK
// EXEC DFHEAP1$,SIZE=512K
*ASM XOPTS(CICS)
* $$ END
----------------------------------------------- JOB 1 (Part 2) -------------------
// ON $CANCEL OR $ABEND GOTO ENDJ2
// LIBDEF *,SEARCH=PRD2.DB2740
// EXEC PROC=ARIS74DB
// EXEC PROC=ARIS74PL
// EXEC ARIPRPA,SIZE=AUTO,PARM='ISOL(&ISOL),&BLOCK,PREP=&PROGNAME,
               DBNAME=&DBNAX0&DBNAX1&DBNAX2,USER=&SQLUSERID/&SQLPW'
* $$ SLI ICCF=(&PROGNAME,&PASSWORD),LIB=(&LIBNO)
/*
/* ENDJ2
// EXEC IESINSRT
----------------------------------------------- JOB 2 (Part 3) -------------------/*
/*
/. ENDJ3
// EXEC IESINSRT
----------------------------------------------- JOB 3 (Part 2) --------------------
/*
// IF CATALOG NE 1 OR $MRC GT 4 THEN
// GOTO NOLNK
   INCLUDE ARIRRTED
// EXEC LNKEDT,SIZE=256K
/. NOLNK
#&
$ $$ EOJ
$ $$ END
-------------------------------------------------- JOB 2 (Part 4) -----------------
#&
$ $$ EOJ
* $$ END
----------------------------------------------- JOB 1 (Part 3) --------------------
/&
* $$ EOJ
```

Figure 154. Compile Skeleton (C$QASONL) for Online High Level Assembler Programs for DB2

Compile Example (Part 2)

The following 3 jobs are generated by skeleton C$QASONL (shown in Figure 154 on page 607).

```
---------------------------------------------- JOB 1 -------------------------------
* $$ JOB JNM=COMUSER,DISP=D,CLASS=A,NTFY=YES
* $$ LST DISP=D,CLASS=Q,PRI=3
* $$ PUN DISP=I,PRI=9,CLASS=A
// JOB COMUSER DB2 PRE PROCESS TEST
// ASSGN SYSIPT,SYSRDR
// EXEC IESINSRT
// ON $CANCEL OR $ABEND GOTO ENDJ2
// LIBDEF *,SEARCH=PRD2.DB2740
// EXEC PROC=ARIS74DB
// EXEC PROC=ARIS74PL
// EXEC ARIPRPA,SIZE=AUTO,PARM='ISOLATION(CS),NOBLK,PREP=TEST           *
               DBNAME=SQLDB,USERID=SQLDBA/SQLDBAPW'
* $$ SLI ICCF=(TEST),LIB=(0099)
/*
/. ENDJ2
// EXEC IESINSRT
/&
* $$ EOJ
------------------------------------------- JOB 2 ----------------------------
* $$ LST DISP=D,CLASS=Q,PRI=3
* $$ PUN DISP=I,PRI=9,CLASS=A
// JOB COMUSER CICS PRE PROCESS TEST
// ASSGN SYSIPT,SYSRDR
// EXEC IESINSRT
// ON $CANCEL OR $ABEND GOTO ENDJ3
// OPTION NOLIST,NODUMP,DECK
// EXEC DFHEAP1$,SIZE=512K
*ASM XOPTS(CICS)
/*
/. ENDJ3
// EXEC IESINSRT
/&
* $$ EOJ -------------------------------------------- JOB 3 -----------------------------
* $$ LST DISP=D,CLASS=Q,PRI=3
// JOB COMUSER COMPILE PROGRAM TEST
// SETPARM CATALOG=2
// IF CATALOG = 1 THEN
// GOTO CAT
// OPTION ERRS,SXREF,SYM,LIST,NODECK
// GOTO ENDCAT
/. CAT
// LIBDEF PHASE,CATALOG=lib.sublib
// LIBDEF *,SEARCH=PRD2.DB2740
// OPTION ERRS,SXREF,SYM,CATAL,NODECK
   PHASE TEST,*
   INCLUDE DFHEAI
/. ENDCAT
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),SIZE(MAXC
               -200K,ABOVE)'
/*
// IF CATALOG NE 1 OR $MRC GT 4 THEN
// GOTO NOLNK
   INCLUDE ARIRRTED
// EXEC LNKEDT,SIZE=256K
/. NOLNK
/&
* $$ EOJ
```

Figure 155. Jobs Generated by Compile Skeleton C$QASONL

For EXEC CICS batch client programs, skeletons are available in VSE/ICCF library 59 as follows:

SKEXCIAS
: High Level Assembler for VSE

SKEXCICV
: COBOL for VSE/ESA

SKEXCIPL
PL/I for VSE/ESA

SKEXCICN
C for VSE/ESA

Creating an Application Job Stream

The *Create Application Jobstream* dialog helps you create job streams. You can save the input parameters that you specify in a VSE/ICCF library member for future use. If you create another job stream with similar parameters, you can use the saved input for default values.

To access the dialog, start with the Administrator *z/VSE Function Selection* panel and select:

5 (Program Development)

2 (Create Application Job Stream)

The programmer (PROG) must choose selection **3** on the *z/VSE Function Selection* panel. The dialog displays the *Create Application Job Stream: Create or Modify* panel.

If you used the dialog before and saved your input parameters, you can use them as defaults. This is helpful if you are creating a job stream with parameters that are similar to ones in a previous job stream. The input was saved in a VSE/ICCF library member. Specify the name of the member on the panel.

If you are creating a new job stream, simply press **ENTER**. You are asked to define new parameters. On the next panel, specify the following:

PROGRAM NAME
Enter the name of the program. The name cannot be:
- ALL
- ROOT
- S

LIBRARY and SUBLIBRARY NAME
Specify the library and sublibrary where the program resides. The dialog searches for the program in this library and sublibrary.

If the sublibrary is defined in the LIBDEF PHASE,SEARCH= statement for the partition where the job runs, you can leave these fields blank.

From the *Select Functions* panel, select any optional functions for your job stream. If your program uses any Input/Output (I/O) devices, other than disk, you can provide specifications for individual devices. Enter:

1 - YES (Provide specifications)

2 - NO (Do not provide specifications)

If you specify 1 (YES), the dialog displays additional panels. The options you can specify are described in the following topics, beginning with “Printer Specifications” on page 610.

When you finish entering your options (or choose not to specify any), you can do one of two things:

1. Return to the beginning of the dialog to review and possibly change the options.

The dialog redisplays the *Select Functions* panel. Again, indicate the options you want to review or specify.

2. Continue with the dialog.

After you continue, you are asked whether you want to save the parameters. If you want to save them, enter the name of a VSE/ICCF library member. The dialog stores the parameters in the member in your default primary library.

If you do not want to save the values, just press ENTER.

The dialog creates a job with the default name PRGEXE. On the *Job Disposition* panel, you can submit the job to batch, file it in your default primary library, or both.

Printer Specifications

You can define the following requirements for up to three printouts:

- Printer address
- Logical unit

  Specify **SYSLST** or **SYS000 - SYS254**.[4]
- Output class
- Number of copies
- Form number

  This specifies that a special form is used for the output.
- Forms control buffer (FCB)
- Train image buffer (UCB)

Reader or Punch Specifications

You can specify the following units to be used by your program:

- Reader logical unit

  If your program has card input from a reader other than SYSRDR, specify the logical unit (**SYS000 - SYS254**).[4]
- Punch or punch to tape
  - Logical unit

    If your program does not write to SYSPCH, specify the logical unit (**SYS000 - SYS254**).
  - Tape address

    If your program punches to tape, enter the physical tape address.

4. Please take account of the NPGR value that is set in the ALLOC procedure. This value might be lower than 255.

Tape Specifications

You can define the following tape I/O specifications for up to four tapes:

- Tape address
- Logical unit

  Specify the logical unit which your program uses to reference the tape (**SYS000 - SYS254**).[4]
- Tape volume ID

  If you specify an ID, write down the value you use. You will need to know it later, if you use the tape for input.

  You should have comments and a PAUSE statement in the job stream for tape mount instructions.
- File name (name which your program uses to reference the tape file).
- File ID

  Specify an optional name that is associated with the file on the tape. If you enter a file ID, write down the value you specify. You will need to know it later when you process the file.
- File date

  For output tapes, this is the expiration date. For input tapes, it is the creation date.

  The date format is **YYYY/DDD**, where YYYY is the year and DDD is the day of the year.

Data Specifications

You can choose how data is included in your job stream:

1. From a VSE/ICCF library member when the job stream runs.

   You are asked for the name of the library member that contains the data. Specify the password, if the member is password-protected.
2. Data entered from the dialog.

   You can enter up to three lines of data. The dialog includes the data in the job stream.

Job Information Specifications

You can specify the following job options:

- UPSI

  You can set up to eight user program switches. Positions 0 - 7 of the UPSI byte are set from left to right. For each program switch, specify:

  0 - Switch is set off

  1 - Switch is set on

  x - Switch is unchanged
- Job date

  Specify a date to override the system date.

In addition, you may include COMMENT and PAUSE statements.

Chapter 50. Regenerating the Supervisor, VSE/POWER, or VSE/ICCF

This chapter describes various tasks related to the regeneration of z/VSE functions.

The generation of the supervisor should only be performed if you have modified any generation macros. For example, you might be using generation macros that are supplied by a vendor. There are no parameters available that you can modify.

To generate the supervisor it is necessary to first install the *Generation Feature*. This can be done 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.

This chapter contains these main topics:
- “Installing the Generation Feature”
- “Regenerating the Supervisor”
- “Regenerating VSE/POWER” on page 614
- “Regenerating VSE/ICCF” on page 617

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 will normally not need to regenerate the supervisor, unless you have modified any generation macros. The topic “Installing z/VSE Generation Feature” in the manual *z/VSE Installation* provides details for installing the Generation Feature.

Regenerating the Supervisor

From z/VSE 4.1 onwards, only **one** supervisor ($$A$SUPI) is shipped with z/VSE. You should not normally need to change this supervisor.

If, you *do* wish to change the supervisor and/or generate a listing of the supervisor, you can use the generation feature to do so. However, you *cannot* modify any of the generation options.

The value of the TRKHLD parameter (of the FOPT macro) is specified using the `IPL SYS` command. This parameter specifies the “number of hold requests” and has a **default value of 12**.

To change the value of the TRKHLD parameter in the SYS command, select the Interactive Interface dialog *Tailor IPL Procedure* (Fast Path **242**), and then option **Modify SYS command parameters** for the procedure you wish to change. For details, see “Tailoring the IPL Procedure” on page 14.

For an example of how to generate the supervisor, refer to the skeleton SKSUPASM provided in VSE/ICCF library 59.

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 156 on page 615 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, see “Skeletons for Starting Up VSE/POWER” on page 47.

```
* $$ JOB JNM=POWERGEN,CLASS=0,DISP=D
* $$ 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 156. 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.

You may add the following parameters to the skeleton (after the line --V100-- POWER shown in Figure 156):

- 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, which you can find in VSE/ICCF Library 59.

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 SNA 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 157 on page 618 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 157. 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.

