IBM VSE/Enterprise Systems Architecture

Programming and Workstation Guide

Version 2 Release 7

SC33-6709-01

IBM VSE/Enterprise Systems Architecture

Programming and Workstation Guide

Version 2 Release 7

SC33-6709-01

Note!

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

Third Edition (September 2003)

This edition applies to Version 2 Release 7 Modification Level 1 of IBM Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA), Program No. 5690-VSE, and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

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

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

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

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

Chapter 6. File Transfer To and From VSE Libraries 89

Chapter 7. File Transfer To and From VSE/POWER Queues 107

Chapter 8. File Transfer To and From the Host Transfer File (HTF) 137

Figures

Notices

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

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

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

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

IBM Deutschland Informationssysteme GmbH
Department 0215
Pascalstr. 100
70569 Stuttgart
Germany

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

Programming Interface Information

This manual is intended to help the customer to use functions provided by VSE/ESA for file transfer between VSE/ESA resources and programmable workstations. It also contains information on functions of VSE/ESA's Interactive Interface for program development. This manual is intended for application programmers and system administrators.

This *VSE/ESA Programming and Workstation Guide* documents intended Programming Interface Information that allow the customer to write programs to obtain the services of VSE/ESA.

Programming Interface Information is identified in the chapters it occurs by the following marking:

Programming Interface Information

Description of Programming Interface Information ...

End of Programming Interface Information

Trademarks and Service Marks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
Advanced Function Printing
C/370
CICS
CICS/VSE
Common User Access
DB2
DB2 Connect
DB2 Server for VSE
DFSORT
ECKD
Enterprise Systems Architecture/390
ESA/390
IBM
Language Environment
MQSeries
Netfinity
NetView
OS/2
OS/390
OS/400
Print Service Facilities
PSF
RISC System/6000
System/370
System/390
S/390
Virtual Machine/ Enterprise Systems Architecture
VM/ESA
VSE/ESA
VTAM
WebSphere
xSeries
z/OS
zSeries

Java, Java Beans, and JavaScript are trademarks or registered trademarks of Sun Microsystems, Inc..

Microsoft, Windows, the Windows 95 logo, Windows NT and Windows XP, are trademarks or registered trademarks of Microsoft Corporation.

Visual Basic is a registered trademark of Microsoft Corporation.

Lotus, 1-2-3, Domino, and Notes are trademarks or registered trademarks of the IBM-Lotus Corporation.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

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

About This Book

This manual gives application programmers, administrators and other users of programmable workstations information about:

- Application development with VSE/ESA's Interactive Interface
- File transfer between VSE/ESA resources and IBM programmable workstations.

Please note that the remainder of this manual uses the term workstation to mean programmable workstation. In some cases the term PC has been used. It also means programmable workstation.

> **Note!**
>
> The support described in this manual is available with the CICS Transaction Server for VSE/ESA but not with CICS/VSE.
>
> Whenever the term CICS is used, it refers to the CICS Transaction Server for VSE/ESA.

The manual has two parts:

- **Part 1** describes the use of the **VSE/ESA Interactive Interface** for application development using VSE/ICCF. It covers topics such as:
  - **Communicating with the System and other Users**

    This is covered in Chapter 2, "Communicating with the System and Other Users", on page 19.
  - **Using Program Development Functions.**

    This is covered in Chapter 3, "Using Program Development Functions", on page 25.

  Note that this manual **does not** contain "Interactive Interface information for administrators or operators". For this information, refer to the IBM manuals *VSE/ESA Administration* and *VSE/ESA Operation*.
- **Part 2** describes VSE/ESA's **Workstation File Transfer Support**. This part covers the whole scope of file transfer functions available for VSE/ESA's resources. It addresses application programmers as well as system administrators and other users.

  The Workstation File Transfer Support includes SEND/RECEIVE command capabilities for transferring user data to and from the following VSE/ESA resources:
  - **VSE libraries:** Give you a central data repository, for example, for backup purposes and as permanent storage. Data that you want to keep for a longer time should be stored here.
  - **VSE/POWER queues:** Let you keep data for an intermediate time (for example, offloading large amounts of workstation data for unformatted printing at the host).
  - **Host Transfer File (HTF):** Used to access VSE/ICCF libraries or VSE/VSAM data.
  - **CICS temporary storage (TS):** Used for immediate processing (CICS TS is located on disk).

In addition, VSE/ESA's Workstation File Transfer Support provides **service function** capabilities for:

- VSE libraries.
- VSE/POWER queues.
- VSE/ESA user profile.

Note that this manual **does not** have information on how to "communicate between nodes" in a network. This topic is covered in the *VSE/ESA Networking Support* manual.

Removal of the VSE/ESA Distributed Workstation Feature!

This manual no longer contains information about the VSE/ESA Distributed Workstation Feature. From VSE/ESA 2.5 onwards, this feature is no longer supported and has been replaced by the *VSE Navigator* application, which uses VSE Java Beans and has advanced functionality. For example, using the VSE Navigator you can:

- Access VSE file systems (POWER, Librarian, ICCF, VSAM).
- Create and submit jobs, including generating jobs based upon the skeletons stored in ICCF library 2.
- Work with the VSE operator console.
- Compare files, and perform a full-text search in VSE-based file systems.
- Interactively insert and edit VSAM records.
- Display the VSE hardware configuration, the VSE system activity (CPU usage, and so on), the current VSE service level, the system labels, the system tasks, the used and free VSAM space, and VSAM data (via maps and views).

For details about the VSE Navigator, refer to the chapter "Using VSE Java Beans to Implement Java Programs" in the *VSE/ESA e-business Connectors User's Guide*.

How to Use This Book

Terms and Abbreviations

In this manual the following terms and abbreviations are often used:

DBCS workstations
are programmable workstations that support the **D**ouble **B**yte **C**haracter **S**et (DBCS), for example, the IBM 55xx and PS/55.

VSE libraries
are VSE private libraries and the system library IJSYSRS.

Workstation
is a programmable workstation (including PCs).

For other unfamiliar terms or abbreviations, refer to the *Glossary* located at the back of the book.

Examples and Additional Instructions

Using Examples in this Manual

This manual contains many examples. There are examples which describe various file transfer cases from and to the:

- VSE libraries
- VSE/POWER queues
- Host Transfer File
- CICS temporary storage queues

Moreover, there are examples describing correct use of the SEND/RECEIVE command syntax and common parameters which are applicable to all of the above VSE/ESA resources.

Please note that the panels in this manual are examples only. They operate with numbers that do not necessarily reflect realistic values.

Looking up Examples in the Index

For your convenience and reference, you may look for an example in the index:

1. All examples are alphabetically listed under entry ’examples’.
2. Also, every topic (for example, VSE libraries or the SEND/RECEIVE option FILE=) has an ’example’ entry.

Using Additional Instructions

Sometimes, there are exceptions or additional instructions for particular users. These are shown in a box with an appropriate heading. An example is:

Output File is a Work File

This is an example of special instructions for users whose output file is a reusable work file.

If the heading of the box applies to the task you are doing, follow the instructions in the box.

You can also use TCP/IP commands to send and receive files over the network. For details, refer to the *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*.

Where to Find More Information

VSE/ESA Home Page

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

You can find the VSE/ESA home page at:

```
http://www.ibm.com/servers/eserver/zseries/os/vse/
```

Your system administrator gives you access to all or some of the following functions:

- Workstation File Transfer Support functions.

- Program development functions.
- File management.

Some functions are not discussed here (for example, “defining a VSE/VSAM file” or “tailoring compile skeletons”). The IBM manual *VSE/ESA Administration* shows the system administrator or any person with administrative authority how to make system resources available to users.

The following IBM manuals contain further **VSE/ESA**-related information on aspects covered in this manual:

- *VSE/ESA Administration*, SC33-6705
- *VSE/ESA Guide for Solving Problems*, SC33-6710
- *VSE/ESA Operation*, SC33-6706
- *VSE/ESA Planning*, SC33-6703
- *VSE/ICCF User’s Guide*, SC33-6739

For further information about **VSE/POWER**, see:

- *VSE/POWER Administration and Operation*, SC33-6733.
- *VSE/POWER Application Programming*, SC33-6736.

For further information about **TCP/IP**, see:

- *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*, SC33-6601.

Summary of Changes

Changes for Third Edition (September 2003)

The third edition of this manual, that is published with VSE/ESA Version 2 Release 7 Modification Level 1, contains these changes:

- References to the *VSE/ESA Distributed Workstation Feature* have been removed from this manual. This feature has been replaced by the *VSE Navigator*, which offers considerable additional functionality. For details about the VSE Navigator, refer to the *VSE/ESA e-business Connectors User's Guide*.
- This information has been added that relates to VSE/ESA Version 2 Release 6 and VSE/ESA Version 2 Release 7, and that was originally included in the March 2003 Edition of the *VSE/ESA Release Guide*:
  - The *Primary Library* and *Secondary Library* dialogs offer a new function that allows the displayed information to be sorted by size. For details, see "Working With the FULIST" on page 28.
  - The *Primary Library* dialog offers a display option for browsing the contents of a library member. For details, see "Working With the FULIST" on page 28.
  - The *Compile Job Generation* dialog has been enhanced with a new parameter TEMPLATE, that allows you to request the creation of an HTML template from a BMS map. For details, see "Compiling a Member" on page 33.
- Updates and corrections have been included that result from documentation APARs that have arisen since the last time this manual was published.

Changes for Second Edition (June 1999)

The second edition of this manual, that was published with VSE/ESA Version 2 Release 4, contained these changes:

- New translation tables for file transfer were added.
- The PC-DOS Workstation File Transfer dialog was removed.
- Year 2000 enhancements were included.

Part 1. Application Development Using VSE/ESA's Interactive Interface

The Interactive Interface of VSE/ESA provides support to help you do program development. This part starts with some introductory information about using VSE/ESA's Interactive Interface for these tasks. In addition, it has information on functions that you can use to communicate with other users and with the system.

This part is structured as follows:

- The VSE/ESA Interactive Interface
- Communicating with the system and other users.
- Using program development functions.

Please note that the *Personal Computer Move Utilities* dialog is described in Part 2.

Chapter 1. The VSE/ESA Interactive Interface

This chapter describes how you use basic functions and components of the VSE/ESA Interactive Interface. It presents the following topics:

- Getting to know the VSE/ESA Interactive Interface.
- The VSE/ESA online panel.
- Signing on.
- Panel types.
- Fast path.
- Synonym function.
- Using program function (PF) keys.
- Interpreting messages.

Getting to Know the VSE/ESA Interactive Interface

The VSE/ESA Interactive Interface allows you to use the facilities of VSE/ESA and its component licensed programs. You select the task you want to perform from *selection panels*. A *dialog* requests input from you to complete the specific task.

Some dialogs perform interactively and display requested information on the terminal screen. Some dialogs create jobs which you submit to the system to complete the task.

Access the appropriate dialog by entering the number of each selection on each panel in the panel hierarchy.

The VSE/ESA Function Selection Panel

This manual does not show the *VSE/ESA Function Selection* panel as it is delivered to you. Instead, this manual uses a customized Selection panel as shown in Figure 2 on page 8. This is because this manual describes only the limited set of functions (accessible through selections 2 - 7). *Your* function selection panel may look different. In this case, the numbering of the selections may be different.

For an overview of the selection panel hierarchy of the individual selections, refer to the IBM manual *VSE/ESA Administration*. It shows the selection panel hierarchy (as it is delivered to you) in the foldout at the back of the book.

To help you access the dialog, this book uses *sequences* which show the selection panel name and selection number you must enter to reach the dialog.

For example, in the *VSE/ESA Function Selection* panel, you can select:
3 (Program Development)
In the *Program Development* panel, select:
1 (Program Development Library)

```
Sample
Fast Path:  31
```

```
Synonym
Default:    ___    Yours:  ICCF
```

The left box under the dialog sequence contains the *Fast Path* for this selection. This facility allows you to go directly to a dialog without going through the entire panel hierarchy to reach the dialog. To use fast path, you enter all the numbers of the selections you would enter on the individual panels in the hierarchy on one single selection panel. For more information on *Fast Path*, refer to “Fast Path” on page 10.

To the right of the fast path box you can find another box into which you may enter a synonym for this selection. In addition to the fast path, you can use a synonym for a certain dialog selection. For the above example, instead of entering the fast path 31, you could also enter *ICCF*. This would invoke the *Program Development Library* dialog.

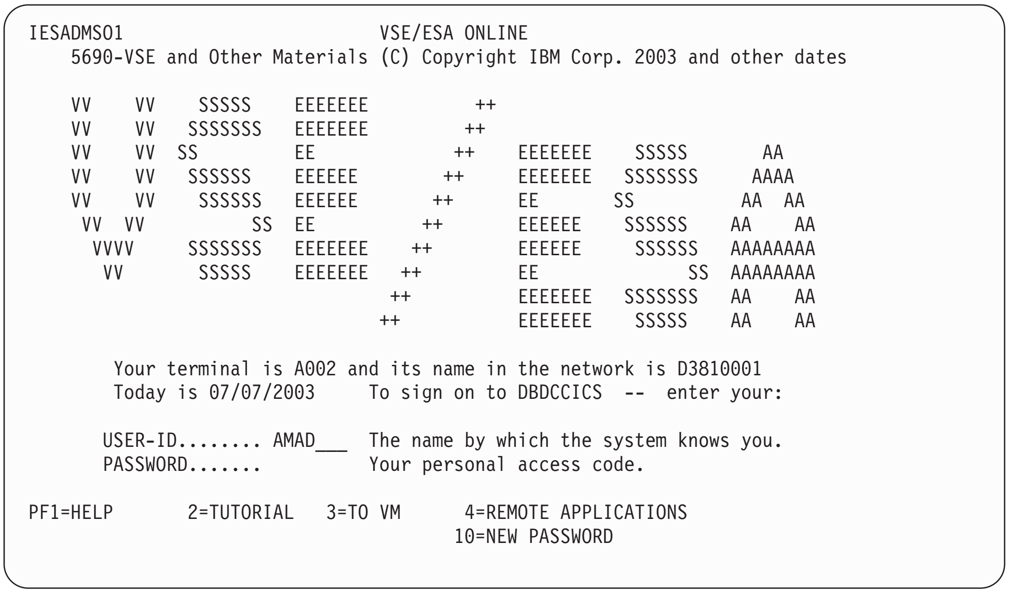
Remember that this book assumes that you use the customized *VSE/ESA Function Selection* panel shown in Figure 2 on page 8. All fast paths shown in this book refer to this panel and may be totally different from the function selection panel created for you.

For further details on the synonym function, refer to “Synonym Function” on page 11.

After you access a dialog, you enter data in specific fields on the panels. Sometimes, you use Program Function (PF) keys. For more information on Program Function keys, refer to “Using Program Function (PF) Keys” on page 14.

The VSE/ESA Online Panel

After you have powered on your terminal, the system will display the *VSE/ESA Online* panel (see Figure 1).



```
IESADMSO1                        VSE/ESA ONLINE
    5690-VSE and Other Materials (C) Copyright IBM Corp. 2003 and other dates

    VV    VV    SSSSS    EEEEEEE          ++
    VV    VV   SSSSSSS   EEEEEEE         ++
    VV    VV  SS         EE             ++    EEEEEEE    SSSSS       AA
    VV    VV   SSSSSS    EEEEEE        ++     EEEEEEE   SSSSSSS     AAAA
    VV    VV    SSSSSS   EEEEEE       ++      EE       SS          AA  AA
     VV  VV          SS  EE          ++       EEEEEE    SSSSSS    AA    AA
      VVVV     SSSSSSS   EEEEEEE    ++        EEEEEE     SSSSSS   AAAAAAAA
       VV       SSSSS    EEEEEEE   ++         EE              SS  AAAAAAAA
                                  ++          EEEEEEE   SSSSSSS   AA    AA
                                 ++           EEEEEEE    SSSSS    AA    AA

        Your terminal is A002 and its name in the network is D3810001
        Today is 07/07/2003     To sign on to DBDCCICS  --  enter your:

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

PF1=HELP       2=TUTORIAL   3=TO VM      4=REMOTE APPLICATIONS
                                        10=NEW PASSWORD
```

Figure 1. VSE/ESA Online Panel

Once the *VSE/ESA Online* panel is on your screen, you may do one of the following, depending on the user profile which your system administrator has set up for you:

- Sign on to the Interactive Interface as described in "Signing on" on page 6.
- Use the first use online tutorial which gives you information about screen symbols, cursor movement keys, special keys and other program function keys.
- Reset the VM terminal allowing you an easy return from VSE/ESA to VM. Note that *PF3=TO VM* is not displayed on the panel if VSE/ESA is not installed under VM.
- Press PF4 to enter the VTAM Remote Applications panel. From there you may enter other applications controlled by your system or by a remote system connected to yours.
- Change your password.

Note: The "VSE/ESA Online" panel may not display the logo for the VSE System Package, *VSE/ESA*, if your system administrator has changed it. For further information, refer to the IBM manual *VSE/ESA Administration*.

Signing on

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 a password. The system administrator creates both for you.

The user ID (4 - 8 characters long and defined by your administrator) identifies you to the system. The password (3 - 8 characters)is a confidential code associated with the user ID. You sign on to the Interactive Interface from the *VSE/ESA Online* panel. An example of the panel is shown in Figure 1 on page 5.

Type in your user ID and password and then press ENTER. The password is **not** displayed on the panel. The system checks the user ID and password. If they are correct, it accesses the selection panel or application which is defined for your user ID.

If you get a message informing you that your user ID or password is incorrect, type in the information again. You may have made a mistake the first time. If it does not work, contact the person responsible for defining user profiles. Pressing PF1 will give you additional information about the error.

If sign-on is correct, the system displays the *VSE/ESA Function Selection* panel, a user selection panel, or an application panel.

If a limit for sign-on attempts has been set via the IESELOGO skeleton and the limit is reached, a message such as the following is issued:

```
USERID 'JONES' WAS REVOKED.
CONTACT YOUR SYSTEM ADMINISTRATOR.
```

Changing Your Password

Password Expiration

User IDs should be defined with an expiration date for the password. If your password expires in seven days or less and you sign-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 your password expires.

When your password has already expired, you have to specify a new one during sign-on. Ask your system administrator for assistance. You can also change your password from the *VSE/ESA Online* panel as described below.

Note: Your password may be valid from 1 to 365 days, depending on how your system administrator has defined your user profile.

Changing the Password for Your User ID

You can change the password yourself from the *VSE/ESA Online* panel. Follow the instructions below.

1. If you are not already signed off, sign off to display the *VSE/ESA Online* panel.
2. On the *VSE/ESA Online* panel, type in your user ID. Press **PF10**. Do **not** press ENTER.
3. On the *Sign-on with new password* panel, enter a new password. Note that you may not reuse a password that is one of the last 12 passwords used for your user ID. The password must be 3 to 8 alphanumeric characters or @, #, or $. It may not contain any blanks.

   Enter your new password in the first field. Then, if your password is less than 8 characters long, press the carriage return key. Type the new password in the second field again.
4. Enter your current password. Then press ENTER.
5. If the entries are correct, the system updates your password and displays the *VSE/ESA Function Selection* panel, a user selection panel, or an application panel.

   Your old password is no longer valid. If you sign off and then sign on again, you must use your new password.

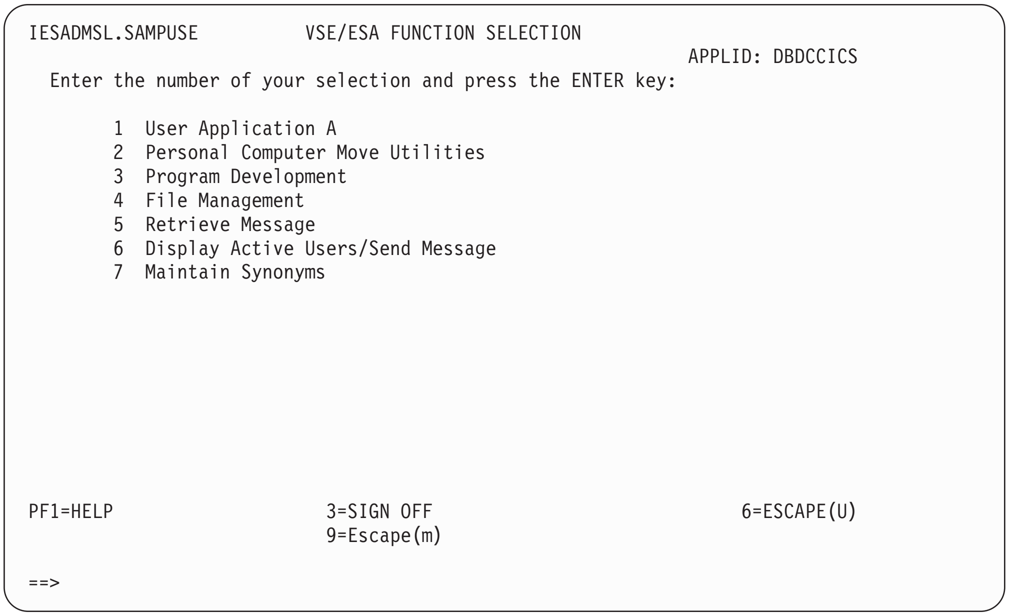
Panel Types

After you have signed on you may access

- the VSE/ESA panel hierarchy via the *Function Selection* panel.
- directly your application.
- your application via a selection panel as shown in Figure 2.

Note: It is assumed that your system administrator has customized the VSE/ESA system and has created the function selection panel shown in Figure 2. Your function selection panel may look different. In this case, the numbering of the selections is also different. The functions accessible through selections 2 - 7 are described in this manual.

Below you find an example of a user-defined function selection panel:



```
IESADMSL.SAMPUSE         VSE/ESA FUNCTION SELECTION
                                                          APPLID: DBDCCICS
  Enter the number of your selection and press the ENTER key:

        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










PF1=HELP                   3=SIGN OFF                             6=ESCAPE(U)
                           9=Escape(m)

==>
```

Figure 2. Example of a Customized Selection Panel

The IBM manual *VSE/ESA Administration* uses this panel as an example to show how to tailor panels according to your needs. For an overview of the selection panel hierarchy of the individual selections, refer to the IBM manual *VSE/ESA Administration*. It shows the selection panel hierarchy (as it is delivered to you) in the foldout at the back of the book.

The Interactive Interface uses several types of panels:

- Selection panels (as shown in Figure 2)
- Data entry panels
- Function lists
- Help panels

These panel types are described in the following section.

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

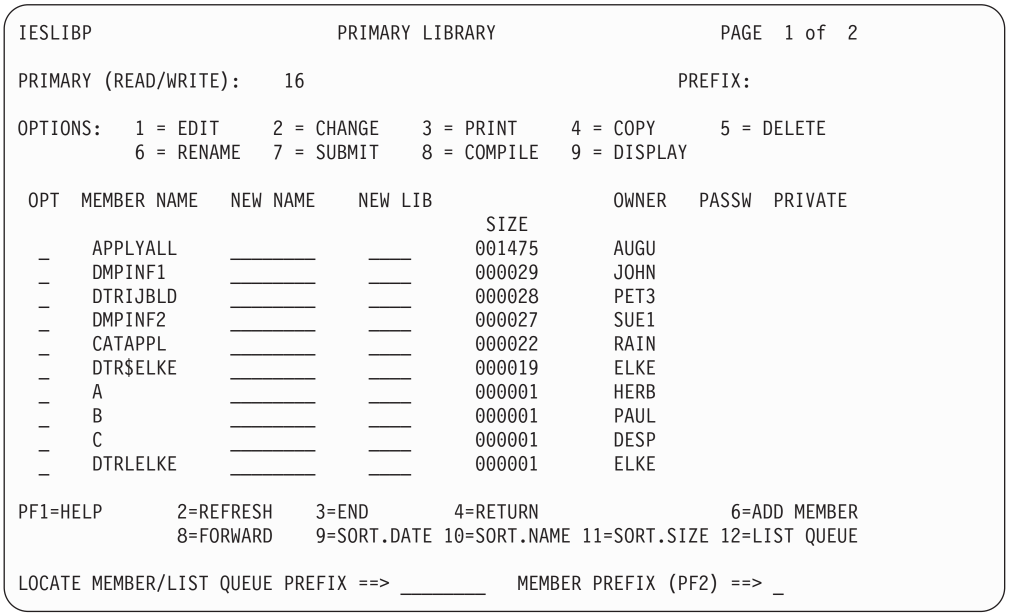
The dialogs use data entry panels to obtain input about the task you are performing. You enter information in particular fields on the panel. For example, if you want to change the synonym for a certain selection path, you must enter the new name in the field provided.

Function Lists

A Function List (FULIST) displays a list of items which you can process. The items could be:
- VSE/ICCF library members
- VSE/VSAM files
- VSE/POWER queue entries
- Files in the Host Transfer File

It also displays a list of options you use to process these items. Some options could be:
- Copy
- Delete
- Rename
- Change
- Print
- Submit
- Compile
- Display

Figure 3 on page 10 shows you an example of a FULIST panel. In the figure, 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. For example, if you want to edit the file CSERV, you must type in **1** in the option column next to the member name CSERV and then press ENTER.



```
IESLIBP                       PRIMARY LIBRARY                     PAGE  1 of  2

PRIMARY (READ/WRITE):    16                                   PREFIX:

OPTIONS:   1 = EDIT     2 = CHANGE    3 = PRINT     4 = COPY      5 = DELETE
           6 = RENAME   7 = SUBMIT    8 = COMPILE   9 = DISPLAY

 OPT  MEMBER NAME   NEW NAME    NEW LIB                 OWNER   PASSW  PRIVATE
                                            SIZE
  _    APPLYALL     ________     ____      001475       AUGU
  _    DMPINF1      ________     ____      000029       JOHN
  _    DTRIJBLD     ________     ____      000028       PET3
  _    DMPINF2      ________     ____      000027       SUE1
  _    CATAPPL      ________     ____      000022       RAIN
  _    DTR$ELKE     ________     ____      000019       ELKE
  _    A            ________     ____      000001       HERB
  _    B            ________     ____      000001       PAUL
  _    C            ________     ____      000001       DESP
  _    DTRLELKE     ________     ____      000001       ELKE

PF1=HELP       2=REFRESH    3=END        4=RETURN                  6=ADD MEMBER
               8=FORWARD    9=SORT.DATE 10=SORT.NAME 11=SORT.SIZE 12=LIST QUEUE

LOCATE MEMBER/LIST QUEUE PREFIX ==> ________   MEMBER PREFIX (PF2) ==> _
```

Figure 3. Example of a FULIST Panel

Application Sign-on Panel

This panel type varies greatly, depending on the application used. Often applications have their own logo, whereas others prefer, for example, a selection panel as the first panel.