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. 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 158 on page 620 shows the FORMAT and ADD statements used for the DTSFILE. For a detailed description of skeleton SKICCFMT, see "Reformatting the VSE/ICCF DTSFILE" on page 174.

z/VSE requires that you define all VSE/ICCF libraries with the DATE option. For detailed information, refer to 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 "Using Skeleton SKDTSEXT" on page 171.

```
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 158. Code Example for Formatting the DTSFILE

Chapter 51. Using RPG II With the CICS Transaction Server

From z/VSE V4R3.0 onwards, DOS/VS RPG II supports *CICS Transaction Server* online applications only. To use this support, you must install PTF UK60655 (for APARs PM16528 and PM22788), which is supplied together with these information APARs:

- II14447 (information/steps for enabling on z/VSE 4.2).
- II14452 (information/steps for enabling on z/VSE 4.3).

This chapter describes how you can migrate DOS/VS RPG II programs, macros, and samples, so they can be used in a *CICS Transaction Server* environment (instead of a *CICS/VSE* environment).

z/VSE provides the Jobs RPGINST and RPGSAMPL in VSE/ICCF Library 59. These Jobs migrate the parts of CICS/VSE that are required to support RPG II in a CICS Transaction Server environment.

This chapter contains these main topics:

- "Running Job RPGINST"
- "Running Job RPGSAMPL" on page 622

Running Job RPGINST

The Job RPGINST performs this processing:

1. Defines a new sublibrary called PRD2.RPGII.
2. Linkedits the phase DFHERP1$ (the CICS/VSE RPGII translator) into the new sublibrary (PRD2.RPGII).
3. Catalogs these CICS/VSE BMS Map structures (A-books) into sublibrary PRD2.RPGII:
   - DFHANRAT
   - DFHANRWC
   - DFHMDC
   - DFHMDCL
   - DFHMDF
   - DFHMDI
   - DFHMRPG
   - DFHMSD
   - DFHPRMCK
   - DFHSYS
4. Catalogs these R-type members into sublibrary PRD2.RPGII:
   - DFHAID
   - DFHEIBLK
   - DFHEIVAR
   - DFHBMSCA
   - DFHMSRCA

Job RPGINST must have completed successfully *before* you can add the new sublibrary PRD2.RPGII to the LIBDEFs of member C$$RPONL.

You can find the member C$$RPONL in VSE/ICCF Library 2.

Running Job RPGSAMPL

Job RPGSAMPL catalogs CICS/VSE-supplied *RPG Sample applications* into sublibrary PRD2.RPGII.

These are the RPG Sample applications that job RPGSAMPL will catalog (as *R-type* members):
- DFHXFILE
- DFHXLOGA
- DFHXL86O
- DFHXRALL
- DFHXRBRW
- DFHXRCOM
- DFHXRMA
- DHFXRMB
- DFHXRMC
- DFHXRMD
- DFHXRMK
- DFHXRML
- DFHXRMNU
- DHFXRREN
- DFHXRREP
- DFH29080

Chapter 52. Displaying System Status and Storage Information

This chapter describes the functionality that is available to you for displaying status and storage information of the z/VSE and CICS systems.

It contains these main topics:
- "Dialogs Available"
- "Using the Display Storage Layout Dialog" on page 624
- "Changing the Dialog Interval Time" on page 629

Related Topics:

For details of the ...	Refer to the following topic in the *z/VSE e-business Connectors User's Guide* ...
VSE Navigator, which uses Java classes to collect data and report-on system activity	"Using VSE Java Beans to Implement Java Programs".
VSE Monitoring Agent, which allows you to collect data from your z/VSE systems	"Collecting Data via the VSE Monitoring Agent".
GDPS® support, which allows a GDPS K-System to monitor a z/VSE system for high availability and disaster recovery purposes	"Using GDPS Support for High Availability".

Dialogs Available

z/VSE provides three dialogs for system status and storage display:
- Display System Activity
- Display Channel and Device Activity
- Display CICS TS Storage
- Display Storage Layout

Display System Activity or Channel and Device Activity

The two dialogs, *Display System Activity* and *Display Channel and Device Activity*, provide system status information for daily operation. For this reason, these two dialogs are described in detail in the manual *z/VSE Operation* in the topics "Displaying System Activity" and "Displaying Channel and Device Activity".

The system administrator can access these dialogs from the *z/VSE Function Selection Panel* as follows:
1. The *Display System Activity* dialog with Fast Path **361**.
2. The *Display Channel and Device Activity* dialog with Fast Path **362**.

The display of the first dialog is updated automatically in intervals. This interval, 10 seconds for example, can be changed by the system administrator. How to change the interval time for this dialog is described under "Changing the Dialog Interval Time" on page 629.

For the second dialog, press ENTER to get an updated display.

Display CICS TS Storage

For tuning and debugging purposes, a detailed display of the CICS TS partition layout may be required. z/VSE provides such a display via Fast Path 364. The display provides detailed information about storage allocation and usage as shown in Figure 159. The help text (PF1) provides a description of the information shown.

For a detailed discussion of CICS TS virtual storage refer to the *CICS Performance Guide*.

Instead of using the dialog, it may be sometimes more convenient entering the command **IEDC** on the CICS TS command line to get the same display.

```
IESADMDCST                    DISPLAY CICS TS STORAGE          Time:  09:40:19
  Applid: DBDCCICS    Sysid: CIC1    Jobname: CICSICCF    CICS TS Level:   111

Storage Protection ...... INACTIVE       Reentrant Programs ..... PROTECT
                                         CICS Trace Table size..      80
Extended DSA:                      (All sizes in kbyte)    LIMIT   25600
                                   ECDSA   EUDSA   ESDSA   ERDSA  Totals
  Current DSA Size ............     2048    1024    1024    6144   10240
  Current DSA used ............     1876      64       8    5220    7168
 *Peak DSA used ...............     1884      64       8    5220
  Peak DSA Size ...............     2048    1024    1024    6144   10240
  Largest free area/Free Storage    0.95    1.00    1.00    0.94
  Times short-on-storage (SOS)..       0       0       0       0       0

DSA:                                                       LIMIT    5120
                                    CDSA    UDSA    SDSA    RDSA  Totals
  Current DSA Size .............     512     256     512     512    1792
  Current DSA used .............     344       8     456     344    1152
 *Peak DSA used ................     352      28     456     344
  Peak DSA Size ................     512     256     512     512    1792
  Largest free area/Free Storage.   0.88    1.00    0.86    0.86
  Times short-on-storage (SOS)...      0       0       0       0       0
PF1=HELP      2=REFRESH   3=END       4=RETURN
```

Figure 159. Display CICS TS Storage Dialog

Using the Display Storage Layout Dialog

The data displayed by the *Display Storage Layout* dialog helps the system administrator optimize storage layout, particularly the partition layout. This is of interest especially for partitions which host permanently running programs, such as VSE/POWER or VTAM or applications installed by the user.

The values displayed may indicate that changes like the following need to be considered:

- A reduction or increase of the partition size.
- A reduction or increase of the GETVIS area of a partition.

Accessing the Dialog

To access the *Display Storage Layout* dialog, start with the Administrator *z/VSE Function Selection Panel* and select:

3 (Operation)

6 (System Status)

3 (Display Storage Layout)

Figure 160 shows the initial display.

The panel is divided into a left part for static partitions and a right part for dynamic classes.

```
IESADMDSS1                     DISPLAY STORAGE LAYOUT

------------------------------------------------------------------------------
| Total SVA (31 bit):     14M   PFIX (31 bit) Used: 2372K Limit:   47M |
-----------------------------------------------------------------------| Data
|       X  X  X        X  X           |                                | Space:
| 16M - - - - - - - - - - - - - - - - | - - - - - - - - - - - - - - -  | 9472K
|       X  X  X        X  X           |                                |
| X     X  X  X        X  X           |                                |
| X     X  X  X        X  X           |                                |
| X  X  X  X  X  X  X  X  X  X  X  X  |                                |
| BG F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB | C  J  P  R  S  Y  Z            |
| 0  1  2  3  4  5  6  7  8  9  A  B  |                                |
-----------------------------------------------------------------------|
|       |  SHARED PARTITIONS:   none                                   | Avail-
| 5120K |  Total SVA (24 bit):  3628K Unused:                     768K | able:
|       |  SUPERVISOR:           724K PFIX (24) Used:  336K Lim:   12M |   70M
------------------------------------------------------------------------------
Private space size:      70M                     Total virtual storage:  264M
PARTITION ID / DYNAMIC CLASS: __   Specify for more information

PF1=HELP      2=REFRESH   3=END       4=RETURN              6=SVA
```

Figure 160. Dialog Entry Panel

Address space IDs connected by **--** indicate that the corresponding partitions share their address space.

The 16M line symbolically marks the transition to 31-bit addresses (only shown for systems with an address space > 16MB). An **X** above the 16M line indicates that space beyond 16M has been allocated for the corresponding partition. A **|** indicates that this space is used by another static partition in the same address space. The address space IDs are shown below the partition IDs.

The size of the private area in each address space is shown at the bottom of the display.

The allocations of the dynamic classes are shown on the right part with the corresponding dynamic class IDs.

The panel allows you to specify the ID of a static partition or the class of a dynamic partition or even a dynamic partition ID directly you want to display. After pressing ENTER, either the storage values of the partition selected are shown together with the name of the job currently running in the partition or a table with the active dynamic partitions of this class. Pressing PF6 on any panel gives you an additional display showing the storage values for the SVA (shared virtual area).

Static Partition Layout Panel

Figure 161 shows the static partition layout for an F8 (CICS2) partition. You get this display by entering F8 in the initial panel shown in Figure 160 on page 625.

```
IESADMDSPL                STATIC PARTITION LAYOUT

            PARTITION: F8                           JOB NAME: CICS2
                                                    PHASE:    DFHSIP
------------------------------------------------------------------------------
'036FFFFF'X                               | GETVIS ANY              |
                                          | USED:               34M |
                                          | FREE:               16M |
- -(16M)- - - - - - - - - - - - - - - - - | HIGH WATER MARK:    40M |
            A GETVIS BELOW                | LARGEST FREE BLOCK: 16M |
            | USED:              5560K    | PFIX (31 BIT) USED:  0B |
            | FREE:              5700K    | PFIX (31 BIT) LIM.:  0B |    50M
            | HIGH WATER MARK:     11M    |                         |
'00501000'X | LARGEST FREE BLOCK: 5632K  V                          |
------------------------------------------------------------------------------
              PROGRAM AREA (EXEC SIZE)         LOADED PHASE:   582B |
'00500000'X   PFIX (24) USED:    0B LIM.:  256K AVAILABLE:     3514B | 4096B
------------------------------------------------------------------------------

                                                             PARTITION:   50M

PF1=HELP      2=REFRESH   3=END       4=RETURN               6=SVA
```

Figure 161. Static Partition Layout Panel

The panel is divided into the GETVIS part and the program part. The left part of the GETVIS part shows GETVIS BELOW (<16M) information, whereas the right part shows information about GETVIS ANY.

Note:

1. The GETVIS ANY area may reach into the GETVIS BELOW area, that means, the USED area (31 bit) and the FREE area (31 bit) include the corresponding 24-bit areas, whereas the largest free block can include the 24-bit area.
2. Any GETVIS area not initialized appears as * on the panel together with a message.
3. High Water Mark is the largest partition GETVIS size used in the current job step.
4. PFIX is the size of the fixed storage pages used by the operating system.

The addresses shown in Figure 161 have the following meaning:

```
036FFFFF = end address of partition
00501000 = start address of partition GETVIS
00500000 = start address of partition
```

You may enter the ID of any static or dynamic partition directly on this panel. You can also enter the ID of a dynamic class to get the list of the active dynamic partitions of that class displayed.

Dynamic Partition Layout Panel

Figure 162 shows the layout of dynamic partition Y1 which is used as example. You get this display by entering in the initial panel (Figure 160 on page 625) class Z and selecting on the subsequent panel Y1.

```
IESADMDDPL               DYNAMIC PARTITION LAYOUT
     PARTITION: Y1         JOB NAME: PAUSEF7        PHASE: LIBR
------------------------------------------------------------------------------
                                       | GETVIS ANY                |
                                       | USED:                 24K |
                                       | FREE:               1896K |
- - - - - - - - - - - - - - - - - - - -| HIGH WATER MARK:      24K |
'007FFFFF'X A GETVIS BELOW             | LARGEST FREE BLOCK: 1896K |
            | USED:                24K | PFIX (31 BIT) USED:    0B |
            | FREE:              1896K | PFIX (31 BIT) LIM.:    0B | 1920K
            | HIGH WATER MARK:     24K |                           |
'00620000'X | LARGEST FREE BLOCK: 1896K  V                          |
------------------------------------------------------------------------------
              PROGRAM AREA (EXEC SIZE)          LOADED PHASE:    80K |
'00520000'X   PFIX (24) USED:    0B LIM.:   48K AVAILABLE:      944K | 1024K
------------------------------------------------------------------------------
              DYNAMIC SPACE GETVIS        USED:                 24K |
                                          FREE:                104K |  128K
'00500000'X                               HIGH WATER MARK:      44K |
------------------------------------------------------------------------------
                                                          PARTITION: 3072K
PF1=HELP      2=REFRESH   3=END       4=RETURN               6=SVA
```

Figure 162. Dynamic Partition Layout Panel

The GETVIS information shown is the same as for static partitions. In addition, the dynamic space GETVIS area is shown. The meaning of the addresses shown is as follows:

```
007FFFFF = end address of partition
00620000 = start address of partition GETVIS
00520000 = start address of partition
00500000 = start address of dynamic space GETVIS
```

Note:

1. The value 0 in GETVIS ANY means that the total GETVIS area is below 16MB.
2. High Water Mark is the largest dynamic space GETVIS area size used in the current VSE/POWER job.

If the value for FREE is frequently very low you should consider a larger size for the dynamic space GETVIS area by decreasing the partition size. If the value for FREE is frequently very high, consider an increase of the partition size. For additional information, you may also refer to the chapter "Storage Management" in the manual *z/VSE Guide to System Functions*.

The panel allows you to specify for display the ID of a static partition, or the class of a dynamic partition, or a dynamic partition ID.

SVA Layout Panel

By pressing PF6 on one of the previous panels you get the SVA layout of your system; Figure 163 shows an example.

By pressing PF6 on this panel, you get back to the partition layout panel.

```
IESADMDSV1              SHARED VIRTUAL AREA LAYOUT

-------------------------------------------------------------------------------
'058FFFFF'X SYSTEM GETVIS(31) USED:   (HWM: 2788K ) |  1796K |        |
'05269000'X                   FREE:                 |  4952K |  6748K |
--------------------------------------------------------------------------|    14M
            PROGRAM AREA(31)  USED:                 |  6615K |        |
'04B00000'X                   AVAILABLE:            |   973K |  7588K |
===============================================================================
'0043FFFF'X V-POOL                                  |    64K |        |
'00415000'X SYSTEM LABEL AREA (SLA)                 |   108K |   172K |
--------------------------------------------------------------------------|
            SYSTEM GETVIS(24) USED:   (HWM:  984K ) |   868K |        |
'00288000'X                   FREE:                 |   720K |  1588K |
--------------------------------------------------------------------------|  3628K
            PROGRAM AREA(24)  USED:                 |  1482K |        |
'000C5000'X                   AVAILABLE:            |   322K |        |
-----------------------------------------------------------------|  1868K |
            SYSTEM DIRECTORY LIST (SDL)             |    64K |        |
'000B5000'X                                         |        |        |
-------------------------------------------------------------------------------
PF1=HELP      2=REFRESH   3=END       4=RETURN                6=PARTITION
```

Figure 163. SVA Layout Panel

High Water Mark (HWM) is the largest size used since the last IPL. The meaning of the SVA addresses and size values is as follows:

ADDRESSES:

```
058FFFFF = End address of SVA
05269000 = Start address of system GETVIS area (31 bit)
04B00000 = Start address of system program area (31 bit)
0043FFFF = End address of virtual pool (V-POOL)
00415000 = Start address of system label area (SLA)
00288000 = Start address of system GETVIS area (24 bit)
000C5000 = Start address of system program area (24 bit)
000B5000 = Start address of SVA (system directory list, SDL)
```

GETVIS SIZE VALUES:

```
 6748K   = Size of the system GETVIS area 31 bit (used and free,
           including GETVIS control area).
 1588K   = Size of the system GETVIS area 24 bit (used and free,
           including GETVIS control area).
```

SVA SIZE VALUES:

```
  64K     = Size of the virtual pool area (V-POOL)
 108K     = Size of the system label area (SLA)
  64K     = Size of the system directory list (SDL)
7588K     = Size of the system program area (31 bit, used and free)
 172K     = Sum of the size of the virtual pool area (V-POOL) and
            size of the system label area (SLA)
1868K     = Size of the system program area (24 bit, used and free,
            plus the size of the system directory list, SDL)
```

In the rightmost column the display shows;

```
  14M     = Size of SVA (31 bit)
3628K     = Size of SVA (24 bit)
```

Changing the Dialog Interval Time

The *Display System Activity* dialog automatically redisplays current system activity every 15 seconds. You can change the interval time or have the display updated only when the user presses ENTER. The information below outlines what you must do to make such changes.

1. Use the *Maintain Application Profiles* dialog and locate the application profile **IESLA**.
2. Create a new application profile. Use profile IESLA as a model. You cannot change the IESLA application profile itself.
3. In the DATA field, enter the value you want for the time interval:

   0 Use ENTER key to refresh the display.

   10-59 Number of seconds for time interval.

   If you do not enter a value or you enter an incorrect one, the dialog does not display an error message. It uses 15 as the default.
4. You must include the new profile either in a user profile, a selection panel, or both. You can use the following dialogs:
   - *Maintain User Profiles*
   - *Maintain Selection Panels*

The dialog calculates information using VSE job accounting tables and other system control blocks. Therefore, VSE job accounting must be active in the system (SYS JA=YES at IPL time). If job accounting is not active, you cannot access the dialog. In this case, if you try to access the dialog, the system displays an information message on the selection panel.

All jobs should include // JOB and /& statements for correct job accounting and a more accurate display of system activity. This includes startup jobs and jobs that are initiated during startup.

Chapter 53. Collecting Additional CICS Activity Data

This chapter describes how you can collect z/VSE activity in a CICS environment.

Related Topics:

For details of the ...	Refer to the following topic in the *z/VSE e-business Connectors User's Guide* ...
VSE Navigator, which uses Java classes to collect data and report-on system activity	"Using VSE Java Beans to Implement Java Programs".
VSE Monitoring Agent, which allows you to collect data from your z/VSE systems	"Collecting Data via the VSE Monitoring Agent".
GDPS support, which allows a GDPS K-System to monitor a z/VSE system for high availability and disaster recovery purposes	"Using GDPS Support for High Availability".

The CICS-based enhancement takes measurements of the **system** and **channel/device** activity of a z/VSE system. It stores the resulting data in temporary storage queues of the CICS Transaction Server (CICS TS) and activates a user exit program for further processing and analysis of the data. Note that the data provided is almost identical to the data collected by the *Display System Activity* and the *Display Channel and Device Activity* dialogs. Both dialogs are described in detail in the *z/VSE Operation* manual. You are recommended to familiarize yourself with these dialogs before working with the enhancements described in this chapter.

The advantage of the enhanced support is that activity data can be collected over a longer period of time, and can be saved for later analysis. Compared to the dialogs, the support is terminal-independent, because it communicates with the user exit program only.

Note:

1. The enhancement applies to data collection, but not to the interpretation of the data collected. It is **not** a replacement for any kind of performance tool.
2. The activity measurements require the Job Accounting parameter in the IPL SYS command to be set (JA=YES). This is the z/VSE default in a newly installed system.

The enhanced support includes CICS TS transactions, programs, queues, and a skeleton as follows:

IEXM This is the main transaction used to start a measurement cycle. IEXM requires measurement parameters as input.

IEXA IEXM invokes this transaction if **system activity** is to be measured. IEXA stores the measurement data in the temporary storage queue IESAIEXA and activates a user exit program; either IESDALOG, which is the default, or the one specified as input parameter for IEXM.

IEXS IEXM invokes this transaction if **channel/device activity** is to be measured. IEXS stores the measurement data in the temporary storage queue

IEDSIEXS and activates a user exit program; either IESCHLOG, which is the default, or the one specified as input parameter for IEXM.

IESAIEXA
This is the CICS TS temporary storage queue in which transaction IEXA saves the current system activity data of a single measurement sample for further processing by a user exit program. The data is overwritten when the next sample is taken.

IEDSIEXS
This is the CICS TS temporary storage queue in which transaction IEXS saves the current channel/device activity data of a single measurement sample for further processing by a user exit program. The data is overwritten when the next sample is taken.

IESDAOUT
This is the default name of the **temporary storage queue** in which the sample user exit program in skeleton SKEXITDA saves the contents of the temporary storage queue IESAIEXA (which contains system activity data of a single measurement sample only).

IESCHOUT
This is the default name of the **temporary storage queue** into which the sample user exit program in skeleton SKEXITDA saves the contents of the temporary storage queue IEDSIEXS (which contains channel/device activity data of a single measurement sample only).

IESDALOG
IESDALOG is the default name of the **user exit program** to be activated in case of system activity measurements.

IESCHLOG
IESCHLOG is the default name of the **user exit program** to be activated in case of channel/device activity measurements.

SKEXITDA
This skeleton provides a sample user exit program for saving measurement data. You can use the skeleton to modify the user exit program provided to meet the specific needs of your environment. The skeleton is available in VSE/ICCF library 59.

IESX This is the CICS TS abend code, which is set if a link to a user exit program is not possible.

Taking Measurements

Main transaction IEXM is used to start a measurement cycle and requires the following input parameters:

- Name of user exit program (EXIT)
- Activity to be measured (MEASURE)
- Device range for channel/device (DEVRANGE)
- Start time for measurement (STARTTIME)
- Stop time for measurement (STOPTIME)
- Interval for measurement (INTERVAL)
- Cancel request (optional) (FORCE)

To start a measurement cycle, you must call transaction IEXM and provide the input parameters needed for the measurement. You can enter the information string at the system console or at any other console of your z/VSE system. Access is also possible from a native CICS TS subsystem. An example is given below:

```
IEXM E(IESDALOG) M(S) STA(070000) STO(073000) I(000100)
```

The input requests transaction IEXM to initiate a measurement cycle for system activity starting at 7:00 and ending at 7:30 and take these measurements in intervals of 60 seconds. The example uses the short forms of the input parameters which are described in detail below.

Format of Input Parameters for Transaction IEXM

An input parameter consists of a keyword plus its value surrounded in parentheses. If the value is not specified, the defaults become effective. Parameters are separated by one or more blanks. The maximum length allowed for the string of input parameters is 100 characters.

If an error occurs in the parameter specifications, the user exit program is notified and no measurement transaction is activated.

Following is a description of the keywords allowed and the values that can be specified for them. The characters shown in uppercase are required, those in lowercase are optional:

Exit(*cccccccc*)
This parameter defines the name of the user exit program invoked by transactions IEXM, IEXA, and IEXS.

The default name is IESDALOG for measuring system activity, and IESCHLOG for measuring channel and device activity.

Measure(Sa|Cda|All)
This parameter defines the type of measurement to be taken:

Sa – System Activity

Cda – Channel and Device Activity

All – Both activities

The default is "All".

Devrange(*ccc[-ccc]*)
This parameter is only meaningful if Channel and Device activity is to be measured. The maximum range is from 000 to FFF, which is also the default.

STArttime(*hhmmss*)
This parameter defines the measurement starting time. Allowed ranges are:

- hh – from 00 to 99

  A value greater 23 specifies a time on a day following the current one.
- mm – from 00 to 59
- ss – from 00 to 59

Note that the CICS TS rules for expiration times apply here. This means that if STArttime is smaller than the current time, two cases are possible:

- If STArttime + 6 hours is greater than the current time, the task is started immediately.
- Otherwise the task is started on the next day.

The default is to start measurements immediately.

Example:

If the STArtime is specified as 100000 (10am) and the current time is 5pm (7 hours greater), the task will be started the next day. With the same STArtime and a current time of 3pm, the task would be started immediately.

STOptime(*hhmmss*)
This parameter defines the stop time of measurement. Allowed ranges are:
- hh – from 00 to 99

  A value greater 23 specifies a time on a day following the current one.
- mm – from 00 to 59
- ss – from 00 to 59

The STOptime has to be larger than the STArttime. A STOptime on a day following the STArtime day is possible with one exception: the crossing of a year boundary is not allowed.

If the STArttime given causes the task to be started on the next day, 24 hours are added to the given STOptime automatically. If the STOptime is smaller than the current time, measurement is not started.

The default is not to stop measurements, which means that the task may be stopped by the user exit program or the operator, for example.

Interval(*hhmmss*)
This parameter defines the interval from one measurement sample to the next. Allowed ranges are:
- hh – from 00 to 99
- mm – from 00 to 59
- ss – from 00 to 59

The lower limit is 10 seconds; the default is 15 seconds.

Force(Yes|No)
This parameter allows you to cancel a running measurement to enforce, for example, the start of a new measurement.
- Yes – Cancel running measurement and start new measurement.
- No – Start new measurement only if no previously defined measurement is still scheduled. This is also the default.

Transactions IEXA and IEXS

Transactions IEXA and IEXS are called by transaction IEXM to perform the actual activity measurements.

IEXM invokes IEXA for system activity measurements, IEXS for channel and device activity measurements. Both transactions store the measurement data in a CICS TS temporary storage queue, either in IESAIEXA or in IEDSIEXS, and activate the corresponding user exit program.

The transactions activate the user exit program each time a measurement sample is taken and written to the CICS TS temporary storage. The exit program should save all or important performance data stored in IESAIEXA or IEDSIEXS since the data is overwritten when the next measurement sample is taken.

The user exit program specifies a return code for the transaction by which it was invoked. This return code indicates either to continue measurements or to terminate them, even if ending time has not been reached yet. When the specified ending time is reached, the exit program is invoked a last time to signal end of measurements.

User Exit Description

Note: User-written exit programs with a name other than IESDALOG or IESCHLOG must be added to the CICS TS table PPT as follows:

```
DFHPPT TYPE=ENTRY,PROGRAM=name,RSL=PUBLIC
```

User Exit Linkage Definition

When transactions IEXM, IEXA, or IEXS establish linkage to a user exit program, they provide linkage information in the CICS TS COMMAREA. The layout of the COMMAREA used to link to a user exit program is as follows:

```
Name      Length  Type    Contents
______________________________________________________________________________

UXPL       0      B      Introducer
UXPLNAME   8      C      'IESUXPL ', name of area
UXPLLENG   2      B      Length of area
UXPLEC     1      B      Event code, event which invoked exit program
  UXECFITI               X'01' first time invocation
  UXECLATI               X'02' last time invocation
  UXECSYST               X'11' measurement data of system activity
  UXECCHDE               X'21' measurement data of channels and devices
UXPLRC     1      B      Return code of the exit routine
                         X'00' ok - continue with measurement
                            anything else - stop measurements
UXPLERRC   1      B      Error code
                         X'00' ok - no error to report
                            anything else - refer to

Figure 169 on page 639
UXPLERRI   8      C      Further error information
UXPLSTAR   6      C      Input parameter: start time    (hhmmss)
UXPLSTOP   6      C      Input parameter: stop time     (hhmmss)
UXPLINTE   6      C      Input parameter: interval      (hhmmss)
UXPLDEVR   7      C      Input parameter: device range  (ccc-ccc)
UXPLTSQU   8      C      Name of TS-Queue containing measurement data
                         (IESAIEXA or IEDSIEXS)
```

Figure 164. COMMAREA Layout for Linkage to User Exit Program

Sample User Exit Program Provided by Skeleton SKEXITDA

Skeleton SKEXITDA in VSE/ICCF library 59 provides a sample user exit program for both types of measurement data: **system activity** and **channel/device activity**. The primary purpose of the program is to save the data collected before a new measurement takes place. You can use SKEXITDA to tailor the exit program according to the needs of your installation.

You may activate the user exit program provided to see how it works. Proceed as follows:

1. Copy skeleton SKEXITDA from VSE/ICCF library 59 to your primary VSE/ICCF library.
2. Rename SKEXITDA to IESDALOG.
3. Compile the sample program (without modifying it) by using the compile option (8) of the *Program Development Library* dialog for online assembler programs.

z/VSE creates a phase named IESDALOG.PHASE and catalogs it into the sublibrary specified in compile skeleton C$$ASONL. This skeleton is stored in VSE/ICCF library 2. The sublibrary should be defined in the active LIBDEF PHASE chain for the CICSICCF partition (as shipped, this is partition F2).

4. Activate the new CICS TS phase by calling transaction CEMT as follows:

   CEMT SET PROG(IESDALOG) NEW

5. Invoke transaction IEXM as shown under "Taking Measurements" on page 632.

   For channel and device activity data you must specify EXIT(IESDALOG) as for system activity. This is because IESDALOG, as provided in skeleton SKEXITDA, can handle both: system activity and channel/device activity data. If you do not specify IESDALOG explicitly, IEXM tries to invoke the default user exit program IESCHLOG.

Transactions IEXA and IEXS save the measurement data of a **single measurement** sample in a temporary storage queue. The name of this queue is IESAIEXA for system activity data, and IEDSIEXS for channel and device activity data. The temporary storage queues IESDAOUT and IESCHOUT are used to save the CICS TS COMMAREA and the measurement records of a **complete measurement cycle**. Therefore, program IESDALOG moves the data of a single measurement sample from IESAIEXA into IESDAOUT and from IEDSIEXS into IESCHOUT.

Program IESDALOG communicates with IEXM and IEXA via the CICS TS COMMAREA by receiving event codes and possibly error codes. IESDALOG responds by providing a return code.

When modifying the user exit program, it is helpful to analyze the content of IESDAOUT and IESCHOUT by using the CICS TS transaction CEBR. Examples of CEBR displays for IESDAOUT and IESCHOUT are shown in Figure 165 through Figure 168 on page 638. Figure 165 shows a sample CEBR IESDAOUT display in character representation:

```
 CEBR           TS QUEUE  IESDAOUT RECORD     1 OF    10    COL     1 OF  2052
 ENTER COMMAND ===>
      **************************  TOP OF QUEUE  *******************************
00001 IESUXPL .....             143740000015000-FFFIESAIEXA
00002 IESUXPL .....        .........................IESAIEXA
00003 IESAIEXA01/12/0014:37:12...3.......................................r....
00004 ....... C................................. .... ..........C.......... P
00005 ....... Y100DITESYSADITTO   ................ .... .....N................
00006 IESUXPL .....        .........................IESAIEXA
00007 IESAIEXA01/12/0014:37:27...........................................r....
00008 ....... C................................. .... ..........C.......... P
00009 ....... Y100DITESYSADITTO   ................ .... .....N................
00010 IESUXPL .....        .........................IESAIEXA
      *************************  BOTTOM OF QUEUE  *****************************





PF1 : HELP               PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE
PF4 : VIEW TOP           PF5 : VIEW BOTTOM         PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : VIEW RIGHT
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

Figure 165. IESDAOUT Display in Character Representation

Figure 166 on page 637 shows a sample CEBR IESDAOUT display in hexadecimal representation:

```
 CEBR           TS QUEUE  IESDAOUT RECORD     1 OF    10    COL     1 OF  2052
 ENTER COMMAND ===>
      **************************  TOP OF QUEUE  *******************************
00001 C9C5E2E4E7D7D34000360100004040404040404040404040404040F1F4F3F7F4F0F0F0F0
00002 C9C5E2E4E7D7D340003611000040404040404040400000000000000000000000000000000
00003 C9C5E2C1C9C5E7C1F0F161F1F261F9F6F1F47AF3F77AF1F2AC42CBF359B2BA0100000000
00004 0004000000000040C3FF000000000000000000000000000000000000000000000000000000
00005 0001000000000040E8F1F0F0C4C9E3C5E2E8E2C1C4C9E3E3D640404000000000000000000
00006 C9C5E2E4E7D7D340003611000040404040404040400000000000000000000000000000000
00007 C9C5E2C1C9C5E7C1F0F161F1F261F9F6F1F47AF3F77AF2F7AC42CC01AFFA3E000000000F
00008 0004000000000040C3FF000000000000000000000000000000000000000000000000000000
00009 0001000000000040E8F1F0F0C4C9E3C5E2E8E2C1C4C9E3E3D640404000000000000000000
00010 C9C5E2E4E7D7D340003602000040404040404040400000000000000000000000000000000
      *************************  BOTTOM OF QUEUE  *****************************




PF1 : HELP               PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE
PF4 : VIEW TOP           PF5 : VIEW BOTTOM         PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : VIEW RIGHT
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

Figure 166. IESDAOUT Display in Hexadecimal Representation

Figure 167 shows a sample CEBR IESCHOUT display in character representation:

```
 CEBR           TS QUEUE  IESCHOUT RECORD     1 OF    12    COL     1 OF   348
 ENTER COMMAND ===>
      **************************  TOP OF QUEUE  *******************************
00001 IESUXPL .....             144420000015100-400IEDSIEXS
00002 IESUXPL .....       .........................IEDSIEXS
00003 14:43:54....130 F1..POWSTART.......*....130 F3..VTAMSTRT.......*....130 F
00004 14:43:54....170 F3..VTAMSTRT............170 F2..CICSICCF...2........170 F
00005 14:43:54....220 F3..VTAMSTRT...........!220 F2..CICSICCF...........!220 F
00006 14:43:54....387 F3..VTAMSTRT.......U...U388 F3..VTAMSTRT.......U...U389 F
00007 IESUXPL .....       .........................IEDSIEXS
00008 14:44:09....130 F1..POWSTART.......*....130 F3..VTAMSTRT.......*....130 F
00009 14:44:09....170 F3..VTAMSTRT............170 F2..CICSICCF...2........170 F
00010 14:44:09....220 F3..VTAMSTRT...........!220 F2..CICSICCF...........!220 F
00011 14:44:09....387 F3..VTAMSTRT.......U...U388 F3..VTAMSTRT.......U...U389 F
00012 IESUXPL .....       .........................IEDSIEXS
      *************************  BOTTOM OF QUEUE  *****************************




PF1 : HELP               PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE
PF4 : VIEW TOP           PF5 : VIEW BOTTOM         PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : VIEW RIGHT
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