HELP Panels

From 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 example, if you have entered incorrect data, the message informs you of the error. If there is a HELP panel available for this particular message and you press PF1, 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.

Fast Path

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 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 fast path works.

After signing on the system displays the *VSE/ESA Function Selection* panel shown in Figure 2 on page 8. Suppose you want to access the *Program Development Library* dialog. To reach the dialog by going through the entire hierarchy, you would do the following:

In the *VSE/ESA Function Selection* panel, select:
3 (Program Development)
In the *Program Development* panel, select:
1 (Program Development Library)

The first panel of the dialog is displayed.

You can use the fast path method to go directly to the *Program Development Library* dialog. In the *VSE/ESA Function Selection* panel, enter:

31

The first panel of the dialog is displayed.

Keep in mind that your system administrator may define a different numbering for your selections. This book uses the selection numbers that refer to the sample panel shown in Figure 2 on page 8.

Note that if you have used the fast path facility or the synonym function to enter a dialog, PF3 will exit as far as the fast path goes. For example, you selected the fast path *31* (Program Development Library) from the *VSE/ESA Function Selection* panel. The *Program Development Library* panel is displayed. If you now press PF3 (End) to leave the dialog, the system will not go just one level back in the hierarchy but to the original panel from which the fast path was issued, in the above example, to the *VSE/ESA Function Selection* panel.

Synonym Function

This function gives you the possibility to equate a selection path to a private synonym. The synonym must consist of 1-8 alphameric characters (the characters $, #, and & are allowed). The first character may not be a number. For example, instead of specifying the fast path **31** to access your primary library, you can also enter a word which you may remember better, for example, **library** into the command line of the corresponding selection panel.

There are default synonyms shipped with the system. Ask your system administrator for the default synonyms. You can also define other synonyms if you prefer different ones. The *Maintain Synonyms* dialog is offered to you via a selection panel. This function lets you change the default synonyms or enter new ones. For more information on maintaining synonyms or adding new ones, refer to “Maintaining Synonyms” below.

Under the dialog sequences in this manual you will find a ’synonym box’ next to the box into which you can enter your fast path. You can use the default synonyms or your own synonyms and enter them in the ’Synonym box’ next to fast path for later reference.

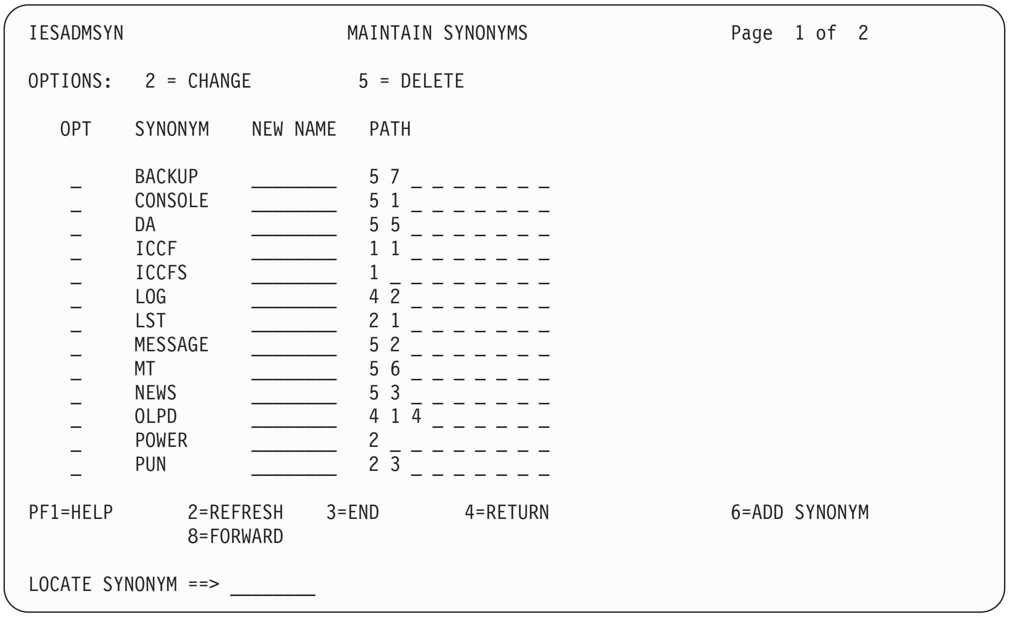
Maintaining Synonyms

If you prefer other synonyms than those offered to you by your system administrator or if you want to add or delete any synonyms, you must use the *Maintain Synonyms* dialog.

To access the dialog, start with the *VSE/ESA Function Selection* panel shown in Figure 2 on page 8 and select:
7 (Maintain Synonyms)

Sample Fast Path: **7**	Synonym Default: —— Yours:

If your system administrator has defined any synonyms for you, you will get a list of the available synonyms in alphabetical order as shown in the example in Figure 4.



```
IESADMSYN                    MAINTAIN SYNONYMS                    Page  1 of  2

OPTIONS:   2 = CHANGE         5 = DELETE

   OPT    SYNONYM    NEW NAME   PATH

    _     BACKUP     ________   5 7 _ _ _ _ _ _ _
    _     CONSOLE    ________   5 1 _ _ _ _ _ _ _
    _     DA         ________   5 5 _ _ _ _ _ _ _
    _     ICCF       ________   1 1 _ _ _ _ _ _ _
    _     ICCFS      ________   1 _ _ _ _ _ _ _ _
    _     LOG        ________   4 2 _ _ _ _ _ _ _
    _     LST        ________   2 1 _ _ _ _ _ _ _
    _     MESSAGE    ________   5 2 _ _ _ _ _ _ _
    _     MT         ________   5 6 _ _ _ _ _ _ _
    _     NEWS       ________   5 3 _ _ _ _ _ _ _
    _     OLPD       ________   4 1 4 _ _ _ _ _ _
    _     POWER      ________   2 _ _ _ _ _ _ _ _
    _     PUN        ________   2 3 _ _ _ _ _ _ _

PF1=HELP       2=REFRESH    3=END       4=RETURN                 6=ADD SYNONYM
               8=FORWARD

LOCATE SYNONYM ==> ________
```

Figure 4. FULIST of Available Synonyms

Changing and Deleting Synonyms

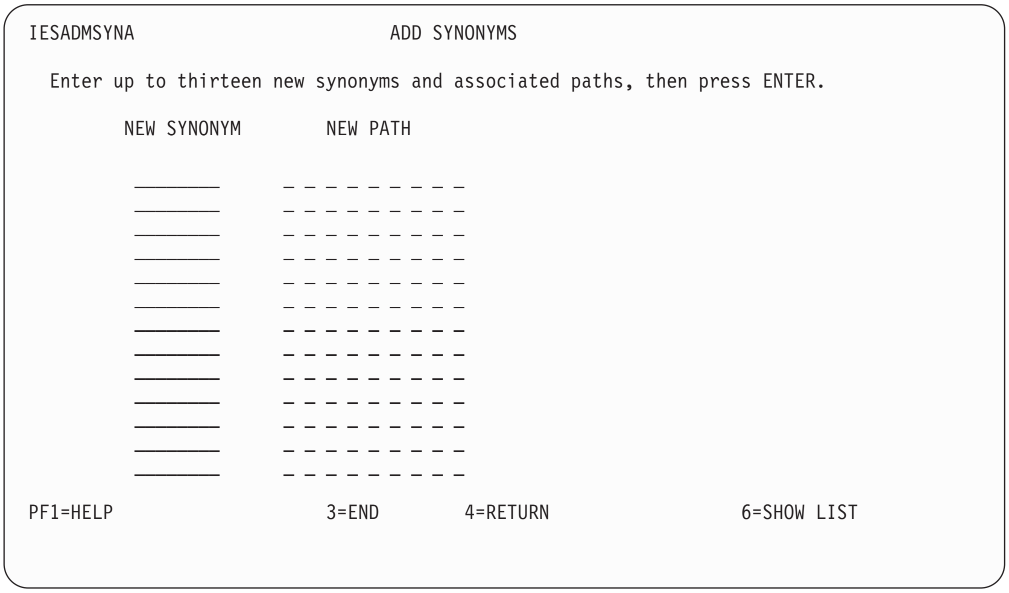
To change any of the listed synonyms, enter **2** in the option column next to the synonym(s) you want to change. Then type in the desired synonym(s) in the *New Name* column. Then press ENTER.

If you want to locate a particular synonym, type in the name into the command line at the bottom of the panel and then press ENTER.

If you decide to delete a synonym from the list, type in **5** in the option column next to the synonym you want to delete. Then press ENTER.

Adding Synonyms

If you want to add one or more synonyms, press **PF6** on the panel shown in Figure 4. The *Add New Synonyms* panel will be displayed. A sample of this panel is shown in Figure 5 on page 13.



```
IESADMSYNA                        ADD SYNONYMS

  Enter up to thirteen new synonyms and associated paths, then press ENTER.

         NEW SYNONYM       NEW PATH

          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _
          ________     _ _ _ _ _ _ _ _ _

PF1=HELP                    3=END        4=RETURN                  6=SHOW LIST
```

Figure 5. Add New Synonyms Panel

Type in the new synonym definitions in the fields offered on the panel. Then press ENTER. Pressing PF6 will take you back to the list panel shown in Figure 4 on page 12 to show you the updated list.

You must press ENTER before you press PF6, otherwise all input will be lost.

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. To invoke the function for a particular PF key, you simply press the PF key on your terminal. You do not have to press ENTER.

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/ESA environment. This may not be true for some applications which may use all 24 PF keys.

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 information explains the PF keys which the Interactive Interface uses.

PF1 Throughout the Interactive Interface, PF1 is **always** used to invoke the HELP function. This function may consist of one or more HELP panels. The HELP explains the panel that is currently displayed (the panel from which you pressed PF1).

In some cases, other types of HELP panels are displayed. For example, if there is a system message on the panel, PF1 will explain that message if online help is available.

PF2 Throughout the Interactive Interface, PF2 is **always** used as ’REDISPLAY’ or ’REFRESH’ if this function is offered on the panel.

This function deletes input, shows the original panel, updates a list, or shows a new list.

PF3 Throughout the Interactive Interface, PF3 is **always** defined as ’END’ if this function is offered on the panel.

This means that you end a dialog and/or go back one level in the panel hierarchy. If you have used the fast path facility or the synonym function to enter a dialog, PF3 will exit as far as the fast path goes. Any data you have typed in during the current dialog will be **ignored** as soon as you press PF3.

On the *VSE/ESA Online* panel (see Figure 1 on page 5), PF3 resets the VM terminal, thus allowing VM/VSE users an easy return from VSE/ESA to VM. However, *PF3=TO VM* is **only** shown if you are running VSE/ESA under VM.

On the *VSE/ESA Function Selection* panel (see Figure 2 on page 8), PF3 is used to sign off.

If you are working in native CICS, PF3 takes you back to the host mode.

PF4 Throughout the Interactive Interface, PF4 is **always** defined as ’RETURN’ if this function is offered on the panel.

This returns you to the first selection panel in the hierarchy (*VSE/ESA Function Selection*). Any data you typed in will be **ignored** as soon as you press PF4.

From the *VSE/ESA Online* panel, pressing PF4 allows you to enter the *VTAM Remote Applications* panel. From there you may enter other applications controlled by your system or by a remote system connected to yours.

PF5 This invokes various functions depending on the task you are performing.

Whenever an update or process function is offered, PF5 is used for it. This means that the function that you selected is processed. For example, PF5 may dynamically update a table.

For other uses of PF5, review the HELP panel from the appropriate dialog.

PF6 This invokes various functions depending on the task you are performing.

When working with the Move Utilities, this PF key is used to set the host session in a mode to accept SEND/RECEIVE commands from the PC.

With PF6=Escape(U) you turn on uppercase translation and escape to CICS. This function is displayed on selection panels only if you have the authorization for it defined in your user profile. If you press PF6, all data which you type in will be translated to uppercase by CICS. This may be useful for certain applications, for example, if you want to create a source program.

You can return to the Interactive Interface from CICS by pressing either PF1 or PF3.

Note: The escape facility may also be allowed from the *VSE/ESA Online* panel. In this case, your system administrator must implement the "escape facility" by modifying the Interactive Interface sign-on panel (refer to *VSE/ESA Administration*). You then can escape before signing on.

For other uses of PF6, review the HELP panel from the appropriate dialog.

PF7 Throughout the Interactive Interface, PF7 is **always** used to move backwards one screen at a time if this function is offered on the panel.

PF8 Throughout the Interactive Interface, PF8 is **always** used to move forwards one screen at a time if this function is offered on the panel.

PF9 This invokes various functions depending on the task you are performing.

With PF9=Escape(m) you turn off uppercase translation and escape to CICS. This function is displayed on selection panels only if you have the authorization for it defined in your user profile. If you press PF9, all data which you type in will *not* be translated to uppercase by CICS.

You can return to the Interactive Interface from CICS by pressing either PF1 or PF3.

If there is a TOP function offered on the panel, PF9 will always be used for it. Pressing PF9 takes you to the top of the list.

For other uses of PF9, review the HELP panel from the appropriate dialog.

PF10 This invokes various functions depending on the task you are performing.

PF10 on the *VSE/ESA Online* panel is used to change the password for your user ID. “Changing Your Password” on page 6 has information on changing passwords.

If there is a SHIFT function offered on the panel, PF10 will always be used for it. Pressing PF10 shifts to the left of the list.

For other uses of PF10, review the HELP panel from the appropriate dialog.

PF11 This invokes various functions depending on the task you are performing.

If there is a SHIFT function offered on the panel, PF11 will always be used for it. Pressing PF11 shifts to the right of the list.

Note that on a PC keyboard with **only ten (P)F keys**, PF11 is simulated by pressing PF9 and the shift key at the same time.

On a normal display with only ten PF keys offered on the keyboard, PF11 can be simulated by typing in *PF11* in the command line and then pressing ENTER. The command line is located at the bottom of the panel.

PF12 This invokes various functions depending on the task you are performing.

If there is a BOTTOM function offered on the panel, PF12 will always be used for it.

Note that on a PC keyboard with **only ten (P)F keys**, PF12 is simulated by pressing PF10 and the shift key at the same time.

On a normal display with only ten PF keys offered on the keyboard, PF12 can be simulated by typing in *PF12* in the command line and then pressing ENTER. The command line is located at the bottom of the panel.

Other Keys

PA2 The PA2 key works on the Interactive Interface panels like the VSE/ICCF command */CANCEL*.

CLEAR

Within the panels of the Interactive Interface, the CLEAR key may function differently from what you may be used to. When you press this key, the system will not display a refreshed version of the screen on which you pressed CLEAR. Instead, the system will take you back one level in the hierarchy. This is the same function as PF3.

Interpreting Messages

If you are using the VSE/ESA Interactive Interface, you may get error messages because of wrong input. Usually they have no number, they just consist of plain text. For example, if you pressed a wrong PF key, the following message will be displayed

```
USE ONLY THE ENTER KEY OR ONE OF THE KEYS LISTED ABOVE
```

For these messages you may press the PF1 key to display additional text describing the error. In exceptional error situations you may get a message telling you to display a certain VSE/ICCF member for further error information. If you receive such a message, take the action which you are asked for in the message or copy the message and talk to your system administrator.

Note that there are also information messages which do not necessarily require a user action, for example

```
There is at least one message waiting for you to retrieve.
```

You may now decide whether to review the message(s) or you may continue normal operation.

Chapter 2. Communicating with the System and Other Users

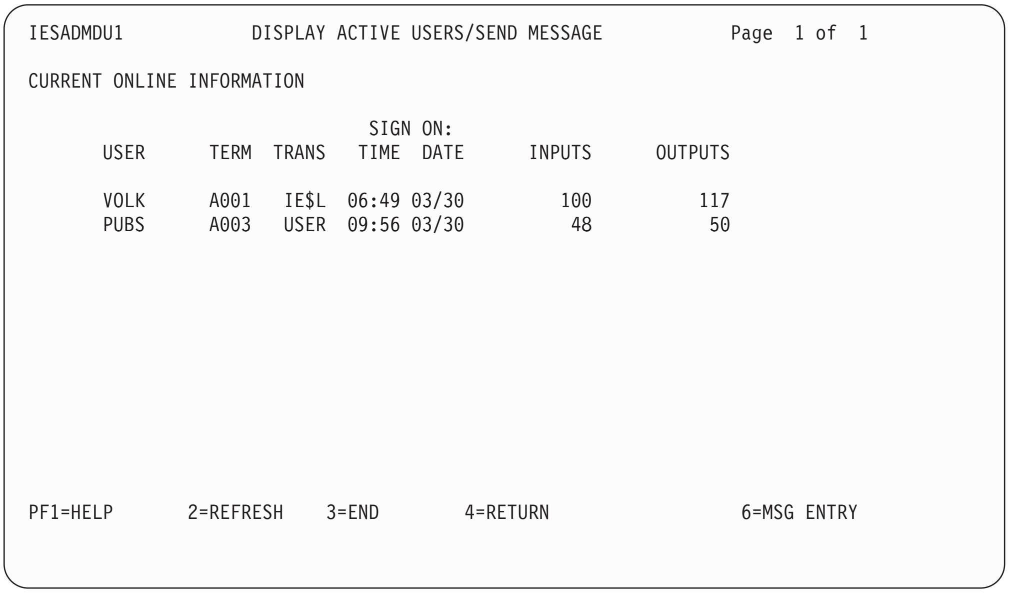
The Interactive Interface of VSE/ESA enables you to communicate with the system and other users. It provides dialogs which give you a comfortable way for performing those tasks.

This chapter describes:
- Displaying active users.
- Sending a message to another user.
- Receiving a message from another user.
- Retrieving a system message.

Displaying Active Users

The *Display Active Users/Send Message* dialog displays the users signed on to VSE/ESA. From the dialog, you can send a message to selected users or to all users who are signed on. To access the dialog from the *VSE/ESA Function Selection* panel shown in Figure 2 on page 8, select **6** (Display Active Users/Send Message).

A panel displays the users who are signed on at that time.



```
IESADMDU1           DISPLAY ACTIVE USERS/SEND MESSAGE           Page  1 of  1

CURRENT ONLINE INFORMATION

                                SIGN ON:
       USER      TERM  TRANS   TIME  DATE      INPUTS      OUTPUTS

       VOLK      A001   IE$L  06:49 03/30         100          117
       PUBS      A003   USER  09:56 03/30          48           50










PF1=HELP      2=REFRESH    3=END       4=RETURN                  6=MSG ENTRY
```

Figure 6. Display Active Users/Send Message Panel

The following information is displayed for each user:

USER This is the user ID of the user signed on. In case a user is **not** signed on correctly, eight question marks (?) are displayed.

TERM This is the terminal ID, which is a unique identifier for each terminal connected to the online system.

TRANS The name of the transaction the user is currently running.

SIGN ON The TIME and DATE of when the user has signed on to the system.

INPUTS The number of input messages received from the user's terminal.

OUTPUTS The number of output messages sent to the user's terminal.

In addition the following options may be offered at the top of the panel:

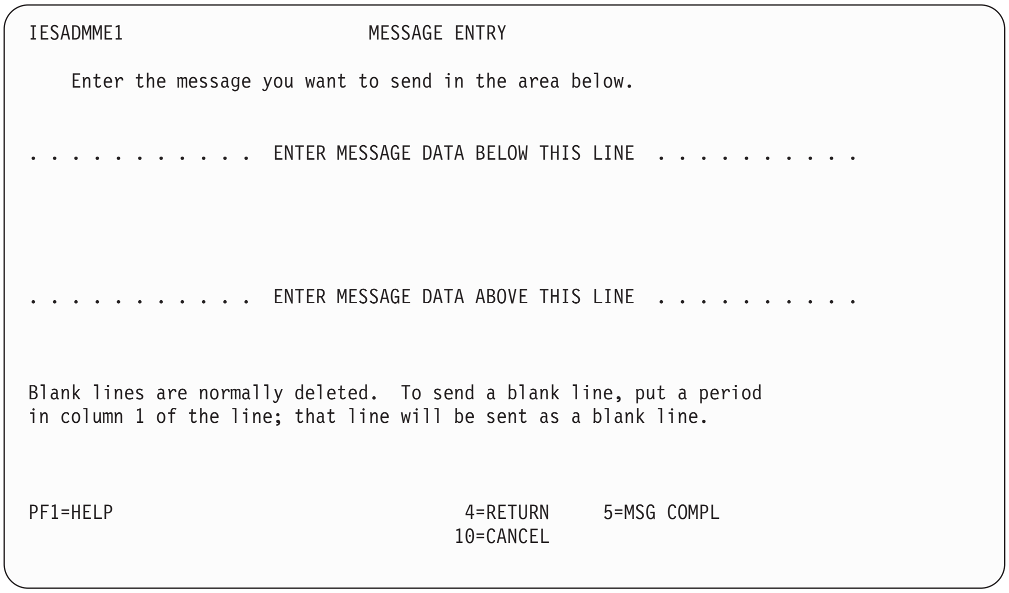
2=SIGN OFF A USER
Displayed only if you have administrator authority. Enter this option next to the user whose terminal is to be disconnected (/DISCONN) or signed off. Note that this function cannot perform properly in case that the selected terminal is out of service (status code 1).

3=SEND TO USER
Displayed only if you have prepared a message to be sent (via PF6=MSG ENTRY). Enter this option next to the user whom you want to send a message.

You can press **PF2** to refresh the panel and display the most current information about users who are signed on.

Sending a Message to Another User

To send a message to one or more users select **6** (Display Active Users/Send Message) in the *VSE/ESA Function Selection* panel. In the *Display Active Users/Send Message* panel, press **PF6** (Message Entry). On the *Message Entry* panel shown below, type in your message.



```
IESADMME1                      MESSAGE ENTRY

    Enter the message you want to send in the area below.

. . . . . . . . . . .  ENTER MESSAGE DATA BELOW THIS LINE  . . . . . . . . . .




. . . . . . . . . . .  ENTER MESSAGE DATA ABOVE THIS LINE  . . . . . . . . . .



Blank lines are normally deleted.  To send a blank line, put a period
in column 1 of the line; that line will be sent as a blank line.



PF1=HELP                                 4=RETURN     5=MSG COMPL
                                        10=CANCEL
```

Figure 7. Message Entry Panel

Press **PF5** to tell the system you have finished typing the message.

The dialog redisplays the *Display Active Users/Send Message* panel. It now displays an additional column (OPT) to the left of the user ID(s), option 3 (Send to user), and the PF keys PF9 and PF10.

Select the user(s) to receive the message. Do one of the following:
If you want to send the message to **all users** signed on just press **PF9**.

If you want to send the message **only to selected users**, enter **3** in the OPT column to the left of the user ID(s) who should receive the message. Then press **PF10**.

When your message is sent, the dialog displays the following message:

```
MESSAGE NUMBER # WAS SCHEDULED FOR y DESTINATION(S)
```

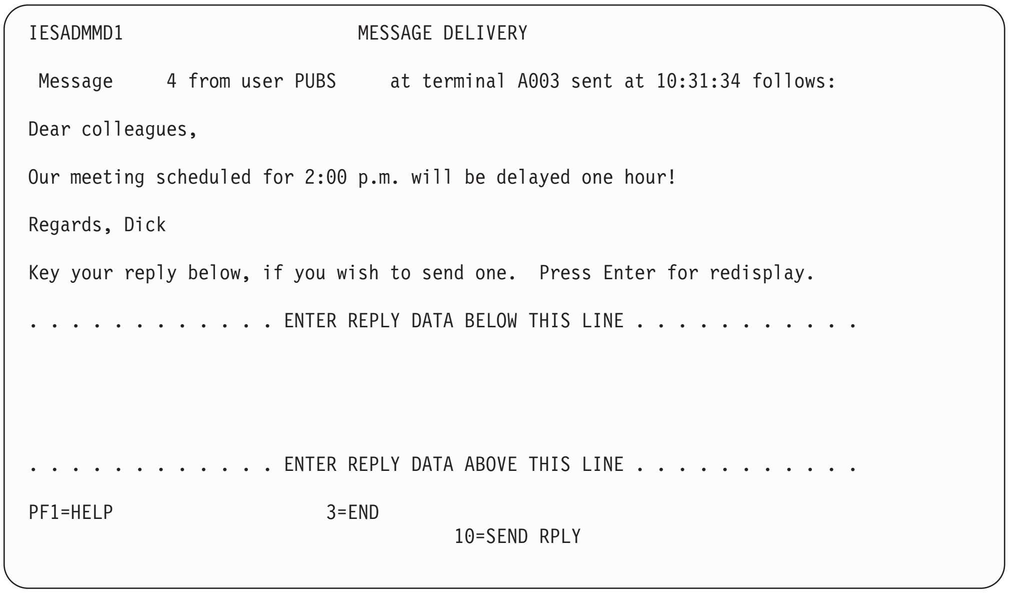
where

the number of the message you are sending during the current session (for example, *2* if it is the second message).

y the number of the users you are sending the message to (for example, *3* if you are sending the message to three users).

Receiving a Message from Another User

Sometimes, another user who is signed on to the Interactive Interface will send you a message. If the message is sent by the *Display Active Users/Send Message* dialog, the system interrupts your current task and displays the *Message Delivery* panel on your terminal. The panel shows the message and the user ID of the person who sent the message. A sample of the panel is shown in Figure 8.



```
IESADMMD1                      MESSAGE DELIVERY

 Message     4 from user PUBS     at terminal A003 sent at 10:31:34 follows:

Dear colleagues,

Our meeting scheduled for 2:00 p.m. will be delayed one hour!

Regards, Dick

Key your reply below, if you wish to send one.  Press Enter for redisplay.

. . . . . . . . . . . . ENTER REPLY DATA BELOW THIS LINE . . . . . . . . . . .




. . . . . . . . . . . . ENTER REPLY DATA ABOVE THIS LINE . . . . . . . . . . .

PF1=HELP                    3=END
                                   10=SEND RPLY
```

Figure 8. Message Delivery Panel

You can send a reply to the user. Type in your **reply** on the panel and then press **PF10**. The system sends the reply to the user.

If you do not want to send a reply, press **PF3** (END) and the system returns you to the task you were doing.

Retrieving a System Message

The *Retrieve Message* dialog displays messages that the **system** has sent you. Such messages can be:

- System error information.
- System informational messages.

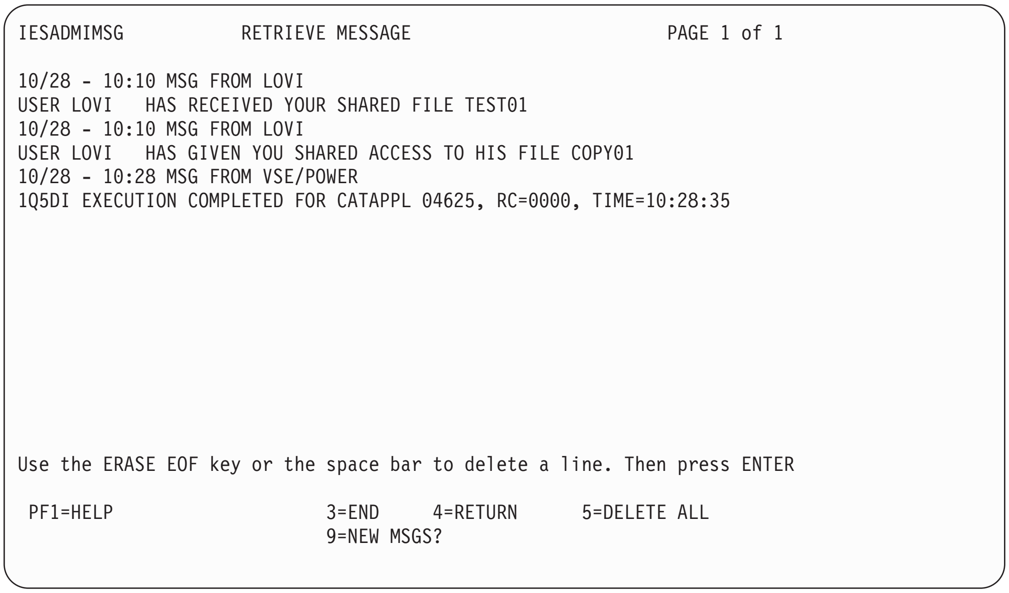
- Notification messages from Workstation File Transfer Support.
- Messages sent by VSE/ICCF users using the /SEND command.
- Notification messages from applications packages.

When you sign on, the system will issue the following message if there are any system messages for you:

```
There is at least one message waiting for you to retrieve.
```

To access the *Retrieve Message* dialog,select **5** (Retrieve Message) in the *VSE/ESA Function Selection* panel shown in Figure 2 on page 8.

The *Retrieve Message* panel displays the messages. An example of this panel is shown below.



```
IESADMIMSG          RETRIEVE MESSAGE                        PAGE 1 of 1

10/28 - 10:10 MSG FROM LOVI
USER LOVI   HAS RECEIVED YOUR SHARED FILE TEST01
10/28 - 10:10 MSG FROM LOVI
USER LOVI   HAS GIVEN YOU SHARED ACCESS TO HIS FILE COPY01
10/28 - 10:28 MSG FROM VSE/POWER
1Q5DI EXECUTION COMPLETED FOR CATAPPL 04625, RC=0000, TIME=10:28:35











Use the ERASE EOF key or the space bar to delete a line. Then press ENTER

 PF1=HELP                    3=END     4=RETURN      5=DELETE ALL
                             9=NEW MSGS?
```

The messages are held in a file. When you access the dialog, your messages are queued for display on your terminal screen. Use the appropriate **PF keys** to scroll forward or backward. The PF keys are described at the bottom of the panel.

You can delete individual message lines. Use the **ERASE EOF** key, or blank out the line using the space bar and then press **ENTER**.

If you want to delete all messages in the queue, press **PF5**.

While you review the panel, it is possible that the system sends you new messages. You can update the panel display to see any new messages by pressing **PF9**. If the system has not sent you any new messages, it displays the following:

```
NO NEW MESSAGES WERE RETURNED
```

You can end the dialog by pressing **PF3**.

Notes:

1. Please be aware that messages are deleted automatically when you leave the dialog and **sign off**.
2. If you access the dialog and no messages have been sent to you, the system displays the following message:

```
THERE ARE NO MESSAGES AVAILABLE
```

Chapter 3. Using Program Development Functions

Developing a program typically includes the following activities:

- Creating and compiling the source of the program.
- Looking at the list output of the compilation.
- Correcting any programming errors found during compilation.
- Preparing data files for program execution.
- Testing and running a newly compiled program.

The VSE/ESA Interactive Interface supports you in these activities. This chapter describes:

- VSE/ICCF libraries.
- Handling VSE/ICCF library members.
- Accessing the VSE/POWER list queue.
- File management
- Running a newly compiled program.

VSE/ICCF Libraries

When developing a program at the VSE host you need to store your program file in a library. Under VSE/ESA, this normally is the VSE/ICCF library.

The VSE/ICCF library, also known as the *Program Development Library*, is actually a set of libraries that VSE/ICCF manages. These libraries may contain programs and/or other files (for example, text).

The User's Point of View

You can access libraries in two ways:

- For reading and writing.
- For reading only.

For every user who is authorized to access these functions, VSE/ICCF has one default *primary* library. The primary library provides read and write access. In addition, via your user profile the system administrator may have given you one or more secondary libraries, which provide read access only.

When you do program development via the *Program Development Library* dialog, the first panel displayed lets you specify the library you want to work with.

- VSE/ICCF *primary* libraries

  With write access, you have the following options in addition to the ones listed under secondary libraries below:
  - Edit
  - Change
  - Rename
  - Delete
  - Copy
- VSE/ICCF *secondary* libraries

With read access, you have the following options:
- Display
- Print
- Copy to primary
- Submit
- Compile

You may have more than one VSE/ICCF primary or secondary library. Several users may have the same VSE/ICCF primary library.

When working with the *Program Development Library* panel (Figure 9 on page 27), you may select which library to use as primary or secondary. You always can select only one library at a time. Depending on how your system administrator has set up your user profile you may share primary and secondary libraries with one or more users.

VSE/ICCF allows to restrict the access within a library further. For example, a programmer (type 2) profile has the following access for library members:

Write access to members the user owns and to all public members. If the member is password-protected, you must specify the password. For details about "library access rights," refer to the IBM manual *VSE/ICCF Administration and Operation*.

Library members have an owner and can be password-protected. Members, as well as libraries, can be public or private.

The System's Point of View

The above two categories of *primary* and *secondary* reflect the user's point of view. You can look at the set of VSE/ICCF libraries also from the system's point of view. You would then have the categories of private, public and common libraries.

- **Private library**

  A library for user data. Several users may share one private library.
- **Public library**

  All users have read access to public libraries. For example, the libraries 50 - 69 are defined as public libraries.
- **Common library**

  The common library is implicitly accessible by all users. This means, you do not have to create an access to it. It contains data available for execution by several users (for example, macros or utilities).

  Data in the common library are normally available only for read access, but the administrator or other authorized users may update members in the common library.

  It is customary to define library 2 as common library.

Handling VSE/ICCF Library Members

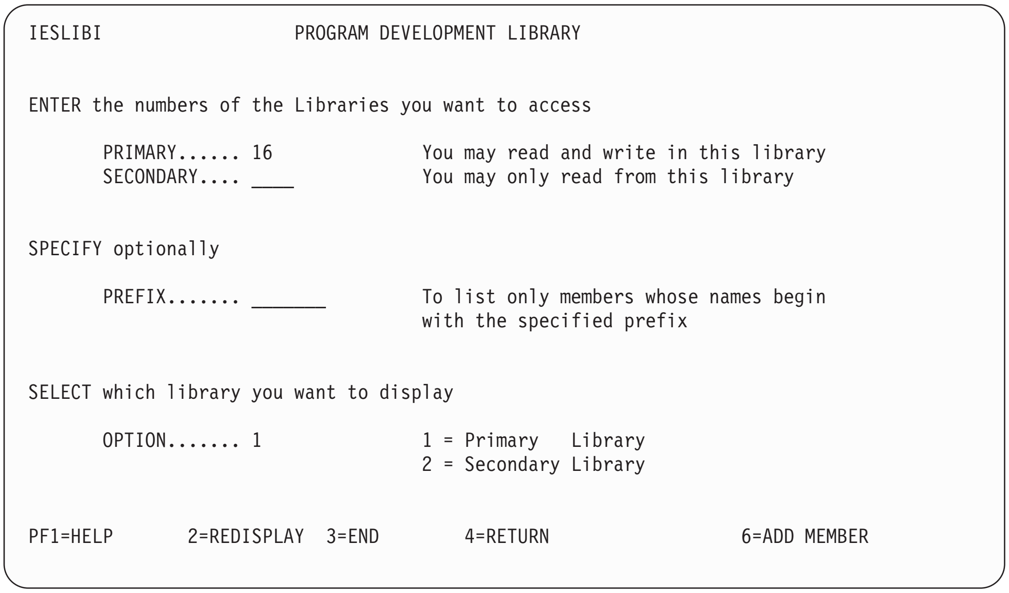
Please note

To be able to access VSE/ICCF libraries under the Interactive Interface, you must be defined with a user ID that is at most 4 characters long.

The *Program Development Library* dialog uses a FULIST to display library members. From the FULIST, you select the member you want to process. You can access the dialog as follows:

In the *VSE/ESA Function Selection* panel shown in Figure 2 on page 8, select:
3 (Program Development)
In the *Program Development* panel, select:
1 (Program Development Library)

Sample Fast Path: **31**	Synonym Default: —— Yours:



```
IESLIBI                  PROGRAM DEVELOPMENT LIBRARY


ENTER the numbers of the Libraries you want to access

       PRIMARY...... 16              You may read and write in this library
       SECONDARY.... ____            You may only read from this library


SPECIFY optionally

       PREFIX....... _______         To list only members whose names begin
                                     with the specified prefix


SELECT which library you want to display

       OPTION....... 1               1 = Primary   Library
                                     2 = Secondary Library


PF1=HELP      2=REDISPLAY  3=END       4=RETURN                6=ADD MEMBER
```

Figure 9. Program Development Library Panel

You can access a library as either a primary or secondary library in your search chain.

The *Program Development Library* panel (see Figure 9) shows your default primary library (as defined in your user profile) in the PRIMARY field. To access your default primary library, you simply have to press ENTER.

To access another primary library for which you are authorized to have write access, enter the **number of the library** in the ’Primary’ field. Then press ENTER.

If the primary library you want to access is empty, the message

```
YOUR PRIMARY LIBRARY IS EMPTY - USE PF6 TO CREATE THE FIRST MEMBER
```

is issued as soon as you press ENTER. In this case, *PF6=ADD MEMBER* is displayed on the panel in Figure 9. You may now create a new member as described under “Adding a New VSE/ICCF Library Member” on page 29.

To access a secondary library, enter the **number of the library** in the ’Secondary’ field and type **2** in the option field. Then press ENTER.

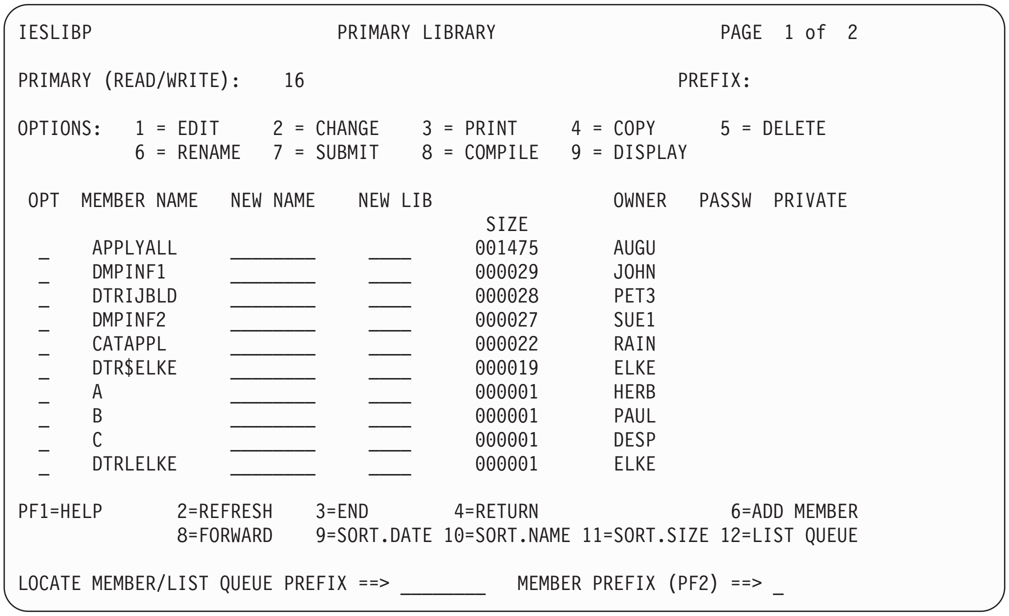
Also, you can display only members that begin with certain characters. Enter the prefix in the PREFIX field. For example, to display the members PRB$DMPQ, PRB$DMPT, PRBTEST, and PRBINIT, you would enter **PRB** in the PREFIX field.

Working With the FULIST

After you have selected the library you want to access, a FULIST displays the library members (for example, Figure 10). The FULIST sorts the members by date in descending order. Members with the same date appear by name in alphabetical order.

Besides the member name, the FULIST shows the following attributes for each member:

- LAST ACCESSED: This shows when the member was last edited.
- OWNER: This is the user ID of the owner of the member. For users with programmer (type 2) profiles, the user ID is displayed only for the user's own members.
- PASSW: If the member is password protected, an asterisk (*) is displayed.
- PRIVATE: If the member has the private attribute, an asterisk (*) is displayed.



```
IESLIBP                       PRIMARY LIBRARY                     PAGE  1 of  2

PRIMARY (READ/WRITE):    16                                    PREFIX:

OPTIONS:   1 = EDIT     2 = CHANGE    3 = PRINT     4 = COPY      5 = DELETE
           6 = RENAME   7 = SUBMIT    8 = COMPILE   9 = DISPLAY

 OPT  MEMBER NAME   NEW NAME    NEW LIB                 OWNER   PASSW  PRIVATE
                                            SIZE
  _    APPLYALL     ________     ____      001475       AUGU
  _    DMPINF1      ________     ____      000029       JOHN
  _    DTRIJBLD     ________     ____      000028       PET3
  _    DMPINF2      ________     ____      000027       SUE1
  _    CATAPPL      ________     ____      000022       RAIN
  _    DTR$ELKE     ________     ____      000019       ELKE
  _    A            ________     ____      000001       HERB
  _    B            ________     ____      000001       PAUL
  _    C            ________     ____      000001       DESP
  _    DTRLELKE     ________     ____      000001       ELKE

PF1=HELP       2=REFRESH    3=END        4=RETURN                  6=ADD MEMBER
               8=FORWARD    9=SORT.DATE 10=SORT.NAME 11=SORT.SIZE 12=LIST QUEUE

LOCATE MEMBER/LIST QUEUE PREFIX ==> ________   MEMBER PREFIX (PF2) ==> _
```

Figure 10. FULIST of Primary Library (sorted by SIZE)

The options you can choose are at the top of the FULIST. Enter the option number in the OPT column to the left of the member(s) you want to process. The options differ for primary and secondary libraries. They are described later in this chapter.

Note: If a member is password protected, you must enter the password to process the member. You can either:

1. Enter the password in the PASSWORD column on the FULIST.
2. Enter the password on the next panel.

The following paragraphs have information about how you can manage the FULIST information displayed. You can manage the FULIST information in various ways. For this, you either:

- Make a direct request via a PF key or

- Enter some information for the system to process.

  How the system processes your entry depends on whether you just press ENTER or whether you press PF2 or PF12.

Sorting the FULIST

Sorting the FULIST Alphabetically: To refresh the FULIST and sort the entries in alphabetical order, press PF10.

You also can enter a prefix in the LOCATE MEMBER field. The dialog displays a page whose first member starts with the prefix. If the FULIST is not sorted alphabetically, you must enter the entire name of the member. You cannot just enter a prefix.

Sorting the FULIST By Date: To refresh the FULIST and sort the entries by date (in descending order), press PF9.

Sorting the FULIST By Size: To refresh the FULIST and sort the entries by size (in descending order), press PF11. The SIZE information is displayed by changing the heading line for LAST ACCESSED into SIZE. Sorting by SIZE may take a few seconds longer than other sort options.

Locating a Particular Member

To locate a particular member, enter the member name in the LOCATE MEMBER field. If the member is found, it is displayed with an * in the OPT column.

If the member is **not** found and the library members are sorted alphabetically, the system scrolls to the page on which the desired library member would be listed had it been included in the list. If the member is not found and the library members are not sorted alphabetically, the same panel is displayed on which the locate command was issued.

Displaying Selected Members Only

To refresh the FULIST and display only entries starting with the given prefix, enter the prefix in the MEMBER PREFIX field **and** press **PF2**.

Displaying Entries of the VSE/POWER List Queue

To display a FULIST of your output for your jobs in the VSE/POWER list queue, press PF12. From there you can return to the *Program Development Library* dialog by pressing **PF3**.

Displaying Selected Entries of the VSE/POWER List Queue: To display entries in the VSE/POWER list queue that start with a given prefix, enter the prefix in the LOCATE MEMBER/LIST QUEUE PREFIX field at the bottom of the panel, and then press **PF12**.

Adding a New VSE/ICCF Library Member

To add a new member to the library, press PF6. Note that PF6 only is displayed for VSE/ICCF primary libraries. Pressing PF6 displays the *Create New Member* panel. In this panel, type in the **member name**, and optionally a **password** which may be up to four characters long. Then press ENTER. The new member is created. The dialog invokes the VSE/ICCF editor.

Notes:

1. You can use fast path 311 from the *VSE/ESA Function Selection* panel to go directly to the *Create New Member* panel.

2. For a detailed description of the "VSE/ICCF editor functions," refer to the *VSE/ICCF User's Guide*.

Program Function Key Summary

PF1 displays a help panel. Any input is saved.

PF2 refreshes the FULIST and displays only entries starting with the given prefix. Enter the prefix in the MEMBER PREFIX field **and** press PF2.

PF3 leads back to the *Program Development Library*. Any input is ignored.

PF4 leads back to the initial *VSE/ESA Function Selection*. Any input is ignored.

PF6 displayed for primary libraries. Press PF6 to add a new member to the library.

PF7 displays the previous page.

PF8 displays the next page.

PF9 refreshes the FULIST and sorts the entries by date in descending order.

PF10 refreshes the FULIST and sorts the entries by name in alphabetical order.

PF11 refreshes the FULIST and sorts the entries by size in descending order.

PF12 displays a FULIST of output for your jobs in the VSE/POWER list queue. From there you can return to the *Program Development Library* dialog by pressing **PF3**.

Enter a list queue prefix in the LOCATE MEMBER field at the bottom of the panel and then press **PF12** to display your entries in the VSE/POWER list queue starting with that prefix.

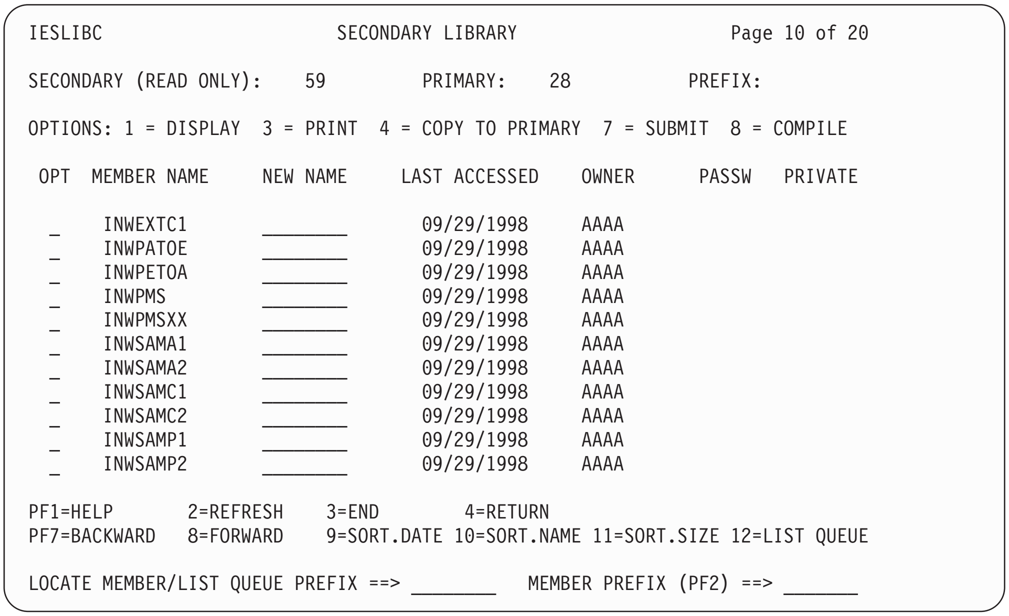
Primary and Secondary Library Options

For the **primary** library, the options shown in Figure 10 on page 28 are offered:

- Edit
- Change
- Print
- Copy
- Delete
- Rename
- Submit
- Compile
- Display

For the **secondary** library, the options shown in Figure 11 on page 31 are offered:

- Display
- Print
- Copy to primary
- Submit
- Compile



```
IESLIBC                    SECONDARY LIBRARY                  Page 10 of 20

SECONDARY (READ ONLY):    59        PRIMARY:    28          PREFIX:

OPTIONS: 1 = DISPLAY  3 = PRINT  4 = COPY TO PRIMARY  7 = SUBMIT  8 = COMPILE

 OPT  MEMBER NAME     NEW NAME     LAST ACCESSED    OWNER      PASSW   PRIVATE

  _    INWEXTC1       ________       09/29/1998     AAAA
  _    INWPATOE       ________       09/29/1998     AAAA
  _    INWPETOA       ________       09/29/1998     AAAA
  _    INWPMS         ________       09/29/1998     AAAA
  _    INWPMSXX       ________       09/29/1998     AAAA
  _    INWSAMA1       ________       09/29/1998     AAAA
  _    INWSAMA2       ________       09/29/1998     AAAA
  _    INWSAMC1       ________       09/29/1998     AAAA
  _    INWSAMC2       ________       09/29/1998     AAAA
  _    INWSAMP1       ________       09/29/1998     AAAA
  _    INWSAMP2       ________       09/29/1998     AAAA

PF1=HELP       2=REFRESH    3=END        4=RETURN
PF7=BACKWARD   8=FORWARD    9=SORT.DATE 10=SORT.NAME 11=SORT.SIZE 12=LIST QUEUE

LOCATE MEMBER/LIST QUEUE PREFIX ==> ________   MEMBER PREFIX (PF2) ==> _______
```

Figure 11. FULIST of Secondary Library

Editing a Member

You can edit a member in a primary library only if you have write access to the member. To edit a VSE/ICCF library member from the *Primary Library* panel enter **1** in the option column next to the member.

The member is displayed for editing. How to “use the VSE/ICCF editor” is described in the IBM manual *VSE/ICCF User's Guide*. This manual describes, for example, how you can set your tabs *each time* you edit a member.

Displaying a Member

In the *Primary Library* panel, enter **9** in the option column next to the member to be displayed. In the *Secondary Library* panel, enter **1** in the option column next to the member to be displayed. You can only browse a member and not edit it.

Changing a Member

You can change a member only in a primary library. The following characteristics can be changed:

- OWNER: Enter the new user ID.
- PASSW: Enter a password for password protection or key over the asterisk (*) to remove password protection.
- PRIVATE: Add or remove the asterisk (*). An asterisk indicates that the member is private.

To change a member, proceed as follows:

On the panel shown in Figure 10 on page 28, type in

  2 (Change)

in the option column next to the desired member. Key in (over) the value you want to change, and then press **ENTER**.

If you have a programmer (type 2) profile, you cannot change a member that another user owns. This can only be done by the system administrator.

Printing a Member

You can print a member of a primary or of a secondary library. The member to be printed gets **job name P$xxxx** (xxxx is your user ID).

If job P$xxxx cannot be printed right away, for example because of other jobs running, the system will print the most current version of the member. For example, in case you edit a member after you entered the print request but before you get the member as P$xxxx job printed, the system will print the edited (updated) version of this member.

On the panel shown in Figure 10 on page 28, to print a member, enter **3** (Print) in the option column next to the member you like to print. Then press ENTER.

Now the system creates the print batch job P$xxxx for you. After processing, the job output is directed to the VSE/POWER LST queue with the default entry **A** for class and **D** for disposition. Job P$xxxx will be printed -without your intervention- if a printer is started for class A.

In some rare cases you may not get a printout; for instance, when the printer has not been started for class A. In these cases, ask your system administrator for help.

Copying a Member

From a primary library, you can copy members to a library to which you have write access. To copy a member, proceed as follows:

On the panel shown in Figure 10 on page 28, type in

4 (Copy)

in the option column next to the desired member. Specify the new member name in the NEW NAME column. In the NEW LIB column, enter the library number to where you want to copy the member.

If you want to use the same member name, enter = in the NEW NAME column.

To copy a member to the same library, leave the NEW LIB field blank or enter =. Then press ENTER.