Figure 167. IESCHOUT Display in Character Representation

Figure 168 on page 638 shows a sample CEBR IESCHOUT display in hexadecimal representation:

```
 CEBR           TS QUEUE  IESCHOUT RECORD     1 OF    12    COL     1 OF   348
 ENTER COMMAND ===>
      **************************  TOP OF QUEUE  *******************************
00001 C9C5E2E4E7D7D34000036010000404040404040404040404040404040F1F4F4F4F2F0F0F0F0
00002 C9C5E2E4E7D7D340000362100004040404040404040400000000000000000000000000000000
00003 F1F47AF4F37AF5F400000000F1F3F040C6F10000D7D6E6E2E3C1D9E3000001350000035C
00004 F1F47AF4F37AF5F400000000F1F7F040C6F30000E5E3C1D4E2E3D9E3000003FA00000868
00005 F1F47AF4F37AF5F400000000F2F2F040C6F30000E5E3C1D4E2E3D9E30000000B00003823
00006 F1F47AF4F37AF5F400000000F3F8F740C6F30000E5E3C1D4E2E3D9E300000003000014E4
00007 C9C5E2E4E7D7D340000362100004040404040404040400000000000000000000000000000000
00008 F1F47AF4F47AF0F90000000FF1F3F040C6F10000D7D6E6E2E3C1D9E3000001350000035C
00009 F1F47AF4F47AF0F90000000FF1F7F040C6F30000E5E3C1D4E2E3D9E3000003FA00000868
00010 F1F47AF4F47AF0F90000000FF2F2F040C6F30000E5E3C1D4E2E3D9E30000000B00003823
00011 F1F47AF4F47AF0F90000000FF3F8F740C6F30000E5E3C1D4E2E3D9E300000003000014E4
00012 C9C5E2E4E7D7D340000360200004040404040404040400000000000000000000000000000000
      *************************  BOTTOM OF QUEUE  ******************************




PF1 : HELP               PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE
PF4 : VIEW TOP           PF5 : VIEW BOTTOM         PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : VIEW RIGHT
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

Figure 168. IESCHOUT Display in Hexadecimal Representation

Flow of Events

The flow of events for a system activity measurement cycle, for example, is as follows:

1. A first time invocation of IESDALOG (start of measurement cycle) is initiated by transaction IEXM and indicated by X'01' in field UXPLEC of the CICS TS COMMAREA. IESDALOG responds by writing message "IESDALOG first" to the console and by passing return code X'00' to IEXM: continue measurement cycle.
2. Transaction IEXM calls transaction IEXA for measuring the system activity.

   After the measurement, IEXA calls exit program IESDALOG. X'11' in field UXPLEC in the CICS TS COMMAREA indicates that the call is from IEXA.

   IESDALOG saves the current measurement data (stored in IESAIEXA) in temporary storage queue IESDAOUT. Exit program IESDALOG writes message "IESDALOG measure" to the console and passes return code X'00' to IEXA: continue taking measurements.
3. A last time invocation of IESDALOG (end of measurement cycle) is indicated by X'02' in field UXPLEC of the CICS TS COMMAREA. IESDALOG responds by writing message "IESDALOG last" to the console.

Error Processing

Errors recognized by the starting transaction IEXM are, for example, syntax errors of the input parameters and CICS TS GETMAIN problems. Error information is passed to one or both user exit programs (IESDALOG and IESCHLOG) depending on the measurements currently being taken.

Error codes (in decimal) are passed in field UXPLERRC of the CICS TS COMMAREA to the user exit program. If necessary, additional information is contained in UXPLERRI. In the following error code list, the kind of additional information is appended in parentheses.

If a link to a user exit program is not possible, the task abends with the CICS TS abend code IESX.

Error Code	Explanation
0	Function completed successfully
1	Unexpected CICS TS error (EIBFN, EIBRCODE)
4	Input data is longer than 100 bytes
5	Input data not found
8	Syntax error (invalid string)
9	Invalid keyword (invalid keyword)
10	Invalid DEVRANGE specification (invalid DEVRANGE)
11	Invalid time specification (invalid time)
12	Duplicate parameter (parameter)
13	Invalid EXIT specification
16	The range limits must be from low to high (invalid DEVRANGE)
17	STARTTIME must be lower than STOPTIME
18	Interval smaller than 10 seconds (invalid INTERVAL)
20	Measure system activity attempted to start twice
21	Measure channel/device activity attempted to start twice
22	Measure system activity canceled
23	Measure channel/device activity canceled
24	Job accounting is not active
25	Internal error, unable to continue, dump taken (dump name)
48	Measure channel/device activity: no SIOs found for range
49	Measure channel/device activity: given range too large, device list incomplete

Figure 169. Error Codes Passed to the User Exit Program

Format of Measurement Data

The following topics show the layout and the format of the logging records as they are stored in the CICS TS temporary storage queues.

Format of System Activity Data

At least one record is available in the CICS TS temporary storage queue IESAIEXA for each single measurement sample. This record contains the measurement data for the **whole system** and the data for **static partitions**.

If **dynamic classes** are defined for a z/VSE system, another record follows containing measurement data for dynamic classes. If **dynamic partitions** are active, more records are created. They provide performance details about dynamic partitions (one record for each dynamic class with active dynamic partitions).

Format of Static Partitions Data

This record format provides:

- Data about overall system performance.
- Detailed performance data about static partitions.
- A summary of data about dynamic classes.

The record format reserves space for 12 static partitions and 23 dynamic classes. If there are fewer static partitions or dynamic classes defined, the rest of the record is filled with binary zeroes and blanks.

Note: In the following example:

- A "d" in the Display column indicates that this information is also provided by the activity dialogs described in the *z/VSE Operation* manual.
- The first byte of LDPARTID is the reply indicator. It is usually blank, but contains an asterisk, if there is an open reply at the console for the corresponding partition.

```
Name     Type     Display  Offset  Explanation
________________________________________________________________________________

*        DATE/TIME INFORMATION
*
LOGREC   DS    0D                  BEGINNING OF THE LOG RECORD
LQID     DS    CL8         000     LOG RECORD QID             (EBCDIC)
LDATE    DS    CL8         008     DATE IN FORMAT:  MM/DD/YY  (EBCDIC)
LTIME    DS    CL8    d    010     DISPLAY TIME:    HH:MM:SS  (EBCDIC)
LTIME2   DS    XL8         018     BEGIN THIS INTERVAL    (STCK BINARY)
LSECS    DS    XL4    d    020     ACTUAL INTERVAL IN SECS     (BINARY)
         SPACE 2
*
*        CPU UTILIZATION
*
LCPUNUM  DS    XL2    d    024     NUMBER OF CPUS IN CEC       (BINARY)
LCPUACT  DS    XL2    d    026     NUMBER OF ACTIVE CPUS       (BINARY)
LCPUQUI  DS    XL2    d    028     NUMBER OF QUIESCED CPUS     (BINARY)
         DS    0F                  FORCE ALIGNMENT
LCPU     DS    XL2    d    02C     SYSTEM CPU TIME IN %       (BINARY)
LWAIT    DS    XL2         02E     SYSTEM WAIT TIME IN %      (BINARY)
LIORATE  DS    XL4    d    030     SYSTEM SIO RATE             (BINARY)
         SPACE 2
*
*        SYSTEM PAGING ACTIVITY
*
LPAGEOUT DS    XL4    d    034     SYSTEM PAGE OUTS            (BINARY)
LOUTRATE DS    XL4    d    038     PAGE-OUT RATE PER SEC       (BINARY)
LPAGEIN  DS    XL4    d    03C     SYSTEM PAGE INS             (BINARY)
LINRATE  DS    XL4    d    040     PAGE-IN RATE PER SEC        (BINARY)
         SPACE 2
*
*        CICS TASK/STORAGE CONTROL DATA
*
LMAXTASK DS    CL4         044     CURRENT MAXTASKS (MXT) LIMIT
*                                  *                  (XMGMXT)  (BINARY)
LPEAKACT DS    CL4    d    048     PEAK # ACTIVE USER TRANSACTIONS
*                                                     (XMGPAT)  (BINARY)
LMXTLIMI DS    CL4    d    04C     TIMES MXT LIMIT REACHED
*                                  *                  (XMGTAMXT)(BINARY)
LMXTLIM2 DS    CL4         050     # USER TRANSACTIONS AT MXT LIMIT
*                                                     (XMGTDT)  (BINARY)
LTASKCNT DS    CL2         054     CURR. TASK COUNT  (DSGCNT  )
*                                  *                            (BINARY)
LTASKMAX DS    CL2         056     MAX. TASK ACCUM.  (DSGPNT  )
*                                  *                            (BINARY)
LTASKNUM DS    CL4    d    058     NO. ACTIVE USER TRANSACTIONS
*                                  *                  (XMGTAT)  (BINARY)
LTASKRAT DS    CL4    d    05C     CICS TASKS PER SEC(TENTHS)  (BINARY)
```

```
LDISPTCH DS    CL4    d    060     CICS TASKS DISPATCHABLE     (BINARY)
LSUSPEND DS    CL4    d    064     CICS TASKS SUSPENDED        (BINARY)
          SPACE 2
*
*   CICS VSAM FCT STATISTICS (removed during CICS/ESA adaptions)
*
         DS    CL10        068     unused
         SPACE 2
*
*        VSE PARTITION/DYN CLASS DATA PER JOB ACCOUNTING TABLE
 *
LNPART   DS    XL2         072     SAVE NUMBER OF PARTITIONS + CLASSES
         SPACE 1
*        LOCAL STATISTICS TABLE FOR 12 PARTITIONS AND 23 CLASSES
         DS    0F          074     FORCE WORD ALIGNMENT
LPARTAB  EQU    *                  PARTITION STATISTICS TABLE
         DS    CL2         074     dummy for alignment
LDPARTID DS    CL3    d    076     PARTITION/CLASS ID + REPLY IND
LSPACEID DS    CL1    d    079     ADDRESS SPACE ID
LPIBFLAG DS    CL2         07A     PIB FLAG
LJOBNAME DS    CL8    d    07C     JOB NAME
LPHANAME DS    CL8    d    084     PHASE NAME
LPERCENT DS    XL4    d    08C     CPU UTILIZATION IN %
LSIORATE DS    XL4         090     SIO RATE/SECOND
LCPUOVHT DS    XL4    d    094     CPU OVERHEAD TIME
LCPUTIME DS    XL4    d    098     CPU TIME IN 1/300 SECONDS
LCPUFLAG DS    CL1         09C     unused
LIOCOUNT DS    XL4    d    09D     I/O COUNT
LIOCFLAG DS    CL1         0A1     unused
LELAPSET DS    CL6         0A2     JOB START TIME FOR ELAPSE CALC
LELAPTIM DS    XL4    d    0A8     JOB ELAPSED TIME IN SECONDS
         DS    0F          0AC     FORCE WORD ALIGNMENT
LTABSIZE EQU   *-LPARTAB    38     STATISTICS TABLE SIZE
         DS    CL(34*LTABSIZE)     BUILD DISPLAY STATISTICS
LOGEND   EQU   *-LOGREC    544
*        END OF LOG RECORD
```

Format of Dynamic Classes/Dynamic Partitions Data

There are two record layouts, one is used for data on dynamic classes, the other for data on dynamic partitions of a specific dynamic class.

The following example shows the record layout for data on dynamic classes.

```
Name     Type     Display  Offset  Explanation
_____________________________________________________________________

*        LOCAL STATISTICS TABLE FOR 35 ENTRIES (PARTITIONS/CLASSES),
*        SERVES AS REPOSITORY FOR JOB ACCOUNTING DATA/DISPLAY DATA
_____________________________________________________________________

LNDATA   DS    XL2         000     NUMBER OF ENTRIES
*
*        VSE PARTITION DATA PER JOB ACCOUNTING TABLE
         DS    0F          004     FORCE WORD ALIGNMENT
LPARTAB  EQU    *                  PARTITION STATISTICS TABLE
         DS    CL3         004     dummy for alignment
LDPARTID DS    CL3         007     PARTITION ID  here: CLASS ID
LPIBFLAG DS    CL2         00A     unused
LJOBNAME DS    CL8         00C     unused
LPHANAME DS    CL8         014     unused
LPERCENT DS    XL4    d    01C     CPU UTILIZATION IN %
LSIORATE DS    XL4         020     SIO rate/second
LCPUOVHT DS    XL4    d    024     CPU OVERHEAD TIME
LCPUTIME DS    XL4    d    028     CPU TIME IN 1/300 SECONDS
LCPUFLAG DS    CL1         02C     unused
LIOCOUNT DS    XL4    d    02D     I/O COUNT
LIOCFLAG DS    CL1         031     unused
LELAPSET DS    CL6         032     unused
LELAPTIM DS    XL4         038     unused
```

```
*
LCLASS   DS    CL1    d    03C     DYNAMIC CLASS ID
LCLSFLG  DS    CL1         03D     DYNAMIC CLASS FLAG
LACTIV   DS    XL2    d    03E     ACTIVE PARTITIONS/CLASS
LMAXDP   DS    XL2    d    040     MAX ALLOCATED PARTITIONS/CLASS
         DS    0F          044     FORCE WORD ALIGNMENT
LTABSIZE EQU   *-LPARTAB   040     STATISTICS TABLE SIZE
         DS    CL(34*LTABSIZE)     BUILD DISPLAY STATISTICS
```

The following example shows the record layout for data on all dynamic partitions of one dynamic class.