Copying to Primary

From a *secondary* or from a *common* library, you can copy a member to the primary library. Specify the new member name in the NEW NAME column. To use the same member name, enter = in the NEW NAME column.

The member is copied to the library you specified as the primary library when you selected the dialog. The library number is displayed at the top of the FULIST.

Deleting a Member

Always keep the contents of your library as small as possible, because this makes library searches quicker.

You can only delete members from a primary library. If you have a programmer (type 2) profile, you cannot delete a member which another user owns. To delete a member, proceed as follows:

On the panel shown in Figure 10 on page 28, type in

5 (Delete)

in the option column next to the desired member. Then press ENTER.

Renaming a Member

You can only rename members in a primary library. Specify the new member name in the NEW NAME column. If you have a programmer (type 2) profile, you cannot rename a member which another user owns. To rename a member, proceed as follows:

On the panel shown in Figure 10 on page 28, type in

6 (Rename)

in the option column next to the desired member. Then press ENTER.

Submitting a Member

You can submit a member to the VSE/POWER reader queue from either the primary or secondary library.

The Interactive Interface supplies a VSE/POWER job statement, if one is not found in the first sixteen records of the member. It also includes an EOJ (End-of-Job) statement. However, it assumes the member has proper VSE job control statements. To submit a member, proceed as follows:

On the panel shown in Figure 10 on page 28, type in

7 (Submit)

in the option column next to the desired member. Then press ENTER.

Notes:

1. If you submit a print job and take the defaults in the POWER LST statement, the output will go to the VSE/POWER list queue with disposition D and class A. See "Printing a Member" on page 32 for more information.
2. If a user-written VSE/POWER reader exit routine is running and writes messages to the system console, the following message will be displayed:

   ```
   IMPROPER VSE/POWER CONTROL STATEMENTS DETECTED
   ```

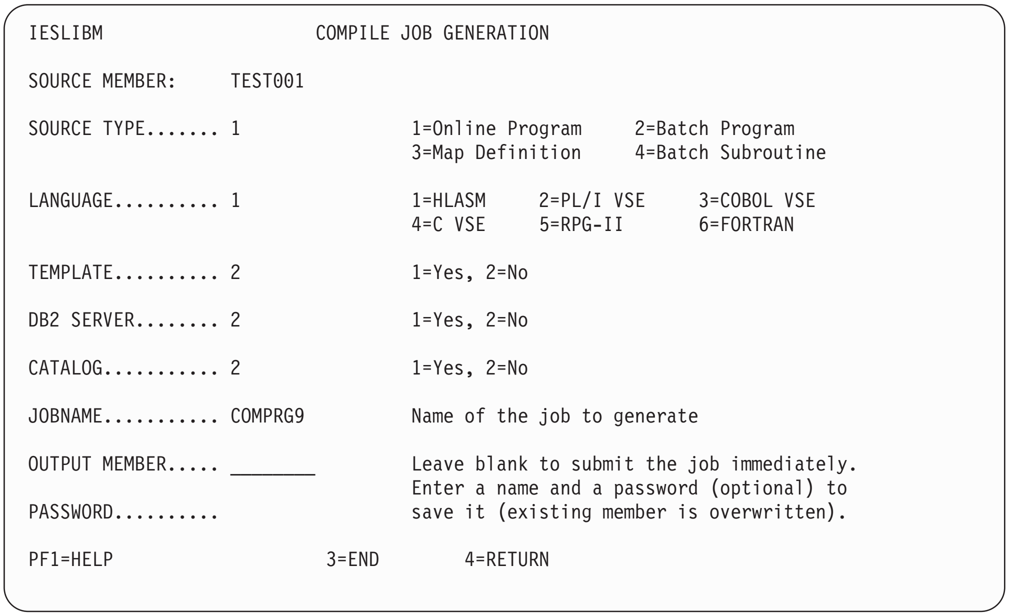
Compiling a Member

You can compile a primary or a secondary library member. Before you use this option, ask your system administrator to set up a correct compile skeleton for you. Manual *VSE/ESA Administration* has information on how to "set up a complile skeleton." To compile a member, proceed as follows:

On the panel shown in Figure 10 on page 28, type in

8 (Compile)

in the option column next to the desired member. Then press ENTER. The following panel will be displayed:



```
IESLIBM                   COMPILE JOB GENERATION

SOURCE MEMBER:     TEST001

SOURCE TYPE....... 1                1=Online Program     2=Batch Program
                                    3=Map Definition     4=Batch Subroutine

LANGUAGE.......... 1                1=HLASM     2=PL/I VSE     3=COBOL VSE
                                    4=C VSE     5=RPG-II       6=FORTRAN

TEMPLATE.......... 2                1=Yes, 2=No

DB2 SERVER........ 2                1=Yes, 2=No

CATALOG........... 2                1=Yes, 2=No

JOBNAME........... COMPRG9          Name of the job to generate

OUTPUT MEMBER..... ________         Leave blank to submit the job immediately.
                                    Enter a name and a password (optional) to
PASSWORD..........                  save it (existing member is overwritten).

PF1=HELP                   3=END        4=RETURN
```

Figure 12. Compile Job Generation

In the panel, select the source type, the programming language used and whether the phase (which is the result of the compile) is to be cataloged. For the compile job the default jobname *COMxxxx* (where xxxx is your user ID) is used. You can type another name over the default name if you do not like it.

To create and submit the job immediately, simply press ENTER. To generate a compile job and save it in your primary libary, type in an output member name and optionally a password. Then press ENTER. The following list gives you some additional information on what to enter into the panel:

SOURCE TYPE

Online Program
This can be any application running under the control of CICS.

Batch Program
This can be any application running in a batch partition. Note that CICS tables are treated as batch programs.

Map definition
A map is the screen image which lets you define the input/output fields for a certain panel. When defining a map, specify this parameter for *SOURCE TYPE*.

Batch Subroutine
This can be any subroutine that will be linked together with another subroutine to form one batch program.

LANGUAGE
This is the programming language used for your program. For maps specify the language of the program using the map.

TEMPLATE
By specifying **1** (valid only for 3=Map Definition), you can request the creation of an HTML template from a BMS map. This makes the compile output suitable for a Web browser environment. Maps of type HTML are stored in library PRD2.DFHDOC.

DB2 SERVER
For DB2 SERVER support another panel (see Figure 13) will be displayed that allows you to specify DB2 SERVER preprocessor options. For this, enter **1** in the entry field.

CATALOG
Specify whether the output of the compiler is to be cataloged into the specified library. In most cases your system administrator has defined the library with a *LIBDEF* statement in member “C$$xxxx” of the common library. The manual *VSE/ESA Administration* has more information available.

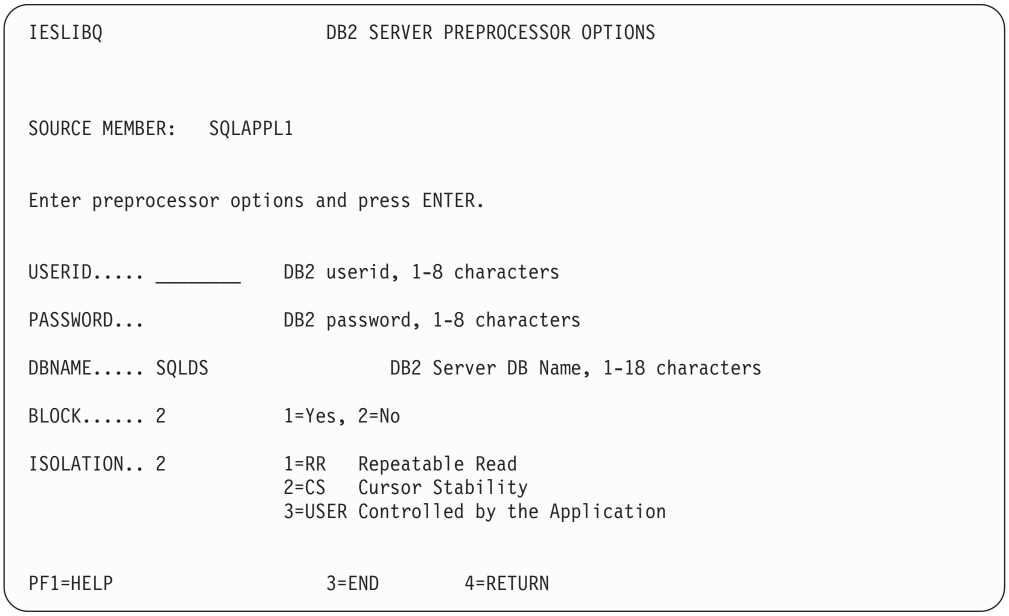
JOBNAME
The jobname should consist of 2 to 8 alphanumeric characters including (# $ @ - . /). You can use the default jobname COMxxxx (where xxxx is your user ID) or type in your own jobname.

OUTPUT MEMBER
This is the name your compile job will receive in your primary library. It must be 1 to 8 characters long and may not contain any commas or $ signs. The first character must be alphabetic. An existing member with the same name will be overwritten without warning.

Note: Do **not** use an output member name which is the same as the source member name. If you do, the source code will be replaced with the compile job stream. For example, in Figure 12 on page 34, the source member name *TEST001* may not be used as the output member name.

PASSWORD
The password for the output member is optional. It should consist of 1 to 4 characters without any commas. The first character must not be numeric.

For DB2 support, the following panel appears for additional DB2 preprocessor information. To display this panel, enter **1** in the entry field ″DB2 SERVER″ that is displayed in the panel *Compile Job Generation* shown in Figure 12 on page 34.



```
IESLIBQ                    DB2 SERVER PREPROCESSOR OPTIONS



SOURCE MEMBER:   SQLAPPL1


Enter preprocessor options and press ENTER.


USERID..... ________    DB2 userid, 1-8 characters

PASSWORD...             DB2 password, 1-8 characters

DBNAME..... SQLDS                 DB2 Server DB Name, 1-18 characters

BLOCK...... 2           1=Yes, 2=No

ISOLATION.. 2           1=RR   Repeatable Read
                        2=CS   Cursor Stability
                        3=USER Controlled by the Application


PF1=HELP                    3=END        4=RETURN
```

Figure 13. DB2 SERVER Preprocessor Options

Handling VSE/ICCF Library Members

In the above panel you have to supply the following information:

USERID
Identifies the creator of the program to DB2 for VSE.

PASSWORD
Established for this user by a DB2 GRANT statement. Valid DB2 passwords can be from 1 to 8 characters long. The first character has to be alphabetic. Characters 2 through 8 have to be alphanumeric or '_' .

Both the user ID and password are used in executing a CONNECT statement to gain access to the DB2 for VSE data base.

DBNAME
Name of the DB2 for VSE application server to be accessed by the DB2 Server statements in the preprocessor source file.

BLOCK
Indicates whether you want DB2 for VSE to insert and retrieve rows in groups.

ISOLATION
Specifies the isolation level that the program runs at:

RR Is repeatable read: DB2 holds a lock on all data read by the program in the current logical unit of work.

CS Is cursor stability: DB2 holds a lock on the row or page of data pointed to by a cursor.

USER Indicates that the application program controls its isolation level itself.

Note that compile skeletons delivered with VSE/ESA are adapted to the most current version of DB2 available. In case you want to use another version of DB2 you have to tailor the compile skeletons. **You should consider the following:**

1. The assembler is part of VSE/ESA. The compilers listed below are optional licensed programs and must be installed separately:
   - PL/I VSE
   - COBOL VSE
   - C VSE
   - RPG-II

     Note: RPG online programs can only be used with CICS/VSE in the "CICS coexistence environment". For details of this environment, refer to the *LE/VSE Customization Guide*
   - FORTRAN
2. For the compile run, it is assumed that the following setup exists for the partition(s) used by the compile jobs:
   - Work files defined and assigned.
   - Library search chain includes the sublibrary containing the compiler used.

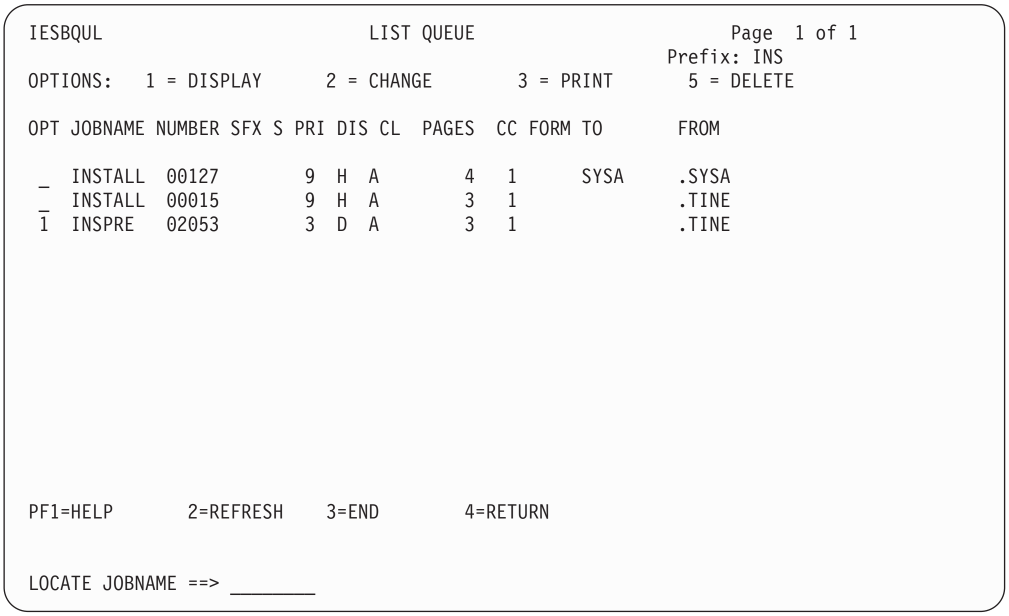
Accessing the VSE/POWER List Queue

To access the VSE/POWER list queue, press **PF12** from either the Primary Library panel (see Figure 10 on page 28) or from the Secondary Library panel (Figure 11 on page 31).

If you want only a group of list queue entries to be displayed, enter the common prefix of those entries in the LOCATE MEMBER/LIST QUEUE PREFIX field. For example, if you want all entries with the prefix *INS* to be displayed, enter INS into the locate field at the bottom of the panel. Then press PF12.

A FULIST panel (see Figure 14) is displayed which shows you the entries in the queue at the time you made your selection. Press **PF2** to refresh the panel. This updates the FULIST and displays the most current information.

Depending on your user profile, only your own entries (instead of the complete list queue) are displayed.



```
IESBQUL                         LIST QUEUE                        Page  1 of 1
                                                            Prefix: INS
OPTIONS:   1 = DISPLAY      2 = CHANGE        3 = PRINT       5 = DELETE

OPT JOBNAME NUMBER SFX S PRI DIS CL  PAGES  CC FORM TO       FROM

 _  INSTALL  00127        9  H  A        4   1      SYSA     .SYSA
 _  INSTALL  00015        9  H  A        3   1               .TINE
 1  INSPRE   02053        3  D  A        3   1               .TINE











PF1=HELP       2=REFRESH    3=END       4=RETURN


LOCATE JOBNAME ==> ________
```

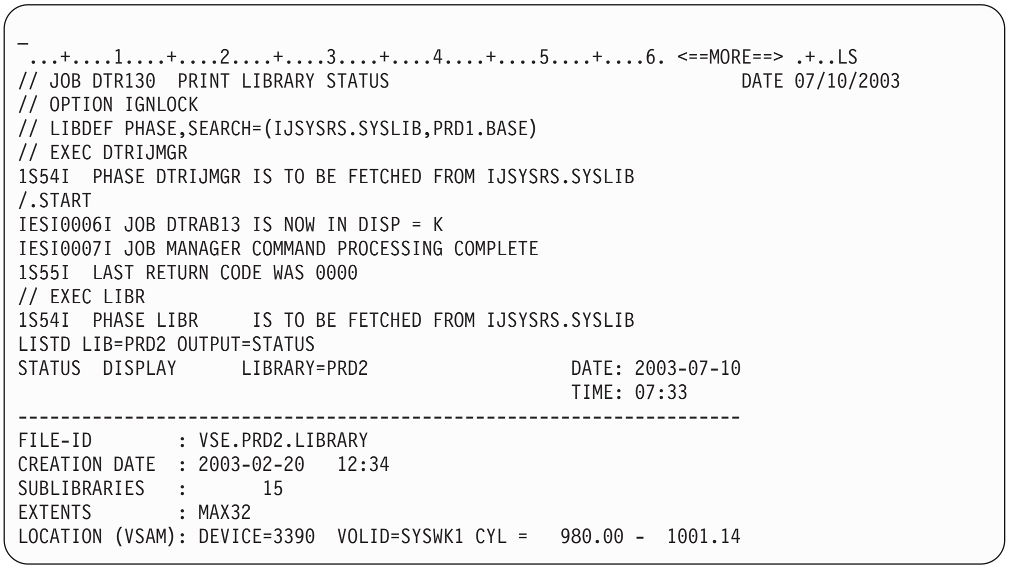
Figure 14. FULIST of VSE/POWER List Queue

To locate a particular queue entry, enter the **jobname** in the LOCATE JOBNAME field. The dialog searches the group which you selected for display (in our example all jobs with the prefix *INS*). If the entry is found, it is displayed with an * in the OPT column. If it is not in the queue, the dialog displays a message.

Note: To locate a particular entry, enter the entire name of the job. You cannot just enter a prefix.

You can use option DISPLAY to display job output. Enter 1 in the OPT column next to the desired queue entry. A panel such as the following will be displayed.

Accessing VSE/POWER List Queue



```
_
 ...+....1....+....2....+....3....+....4....+....5....+....6. <==MORE==> .+..LS
// JOB DTR130  PRINT LIBRARY STATUS                                 DATE 07/10/2003
// OPTION IGNLOCK
// LIBDEF PHASE,SEARCH=(IJSYSRS.SYSLIB,PRD1.BASE)
// EXEC DTRIJMGR
1S54I  PHASE DTRIJMGR IS TO BE FETCHED FROM IJSYSRS.SYSLIB
/.START
IESI0006I JOB DTRAB13 IS NOW IN DISP = K
IESI0007I JOB MANAGER COMMAND PROCESSING COMPLETE
1S55I  LAST RETURN CODE WAS 0000
// EXEC LIBR
1S54I  PHASE LIBR     IS TO BE FETCHED FROM IJSYSRS.SYSLIB
LISTD LIB=PRD2 OUTPUT=STATUS
STATUS  DISPLAY      LIBRARY=PRD2                   DATE: 2003-07-10
                                                    TIME: 07:33
--------------------------------------------------------------------
FILE-ID        : VSE.PRD2.LIBRARY
CREATION DATE  : 2003-02-20   12:34
SUBLIBRARIES   :       15
EXTENTS        : MAX32
LOCATION (VSAM): DEVICE=3390  VOLID=SYSWK1 CYL =   980.00 -  1001.14
```

Figure 15. Display of VSE/POWER List Queue Member

For DISPLAY, PF keys are available that help you scroll and search the data. The PF keys are shown in Table 1. From within the displayed list ouput, you also may view the PF key settings by entering:

```
/SHOW PFLS
```

at the top left corner of the screen.

Table 1. PF Keys for DISPLAY Option in VSE/POWER List Queue

PF Key	PF Key Setting
PF1	Show PF Key setting
PF2	Redisplay current page
PF3	End
PF4	End
PF5	Locate; search columns 1-9 only
PF6	Locate search all columns
PF7	Scroll backward
PF8	Scroll forward
PF9	TOP
PF10	Shift left
PF11	Shift right
PF12	BOTTOM

Notes:

1. The characters you enter for the LOCATE commands (PF5 and PF6) must be entered in uppercase. Enclose with delimiters (typically ′/′) a search string that includes blanks.
2. The settings for PF13 - PF24 correspond to the settings for PF1 - PF12.
3. For further information about how to use the "PF keys" in the VSE/ICCF editor, refer to the IBM manual *VSE/ICCF User's Guide*.

File Management

For the default programmer hierarchy, the Interactive Interface offers the *File Management* dialog. This helps you create, delete, and use VSE/VSAM files, but not user catalogs. You can access the dialog as follows:

In the *VSE/ESA Function Selection* panel shown in Figure 2 on page 8, select:
4 (File Management)

```
Sample                 Synonym
Fast Path:   4         Default:    ——    Yours:
```

A panel displays four tasks which you can select.

1. Display or Process a File
2. Define a New File
3. Define a Library
4. Define an Alternate Index or Name

Some dialog tasks process the information immediately. Other tasks create a job. You can submit the job to run or store it as a VSE/ICCF library member in your default primary library.

Note: The access to selections 2, 3, and 4 is dependent on the authorization in your user profile. These selections **should** only be done with administrative authority. Refer to the IBM manual *VSE/ESA Administration* for more information about the "file management dialog" or contact your system administrator.

At the top of the panel the default catalog is displayed. If it is not satisfactory, type in the desired catalog name over the default catalog, type in the desired selection number (in this case **1**), and then press ENTER.

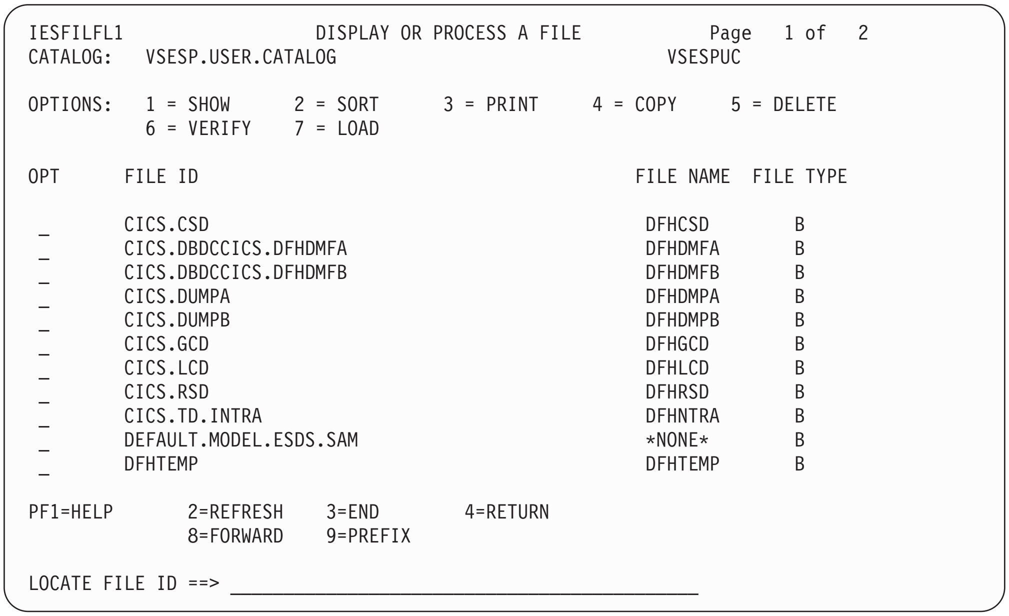
Display or Process a File

To access this dialog, proceed as follows:

In the *VSE/ESA Function Selection* panel shown in Figure 2 on page 8, select:
4 (File Management)
In the *File Management* panel, select:
1 (Display or Process a File)

```
Sample                 Synonym
Fast Path:   41        Default:    ——    Yours:
```

The following panel will be displayed:



```
IESFILFL1                  DISPLAY OR PROCESS A FILE           Page   1 of   2
CATALOG:   VSESP.USER.CATALOG                              VSESPUC

OPTIONS:   1 = SHOW      2 = SORT      3 = PRINT     4 = COPY     5 = DELETE
           6 = VERIFY    7 = LOAD

OPT      FILE ID                                         FILE NAME  FILE TYPE

 _       CICS.CSD                                         DFHCSD       B
 _       CICS.DBDCCICS.DFHDMFA                            DFHDMFA      B
 _       CICS.DBDCCICS.DFHDMFB                            DFHDMFB      B
 _       CICS.DUMPA                                       DFHDMPA      B
 _       CICS.DUMPB                                       DFHDMPB      B
 _       CICS.GCD                                         DFHGCD       B
 _       CICS.LCD                                         DFHLCD       B
 _       CICS.RSD                                         DFHRSD       B
 _       CICS.TD.INTRA                                    DFHNTRA      B
 _       DEFAULT.MODEL.ESDS.SAM                           *NONE*       B
 _       DFHTEMP                                          DFHTEMP      B

PF1=HELP       2=REFRESH    3=END        4=RETURN
               8=FORWARD    9=PREFIX

LOCATE FILE ID ==> ____________________________________________
```

Figure 16. Display or Process a File

Figure 16 shows the file IDs and file names of all files in the catalog specified in the previous panel. Pressing PF2 updates the list to show you the latest information. Pressing PF9 displays another panel which lets you define a subset of this FULIST, allowing you to work with that subset. 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.

Show a File

This is option 1 in the panel *Display or Process a File* shown in Figure 16. Type in **1** in the option column next to the file you want to show. Then press ENTER.

The task *Show a File* displays details about the characteristics of the file or alternate index. There can be several panels of information.

Use PF7 and PF8 for scrolling through the panels. If you select the show option for a base file (indicated by a *B* in the file type column), you may press PF9=ASSOCIAT to receive base file information and a list of alternate indexes and alternate names. If you select the show option for an alternate index (indicated by an *A* in the file type column), pressing PF9=ASSOCIAT gives you a list of alternate names.

Note: PF9=ASSOCIAT is only displayed if there is an alternate name/index defined for the file.

Use PF10 (SPACE INFO) to display detailed information about the space occupied by the file.

Sort a File

This is option 2 in the panel *Display or Process a File* shown in Figure 16 on page 40. Type in **2** in the option column next to the file you want to sort. Then press ENTER.

The task *Sort a File* sorts a VSE/VSAM file. You must have:
- The IBM Sort-Merge program product or a compatible product installed.
- Both input and output files already defined in the catalog.

The dialog displays several panels. You need the following information:
- Is the selected file input or output file for SORT?
- File ID or file name of the second file. You may sort one input and one output file. In this case you have to specify two files.
- Format. Data representation within the sort field. You can have up to six sort fields consisting of format, start, length, and sorting order.
- Start of the sort field.
- Length of the sort field.
- Sorting order: ascending or descending.