```
Name     Type     Display  Offset  Explanation
________________________________________________________________________________

*        LOCAL STATISTICS TABLE FOR 35 ENTRIES (PARTITIONS/CLASSES),
*        SERVES AS REPOSITORY FOR JOB ACCOUNTING DATA/DISPLAY DATA
________________________________________________________________________________

LNDATA   DS    XL2         000     NUMBER OF ENTRIES
*
*        VSE PARTITION DATA PER JOB ACCOUNTING TABLE
         DS    0F          004     FORCE WORD ALIGNMENT
LPARTAB  EQU    *                  PARTITION STATISTICS TABLE
         DS    CL3         004     dummy for alignment
LDPARTID DS    CL3    d    007     PARTITION ID
LPIBFLAG DS    CL2         00A     PIB FLAG
LJOBNAME DS    CL8    d    00C     JOB NAME
LPHANAME DS    CL8    d    014     PHASE NAME
LPERCENT DS    XL4    d    01C     CPU UTILIZATION IN %
LSIORATE DS    XL4         020     SIO rate/second
LCPUOVHT DS    XL4         024     CPU OVERHEAD TIME
LCPUTIME DS    XL4         028     CPU TIME IN 1/300 SECONDS
LCPUFLAG DS    CL1         02C     unused
LIOCOUNT DS    XL4    d    02D     I/O COUNT
LIOCFLAG DS    CL1         031     unused
LELAPSET DS    CL6         032     JOB START TIME FOR ELAPSE CALC
LELAPTIM DS    XL4         038     JOB ELAPSED TIME IN SECONDS
*
LCLASS   DS    CL1         03C     unused
LCLSFLG  DS    CL1         03D     unused
LACTIV   DS    XL2         03E     unused
LMAXDP   DS    XL2         040     unused
         DS    0F          044     FORCE WORD ALIGNMENT
LTABSIZE EQU   *-LPARTAB   040     STATISTICS TABLE SIZE
         DS    CL(34*LTABSIZE)     BUILD DISPLAY STATISTICS
                           804
```

Format of Channel and Device Activity Data

CICS TS QUEUE IEDSIEXS contains for every measurement point as many records as necessary for the given device range. Device activity information is given for every device in the device range and for every job using this device. 12 device-job entries fit into 1 TSQUEUE record.

The following example shows record layout for data on channel and device activity.

```
Name     Type     Display  Offset  Explanation
________________________________________________________________________________

TSREC    DS    0F          000     START OF TSQUEUE RECORD
TSDTTIME DS    CL8         000     MEASUREMENT TIME
TSDFTIME DS    F           008     MEASUREMENT INTERVAL
TSLINE   DS    0F          00C
TSDCUUU  DS    CL3         00C     DEVICE NUMBER
```

```
TSDSPOOL DS    CL1          00F     DEVICE SPOOLED
TSDPART  DS    CL2          010     PARTITION ID
*        DS    CL2          012     FILLER
TSDJOBN  DS    CL8          014     JOBNAME
TSDSCUU  DS    F            01C     SIO PER DEVICE
TSDSCUX  DS    F            020     SIO PER CONTROL UNIT
TSDSCXX  DS    F            024     SIO PER CHANNEL
         DS    0F           028     FORCE WORD ALIGNMENT
TSLISIZE EQU   *-TSLINE             TSLINE SIZE
         DS    CL(11*TSLISIZE)      STORAGE FOR NEXT 11 ENTRIES
                            15C
```

Examples of Measurement Data

Example 1: Data of Two Static Partitions and One Dynamic Class

The following example shows overall system activity data and specific data for two static partitions and one dynamic class.

```
Name      Contents          Contents   Explanation
          (hexadec)         (char/dec)
________________________________________________________________________________

LQID      C9C5E2C1C9C5E7C1  IESAIEXA  Name of the logging CICS TS queue
LDATE     F0F261F1F061F9F5  02/10/00  Date in format  MM/DD/YY
LTIME     F0F87AF5F17AF1F5  08:51:15  Begin of the next interval
                                         (readable form)
LTIME2    AABCDBF43D6C7102            Begin of the next interval
                                         (internal STCK format)
LSECS     00000000F                15  Length of interval between
                                         2 measurement points in seconds
**         CPU UTILIZATION
*
LCPUNUM   0006                      6  Number of CPUs in CEC
LCPUACT   0006                      6  Number of active CPUs
LCPUQUI   0000                      0  Number of quiesced CPUs
          0000                         ... force alignment ...
LCPU      000D                     13  CPU utilization in % (CPU time):
                                        sum of all LPERCENT values of
                                        all partitions and classes
LWAIT     024B                    587  CPU utilization in % (WAIT time):
                                        100 × LCPUACT - LCPU
LIORATE   0000003A                 58  SIO rate:
                                        (SUM.current-SUM.previous) / LSECS
                                         where SUM =
                                           sum of all LIOCOUNT values
                                           of all paritions and classes
*
*         SYSTEM PAGING ACTIVITY
*
LPAGEOUT 00000003                   3  Number of pages put on secondary
                                         storage
LOUTRATE 00000000                   0  LPAGEOUT.current - LPAGEOUT.previous
                                         devided by LSECS
LPAGEIN  00000001                   1  Number of pages put into main
                                         storage
LINRATE  00000000                   0  (LPAGEIN.current - LAPGEIN.previous)
                                         devided by LSECS
*
*         CICS TASK/STORAGE CONTROL DATA
```

```
*
LMAXTASK 00000014             20  CURRENT MAXTASKS (MXT) LIMIT
*                                 *                  (XMGMXT)  (BINARY)
LPEAKACT 00000006              6  PEAK # ACTIVE USER TRANSACTIONS
*                                                    (XMGPAT)  (BINARY)
LMXTLIMI 00000000              0  TIMES MXT LIMIT REACHED
*                                 *                  (XMGTAMXT)(BINARY)
LMXTLIM2 00000000              0  # USER TRANSACTIONS AT MXT LIMIT
*                                                    (XMGTDT)  (BINARY)
LTASKCNT 000E                 14  CURR. TASK COUNT  (DSGCNT  )
*                                 *                            (BINARY)
LTASKMAX 000E                 14  MAX. TASK ACCUM.  (DSGPNT  )
*                                 *                            (BINARY)
LTASKNUM 0000003A             58  NO. ACTIVE USER TRANSACTIONS
*                                 *                  (XMGTAT)  (BINARY)
LTASKRAT 0000000A             10  CICS TASKS PER SEC(TENTHS)  (BINARY)
LDISPTCH 00000002              2  CICS TASKS DISPATCHABLE     (BINARY)
LSUSPEND 00000003              3  CICS TASKS SUSPENDED        (BINARY)
         SPACE 2
 **
*   CICS VSAM FCT STATISTICS (removed during CICS/ESA adaptions)
*
         00000000000000000000      unused
         SPACE 2
 *        VSE PARTITION/DYN CLASS DATA PER JOB ACCOUNTING TABLE
*
LNPART   0010                   16  Number of entries in job accounting
                                       table
LPARTAB  EQU    *                   Partition statistics table
```

First Example of Static Partition Statistics:

```
         0000                       ... dummy for alignment ...
LDPARTID 40C6F4                   F4 First byte: blank or asterisk
                                    Second and third byte: partition id
                                       or class id + 'FF'
LSPACEID F4                        4  Address space ID
LPIBFLAG F9F0                     90  Partition's PIB flag in characters
                                       82:  stopped
                                       80:  unbatched
                                       00:  active
LJOBNAME D5D640D5C1D4C540   NO NAME  Job name
LPHANAME 4040404040404040           Phase name
LPERCENT 00000000                  0  CPU utilization in %:
                                    (CPUDIFF × 100) / (LSECS × 300)
                                       where CPUDIFF =
                                          (LCPUTIME+LCPUOVHT).current -
                                            (LCPUTIME+LCPUOVHT).previous
LSIORATE 00000000                  0  SIO rate:
                                    LIOCOUNT.current-LIOCOUNT.previous
                                       divided by LSECS
LCPUOVHT 00000014                 20  CPU overhead time in 1/300 seconds
LCPUTIME 0000002B                 43  CPU time in 1/300 seconds
LCPUFLAG 40                           unused
LIOCOUNT 00000058                 88  Number of start I/Os
LIOCFLAG 40                           unused
LELAPSET 000000000000                 Job start time
LELAPTIM 00000000                  0  Job elapsed time in seconds
```

Second Example of Static Partition Statistics:

```
         0000                       ... dummy for alignment ...
LDPARTID 40C2C7                   BG First byte: blank or asterisk
                                    Second and third byte: partition id
                                       or class id + 'FF'
LSPACEID F0                        0  Address space ID
LPIBFLAG F0F0                     00  Partition's PIB flag in characters
                                       82:  stopped
                                       80:  unbatched
```

```
                                             00:  active
LJOBNAME C9C5E2E7C4C1C340  IESXDAC   Job name
LPHANAME C1E2D4C1F9F04040  ASMA90    Phase name
LPERCENT 00000009                 9  CPU utilization in %:
                                     (CPUDIFF×100) / (LSECS×*300)
                                        where CPUDIFF =
                                           (LCPUTIME+LCPUOVHT).current -
                                            (LCPUTIME+LCPUOVHT).previous
LSIORATE 0000001F                31  SIO rate:
                                     LIOCOUNT.current-LIOCOUNT.previous
                                        devided by LSECS
LCPUOVHT 00000116               278  CPU overhead time in 1/300 seconds
LCPUTIME 000003E5               997  CPU time in 1/300 seconds
LCPUFLAG 40                          unused
LIOCOUNT 00000497              1175  Number of start I/Os
LIOCFLAG 40                          unused
LELAPSET AA9C09C400C0                Job start time
LELAPTIM 00000025                37  Job elapsed time in seconds
```

Example of Dynamic Class Statistics:

```
         0000                        ... dummy for alignment ....
LDPARTID 40E8FF                   Y  First byte: blank or asterisk
                                     Second and third byte: partition id
                                        or class id + 'FF'
LSPACEID 00                          Address space ID
LPIBFLAG F0F0                    00  Partition's PIB flag in characters
                                        82:  stopped
                                        80:  unbatched
                                        00:  active
LJOBNAME 0000000000000000            Job name
LPHANAME 0000000000000000            Phase name
LPERCENT 00000003                 3  CPU utilization in %:
                                     (CPUDIFF×100) / (LSECS×300)
                                        where CPUDIFF =
                                          (LCPUTIME+LCPUOVHT).current -
                                           (LCPUTIME+LCPUOVHT).previous
LSIORATE 00000019                25  SIO rate:
                                     LIOCOUNT.current-LIOCOUNT.previous
                                        devided by LSECS
LCPUOVHT 00000078               120  CPU overhead time in 1/300 seconds
LCPUTIME 000000BC               188  CPU time in 1/300 seconds
LCPUFLAG 40                          unused
LIOCOUNT 0000036B               875  Number of start I/Os
LIOCFLAG 40                          unused
LELAPSET 000000000000                Job start time
LELAPTIM 00000000                    Job elapsed time in seconds
```

Example 2: Data of One Dynamic Class/One Dynamic Partition

The following example shows data of one dynamic class.