Output File is a Work File

If the output file is a work file (reusable), you can:
- Overwrite the data in the work file. The output file is cleared before the sort output is written.
- Add the sorted output to the data in the work file.

You can *sort in place* by using reusable work files for both the input and output files.

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 to a VSE batch partition. It also stores it in a VSE/ICCF library member in your default primary library.

The default library member name is **F$xxxx**, where xxxx is your user ID. You can change the name on the panel.

Print a File

This is option 3 in the panel *Display or Process a File* shown in Figure 16 on page 40. Type in **3** in the option column next to the file you want to print. Then press ENTER.

The task *Print a File* prints one or more records of a VSE/VSAM file on the system printer. You need the following information:
- PRINT FORMAT
  1 - Character
  2 - Hexadecimal
  3 - Character and hexadecimal

- PRINT ENTIRE FILE
  1 - Yes (Entire file)
  2 - No (Partial print)

Partial Print

If you choose *partial print*, specify where the printing starts and ends.

You can start printing with:
- Particular key (KSDS (key-sequenced data sets) files only).
- Particular record number.

You can stop printing by specifying:
- Last key you want printed (KSDS files only).
- Number of records you want printed (last record number for RRDS (relative record data set) files).

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 in a VSE/ICCF library member in your default primary library.

Option **2** submits the job to a VSE batch partition. It also stores it in a VSE/ICCF library member in your default primary library.

The default library member name is **F$xxxx**, where xxxx is your user ID. You can change the name on the panel.

Copy a File

This is option 4 in the panel *Display or Process a File* shown in Figure 16 on page 40. Type in **4** in the option column next to the file you want to copy. Then press ENTER.

The task *Copy a File* copies all or part of a file to another file. You can also copy a VSE/VSAM file from disk to tape and vice versa. The dialog creates a job stream with the VSE/VSAM REPRO command. You need the following information:

- Is the selected file input or output file for the COPY?
- File ID or file name of the second file (for a VSE/VSAM file),

  or File ID of the second file (for a labeled tape file).
- Start and end positions for the copy.
- Start and end key (keyed-file only). If the file is a KSDS (key-sequenced data set) file, you can specify the start or end key for the copy.
- Start and end record specification. You can specify the start or end record, if the file is:
  – ESDS (entry-sequenced data set)
  – KSDS (key-sequenced data set)
  – RRDS (relative record data set)
  – VRDS (variable-length relative-record data set)

Tape Files

If either file is a **tape** file, you need the following information:
- Volume serial number (for labeled tape files only)
- Tape address
- Record format
- Block size and/or record size, depending on the record format

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 in a VSE/ICCF library member in your default primary library.

Option **2** submits the job to a VSE batch partition. It also stores it in a VSE/ICCF library member in your default primary library.

The default library member name is **F$xxxx**, where xxxx is your user ID. You can change the name on the panel.

Delete a File

This is option 5 in the panel *Display or Process a File* shown in Figure 16 on page 40. Type in **5** in the option column next to the file you want to delete. Then press ENTER. A panel is displayed. You must confirm the DELETE. If the file has an alternate name, you must decide whether only the alternate name or the file including all alternate names should be deleted.

The task *Delete a File* deletes a VSE/VSAM file or an alternate index. The option *Delete a File* is dependent on the authorization in your user profile. It should only be done with administrative authority.

Verify a File

This is option 6 in the panel *Display or Process a File* shown in Figure 16 on page 40. Type in **6** in the option column next to the file you want to verify. Then press ENTER.

The task *Verify a File* is used to compare the end-of-file information as it is stored in a VSE/VSAM catalog with the end-of-file indicator(s) in the file itself. If the information in the catalog does not agree with the end-of-file indicator(s) in the file, the catalog information is corrected. You **cannot** verify an alternate index.

Some VSE/VSAM open errors are caused by an abend or error condition which results in a file not being closed properly. If catalog damage is not severe, the VERIFY task may enable you to open and process the file successfully. You should determine the reasons for any open failures and correct them to avoid more serious problems.

You do not have to enter 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 in a VSE/ICCF library member in your default primary library.

Option **2** submits the job to a VSE batch partition. It also stores it in a VSE/ICCF library member in your default primary library.

The default library member name is **F$xxxx**, where xxxx is your user ID. You can change the name on the panel.

Load

This function is split into two parts, depending on what you selected:

- Load a file.
- Load an alternate index.

Load a File: This is option 7 in the panel *Display or Process a File* shown in Figure 16 on page 40. Type in **7** in the option column next to the file you want to load. Then press ENTER.

You should select the task *Load a File* for a base file. The FILE TYPE field displays the letter *B* to indicate base files. The dialog loads a base file with data from a VSE/ICCF library member.

You can load ESDS, KSDS, RRDS and VRDS files. You cannot load sequential SAM (sequential access method) ESDS files. If the receiving file (the file you want to load) is not empty, data is added to the existing file. If this is not desired, you should first delete and then define the file before running the job created by this dialog.

You need the following information:

- INPUT MEMBER NAME The name of the VSE/ICCF library member that contains the data.
- PRINT Specify whether you want the file printed after it is loaded:
  1 - YES
  2 - NO
- RECORD LENGTH Specify **80** to load the VSE/VSAM file with 80 byte fixed-length records (the record length of the VSE/ICCF library member).

  If the VSE/VSAM file will contain records of more or less than 80 bytes or if the file should have variable length records, then you must segment the records in the VSE/ICCF member. You specify one VSE/VSAM record as a series of lines (in the library member) with 1 - 72 bytes each. You indicate the end of each record with the character string QQQ.

  The line length can be 1 - 72. For example, if you use a line length of 50, you enter your data in positions 1 - 50 and end the record with QQQ.

  1. Select a line length *X* from 1 - 72. Enter your data in the VSE/ICCF member from position 1 to position *X*. At the end of the data, enter **QQQ**.

     You can put QQQ in any position within the line length or in the position:

     ```
     LINE LENGTH + 1
     ```

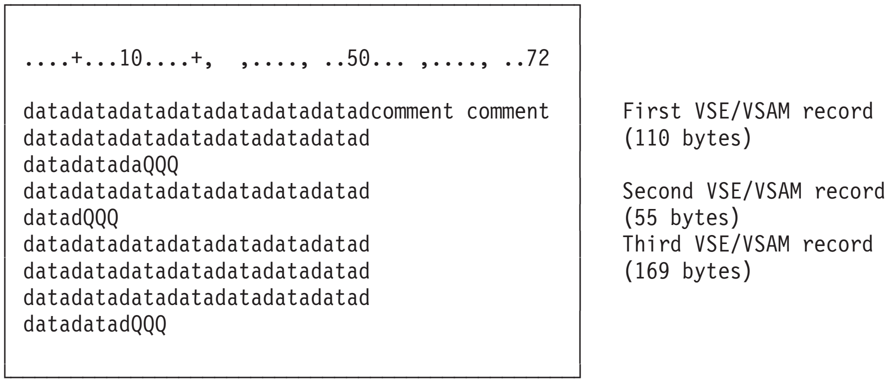
     In addition, you can also put QQQ immediately at the beginning of the next line.
  2. You can put comments in any position to the right of the line length (at position X + 1). For example, if your line length is 50, you can put comments in positions 51 - 80.
  3. For the RECORD LENGTH field, specify the **line length** you used.

  The maximum record length for the file is 2048 bytes.

As an example, suppose you have variable length records. The first records is 110 bytes. You can specify the length as 50. Enter the data in positions 1 - 50 in the first two lines of the library member. In the third line, enter the remaining data in positions 1 - 10 and then enter **QQQ** to indicate the end of the record. The second record is 55 bytes. In the fourth line enter data 1 - 50 and in the fifth line enter data bytes 51 - 55. Enter QQQ to indicate the end of record, and so on.

An example of segmenting the data in the VSE/ICCF library member is shown in Figure 17.

VSE/ICCF Library Member



```
....+...10....+,  ,...., ..50... ,...., ..72

datadatadatadatadatadatadatadcomment comment     First VSE/VSAM record
datadatadatadatadatadatadatad                    (110 bytes)
datadatadaQQQ
datadatadatadatadatadatadatad                    Second VSE/VSAM record
datadQQQ                                         (55 bytes)
datadatadatadatadatadatadatad                    Third VSE/VSAM record
datadatadatadatadatadatadatad                    (169 bytes)
datadatadatadatadatadatadatad
datadatadQQQ
```

Figure 17. Illustration of Segmenting VSE/VSAM Records in VSE/ICCF Member

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 in a VSE/ICCF library member in your default primary library.

Option **2** submits the job to a VSE batch partition. It also stores it in a VSE/ICCF library member in your default primary library.

The default library member name is **F$xxxx**, where xxxx is your user ID. You can change the name on the panel.

Load an Alternate Index: This is option 7 in the panel *Display or Process a File* shown in Figure 16 on page 40. Type in **7** in the option column next to the alternate index you want to load. Then press ENTER.

You should select the task *Load a File* for an alternate index. Note that your system administrator must have defined an alternate index for you. The FILE TYPE field displays the letter *A* to indicate an alternate index. The dialog loads an alternate index from an existing base file (build index function).

You do not have to enter 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 in a VSE/ICCF library member in your default primary library.

File Management

Option **2** submits the job to a VSE batch partition. It also stores it in a VSE/ICCF library member in your default primary library.

The default library member name is **F$xxxx**, where xxxx is your user ID. You can change the name on the panel.

Note: The following tasks are described in *VSE/ESA Administration*. If you need to do one of these tasks, ask your system administrator.

- "Define a new file"
- "Define a library"
- "Define an alternate index or name"

Running a Newly Compiled Program

After creating and compiling your program, you have to take some additional actions to run it. These actions are different, depending on whether your program is

- a batch program, executed in a batch partition.
- an online program, executed under the control of CICS.

Steps Required for Batch Programs

To execute a program designed for batch processing, proceed as follows:

1. If your program accesses disk files, you must define these files to the system and load them, if required. Refer to *VSE/ESA Administration* for more information on "defining a file". Refer to "Load a File" on page 44 for information on how to load a file.
2. In the *VSE/ESA Function Selection* panel, select
   3 (Program Development)
3. In the *Program Development* panel, select
   2 (Create Application Jobstream)

```
Sample                Synonym
Fast Path:   32       Default:    ——    Yours:
```

   This selection allows you to generate the required job control statements for program execution. If you have a previously generated jobstream related to this program in your VSE/ICCF library, you can submit it directly to the VSE/POWER RDR queue. This is described under "Submitting a Member" on page 33.

Steps Required for Online Programs

For programs to be executed in the CICS Transaction Server partition, you must first define the following to CICS:

Transaction
If your program should be invoked as a CICS transaction, the transaction ID must be defined to CICS via the CEDA transaction. If your program is invoked via CICS LINK or XCTL, no transaction definition is required for it.

Programm
There is always an entry required for each program executed in the CICS partition. Define the program to CICS via the CEDA transaction.

File If your program uses a disk data set either for input or output and this data set is not yet defined to CICS, you must define it in the CSD file using CEDA DEFINE. In addition, before using the file, you must define it to the system and then initialize (load) it. Refer to *VSE/ESA Administration* for more information on “defining a file”. Refer to “Load a File” on page 44 for information on how to load a file.

Transient Data Sets
If your program refers to CICS transient data sets, you must modify DFHDCTxx and assemble it.

Others
Some other tables like DFHTSTxx, DFHPLTxx, etc may be required depending what your program is doing.

Normally, your system administrator will do all these definitions for you. If the required definitions for CICS are made via table assembly, CICS must be shut down in some cases and then restarted before the program can be executed.

Part 2. VSE/ESA's Workstation File Transfer Support

VSE/ESA's Workstation File Transfer Support enables users to exchange data between a workstation and a VSE/ESA host system. Workstation data can be stored at the host system and other users can access it. Workstation users also can store host data on a disk or diskette and work with it independently of VSE/ESA.

This part gives information on the scope and functions provided:

- A first look at file transfer functions.
- Getting ready for file transfer.
- File transfer to and from VSE libraries.
- File transfer to and from VSE/POWER queues.
- File transfer to and from the Host Transfer File.
- File transfer to and from CICS temporary storage.
- Data conversion support.
- Using the workstation file transfer trace.
- Using file transfer service functions.

Later sections of this manual provide details about each item. Figure 18 on page 50 shows a graphic that is used throughout to illustrate major functions.

VSE/ESA also provides program **INWMUTIL** which can be run in a batch partition. It allows to exchange files between VSE/VSAM and the HTF without using the *Move Utilities* dialog. Refer to the manual *VSE/ESA System Utilities* for a detailed description of this support.

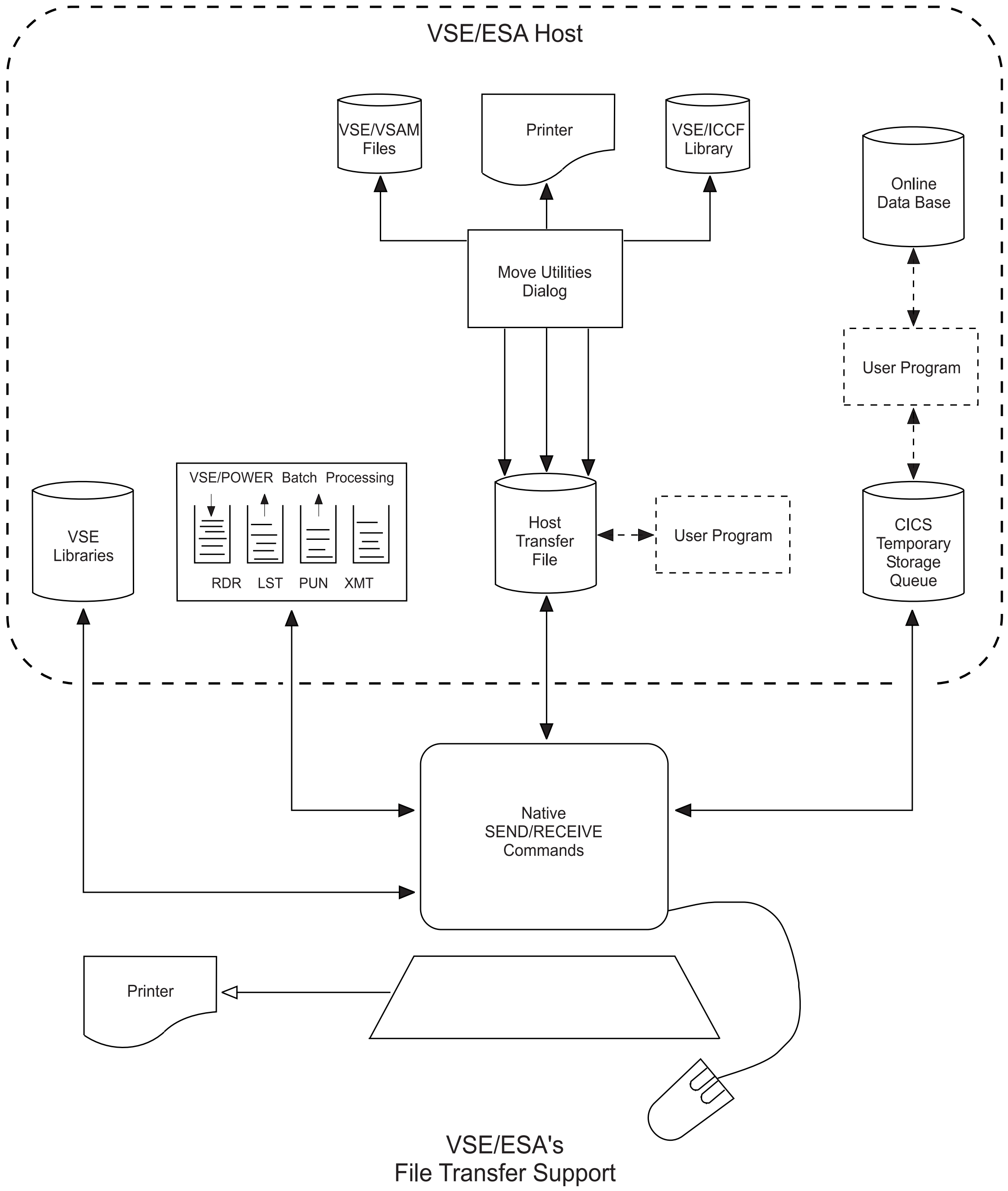


Figure 18. Major Data Paths of VSE/ESA's Workstation File Transfer Support

Chapter 4. A First Look at File Transfer Functions

VSE/ESA Workstation File Transfer Support allows you to transfer many kinds of user data between your programmable workstation and VSE/ESA resources. This chapter introduces the following main functions:

- **File Transfer to and from the VSE libraries**

  For details, refer to chapter Chapter 6, "File Transfer To and From VSE Libraries" , on page 89.
- **File Transfer to and from VSE/POWER queues**

  For details, refer to chapter Chapter 7, "File Transfer To and From VSE/POWER Queues", on page 107.
- **File Transfer to and from the Host Transfer File (HTF)**

  For details, refer to chapter Chapter 8, "File Transfer To and From the Host Transfer File (HTF)", on page 137.
- **File Transfer to and from CICS Temporary Storage (TS)**

  For details, refer to chapter Chapter 9, "File Transfer To and From CICS Temporary Storage", on page 185.

As part of the introduction, this chapter also contains common examples of file transfer using SEND/RECEIVE commands.

To support the above file transfer functions VSE/ESA provides:

- **Data Conversion** to convert transmitted data from workstation to host format and vice versa.

  For details about this topic, refer to:
  - Chapter 10, "Data Conversion Support", on page 205
- An Application Programming Interface (**API**) to access the **Host Transfer File** from within your CICS application programs.

  For details about this topic, refer to "Application Program Interface (API) for the HTF" on page 173.
- The ***Move Utilities*** dialog of the VSE/ESA Interactive Interface to further process the data in the Host Transfer File (HTF) on the host side.

  For details about this dialog, refer to "Move Utilities Dialog for Host Access to the HTF" on page 148.
- A file transfer **trace** to trace the transmission of data between workstation and host in case of problems.

  For details about this topic, refer to Chapter 11, "Using the Workstation File Transfer Trace", on page 225.

In addition to the above data transfer functions, there is a set of file transfer **service functions** for:

- Receiving directory information about VSE libraries.
- Sending a Librarian command for execution to the VSE/ESA host and receiving command output.
- Receiving job completion messages from VSE/POWER.
- Receiving message return codes from VSE/POWER.
- Receiving host user profiles.

Details about the service functions, with descriptions of the corresponding SEND/RECEIVE commands, are in Chapter 12, “Using File Transfer Service Functions”, on page 243.

File Transfer To and From VSE Libraries

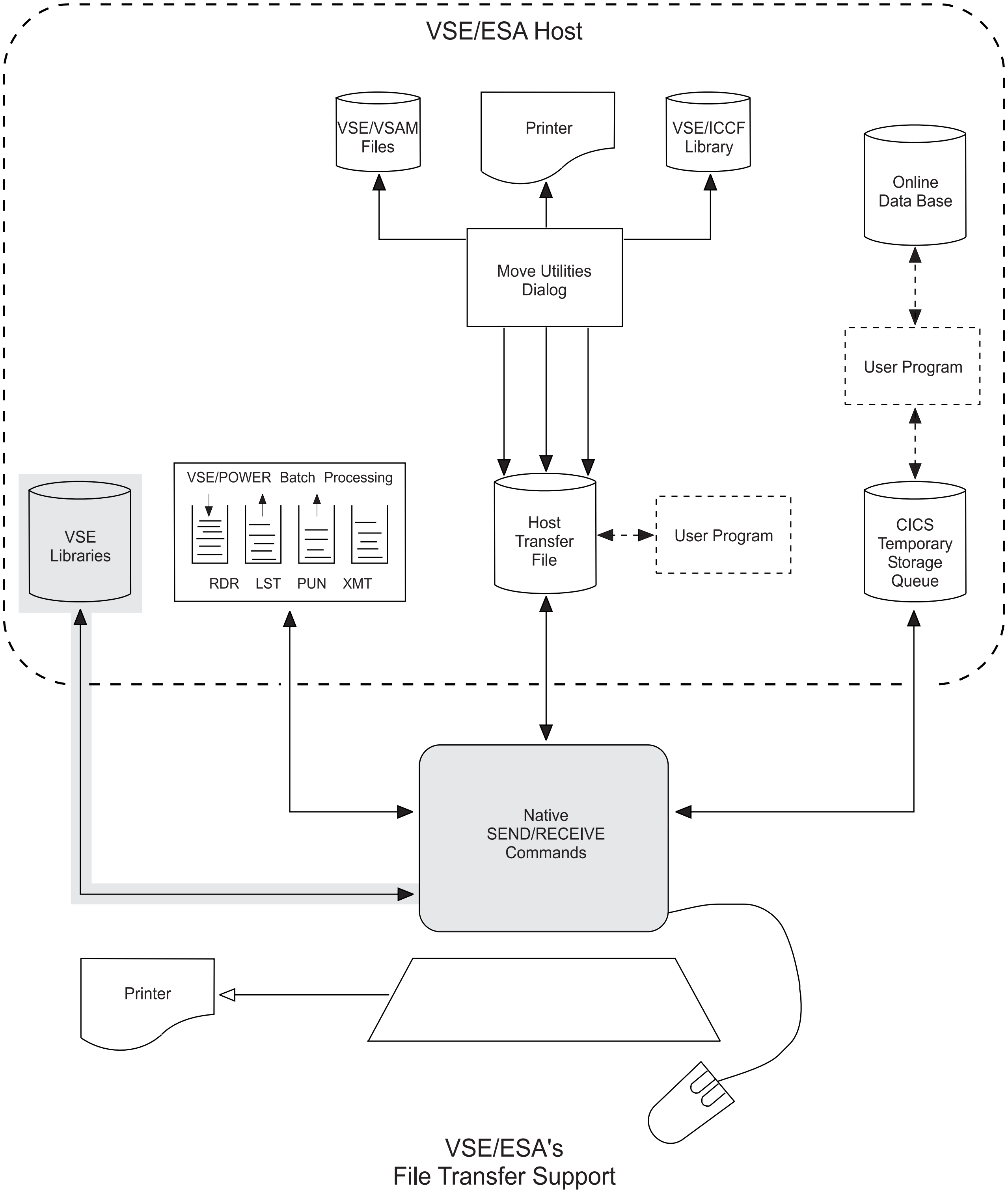


Figure 19. Transferring Data To and From VSE Libraries

IBM workstation users can access the VSE libraries via SEND/RECEIVE file transfer commands. They enable a user to fully utilize the VSE libraries as central repository from the workstation.

File Transfer To and From VSE Libraries

Moreover, the file transfer service functions may be used to do limited library management from the workstation.

Use the SEND/RECEIVE commands for:

- Sending a workstation file to a VSE sublibrary (as string or fixed-80 character data)
- Receiving VSE phases to the workstation (to easily transfer (send) the received phases to another VSE library, for example)
- Receiving a VSE sublibrary member to a workstation file (as binary string or as fixed 80-character records)
- Checking in and out library members (to protect your data when editing library members at the workstation)
- Using your **default** sublibrary **PRIMARY.userid** for file transfer.

The library services at the host are provided by the **Librarian Transaction Server**, a copy of which is loaded into a static or dynamic partition for each SEND/RECEIVE request from the workstation.

For details on how the Librarian Transaction Server is invoked, refer to "The Librarian Transaction Server" on page 90.

The examples in the following section illustrate some commonly used file transfer cases for the VSE libraries.

All **common options** of the SEND and RECEIVE commands are explained in "Common SEND/RECEIVE Command Syntax" on page 70. For **special SEND/RECEIVE options**, refer to "SEND/RECEIVE Command Syntax and Options" on page 103.

Examples

About this Section

This section provides some examples of how you may use SEND/RECEIVE commands with option **FILE=LIB**. This option is used for transferring data to and from a VSE library.

An example case starts with a short heading marked by the '►' character.

► **Checkin/Checkout:**

```
RECEIVE  VTAM.BK ATCSTR00 B  ( FILE=LIB  L=PRD2 S=CONFIG LOCK
```

• **Explanation:**

```
- Receives VSE library member ATCSTR00.B from sublibrary PRD2.CONFIG
  to workstation file VTAM.BK and locks it at the host against write-access by other users.

- The member is checked-out.
```

```
SEND  VTAM.BK  ATCSTR00 B  ( FILE=LIB  UNLOCK  L=PRD2  S=CONFIG
```

• **Explanation:**

```
- The member is checked-in, that is, the updated version of ATCSTR00.B is sent back to
  the host. The lock on the file is removed.

- Note that option UNLOCK implies option REPLACE.
```

▸ **Backup/Restore a Phase:**

```
RECEIVE  MYMOD.PHS  MYMOD PHASE  ( FILE=LIB  BINARY


SEND  MYMOD.PHS  MYMOD PHASE  ( FILE=LIB  S=$$C  BINARY
```

• **Explanation:**

```
- The first example receives member MYMOD.PHASE from the user's primary sublibrary PRIMARY.userid.
  The second example restores it to the common sublibrary PRIMARY.$$C.
```

File Transfer To and From VSE/POWER Queues

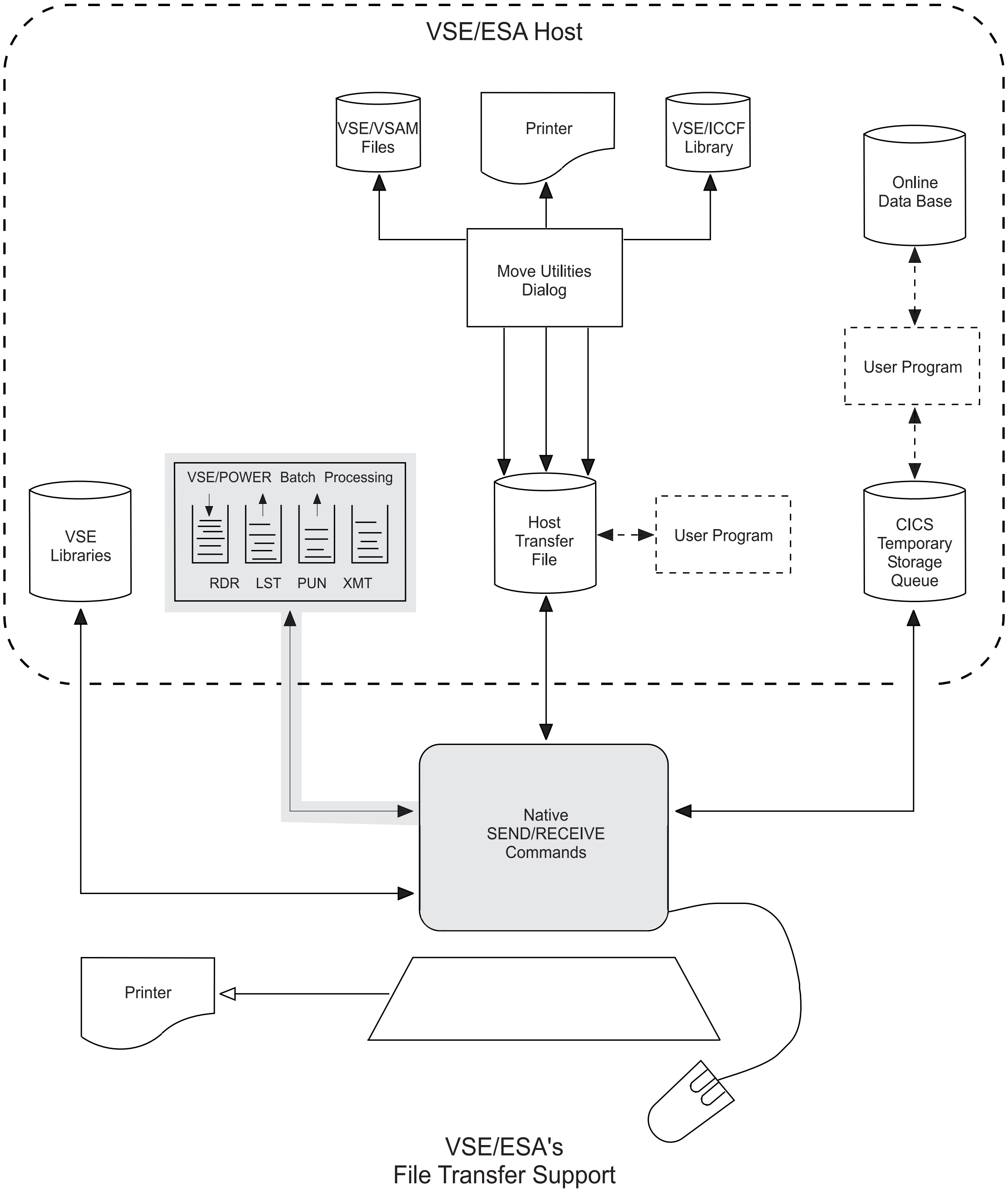


Figure 20. Transferring Data To and From VSE/POWER

Workstation users can access VSE/POWER queues via the SEND/RECEIVE file transfer commands for:

- Sending VSE jobs from the workstation to the VSE/POWER reader queue (RDR queue) for execution.

- Sending data to the VSE/POWER list queue (LST queue) for formatting via the CICS Report Controller Feature (RCF).
- Sending unformatted data to the LST queue (FILE=LSTX).
- Sending data to the VSE/POWER punch queue (PUN queue).
- Receiving entries from the RDR, LST, and PUN queue.

You may use these file transfer functions, to:

- Offload large amounts of workstation data for unformatted printing at the host.
- Submit jobs to the host and subsequently retrieve the list or punch output.
- Receive output produced by VSE batch programs to the workstation via the PUN queue.
- Import/export data on magnetic tape via the PUN queue.
- Route workstation data to remote VSE nodes via VSE/POWER PNET.
- If running under VM, send a workstation file to the reader of a VM user or virtual machine.

The examples in the following section illustrate some commonly used file transfer cases for the VSE/POWER queues.

All **common options** of the SEND and RECEIVE commands are explained in “Common SEND/RECEIVE Command Syntax” on page 70. For **special SEND/RECEIVE options**, refer to “SEND/RECEIVE Command Syntax and Options” on page 127.

Examples

About this Section

This section provides some examples of how you may use the SEND/RECEIVE commands with option **FILE=RDR**, **FILE=PMSG**, and **FILE=LST**. These are 3 of 6 available options for transferring data to and from the VSE/POWER queues.

An example case starts with a short heading marked by the ’►’ character.

The following examples assume that:

- The workstation file **VSE.JOB** contains a VSE job which is to be executed at the host.
- You wish to **submit** the job to the VSE system, **verify** correct execution, and **retrieve** the produced output to the workstation.

► **Submit Job to VSE/POWER:**

To submit the job to VSE, enter the following command:

```
     SEND VSE.JOB  (FILE=RDR
```

You may receive a message like this:

```
     INW0079I   JOB  ABCD  01234 successfully submitted.   RC=0000
```

File Transfer To and From VSE/POWER Queues

▸ **Query Job Status:**

To query the status of your submitted job enter the following command:

```
     RECEIVE  JOB.MSG  (FILE=PMSG
```

• **Explanation:**

- This will return all your job completion messages in workstation file JOB.MSG.
- You may receive one or more messages like this:

```
     JOB ABCD 01234 EXECUTED NODE=LOCAL DATE=03/25/93 TIME=11:13:06 MAXRC=0000 LASTRC=0000
```

▸ **Receive List Output:**

To receive the produced list output from the VSE/POWER LST queue enter the following command:

```
     RECEIVE  JOB.OUT  ABCD 1234  (FILE=LST
```

• **Explanation:**

- You receive the generated list output of job ABCD from the VSE/POWER list (LST) queue to workstation file JOB.OUT.

File Transfer To and From Host Transfer File (HTF)

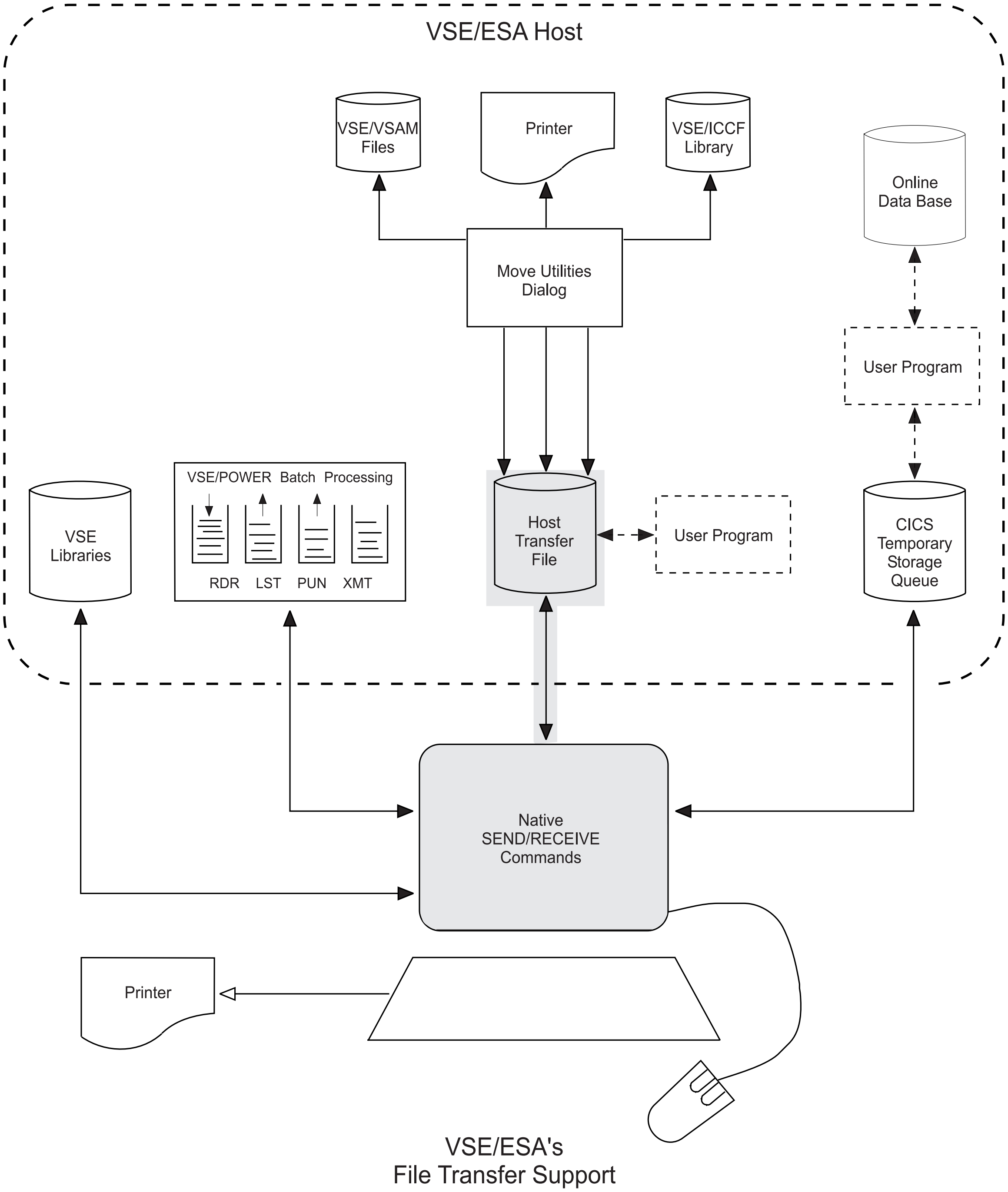


Figure 21. Transferring Data To and From the HTF

The **H**ost **T**ransfer **F**ile (**HTF**) is a VSE/VSAM file. It serves as a "holding area" for files that are transferred to and from workstations and can be used to exchange (share) data with other users.

You would send workstation data to the Host Transfer File if:

File Transfer To and From Host Transfer File (HTF)

- Data is to be moved to a VSE/ICCF library.
- Data is to be moved to VSE/VSAM space.
- You want to keep your data in the Host Transfer File for later retrieval by you or another user.
- You want to distribute your data to other users.

The HTF can contain more than one file at a time, and more than one user can access it at the same time. For this, the file's owner can limit the file access in various ways.

The owner may specify a file in the HTF as:
- Private
- Public
- Shared

For details on how to set these user access levels, refer to Chapter 8, "File Transfer To and From the Host Transfer File (HTF)", on page 137.

When users want to receive VSE/VSAM files or VSE/ICCF members to a workstation, they use the ***Move Utilities*** dialog to move that data to the HTF. Afterwards, users can download the data from the HTF to the workstation.

Similarly, when users want to send a workstation-based file to VSE/VSAM or VSE/ICCF, they can upload it to the Host Transfer File. The *Move Utilities* dialog then can be used to work with the data and/or store it.

For details on how to use the *Move Utilities* dialog, refer to Chapter 8, "File Transfer To and From the Host Transfer File (HTF)", on page 137.

Additional considerations

▸ **Migration Considerations:**

Note that you cannot migrate an existing HTF from releases prior to VSE/ESA 1.3. **A new HTF must be created!** For details about how to create the HTF, refer to "Creating the Host Transfer File (HTF)" on page 139.

▸ **Application Interface for HTF:**

VSE/ESA provides an application interface (**API**) which lets you access the HTF from within your CICS application programs. For details see "Application Program Interface (API) for the HTF" on page 173.

The examples in the following section illustrate some commonly used file transfer cases for the Host Transfer File.

All **common options** of the SEND and RECEIVE commands are explained in "Common SEND/RECEIVE Command Syntax" on page 70. For **special SEND/RECEIVE options**, refer to "SEND/RECEIVE Command Syntax and Options" on page 180.

Examples

About this Section

This section provides some examples of how you may use the SEND/RECEIVE commands with option **FILE=HTF**. This option is used for transferring data to and from the Host Transfer File.

An example case starts with a short heading marked by the '►' character.

► **Store a Private File in the HTF:**

User SYSA, for example, may enter the following command:

```
     SEND SETUP.BAT  SETUP BAT
```

• **Explanation:**

- Workstation file SETUP.BAT is stored in the HTF under the name SETUP BAT as a private file. (Option FILE=HTF is **default.**)
- Only user SYSA has access to this file.

► **Store/Retrieve a Public File in/from the HTF:**

User SYSA, for example, may enter the following command:

```
     SEND PROG1.EXE PROG1 EXEBIN  (PUBLIC BINARY
```

• **Explanation:**

- Workstation file PROG1.EXE is stored in the HTF under the name PROG1 EXEBIN as a public file. User SYSA is the file owner.
- All workstation users have read access to this file.
- Option **BINARY** means that the file has been sent 'as is'. No ASCII/EBCDIC translation has taken place.
- Any workstation user could issue the following command:

  ```
  RECEIVE PROG1.EXE PROG1 EXEBIN  (FROM=SYSA BINARY
  ```

  to receive file PROG1 EXEBIN from the HTF to the workstation.

File Transfer To and From Host Transfer File (HTF)

▸ **Store/Retrieve a Shared File in/from the HTF:**

User SYSA, for example, may enter the following command:

```
SEND TEST.DAT  TEST DATA  (FILE=HTF  FOR=JONES
```

- **Explanation:**
  - Workstation file TEST.DAT is stored in the HTF under the name TEST DATA as shared file.
  - User JONES receives a notification message.
  - User SYSA is sharing access to this file with user JONES.
    No other users can access this file.
  - User JONES can issue the following command:

    ```
    RECEIVE TEST.DAT  TEST DATA  (FROM=SYSA
    ```

  - User JONES receives file TEST DATA from the HTF to the workstation.
  - User SYSA is notified that user JONES received the file.
  - Note that user JONES may also access file TEST DATA via the **Move Utilities** host dialog; for example, to move the data to **VSE/ICCF** or to **VSE/VSAM**.

File Transfer To and From CICS TS Temporary Storage

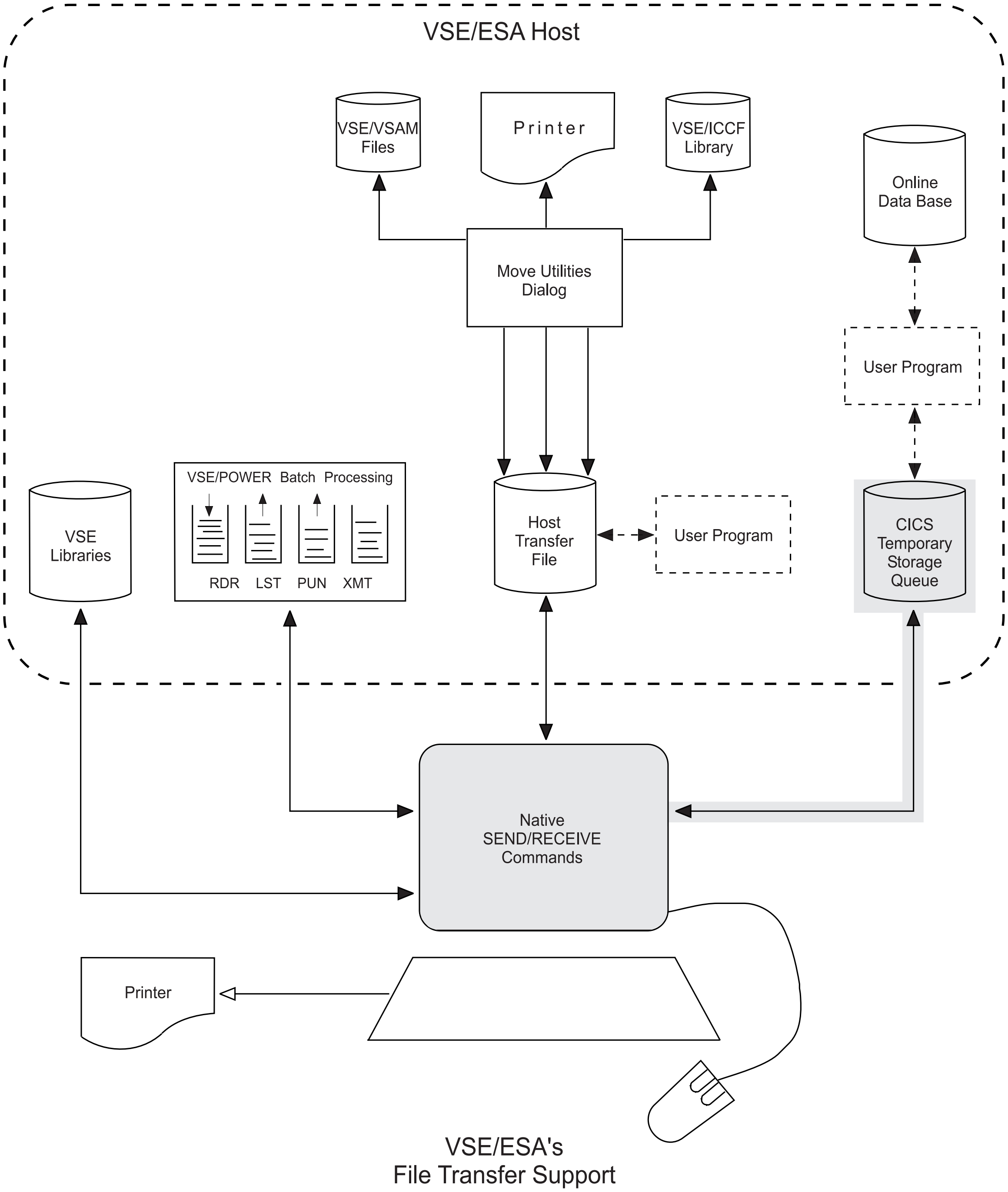


Figure 22. Transferring Data To and From CICS TS

Workstation users can send data to or receive data from CICS temporary storage at the host. Temporary storage is located on disk.

You would send data to CICS TS queues, if:

- You want to store it for retrieval by another user.

- The data is to be retrieved by an online application.

A TS *queue* is created when the **first** file is stored in temporary storage. Additional files can be appended to this same TS *queue*.

Data can be transferred between workstation and temporary storage without further processing. Two workstation users, for example, can exchange data using temporary storage.

You may also want to process data that is sent from a workstation to temporary storage or move data to temporary storage from the host environment. A typical application of the latter would be the retrieval of records from an DL/I data base and storing these records in a temporary storage queue. Workstation users could then retrieve this data from temporary storage.

For such applications, you must provide CICS programs that can retrieve data from temporary storage or store it there. Such programs can be:

- A transaction that is started as a separate task in the host.
- A *user exit* program that the workstation user specifies when sending data to or receiving data from temporary storage. The program must be written according to the specific conventions that are described under “Temporary Storage Conventions” on page 186. VSE/ICCF library 59 has COBOL, PL/I, and Assembler samples of user exit programs.

The examples in the following section illustrate some commonly used file transfer cases for the CICS TS queues.

All **common options** of the SEND and RECEIVE commands are explained in “Common SEND/RECEIVE Command Syntax” on page 70. For **special SEND/RECEIVE options**, refer to “SEND/RECEIVE Command Syntax and Options” on page 199.

Examples

About this Section

This section provides some examples of how you may use the SEND/RECEIVE commands with option **FILE=TS**. This option allows you to transfer files between a workstation and CICS temporary storage.

An example case starts with a short heading marked by the ’►’ character.

► **Sending Data to CICS TS:**

```
User SYSA working with CICS terminal address A002 issues the following command:

     SEND SETUP.BAT  SETUP BAT  ( FILE=TS
```

- **Explanation:**

```
   - Workstation file SETUP.BAT is stored in a new TS queue named CFTRA002.
     The default queue name consists of the characters CFTR and the CICS terminal ID.
```

▸ Receiving Data from CICS TS:

User SYSA working with CICS terminal address A002 issues the following command:

```
RECEIVE SETUP.BAT  SETUP BAT  ( FILE=TS
```

- **Explanation:**
  - User SYSA receives the contents of TS queue **CFTRA002** to workstation file SETUP.BAT.
  - Any other workstation user could issue the command:

RECEIVE SETUP.BAT SETUP BAT (FILE=TS QNAME=CFTRA002 to receive the contents of TS queue CFTRA002.

Chapter 5. Getting Ready for File Transfer

This chapter contains planning and administrative information. It also contains conceptual information about using the file transfer commands and describes all preparatory tasks needed to start with file transfer.

This chapter presents the following topics:

- Supported hardware and 3270 emulation programs.
- Defining the workstation to VSE/ESA.
- Preparing the host session for file transfer.
- Common SEND/RECEIVE command syntax.
- Common SEND/RECEIVE command options.
- Executing SEND and RECEIVE commands from a command procedure.
- File transfer without signon to the VSE/ESA Interactive Interface.
- File transfer on IBM DBCS workstations.
- Data protection.
- Host resource characteristics.

Supported Hardware and 3270 Emulation Programs

3270 Terminal Emulation

VSE/ESA supports any workstation as a 3270 display station, provided that the workstation supports 3270 terminal emulation via a suitable adapter and the corresponding emulation control program.

The workstation is connected and defined to VSE/ESA like a normal 3270 display station (as a local or as a remote SNA or non-SNA workstation).

Workstation File Transfer Support

In addition to 3270 terminal emulation, VSE/ESA provides file transfer support between workstations and a VSE/ESA host system. This support is available for any workstation which runs with one of the *emulation programs* listed below:

Table 2. 3270 Emulation Programs for Workstation File Transfer Support

3270 Emulation Program Name	Supported Version	Operating System	Mode
IBM OS/2 Communication Manager	any	OS/2	DFT
IBM Personal Communication/3270	any	PC-DOS	DFT
IBM PC Emulation Program V3	any	PC-DOS	DFT
IBM PC Emulation Program, Entry Level	any	PC-DOS	CUT
IBM PC/HOST File Transfer and Emulation Program (FTTERM)	any	PC-DOS	CUT
IBM AIX 3270 Host Connection Program/6000 (HCON)	1.4	AIX 3.2	DFT
IBM OS/2 Communication Manager (Japanese Version)	any	OS/2 (Japanese Version)	DFT

Table 2. 3270 Emulation Programs for Workstation File Transfer Support (continued)

3270 Emulation Program Name	Supported Version	Operating System	Mode
IBM PS/55 Japanese 3270 Personal Computer Series	any	PC-DOS (Japanese Version)	DFT

Notes:

1. For information about required **emulation adapter cards** and possible **host connections**, refer to the appropriate programmable workstation and emulation program documentation or contact your IBM marketing representative.
2. The **IBM AIX 3270 Host Connection Program/6000 (HCON)** uses the **XFER** command rather than SEND/RECEIVE commands for file transfer. Note that this manual only describes the format of the SEND/RECEIVE commands. For a description of the XFER command, refer to the *IBM AIX RISC System/6000 Command Reference*
3. VSE/ESA's workstation file transfer supports file transfer buffer sizes from 2 - 32KB, except for **local non-SNA** connections. Here the maximum file transfer buffer size is **31KB** (CICS restriction).

Defining the Workstation to VSE/ESA

Use the Configure Hardware dialog to define a workstation as a local or remote workstation (SNA or non-SNA) to VSE/ESA.

For further information on how to use the "Configure Hardware dialog," refer to the IBM manual *VSE/ESA Networking Support*.

The definition for a workstation specifies a specific *screen size* (for example, 24X80). This screen size must be identical to the screen size that was specified during the configuration (also known as *setup*) of the workstation emulation program.

When defining a workstation to VSE/ESA for file transfer you must know whether it operates in **DFT** or **CUT** mode. If you do not know or cannot find out, assume that your workstation operates in DFT mode.

The *Configure Hardware* dialog clearly indicates how to select either of the two modes.

Note that for workstations operating in **DFT** mode, the Extended Data Stream (EXTDS) feature must be specified for each device associated with a host session in the CICS terminal control table (DFHTCTxx).

If the **CICS Autoinstall feature** is used to define your terminals, the correct VTAM logmode table must be selected. The VSE-supplied logmode tables have an ′**E**′ or ′**Q**′ in position **7** of the table name to indicate extended data stream (for example, SP3272QN).

Note: If you prepare your own logmode table, the query bit X ′80′ must be set in the second byte of the PSERVIC parameter.

Preparing the Host Session for File Transfer

File transfer between a workstation and VSE/ESA is always initiated from the workstation. For this, you can issue a SEND or RECEIVE command in the following ways:

- Enter a SEND/RECEIVE command from the command prompt.
- Via program EHLLAPI.

The 3270 emulation program will convert the SEND/RECEIVE command into an internal format. For example:

```
IND$FILE PUT|GET host-filename (options
```

This converted command will be sent to the host session.

Preparing the Host Session

The VSE/ESA file transfer program is a CICS transaction (IND$). Thus the host session must be in a state where it can execute CICS transactions.

For users signed on to VSE/ESA's Interactive Interface, this means to perform the following preparatory steps:

1. If you are authorized to escape to native CICS, press

   PF6=ESCAPE(U)

   on any panel where this key is offered. In this mode the host session will accept any CICS commands and transactions.
2. If you are **not** authorized to escape to native CICS, follow selection path

   2-6

   from the *VSE/ESA Function Selection* panel, see Figure 2 on page 8 (fast path 3-8-6 as delivered with VSE/ESA).

   or,

   Enter the *Move Utilities* dialog, selection path 2-x (fast path 3-8-x as delivered with VSE/ESA) and press

   PF6=PC-MODE

   wherever this key is offered. In this mode, the host session will only accept the transactions IND$ and INWQ (see below).

Your host session is now ready for file transfer!

Checking the Host Session

If you are using a 3270 emulation program in **DFT** mode and you are using file transfer for the **first time**, you should check whether your host session is properly defined with the EXTDS feature.

For this, enter in the host session:

```
INWQ
```

The host session will respond by displaying:

```
xxxx=DFT or xxxx=CUT
```

in the upper left corner of the host screen, where **xxxx** is the CICS terminal ID (for example, A001=DFT).