```
Name     Content           Content    Explanation
         (hexadec)         (char/dec)
_______________________________________________________________________________

LNDATA   0004                    4   Number of entries
*
*        VSE PARTITION DATA PER JOB ACCOUNTING TABLE
         0000                        ... dummy for alignment ...
LDPARTID 40E8FF                   Y  First byte: blank
                                     Second and third byte:
                                         class id + 'FF'
LPIBFLAG 0000                        unused
LJOBNAME 0000000000000000            unused
LPHANAME 0000000000000000            unused
LPERCENT 00000003                 3  CPU utilization in %:
                                     (CPUDIFF×100) / (LSECS×300)
                                        where CPUDIFF =
```

```
                                       (LCPUTIME+LCPUOVHT).current -
                                        (LCPUTIME+LCPUOVHT).previous
LSIORATE 00000019               25  SIO rate:
                                    LIOCOUNT.current-LIOCOUNT.previous
                                       devided by LSECS
LCPUOVHT 00000078              120  CPU overhead time in 1/300 seconds
LCPUTIME 000000BC              188  CPU time in 1/300 seconds
LCPUFLAG 40                         unused
LIOCOUNT 0000036B              875  Number of start I/Os
LIOCFLAG 40                         unused
LELAPSET 000000000000               unused
LELAPTIM 00000000                   unused
*
LCLASS   E8                      Y  Dynamic class id
LCLSFLG  80                         Class table entry flag:
                                    x'80' - dynamic class enabled
                                    x'40' - class table entry in error
LACTIV   0001                    1  Active partitions/class
LMAXDP   0008                    8  Max. allocated partitions/class
```

The following example shows data of a single dynamic partition

```
Name     Content          Content    Explanation
         (hexadec)        (char/dec)
________________________________________________________________________________

*        LOCAL STATISTICS TABLE FOR 35 ENTRIES (PARTITIONS/CLASSES),
*        SERVES AS REPOSITORY FOR JOB ACCOUNTING DATA/DISPLAY DATA
LNDATA   0001                    1  Number of active partitions/class
*        VSE PARTITION DATA PER JOB ACCOUNTING TABLE
         0000                       ... dummy for alignment ...
LDPARTID 40E8F2                 Y2  First byte: blank or asterisk
                                    Second and third byte: partition id
LPIBFLAG F0F0                   00  PIB flag
                                       82:  stopped
                                       80:  unbatched
                                       00:  active
LJOBNAME C9C5E2E7C4C4C340  IESXDDC   Job name
LPHANAME C1E2D4C1F9F04040  ASMA90    Phase name
LPERCENT 00000003                3  CPU utilization in %:
                                      (CPUDIFF×100) / (LSECS×300)
                                        where CPUDIFF =
                                          (LCPUTIME+LCPUOVHT).current -
                                           (LCPUTIME+LCPUOVHT).previous
LSIORATE 00000019               25  SIO rate:
                                    LIOCOUNT.current-LIOCOUNT.previous
                                       devided by LSECS
LCPUOVHT 00000078              120  CPU overhead time in 1/300 seconds
LCPUTIME 000000BC              188  CPU time in 1/300 seconds
LCPUFLAG 40                         unused
LIOCOUNT 0000036B              875  Number of start I/Os
LIOCFLAG 40                         unused
LELAPSET AAA079330140               Job start time
LELAPTIM 00000020               32  Job elapsed time in seconds
*
LCLASS   00                         unused
LCLSFLG  00                         unused
LACTIV   0000                       unused
LMAXDP   0000                       unused
```

Example 3: Channel and Device Activity Data

The following example shows channel and device activity data of one device and one job using this device.

```
Name     Content           Content    Explanation
         (hexadec)         (char/dec)
________________________________________________________________________________

TSDTTIME F0F67AF5F67AF5F4  06:56:54  Measurement time
TSDFTIME 0000000F                15  Measurement interval
TSLINE
TSDCUUU  F2F3F0                 230  Device number (cuu)
TSDSPOOL 40                          Device spooled
                                        x'40' - no
                                        x'5C' - yes
TSDPART  E8F2                    Y2  Partition ID
         0000                        ... filler ...
TSDJOBN  C9C5E2D7C4C4C340  IESXDDC   Job name
TSDSCUU  00000151               337  SIO per device
TSDSCUX  00005131             20785  SIO per control unit
TSDSCXX  00005131             20785  SIO per channel
```

Chapter 54. Fast Paths and Synonyms for Dialogs

Related Topic:
- "Dialogs of the Interactive Interface" in the manual *z/VSE Planning*, SC34-2635.

This chapter lists the Fast Paths and synonyms that can be used to access dialogs of the Interactive Interface. For information about creating your own synonyms, refer to "Maintaining Synonyms" on page 142.

In Table 31:
- **z/VSE Selection** is the name used for a dialog in a selection panel of the Interactive Interface (*Analyze and Apply PTFs*, for example).
- **Default For** shows where the dialog appears in the default panel hierarchies supplied by z/VSE:
  - **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.
- **Fast Path** shows the number string that can be used to access the dialog from a user's *initial* selection panel.
- **Default Synonym** shows the character string (if any) that is predefined for the dialog. Users can enter the synonym from *any* selection panel of the Interactive Interface to access the dialog.

  Note that predefined synonyms for groups of dialogs (such as the dialogs for backing up data) are also listed in the figure.

Table 31. Fast Paths and Synonyms for Dialogs

z/VSE Selection	Default For	Fast Path	Default Synonym
Analyze and Apply PTFs	A	1422	---
Apply PTFs	A	1423	---
Archive All ICCF Libraries on Tape	A P O	37242 5652 5242	--- --- ---
Backup	A P O	37 56 5	BACKUP BACKUP BACKUP

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Backup a File	A P O	3752 5672 552	--- --- ---
Backup a User Catalog to Disk	A O	37184 5184	--- ---
Backup a User Catalog to Tape	A O	37183 5183	--- ---
Backup a Volume	A P O	3751 5671 551	--- --- ---
Backup History File	A O	373 53	--- ---
Backup the DTSFILE (All ICCF Libraries)	A P O	37241 5651 5241	--- --- ---
Backup the Master Catalog to Disk	A O	37182 5182	--- ---
Backup the Master Catalog to Tape	A O	37181 5181	--- ---
Backup VSAM File	A P O	3713 563 513	--- --- ---
Backup VSE Library on Tape	A O	3721 521	--- ---
BSM Cross Reference Report	A	286	---
BSM Group Maintenance	A	282	---
BSM Resource Profile Maintenance	A	281	---
BSM Security Rebuild	A	283	---
Catalog Printer UCB	A	244	UCB
Change Nicknames	A	146	---
Configure Hardware	A	241	---
Console	A P O S	31 51 2 2	CONSOLE CONSOLE CONSOLE ---

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Copy a File	 A P O	 3772 5692 572	 --- --- ---
Copy a Volume	 A P O	 3771 5691 571	 --- --- ---
Create Application Job Stream	 A P	 52 3	 --- ---
Create Standalone Dump Program on Tape	 A P S	 461 461 161	 --- --- ---
Create Standalone Dump Program on Disk	 A P S	 462 462 162	 --- --- ---
Customize z/VSE Workstation Platform	 A	216	---
Define a Library	 A P	 223 63	 --- ---
Define a New File	 A P	 222 62	 --- ---
Define a New User Catalog	A	226	---
Define an Alternate Index or Name	 A P	 224 64	 --- ---
Define Transaction Security	A	285	---
Defragmentation of History File	A	147	---
Display Active Users/Send Message	 A P O S	 33 52 41 4	 USERS or MESSAGE ---
Display Channel and Device Activity	 A O	 362 72	 SIO SIO
Display CICS TS Storage	 A O	 364 74	 --- ---
Display or Process a Catalog, Space	A	225	---
Display or Process a File	 A P	 221 61	 --- ---

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Display Storage Layout	A O	363 73	--- ---
Display System Activity	A P O	361 55 71	DA DA DA
Display VTOC	A	23	VTOC
Down-Level Check	A	1431	---
Dump Program Utilities	A P S	46 46 16	--- --- ---
Enter News	A P O	34 53 42	NEWS NEWS NEWS
Export-Disconnect a User Catalog	A O	3715 515	--- ---
Export ICCF Library Members to Tape	A P O	37243 5653 5243	--- --- ---
Export VSAM File	A P O	3711 561 511	--- --- ---
Fast Service Upgrade	A	143	---
File and Catalog Management Dialogs	A P	22 6	VSAM VSAM
FlashCopy VSAM Catalog/Files	A O	3719 519	--- ---
Format ICCF Dump Data	A P S	467 467 167	--- --- ---
FSU Installation	A	1433	---
FSU Preparation	A	1432	---
IBM Service	A	14	SERVICE
Import-Connect a User Catalog	A O	3716 516	--- ---

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Import ICCF Library Member	A P O	37254 5664 5254	--- --- ---
Import VSAM File	A P O	3712 562 512	--- --- ---
Inspect Dump Management Output	A P S	44 44 14	--- --- ---
Inspect Message Log	A P S	42 42 12	LOG LOG ---
Install Generation Feature	A	13	---
Install Program(s) from Tape	A	112	---
Install Programs - V1 Format	A	12	---
Install Programs - V2 Format	A	11	---
Invoke CEDA	A	72	RDO
Invoke CEMS	A	73	CEMS
Invoke CEOS	P O	82 62	CEOS CEOS
Invoke CEMT	A P O	71 81 61	MT MT MT
List and Process User Files in Host Transfer File	A P	381 581	--- ---
Look Up PTF/APAR	A P S	1448 458 158	--- --- ---
Maintain Application Profiles	A	213	APM
Maintain Certificate–User ID List	A	284	---
Maintain Dynamic Partitions	A	27	---
Maintain LDAP User Profiles	A	217	LUPM
Maintain PRIMARY Sublibraries	A	215	---
Maintain Printer FCB	A	243	FCB
Maintain Printer UCB	A	244	UCB
Maintain Security Profiles: Resource Class ACICSPCT	A	2812	---

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Maintain Security Profiles: Resource Class APPL	A	2818	---
Maintain Security Profiles: Resource Class DCICSDCT	A	2813	---
Maintain Security Profiles: Resource Class FACILITY	A	2819	---
Maintain Security Profiles: Resource Class FCICSFCT	A	2814	---
Maintain Security Profiles: Resource Class JCICSJCT	A	2815	---
Maintain Security Profiles: Resource Class MCICSPPT	A	2816	---
Maintain Security Profiles: Resource Class SCICSTST	A	2817	---
Maintain Security Profiles: Resource Class TCICSTRN	A	2811	---
Maintain Selection Panels	A	212	SPM
Maintain Synonyms	A P O	214 57 8	SYNONYMS SYNONYMS SYNONYMS
Maintain User Profiles	A	211	UPM
Maintain VTAM Application Names	A	25	---
Maintain VTAM Startup Options	A	26	---
Manage Batch Queues	A P O S	32 2 3 3	POWER POWER POWER ---
Manage List Queue	A P O S	321 21 31 31	LST LST LST ---
Manage Punch Queue	A P O S	323 23 33 33	PUN PUN PUN ---
Manage Reader Queue	A P O S	322 22 32 32	RDR RDR RDR ---

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Manage Transmit Queue	A P O S	324 24 34 34	XMT XMT XMT ---
Manage Wait for Run Subqueue	A P O S	325 25 35 35	--- --- --- ---
Manage In-Creation Queue	A P O S	326 26 36 36	--- --- --- ---
Move Files from Host Transfer File to ICCF	A P	385 585	--- ---
Move Files from Host Transfer File to VSAM	A P	383 583	--- ---
Move ICCF Members to Host Transfer File	A P	384 584	--- ---
Move VSAM Files to Host Transfer File	A P	382 582	--- ---
Online Problem Determination	A P S	414 411 11	OLPD OLPD ---
PC File Transfer	A P	386 586	--- ---
Personal Computer Move Utilities	A P	38 58	IWS ---
Personalize History File	A	145	---
Prepare for Installation	A	111	---
Print IPL Diagnostics	A P S	466 466 166	--- --- ---
Print SDAID Tape	A P S	468 468 168	--- --- ---
Print Service Document	A	1421	---

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Print Stand-Alone Dump	A P O	469 469 169	--- --- ---
Program Development Library	A P O	51 1 1	ICCFS ICCFS ICCFS
Program Development Library (Primary Library)	A P O	511 11 11	ICCF ICCF ICCF
Remove FlashCopy Relation	A	3773	---
Remove PTF Records from History File	A	1424	---
Remove Standalone Dump Program from a SYSRES Disk	A P O	463 463 163	--- --- ---
Restore	A P O	37 56 5	RESTORE RESTORE RESTORE
Restore a File	A P O	3762 5682 562	--- --- ---
Restore a Member of an ICCF Library	A P O	37253 5663 5253	--- --- ---
Restore a User Catalog from Disk	A O	37174 5174	--- ---
Restore a User Catalog from Tape	A O	37173 5173	--- ---
Restore a Volume	A P O	3761 5681 561	--- --- ---
Restore History File	A O	374 54	--- ---
Restore One ICCF Library	A P O	37252 5662 5252	--- --- ---

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Restore the DTSFILE (All ICCF Libraries)	A P O	37251 5661 5251	--- --- ---
Restore the Master Catalog from Disk	A O	37172 5172	--- ---
Restore the Master Catalog from Tape	A O	37171 5171	--- ---
Restore VSAM File	A P O	3714 564 514	--- --- ---
Restore VSE Library from Tape	A O	3722 522	--- ---
Retrace History File	A P S	1441 451 151	--- --- ---
Retrace APARs	A P S	1445 455 155	--- --- ---
Retrace Component ID	A P S	1447 457 157	--- --- ---
Retrace Components	A P S	1443 453 153	--- --- ---
Retrace Members	A P S	1446 456 156	--- --- ---
Retrace Products	A P S	1442 452 152	--- --- ---
Retrace PTFs	A P S	1444 454 154	--- --- ---
Retrieve Message	A P O S	35 54 43 5	RETRIEVE RETRIEVE RETRIEVE ---

Table 31. Fast Paths and Synonyms for Dialogs (continued)

z/VSE Selection	Default For	Fast Path	Default Synonym
Scan Dump Files on Tape	A P O	464 464 164	--- --- ---
Scan Dump Files on Disk	A P O	465 465 165	--- --- ---
Scan VSE Library Backup Tape	A O	3723 523	--- ---
Storage Dump Management	A P S	43 43 13	--- --- ---
Tailor IPL Procedure	A	242	---
TCP/IP Configuration	A	245	---
Verify Location of Involved Serviced Files	A	141	---

Part 6. Appendixes

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
: A function of VSE that ensures that the system and the data and programs stored in it can be accessed only by authorized users in authorized ways.

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 VSE/VTAM.

ACF/VTAM
: See VTAM.

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 Function Printing (AFP)
: A group of IBM licensed programs that support APA printers.

*** alternate index**
: In systems with VSE/VSAM, a collection of index entries related to a given base cluster and organized by an alternate key, that is, a key other than the prime key of the base cluster data records; it gives an alternate directory for finding records in the data component of a base cluster. See also *path*.

*** alternate library**
: An interactively accessible library that can be accessed from a terminal when the user of that terminal issues a connect or switch library request. Synonymous with *public library*.

*** application profile**
: A control block in which the system stores the characteristics of one or more application programs.

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.

*** automated system initialization (ASI)**
A function that allows control information for system startup to be cataloged for automatic retrieval during system startup.

*** autostart**
A facility that starts up VSE/POWER with little or no operator involvement.

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.

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. See also *control block*.

*** catalog**
1. 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)
2. To store a library member such as a phase, module, or book in a sublibrary.

See also *VSAM master catalog, VSAM user catalog*.

*** cataloged procedure**
A set of control statements placed in a library and retrievable by name.

CICS Transaction Server for VSE/ESA
This is the successor system to CICS/VSE.

*** CICS/VSE**
Customer Information Control System/VSE. No longer shipped on the Extended Base Tape.

*** common library**
An interactively accessible library that can be accessed by any user of the system or subsystem that owns the library.

*** compaction**
In SNA, the transformation of data by packing two characters in a byte so as to take advantage of the fact that only a subset of the allowable 256 characters is used; the most frequently sent characters are compacted.

*** compile**
To translate a source program into an executable program (an object program). See also *assembler*.

conditional job control
The capability of the job control program to process or to skip one or more statements based on a condition that is tested by the program.

*** configuration**
The devices and programs that make up a system, subsystem, or network.

connect
To authorize library access on the lowest level. A modifier such as "read" or "write" is required for the specified use of a sublibrary.

*** control area (CA)**
In VSE/VSAM, a group of control intervals used as a unit for formatting a data set before adding records to it. Also, in a key-sequenced data set, the set of control intervals, pointed to by a sequence-set index record, that is used by VSAM for distributing free space and for placing a sequence-set index record adjacent to its data.

control block
An area within a program or a routine defined for the purpose of storing and maintaining control information.

*** control interval (CI)**
A fixed-length area of disk storage where VSE/VSAM stores records and distributes free space. It is the unit of information that VSE/VSAM transfers to or from disk storage. For FBA, it must be an integral multiple, to be defined at cluster definition, of the block size.

DASD sharing
An option that lets independent computer systems use common data on shared disk devices.

*** data entry panel**
A panel in which the user communicates with the system by filling in one or more fields. See also *panel* and *selection panel*.

data file
See *file*.

Data Interfile Transfer, Testing and Operations (DITTO) utility
An IBM program that provides file-to-file services for card I/O, tape, and disk devices. The latest version is called DITTO/ESA for VSE.

data set
See *file*.

default value
A value assumed by the program when no value has been specified by the user.

*** device address**
1. The identification of an input/output device by its channel and unit number. 2. In data communication, the identification of any device to which data can be sent or from which data can be received.

*** device type code**
The four- or five-digit code to be used for defining an I/O device to a computer system.

*** dialog**
1. In an interactive system, a series of related inquiries and responses similar to a conversation between two people. 2. 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*.

directory
1. A table of identifiers and references to the corresponding items of data. (I) (A) 2. In VSE, specifically, the index for the program libraries. See also *library directory* and *sublibrary 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.

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.

DOSRES
Disk operating system residence volume.

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**
1. Data that has been dumped. (T) 2. 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)

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*.

Enterprise Systems Architecture (ESA)
See *ESA/390*.

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.

*** escape**
To return to the original level of a user interface.

extent Continuous space on a disk occupied by or reserved for a particular file or VSAM data space.

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.

*** FCP** See *Fibre Channel Protocol*.

***fibre-channel connection (FICON®)**
A fibre-channel communication protocol designed for IBM mainframe computers and peripherals.

***Fibre Channel Protocol (FCP)**
The serial SCSI command protocol used on fibre-channel networks.

***FICON**
See *fibre-channel connection*.

*** file** A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

*** fragmentation (of storage)**
In virtual system, inability to assign real storage locations to virtual storage addresses because the available spaces are smaller than the page size.

FULIST (FUnction LIST)
A type of selection panel that displays a set of files and/or functions for the choice of the user.

*** generate**
To produce a computer program by selecting subsets of skeletal code under the control of parameters. (A)

generation
See *macro generation*.

*** GETVIS space**
Storage space within a partition or the shared virtual area, available for dynamic allocation to programs.

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.

*** 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.

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.

*** initial program load (IPL)**
The process of loading system programs and preparing the system to run jobs.

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.

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.

job accounting
A system function that lists how much every job step uses of the different system resources.

*** job catalog**
A catalog made available for a job by means of the filename IJSYSUC in the respective DLBL job control statement.

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 step
One of a group of related programs complete with the JCL statements necessary for a particular run. Every job step is identified in the job stream by an EXEC statement under one JOB statement for the whole job.

job stream
The sequence of jobs as submitted to an operating system.

label information area
An area on a disk to store label information read from job control statements or commands. Synonymous with *label area*.

*** librarian**
The set of programs that maintains, services, and organizes the system and private libraries.

library
See *VSE library* and *VSE/ICCF library*.

*** 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).

*** 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.

*** local address**
In SNA, an address used in a peripheral node in place of a network address and transformed to or from a network address by the boundary function in a subarea node.

*** 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 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)
1. A name used in programming to represent an I/O device address. 2. In SNA, a port through which a user accesses the SNA network, to communicate with another user.

master console
In z/VSE, one or more consoles that receive all system messages, except for those that are directed to a particular console. The operator of a master console can reply to all outstanding messages and enter all system commands.

major node
In VTAM, a set of minor nodes that can be activated as a group. See *node* and *minor node*.

*** member**
The smallest unit of data that can be stored in and retrieved from a sublibrary. See also *library member*.

message
1. 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. 2. In telecommunication, a logical set of data being transmitted from one node to another.

*** minor node**
In VTAM programs, a uniquely-defined resource within a major node. See *node* and *major node*.

*** 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)

*** 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 *local address*. See also *network name*.

network addressable unit (NAU)
In SNA, a logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted by the path control network. Each NAU has a network address that represents it to the path control network. See also *network name*, *network address*.

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**
1. 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. 2. In VTAM programs, a point in a network defined by a symbolic name. Synonymous with *network node*. See *major node* and *minor node*.

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*.

object module (program)
A program unit that is the output of an assembler or compiler and is input to a linkage editor.

optional program
An IBM licensed program that a user can install on VSE by way of available installation-assist support.

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 frame
An area of processor storage that can contain a page.

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* and *data entry panel*.

partition
A division of the virtual address area available for running programs. See also *dynamic partition, static partition*.

*** path**
1. In VTAM, the intervening nodes and data links connecting a terminal and an application program in the host processor. 2. In VSAM, a named logical entity providing access to the records of a base cluster either directly or through an alternate index.

personal computer (PC)
A microcomputer for individuals or small businesses.

*** physical record**
The amount of data transferred to or from auxiliary storage. Synonymous with *block*.

*** physical unit (PU)**
In SNA, the component that manages and monitors the resources of a node.

*** primary library**
A VSE library owned and directly accessible by a certain terminal user.

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*.

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)

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*.

prompt
  To issue messages to a terminal or console user, requesting information necessary to continue processing.

protocol
  In SNA, the set of rules for requests and responses between communicating nodes that want to exchange data.

* **queue file**
  A direct access file maintained by VSE/POWER that holds control information for the spooling of job input and job output.

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.

* **restore**
  To write back onto disk data that was previously written from disk onto an intermediate storage medium such as tape.

* **routing**
  The assignment of the path by which a message will reach its destination.

SAM ESDS file
  A SAM file managed in VSE/VSAM space, so it can be accessed by both SAM and VSE/VSAM macros.

* **search chain**
  The order in which chained sublibraries are searched for the retrieval of a certain library member of a specified type.

security
  See *access control*.

* **selection panel**
  A displayed list of items from which a user can make a selection. Synonymous with *menu*.

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 file**
  A file in which records are processed in the order in which they are entered and stored.

* **service program**
  A computer program that performs functions in support of the system. Synonymous with *utility program*.

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 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.

* **software**
  All or part of the programs, procedures, rules, and associated documentation of a data processing system. Software is an

intellectual creation that is independent of the medium on which it is recorded. (T)

*** 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.

*** standard label**
A fixed-format record that identifies a volume of data such as a tape reel or a file that is part of a volume of data.

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*.

*** subarea**
A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all links, and adjacent link stations in attached peripheral or subarea nodes that are addressable within the subarea share a common subarea address and have distinct element addresses.

sublibrary
A subdivision of a library. Members can only be accessed in a sublibrary.

sublibrary directory
An index for the system to locate a member in the accessed sublibrary.

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*.

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 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.

*** tailor**
A process that defines or modifies the characteristics of the system.

*** 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.

*** transaction**
1. In a batch or remote batch entry, a job or job step. 2. 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.

*** transmit**
To send data from one place for reception elsewhere. (A)

*** utility program**
1. 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*. 2. 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 space**
A subdivision of the virtual address area available to the user for the allocation of private, nonshared partitions.

Virtual Machine/Enterprise Systems Architecture (VM/ESA)
The most advanced VM system currently available.

virtual storage
Addressable space image for the user from which instructions and data are mapped into processor (real) storage locations.

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.

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/ICCF (VSE/Interactive Computing and Control Facility)
An IBM program that serves as interface, on a time-slice basis, to authorized users of terminals linked to the system's processor.

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/POWER
An IBM 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.

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.

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.

VTAM (Virtual Telecommunications Access Method)
An IBM program which 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).

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.

z/VM z/Virtual Machine.

z/VSE (z/Virtual Storage Extended)
The most advanced VSE system currently available.

31-bit addressing
Provides addressability for address spaces of up to 2 gigabytes.

Index

Special characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Readers' Comments — We'd Like to Hear from You

IBM z/VSE
Administration
Version 5 Release 1

Publication No. SC34-2627-01

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:
- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: +49-7031-163456
- Send your comments via email to: s390id@de.ibm.com
- Send a note from the web page: http://www.ibm.com/systems/z/os/zvse/

If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	Email address

Readers' Comments — We'd Like to Hear from You
SC34-2627-01

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 Research & Development GmbH
Department 3248
Schoenaicher Strasse 220
71032 Boeblingen
Germany

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Product Number: 5609-ZV5

Printed in USA

SC34-2627-01