Preparing the Host Session for File Transfer

This display depends on whether the EXTDS feature (also known as ″query bit″) is defined (DFT) or **not** defined (CUT) for this terminal.

You are now ready to do file transfer!

Initiating File Transfer

Switch to your workstation session and enter a SEND or RECEIVE command as described in:

- Chapter 6, “File Transfer To and From VSE Libraries”, on page 89.
- Chapter 7, “File Transfer To and From VSE/POWER Queues”, on page 107.
- Chapter 8, “File Transfer To and From the Host Transfer File (HTF)”, on page 137.
- Chapter 9, “File Transfer To and From CICS Temporary Storage”, on page 185.

Common SEND/RECEIVE Command Syntax

SEND/RECEIVE commands are the means by which you use the various functions offered through VSE/ESA’s Workstation File Transfer Support.

To perform file transfer to and from the various VSE/ESA host resources, you enter particular SEND/RECEIVE commands with options special to a resource. However, structure and main syntax elements are the same with all SEND/RECEIVE commands.

The **general format** of SEND/RECEIVE commands is:

```
SEND|RECEIVE pws_name id: file_name file_type (options) comments
```

The above syntax elements signify the following:

pws_name

is the **first** parameter following the SEND or RECEIVE command. It stands for the full file description of the workstation file to be sent to or received from a host resource (**mandatory**).

Specify the location (drive and directory path) specification (filename and file extension/type) of the workstation’s file to be sent to VSE/ESA. Or specify the location and specification at the workstation to where a host file is to be received. For example:

```
d:\root\test.txt
```

id: is the host session that you want to involve during file transfer (**optional**).

The host session ID (**id:**) is the **one-character** short name of the desired session followed by a colon (for example, **a:** or **c:**).

If this option is omitted, the **default** session is used.

file_name

is the name of the file located at the VSE/ESA host to be received at the workstation under pws_name. Or it gives the host’s file name to where the workstation-based file pws_name is to be sent. This parameter is optional. If it is omitted your **user ID** is used (**default**).

Note that for some file transfer functions, a file_name does **not** apply. However, some emulation programs require a file name to be specified. In such a case, specify any name. It will be ignored at the host.

file_type
is the type of the file located at the VSE/ESA host (for example, PROC, PHASE). The filetype is **optional**.

Defaults are assumed depending on the host resource being accessed.

options
are used to specify the actual file transfer requirements (for example, the file transfer **resource** with option FILE=). For the different host resources, there are common and special options available.

Carefully note that your options list must always start with an open parenthesis ′**(**′. The closing parenthesis ′**)**′ is optional except when you use **comments**. Here you also must use the closing parenthesis.

Options can be specified in any order separated by one or more blanks.

comments
can be used for some file transfer functions to pass information to the host.

For details about **common options** available with all VSE/ESA host resources, continue reading the following section.

For information about **special options** that are valid with the SEND/RECEIVE command syntax, refer to the following chapters:

- Chapter 6, "File Transfer To and From VSE Libraries", on page 89.
- Chapter 7, "File Transfer To and From VSE/POWER Queues", on page 107.
- Chapter 8, "File Transfer To and From the Host Transfer File (HTF)", on page 137.
- Chapter 9, "File Transfer To and From CICS Temporary Storage", on page 185.

Examples

About this Section

This section illustrates correct and incorrect use of the syntax available with SEND/RECEIVE.

An example starts with a short heading marked by the ′►′ character.

► **Examples of Valid SEND/RECEIVE Commands:**

```
SEND PCFILE  HOSTFILE

RECEIVE A:PCFILE.TMP  HOSTFILE

SEND D:\MAINDIR\PC.FIL  E:  HOST FILE

SEND VSE.JOB  (  FILE=RDR

SEND VSE.JOB  (FILE=RDR)

RECEIVE A:JOB.MSG  B:  (  FILE=PMSG

RECEIVE PCFILE.PHS HOST PHASE (FILE=LIB BINARY

SEND PCFILE HOSTFILE  (FILE=TS  PROG=CFTREXIT ) this is a comment

RECEIVE PCCOPY HOSTFILE (APPEND FILE=TS

RECEIVE DUMMY (  FILE=PCMD  ) L RDR,*IWS
```

▸ **Examples of Invalid SEND/RECEIVE Commands:**

```
    RECEIVE C:PCFILE
```

• **Explanation:**

```
Host file name (file_name) missing. Although this would be accepted by the host,
the emulation programs require a host file name unless options are also specified.
```

```
    RECEIVE HOSTFILE  C:PCFILE.TXT
```

• **Explanation:**

```
The workstation file name (pws_name)
must always be the first parameter after the SEND/RECEIVE keyword.
```

```
    SEND PCFILE HOSTFILE FILE=PUN BINARY
```

• **Explanation:**

```
Options must be preceded by a left '(' parenthesis.
```

```
    SEND PCFILE (FILE=TS REPLACE this is a comment
```

• **Explanation:**

```
Comments must be preceded by a right ')' parenthesis.
```

```
    SEND PCFILE.EXE HOSTFILE.EXE
```

• **Explanation**

```
Host file name and file type (file_name and file_type) must
be separated by a blank.
```

Common SEND/RECEIVE Command Options

There are two types of *options* which you specify in SEND/RECEIVE commands:

- Common options
- Special options

This section describes common SEND/RECEIVE command options which are applicable to sending and receiving data to and from **all** VSE/ESA host resources.

SEND/RECEIVE command options that are special to host resources are discussed in detail with the description of this specific resource in the appropriate chapter.

The following sections explain the common SEND/RECEIVE options:

```
ILE=
ASCII/BINARY
USERTRT=
CRLF/NOCRLF
BLANK
DCDF
APPEND/REPLACE/NOREPLACE
NLS=
INW1
TRACE
TRACE=
TRACE=
TRACECUT
```

Specifying the VSE/ESA Host Resource (FILE=Option)

The **FILE option** specifies the VSE/ESA resource to which you send or from which you receive your data at the workstation.

For data transfer, use the option FILE= in SEND/RECEIVE commands like this:

DATA TRANSFER:

```
FILE=HTF|LIB|TS|LST|RDR|PUN|LSTX
```

HTF the file is sent/received to/from the Host Transfer File. HTF is the default.

LIB the file is sent/received to/from a VSE library.

TS the file is sent/received to/from a CICS temporary storage queue.

LST the file is sent/received to/from the VSE/POWER list (LST) queue. Here the CICS report controller feature (RCF) is used for formatting and printing the file.

LSTX the file is sent 'as is' to the VSE/POWER LST queue. Here the CICS report controller feature (RCF) is not used.

RDR the file is sent/received to/from the VSE/POWER reader queue.

PUN the file is sent/received to/from the VSE/POWER punch queue.

For file transfer details that are special to using the different VSE/ESA resources, refer to the chapters dealing with that VSE/ESA resource:

- Chapter 6, “File Transfer To and From VSE Libraries”, on page 89
- Chapter 7, “File Transfer To and From VSE/POWER Queues”, on page 107
- Chapter 8, “File Transfer To and From the Host Transfer File (HTF)”, on page 137
- Chapter 9, “File Transfer To and From CICS Temporary Storage”, on page 185

For using the **service functions** offered by VSE/ESA's Workstation File Transfer Support, use option FILE in **RECEIVE commands** by specifying either of the following options:

SERVICE FUNCTIONS:

```
FILE=LDIR|LCMD|PCMD|PMSG|PROF
```

LDIR receive VSE library and sublibrary directory information, that is, the lists of sublibraries and members contained.

LCMD
send Librarian commands for execution to the VSE/ESA host and receive the command output returned by VSE/ESA.

PCMD
send VSE/POWER commands for execution to the VSE/ESA host and receive the command output returned by VSE/ESA.

PMSG
check status of submitted jobs and receive job completion messages.

PROF receive VSE/ESA host information like resource access control information in a host user profile at the workstation.

Specifying the VSE/ESA Host Resource (FILE= Option)

Note: Service functions require the command RECEIVE to be used because the commands may produce output that must be returned to the workstation.

For service function details, refer to chapter Chapter 12, “Using File Transfer Service Functions”, on page 243.

Examples

About this Section: This section illustrates correct use of the FILE option. This option specifies either the host resource in SEND and RECEIVE commands for data transfer or specifies a service function. The examples are given in the order of the FILE parameter descriptions that is given above.

▸ **Examples of Data Transfer:**

▸ Data Transfer to/from the HTF:

```
SEND|RECEIVE PCFILE HOSTFILE
or
SEND|RECEIVE PCFILE HOSTFILE  (FILE=HTF
```

▸ Data Transfer to/from a VSE library:

```
SEND|RECEIVE PCFILE.PRC HOSTFILE PROC  (FILE=LIB
```

▸ Data Transfer to/from CICS TS:

```
SEND|RECEIVE PCFILE HOSTFILE  (FILE=TS
```

▸ Sending a Print File to the VSE/POWER List Queue:

```
SEND PCFILE.LST HOSTFILE  (FILE=LST COPY=3 LINES=50
```

▸ Storing Workstation Data in the VSE/POWER List Queue:

```
SEND PCFILE.EXE HOSTFILE  (FILE=LSTX BINARY
```

▸ Receiving an Entry from the VSE/POWER List Queue:

```
RECEIVE PCFILE HOSTFILE 12345  (FILE=LST
```

▸ Submitting a Job to the VSE/POWER Reader Queue:

```
SEND VSE.JOB  (FILE=RDR
```

▸ Receiving a Job from the VSE/POWER Reader Queue:

```
RECEIVE VSE.JOB  VSEJOB  12345  (FILE=RDR
```

▸ Sending Data to the VSE/POWER Punch Queue:

```
SEND PCFILE.DAT PUNDATA  (FILE=PUN
```

▸ Receiving an Entry from the VSE/POWER Punch Queue:

```
RECEIVE PCFILE.DAT  PUNDATA 12345  (FILE=PUN
```

► **Examples of Service Functions:**

► Receiving a List of all Sublibraries in the VSE PRIMARY Library:

```
RECEIVE PRIMARY.DIR  (FILE=LDIR
```

► Sending a Librarian Command to the VSE Host and Receiving Command Output:

```
RECEIVE LIBR.MSG  (FILE=LCMD)  DEFINE S=PRIMARY.BETA
```

► Sending a VSE/POWER Command to the VSE Host and Receiving Command Output:

```
RECEIVE READER.RDR  (FILE=PCMD) D RDR
```

► Receiving the Job Completion Message for a Job Submitted to the VSE/POWER Reader Queue:

```
RECEIVE VSEJOB.MSG VSEJOB  12345  (FILE=PMSG
```

► Receiving Host User Profile Information to the Workstation:

```
RECEIVE PROFILE  (FILE=PROF
```

Performing ASCII/EBCDIC Translation (ASCII, BINARY, USERTRT=Options)

Data on a workstation exists in ASCII format. Data on the VSE/ESA host has EBCDIC format. When data transferred between a workstation and VSE/ESA is to be further processed at the target system, it must be translated:

- from ASCII to EBCDIC during upload (SEND), and
- from EBCDIC to ASCII during download (RECEIVE).

ASCII/EBCDIC translation is the **default**. The data is automatically translated unless otherwise specified.

ASCII/EBCDIC translation can also be requested explicitly by specifying the ASCII option in the SEND/RECEIVE command for both upload and download.

If ASCII/EBCDIC translation is not desired the **BINARY** option **must** be specified. This would be the case for workstation data that is not processed at the host but stored only for backup purposes or for downloading by other workstation users.

ASCII Option

- Translates transferred data from ASCII to EBCDIC on upload and from EBCDIC to ASCII on download. Translation is done at the host.
- Is the default.
- Implies the CRLF option unless NOCRLF is specified explicitly. (See “Transferring Logical Records and String Data (CRLF, NOCRLF, LRECL=Options)” on page 78).
- Must be specified when the NOCRLF option is active but translation is required.
- For DBCS workstations, ASCII or the DBCS equivalent JISCII must be specified explicitly. The default is ’no translation’. CRLF is **not** implied. Translation is done at the workstation.
- Causes some emulation programs like the IBM OS/2 Communication Manager to do a ’pre-translation’ at the workstation for correct handling of special characters. If this pre-translation is not desired **do not** specify the **ASCII** option explicitly.

BINARY Option

- Must be specified to suppress ASCII/EBCDIC translation for upload and download.
- Implies the NOCRLF option unless CRLF is specified explicitly. (See "Transferring Logical Records and String Data (CRLF, NOCRLF, LRECL=Options)" on page 78).
- Cannot be specified for DBCS workstations. Here 'no translation' is the default.

Examples

▸ **Translate workstation data from ASCII to EBCDIC:**

```
SEND PCFILE HOSTFILE
         or
SEND PCFILE HOSTFILE (ASCII
```

▸ **Translate host data from EBCDIC to ASCII:**

```
RECEIVE PCFILE HOSTFILE
         or
RECEIVE PCFILE HOSTFILE (ASCII
```

▸ **Suppress ASCII/EBCDIC translation:**

```
SEND PCFILE HOSTFILE (BINARY

RECEIVE PCFILE HOSTFILE (BINARY
```

Modifying the ASCII/EBCDIC Translation Tables

ASCII/EBCDIC translation uses standard US translation tables (code pages) which are kept as separate modules in VSE library IJSYSRS.SYSLIB and which are loaded for each file transfer operation that requires translation. These modules are

- **INWPATOE.PHASE** for ASCII to EBCDIC translation
- **INWPETOA.PHASE** for EBCDIC to ASCII translation

If these standard tables do not produce the desired results because, for example:

- Special national characters are not handled correctly or
- You wish to translate the transmitted data to uppercase

you can replace the standard tables using skeletons in VSE/ICCF library 59 or prepare your own tables. Skeletons are available for the following languages:

- US english (INWPENG3)
- UK english (INWPUK3)
- french (INWPFRC3)
- german (INWPGER3)
- italian (INWPITL3)
- spanish (INWPSPN3)

You can do other tables either:

- **Permanently** by storing the new tables with the original names INWPATOE and INWPETOA in the VSE library, or
- **Temporarily** for one file transfer operation by assigning the tables a name of your own choice and specifying this name in the **USERTRT=** option of the SEND/RECEIVE command.

Performing ASCII/EBCDIC Translation (ASCII, BINARY, USERTRT= Options)

When you use your own translation tables **do not** specify the **ASCII** option explicitly. Otherwise, ’pre-translation’ at the workstation might produce undesired results.

Preparing Your Own ASCII/EBCDIC Translation Tables

To help you prepare your own ASCII/EBCDIC translation tables, the source of the standard tables is included in VSE/ICCF library 59:

- **INWPATOE** - for ASCII to EBCDIC translation (SEND)
- **INWPETOA** - for EBCDIC to ASCII translation (RECEIVE)

If you need special translation of any characters, proceed as follows:

1. First copy member INWPATOE and/or INWPETOA to your VSE/ICCF primary library.
2. Edit the member and modify the corresponding characters as needed.
3. If you wish to replace the standard tables **permanently** leave the names INWPATOE/INWPETOA unchanged.
4. If you need multiple copies of the translation tables which are to replace the standard tables **temporarily** for one file transfer operation, change INWPATOE/INWPETOA to names of your own choice.
5. Submit the member to assemble and catalog the tables.
6. If you created new tables with new names, you must define these names in the CICS CSD file (CEDA DEFINE PROGRAM). Otherwise, activate the new tables with the CICS command:

   CEMT SET PRO(xxxxxxxx) NEW

   (where xxxxxxxx is INWPATOE or INWPETOA)

Using Pre-defined Language Support

The skeletons listed on page 76 contain both translate tables in one source member. To activate the translate tables, do the following:

- Submit the member to assemble and catalog the tables.
- If you created new tables with new names, you must define these names in the CICS CSD file (using CEDA DEFINE PROGRAM). Otherwise, activate the new tables with the CICS command:

  CEMT SET PRO(xxxxxxxx) NEW

  (where xxxxxxxx is INWPATOE or INWPETOA)

USERTRT= Option

- Specifies the name of the user-defined translation table which is to **temporarily** override the standard tables for ASCII/EBCDIC translation.
- The translation table must be cataloged as a phase in a VSE/ESA library that can be accessed by the CICS partition.
- The name of the translation table must be defined in the CICS CSD file (using CEDA DEFINE PROGRAM).
- If the specified translation table cannot be loaded the corresponding standard table is used.

Specifying Code Pages (for OS/2 Users)

The IBM OS/2 Communication Manager provides options to specify code pages for ’pre-translation’ at the workstation. These options are processed by the workstation and are not sent to the host as part of the SEND/RECEIVE command.

H(nnn)
Specifies the **host code page** to be used for the host file being sent or received, where ’nnn’ is the code page number.

P(nnn)
Specifies the **workstation code page** to be used for the workstation file being sent or received, where ’nnn’ is the code page number.

For a description of code page numbers see the IBM manual *IBM OS/2 Extended Edition System Administrator’s Guide for Communications, S90X-7946*

Examples

▶ **Specify user-defined translation table:**

```
SEND PCFILE HOSTFILE (USERTRT=SENDTRT

RECEIVE PCFILE HOSTFILE (USERTRT=RCVETRT
```

Transferring Logical Records and String Data (CRLF, NOCRLF, LRECL=Options)

For the purpose of file transfer, workstation files can be grouped in two categories:

- Files that contain **logical records** separated by carriage-return/line-feed (CR/LF) characters. These can be .BAT or .CMD files or any text files created with an editor.
- Files that contain **string data**. That is, the file consists of one contiguous string of characters. Typical examples of string data are .EXE and .COM files.

When transferring files containing logical records, specify the CRLF option to preserve the logical record structure. Each record in the workstation file becomes one record in the host file and vice versa. **CRLF** is the **default**.

To transfer string data specify the NOCRLF option. On upload, the workstation file is, by default, stored in the host file (1) as one record, if the file is smaller than the file transfer buffer (2-32KB) or (2) as *n* records, where *n* is the number of file transfer buffers required to send the file. (You can override this default by defining the size of the host records with the LRECL= option.)

On download, the records from the host file are concatenated to form a contiguous string of characters in the workstation file.

The CRLF and NOCRLF options are described in detail below. The LRECL= option is not a common option and is discussed, wherever applicable, in the chapters describing the individual host resources.

CRLF Option

- CRLF is the default.
- CRLF implies ASCII unless BINARY is specified explicitly.
- On a DBCS workstation CRLF must be specified explicitly. ASCII is not implied.

Upload (SEND):

- The workstation file is assumed to contain logical records delimited by carriage-return/line-feed (CR/LF) characters.
- The CR/LF characters are used by the host to determine where the logical record boundaries are and to break up the data into individual logical host records.

- The CR/LF characters are discarded before the data is stored in the host file.
- Each record in the workstation file becomes one record in the host file.
- If the workstation file has an end-of-file marker (X’1A’), which follows the CR/LF (X’0D’ and X’0A’) characters, this is not stored in the host file.

Download (RECEIVE):

- The host file contains logical records. These are delimited in the workstation file by adding carriage-return/line-feed (CR/LF) characters at the end of each record.
- Each record in the host file becomes one record in the workstation file.
- From each record, trailing blanks are deleted. If the ASCII option is specified with CRLF, EBCDIC blanks (X’40’) are removed. If the BINARY option is specified with CRLF, ASCII blanks (X’20’) are removed.
- An end-of-file marker (X’1A’) is inserted at the end of the workstation file.

NOCRLF Option

- NOCRLF must be specified to treat transferred data as string data rather than as logical records.
- NOCRLF implies BINARY, unless ASCII is specified explicitly
- On a DBCS workstation, NOCRLF is the default. It cannot be specified explicitly.
- On upload, the contents of each file transfer buffer is stored as one record in the host file.
- On download, the host records are concatenated to form one contiguous string of data in the workstation file.

Examples

About this Section: This section provides some examples of how you may use the SEND/RECEIVE commands with the common options CRLF, NOCRLF.

An example starts with a short heading marked by the ’►’ character.

For the following examples, it is assumed that workstation files with a file extension of:

- **TXT** are text files containing logical records delimited by CR/LF characters.
- **STR** are files containing string data.

► **Submit a VSE job from the workstation to the VSE/POWER reader queue:**

```
SEND VSEJOB.TXT (FILE=RDR
         or
SEND VSEJOB.TXT (FILE=RDR CRLF
         or
SEND VSEJOB.TXT (FILE=RDR ASCII
         or
SEND VSEJOB.TXT (FILE=RDR ASCII CRLF
```

- **Explanation:**

```
- The four statements produce identical results: logical record
  handling and ASCII/EBCDIC translation.
```

Transferring Logical Records and String Data (CRLF, NOCRLF, LRECL= Options)

▸ **Send the same job to CICS temporary storage queue CFTRVJOB for retrieval by other workstation users:**

```
SEND VSEJOB.TXT VSEJOB (FILE=TS QNAME=CFTRVJOB NOCRLF
         or
SEND VSEJOB.TXT VSEJOB (FILE=TS QNAME=CFTRVJOB BINARY
         or
SEND VSEJOB.TXT VSEJOB (FILE=TS QNAME=CFTRVJOB BINARY NOCRLF
```

- **Explanation:**

```
- The three statements produce identical results: The contents of VSEJOB.TXT
  are stored in the TS queue as a contiguous string of data
  including CR/LF and end-of-file characters.
```

▸ **Retrieve the job from the CICS TS queue:**

```
RECEIVE JOBCOPY.TXT VSEJOB (FILE=TS QNAME=CFTRVJOB NOCRLF
         or
RECEIVE JOBCOPY.TXT VSEJOB (FILE=TS QNAME=CFTRVJOB BINARY
         or
RECEIVE JOBCOPY.TXT VSEJOB (FILE=TS QNAME=CFTRVJOB BINARY NOCRLF
```

- **Explanation:**

```
- The three statements produce identical results: Since the received string
  still contains CR/LF and end-of-file characters JOBCOPY.TXT contains
  logical records and is identical with the original VSEJOB.TXT.
```

▸ **Retrieve a phase from a VSE sublibrary and send it back to a different sublibrary:**

```
RECEIVE IESAMPLE.STR IESAMPLE PHASE (FILE=LIB NOCRLF L=IJSYSRS S=SYSLIB

SEND IESAMPLE.STR IESAMPLE PHASE (FILE=LIB NOCRLF L=PRD2 S=CONFIG
```

- **Explanation:**

```
- Like in the previous example, the option could be NOCRLF or BINARY or both.
```

▸ **Send logical records but do not translate to EBCDIC:**

```
SEND PCFILE.TXT HOSTFILE (CRLF BINARY
```

▸ **Concatenate host records into a single string and translate from EBCDIC to ASCII:**

```
RECEIVE PCFILE.STR HOSTFILE  (NOCRLF ASCII
```

▸ **Break up a workstation file containing string data into 80-byte records, translate to EBCDIC, and send them to a VSE library:**

```
SEND PCFILE.STR HOSTFILE  (FILE=LIB LRECL=80
```

Keeping Trailing Blanks on Download (BLANK Option)

RECEIVE only: When logical records are downloaded from the host to the workstation with the CRLF option, trailing blanks (EBCDIC X'40') are automatically deleted from the end of each record to save disk space and download time. Specify the

BLANK option

if you do **not** wish trailing blanks to be deleted.

Note: When receiving VSE/POWER queue entries that were not created by a file transfer SEND command but, for example, by a VSE batch job, the records of these entries will **not** contain trailing blanks, even if option BLANK is specified. This is because VSE/POWER trims off trailing blanks of all records created by batch jobs.

Example

▶ **Using the blank option:**

```
RECEIVE PCFILE HOSTFILE (BLANK
```

Specifying a Data Conversion Descriptor File (DCDF Option)

In addition to ASCII/EBCDIC translation (which applies to the entire file) VSE/ESA file transfer offers data conversion on **record** and **field** level. For example, if a host record contains a field with packed decimal numbers, these numbers can be converted on download to ASCII numbers. The format of the records and the type of conversion to be performed on each record field must be defined by the user in a Data Conversion Description File (DCDF).

When field-level data conversion is to be performed during file transfer, the name of the Data Conversion Descriptor File must be specified in the SEND/RECEIVE command with the **DCDF=** option.

DCDF= Option

- Specifies the name of the Data Conversion Descriptor File (DCDF= Option) to be used for field-level data conversion.
- The DCDF must be present in the **Host Transfer File** and must be owned by the user issuing the SEND/RECEIVE command. This is true for file transfer to/from all host resources (HTF, CICS TS, VSE/POWER, VSE libraries).
- In the HTF, the DCDF must have a **filetype** of DCDF or blank. If there are two DCDFs with the same name, one with filetype DCDF and one with a blank filetype, the one with filetype DCDF will be used.
- When the DCDF= option is specified, the options ASCII/BINARY and CRLF/NOCRLF have **no** effect and are ignored, with the following **exception**:

  If on download the converted file is to be appended to an existing workstation file which is delimited by an end-of-file marker (X'1A'), then the **CRLF** option must be specified explicitly. (See also the APPEND option.)
- On a DBCS workstation, ASCII or CRLF must not be specified when the DCDF= option is used.

Examples

▶ **Using the DCDF= option:**

```
Assume that you have prepared a Data Conversion Descriptor File
and stored it in the Host Transfer File as CONVUP DCDF.
You can now request field-level data conversion during
file transfer to any of the host resources:
```

SEND PCFILE HOSTFILE (FILE=TS DCDF=CONVUP

• **Explanation:**

```
  - The system reads file CONVUP from the HTF and then converts each
    PCFILE record according to the specifications in CONVUP before it
    is stored in the TS queue.
```

For detailed information, refer to Chapter 10, “Data Conversion Support”, on page 205.

Appending/Replacing Workstation Files (APPEND, REPLACE Options)

RECEIVE only: When a host file is received at the workstation and a workstation file with the same name already exists, this file is **replaced without warning**. You can, however, also append the new file to an existing file with the **APPEND** option.

APPEND Option

- The downloaded host file is appended to the end of the specified workstation file.
- If the workstation file does not exist, a new file is created.
- If the existing file has an end-of-file marker (X'1A'), the **CRLF** option must be specified **explicitly** to delete the end-of-file marker before the new file is appended.
- If you do not wish the end-of-file marker to be deleted, omit the CRLF option or specify BINARY. **Do not** specify the NOCRLF option.

Examples

▸ **Using the APPEND option:**

```
RECEIVE PCFILE.TXT HOSTFILE (APPEND CRLF

RECEIVE PCFILE.STR HOSTFILE (APPEND BINARY
```

Modifying File Transfer Messages (NLS=, INW1 Options)

For each file transfer operation, a completion or error message is sent to the workstation session as well as to the VSE host session.

The message displayed in the host session is retrieved from the VSE text repository file **IESTRFL**. The message displayed in the workstation session is taken from a message module **INWPMSxx.PHASE** in VSE library IJSYSRS.SYSLIB, where xx is the national language suffix (for example, US for US English, GR for German, SP for Spanish).

Both messages are normally displayed in the language of the VSE host system, regardless of the language on the workstation. For the messages displayed in the workstation session this means that the messages are taken from module INWPMSUS on an English host, from INWPMSGR on a German host, and so on.

VSE/ESA currently offers four national language versions of the file transfer message module:

```
INWPMSUS  - US English
INWPMSGR  - German
INWPMSSP  - Spanish
INWPMSKJ  - Japanese (Kanji)
```

If you want the file transfer messages displayed in the **workstation session** to be in a language different from the host language, you can make this change:

- **Permanent** by renaming the message module. For example, you can rename INWPMSSP to INWPMSUS to get Spanish messages from an English host, or
- **Temporary** for one file transfer operation by specifying the suffix of the desired module in the **NLS=** option of the SEND/RECEIVE command (for example, NLS=SP).

Preparing Your Own Workstation File Transfer Message Module

If you need the workstation file transfer messages in a language other than the ones supported by VSE/ESA, you can generate your own national language version. For this purpose, the source of the US English messages is included in member INWPMS in VSE/ICCF library 59. Proceed as follows:

1. First copy member INWPMS to your VSE/ICCF primary library.
2. Edit the member, and translate the message texts as required. Be careful not to destroy the structure of the member.
3. Rename the US suffix of the module name to a suffix of your own choice. Do not change the INWPMS part of the name.
4. Each message program must be defined to CICS using CEDA. These are the IBM-supplied programs you can use:

   ```
   INWPMSUS - used for US English NLS version
   INWPMSSP - used for Spanish NLS version
   INWPMSGR - used for German NLS version
   INWPMSKJ - used for Japanese Kanji NLS version
   INWPMSCH - used for Chinese ASCII text
   INWPMSKO - used for Korean ASCII text
   INWPMSTA - used for Taiwanese ASCII text
   INWPMSXX - used for messages with TRANSxx prefix instead of INWxxxx
   INWPMSIT - not used
   INWPMSPO - not used
   INWPMSFR - not used
   ```

   If you use a suffix other than those above, you must add a corresponding entry to the CSD file using the CEDA DEFINE PROGRAM.

NLS= Option

- Allows a user to display the file transfer messages sent to the workstation session in a language other than the host language.
- Specifies the two-character suffix of the name of the module which contains the desired messages. The related program must be defined to CICS.
- Results in message INW0098I LOAD ERROR FOR MESSAGE MODULE INWPMSxx, if the specified module does not exist or is not defined to CICS.

Checking for ’File Transfer Complete’ Message (INW1 Option)

This file transfer option is used for workstation programs that generate SEND/RECEIVE commands and must analyze the file transfer messages returned from the host.

VSE file transfer has several different messages which indicate ’file transfer successfully completed’. To facilitate the checking for a ’successful’ condition, the INW1 option can be specified. This will cause all ’successful’ messages to be returned with the same message **number**, INW0001I.

The INW1 option can be specified for all SEND/RECEIVE operations.

Affected messages are:

```
 INW0017I  File transfer complete with BINARY option forced
 INW0022I  File stored in TS queue xxxxxxxx
 INW0024I  File stored in TS queue xxxxxxxx. Control given to program yyyyyyyy
 INW0025I  File received from TS queue xxxxxxxx
 INW0048I  Trace complete - check trace file IWSTRACE in POWER list queue
 INW0049I  Trace complete - check trace file xxxxxxxx in CICS temporary storage
 INW0050I  Trace complete - check TS queue xxxxxxxx or list queue yyyyyyyy
 INW0075I  Queue entry xxxxxxxx yyyyyyy received
*INW0079I  Job xxxxxxxx yyyyyyy successfully submitted. RC=nnnn
 INW0082I  Queue entry xxxxxxxx yyyyyyy created
```

```
 INW0084I  OK
*INW0085I  Librarian command(s) successfully submitted - MAXRC=nnnn

*displayed with prefix INW0001I only if RC=0000 or MAXRC=0000, respectively
```

Examples

▸ **Using the INW1 option:**

```
SEND PCFILE HOSTFILE (FILE=TS

results in message:
INW0022I  File stored in TS queue CFTRA001

SEND PCFILE HOSTFILE (FILE=TS INW1

results in message:
INW0001I  File stored in TS queue CFTRA001
```

Specifying Workstation Emulator Options

Some 3270 emulation programs offer SEND/RECEIVE command options which are processed at the workstation and are not sent to the host system. Such options let you, for example, **define** the size of the **file transfer buffer** or the **period of time** the workstation waits for a response from the host before cancelling the file transfer request.

These options can be specified in SEND/RECEIVE commands but are **not** documented in this manual. For this information, refer to the documentation of the corresponding emulation program.

Examples

About this Section: This section illustrates how you may specify emulator characteristics during file transfer in SEND and RECEIVE commands.

▸ **Setting the File Transfer Timeout Period to One Second:**

```
For IBM OS/2:

     SEND PCFILE HOSTFILE (  T(01)
```

▸ **Setting the File Transfer Timeout Period to 30 Seconds:**

For IBM 3270 Personal Communication:

SEND PCFILE HOSTFILE (TIME(30)

▸ **Set the File Transfer Buffersize (packet size) to 8000 Bytes:**

```
For IBM OS/2:
```

The file transfer buffersize is defined using the Configuration dialog of the Communication Manager (CM).

For IBM 3270 Personal Communication:

SEND PCFILE HOSTFILE (PKTSIZE(8000)

Starting the File Transfer Trace for Problem Determination (TRACE|TRACECUT Option)

The TRACE|TRACECUT option starts the file transfer trace available for problem determination.

You can write the trace records to a CICS temporary storage queue and/**or** to the VSE/POWER list queue (LST). To do this, you can select the following TRACE options:

TRACE	writes to both CICS TS and VSE/POWER list (LST) queue
TRACE=Q	writes to CICS TS queue only
TRACE=LST	writes to VSE/POWER list queue only

Note: Use of the options TRACE, TRACECUT, and TRACE=LST requires the CICS Report Controller Feature (RCF). However, option TRACE=Q does not require the CICS RCF.

The **TRACECUT** option starts the file transfer trace available for problem determination of **CUT-defined workstations**.

Use the TRACECUT option only if you are absolutely sure that your PC runs in CUT mode (Control Unit Terminal attachment). Using the TRACECUT option for a PC in DFT mode will result in a hangup of both the host and the PC sessions.

For details about the trace, refer to Chapter 11, “Using the Workstation File Transfer Trace”, on page 225.

Executing SEND and RECEIVE Commands From a Command Procedure

Command processing at the workstation may at times occur faster than command processing at the CICS host. As a result, if you use workstation command procedures of type BAT or CMD that include series of SEND/RECEIVE commands, the workstation might issue the commands faster than CICS is ready to process them.

To solve this problem, slow down workstation command processing by adding some time-lag to the procedure. You may include a dummy loop between two SEND/RECEIVE commands in the BAT or CMD procedure, for example.

File Transfer Without Signon to the VSE/ESA Interactive Interface

File transfer between VSE/ESA and a workstation generally requires that the user signs on to the Interactive Interface of VSE/ESA. This is because the file transfer function needs information, especially the user ID, from the VSE/ESA user status record which is built during sign-on time.

It is possible, however, to perform file transfer functions to all VSE/ESA resources without signing on to the Interactive Interface. This includes transfer to/from:

- VSE libraries
- VSE/POWER queues
- Host Transfer File
- CICS temporary storage

File Transfer Without Signon to the VSE/ESA Interactive Interface

Using file transfer to and from these resources without signing on to the Interactive Interface entails a number of limitations and restrictions. For details on **restrictions that are special** to a particular VSE/ESA resource, refer to the description given under the corresponding chapter:

- "VSE Libraries Access Without Signon to the Interactive Interface" on page 102
- "VSE/POWER Access without Signon to the Interactive Interface" on page 126
- "HTF Access Without Signon to the Interactive Interface" on page 147
- "CICS TS Access Without Signon to the Interactive Interface" on page 199.

The **common restrictions** which apply to all resources are described in the following sections. They relate to the following areas:

- Required CICS definitions shipped with VSE/ESA
- Use of the CICS Report Controller Feature
- User ID handling
- National language support (NLS)

Do Not Delete ... !

As already stated, file transfer between VSE/ESA and a programmable workstation is designed to work with the Interactive Interface of VSE/ESA and depends on Interactive Interface resources and functions.

However, it is possible to do file transfer to all supported host resources without signon to the Interactive Interface. Be aware, however, that when you delete the Interactive Interface from the VSE/ESA system, you **must not delete** the following CICS definitions:

Name of copybook:	Required entries:
IESWPPT.Z	All
IESZPPT.Z	IESMRTP IESSVL IESCICIN IESNEWS
IESZPPTL.Z	IESMRx IESxSOP, where x is the language indicator (E=English, G=German, S=Spanish J=Japanese)
IESWPCT.Z	IND$ APVU INWQ INWX INWN INWE INWK INWJ INWL IESN
IESZFCTO.A	IESTRFL IESCNTL
IESZPLTI.A	IESCICIN

CICS Report Controller Feature (RCF)

The CICS Report Controller Feature (RCF) is required for the following functions:

- Send to the VSE/POWER list queue with option FILE=LST.
- File transfer trace with the TRACE, TRACECUT, and TRACE=LST options.

If you intend to use any of these functions, **do not** delete the CICS RCF.

User ID

The user ID plays a key role in most file transfer operations. When working without signon to the Interactive Interface it is, therefore, recommended to at least sign on to CICS. In this case, the 1-8 character user ID returned by the EXEC CICS ASSIGN command will be used.

If the user has **not** signed on to CICS, a dummy user ID will be created. It consists of the characters NOII and the CICS terminal ID. For example, when working with terminal D380, the user ID will be NOIID380.

National Language Support (NLS)

Normally, the type of the host language is extracted from the user status record. Without signing on to VSE/ESA, the host language is always assumed to be US English.

The file transfer messages displayed in the workstation session are always in US English. Use the NLS option of the SEND/RECEIVE commands to specify the correct language.

File Transfer on IBM DBCS Workstations

File transfer on an IBM DBCS workstation differs from that on an IBM 3270 workstation in two ways:

- Format of SEND/RECEIVE commands
- Handling of ASCII/EBCDIC translation

Format of SEND/RECEIVE Commands

The host filename specification must always be followed by a left bracket **[** regardless of whether or not any options are specified:

```
SEND PCFILE HOSTFILE [
RECEIVE PCFILE HOSTFILE [ (FILE=TS
```

In the above commands, the left bracket is **not** a syntax notation symbol. It is part of the command.

ASCII/EBCDIC Translation

On a 3270 workstation, ASCII/EBCDIC translation is done on the host. On a DBCS workstation, translation is done on the workstation.

The ASCII option must be specified explicitly if translation is desired. If ASCII is omitted, BINARY is assumed.

The CRLF option must also be specified explicitly. The default is NOCRLF. Thus, for the command

```
SEND PCFILE HOSTFILE [ (FILE=TS
```

the following options are active:

```
FILE=TS BINARY NOCRLF QNAME=CFTRxxxx NOREPLACE
```

Note that on a Japanese DBCS workstation, the keyword ASCII is replaced by JISCII. For example:

```
SEND PCFILE HOSTFILE [ (JISCII CRLF
```

The BINARY and NOCRLF options cannot be specified explicitly.

Data Protection

VSE/ESA provides access protection for VSE libraries, sublibraries and members from unauthorized access by workstation users.

For a detailed description of VSE/ESA security support refer to the *VSE/ESA Administration* and *VSE/ESA Planning* manuals.

Host Resource Characteristics

The following table shows characteristics that are of interest for file transfer purposes.

Table 3. File Transfer - Summary of Host Resource Characteristics

File transfer to/from:	**VSE Library**	**Host Transfer File**	**VSE/POWER**	**CICS TS Queue**
Works w/o signon to VSE/ESA **(2)**	YES	YES	YES	YES
Setup required	NO	YES	NO	NO
Host data storage	permanent	permanent	temporary	temporary
Host record format	fixed-80	variable	variable	variable
Maximum size of one host record when CRLF or LRECL= option is specified	80	32600	32747	32760
Maximum number of host records in one host file	**(3)**	999999	**(3)**	32767
Maximum number of files in host resource	**(3)**	**(3)**	**(3)**	**(4)**
: (1) For almost all file transfer cases no CICS RCF is required. However, the function **SEND (FILE=LST** requires the use of CICS RCF. Also note that the **trace** function (see Chapter 11, “Using the Workstation File Transfer Trace”, on page 225) uses the CICS Report Controller Feature (RCF). However, option TRACE=Q works without using the RCF.				
: (2) For further information about common restrictions refer to “File Transfer Without Signon to the VSE/ESA Interactive Interface” on page 85. For details about resource specific restrictions, refer to: • “VSE Libraries Access Without Signon to the Interactive Interface” on page 102 • “VSE/POWER Access without Signon to the Interactive Interface” on page 126 • “HTF Access Without Signon to the Interactive Interface” on page 147 • “CICS TS Access Without Signon to the Interactive Interface” on page 199				
: (3) Limited only by the space available				
: (4) When files are appended to one *queue*, the sum of the number of records of all files (plus header records) cannot exceed 32767. The number of queues is limited by the space available				

Chapter 6. File Transfer To and From VSE Libraries

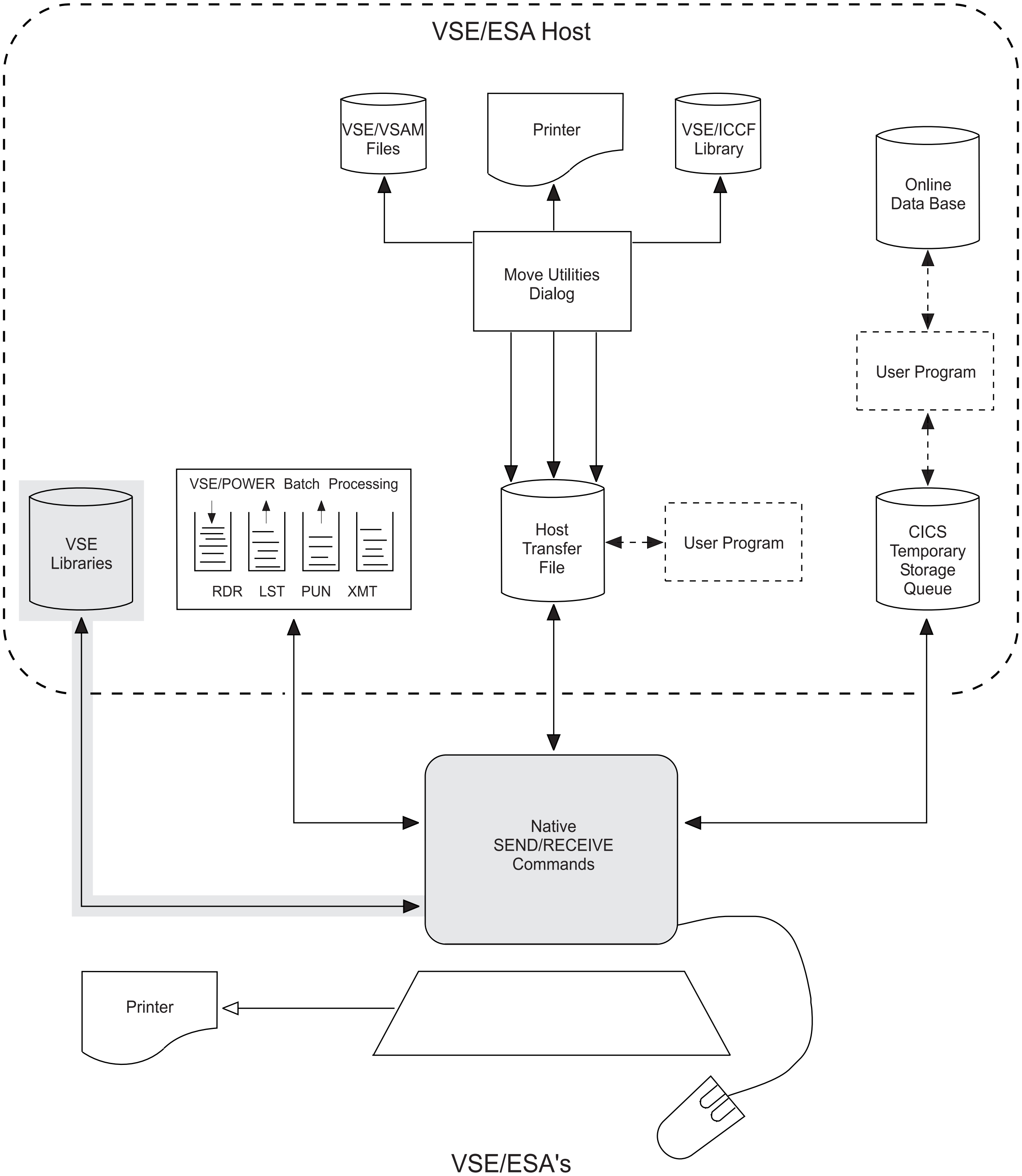


Figure 23. Transferring Data To and From the VSE Libraries

Overview

VSE/ESA provides file transfer functions to and from the VSE libraries. These functions transfer user data via SEND/RECEIVE commands and let you:

- Send a workstation-based file to a VSE sublibrary.
- Receive a VSE sublibrary member to the workstation's file system.

Additional considerations

Note that VSE library access is also supported by the *VSE Navigator* application. For details about this application, refer to the *VSE/ESA e-business Connectors User's Guide*.

The following sections describe data transfer to and from VSE libraries by using SEND and RECEIVE commands.

The Librarian Transaction Server

Library services at the host are provided by the Librarian Transaction Server (LTS) program a copy of which is loaded into a dynamic or static partition for each SEND/RECEIVE request from the workstation. This requires that VSE/POWER and CICS are active.

Attention System Console Operators!

In VSE/ESA batch jobs cannot run while the system console is in redisplay mode. This applies also for the job that executes the Librarian Transaction Server.

Thus, if the system console is left in redisplay mode for a longer period of time, file transfer to/from the VSE libraries may fail with message:

```
INW0074I XPCC function CONNECT failed.  Retcode/Reason = 000C
```

It indicates that no connection could be established between the file transfer program and the Librarian Transaction Server.

Partition Classes

By default the Librarian Transaction Server is loaded into a dynamic partition of class **P**. The VSE predefined system environments 4 and 6 automatically activate class **P** with 32 dynamic partitions.

In a system or environment where this partition class does not exist or is reserved for other tasks, the class for the Librarian Transaction Server must be specified either by:

- The **CLASS= option** in the SEND/RECEIVE command, or by
- Defining up to three partition classes with the *Customize VSE/ESA Workstation Platform* dialog (fast path 216 as delivered with VSE/ESA) of VSE/ESA's Interactive Interface. The corresponding panels are shown in “Defining Partition Classes Via the Interactive Interface” on page 92.

The system determines the partition class for the Librarian Transaction Server as follows:

1. If the **CLASS= option** was specified in the SEND/RECEIVE command, the system attempts to start the server in the specified class. If the server cannot be started in this class, the file transfer request is rejected.
2. If the CLASS= option **was not** specified in the SEND/RECEIVE command, the system checks if partition classes were defined via the *Customize VSE/ESA Workstation Platform* dialog. If so, the system attempts to start the server in one of these classes, trying each class in the order in which they were defined. If the server cannot be started in any of these classes, the file transfer request is rejected.
3. If **no** partition classes were defined as described in 1. and 2., the system attempts to start the server in default class **P**. If the server cannot be started in this class, the file transfer request is rejected.

Whenever a SEND/RECEIVE request to or from VSE libraries is rejected because the Librarian Transaction Server could not be started, message INW0086I is issued. It gives the reason for the failure. For example:

▸ **Message INW0086I:**

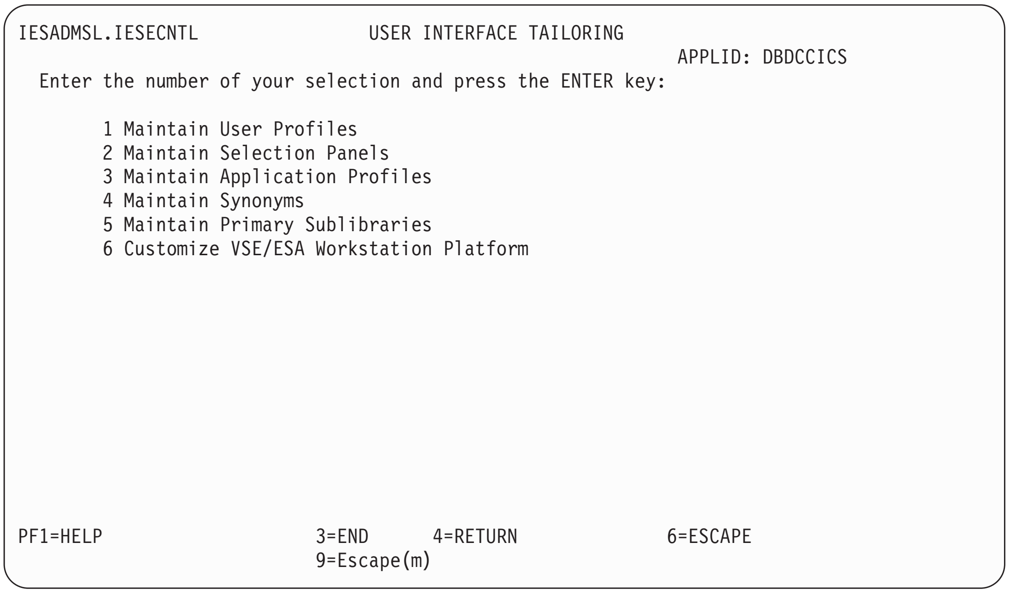
```
INW0086I LTS job could not be started in class(es) CYZ RC=0000 Reason=0800
```

- **Explanation:**

```
- The above message means that the system tried to start the server
  in classes C, Y, and Z without success.

- Reason=0800 means that class Z is disabled or does not exist at all.
  Note that this reason code pertains to the last of the classes only.
  About classes C and Y no definite information is available.
  These classes may have been busy (Reason=0400), for example.
```

Details about return and reason codes for message "INW0086I" are given in the IBM manual *VSE/ESA Messages and Codes, Volume 2*.

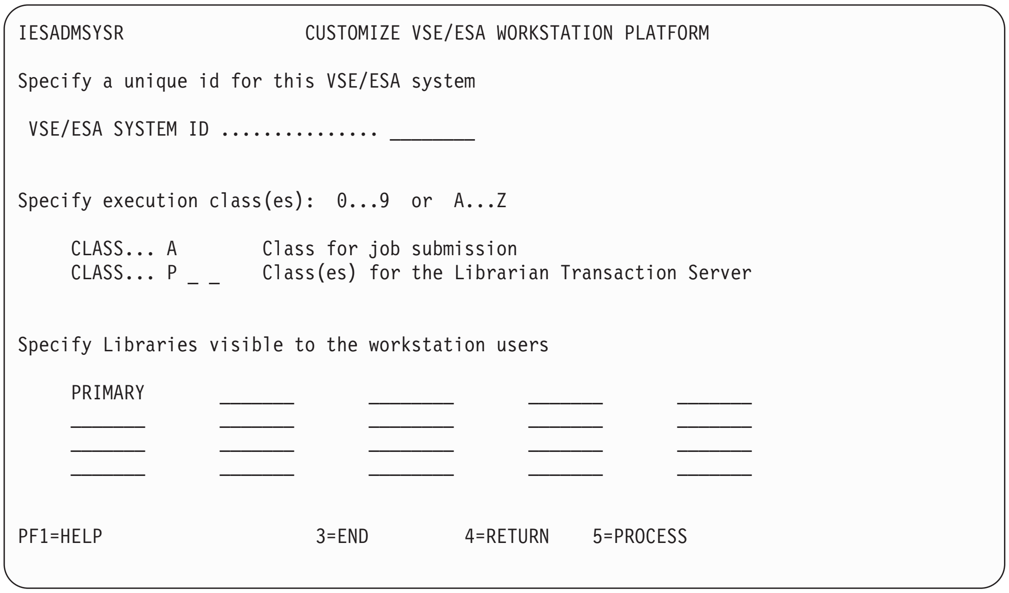
Defining Partition Classes Via the Interactive Interface



```
IESADMSL.IESECNTL                USER INTERFACE TAILORING
                                                             APPLID: DBDCCICS
  Enter the number of your selection and press the ENTER key:

        1 Maintain User Profiles
        2 Maintain Selection Panels
        3 Maintain Application Profiles
        4 Maintain Synonyms
        5 Maintain Primary Sublibraries
        6 Customize VSE/ESA Workstation Platform











PF1=HELP                   3=END      4=RETURN             6=ESCAPE
                           9=Escape(m)
```

Figure 24. User Interface Tailoring

To define partition classes to be used by the Librarian Transaction Server for file transfer, use selection field **CLASS** for the Librarian Transaction Server in the panel *Customize VSE/ESA Workstation Platform* shown below. All other entry fields are not applicable at this point in time (VSE/ESA 1.3.0).



```
IESADMSYSR                 CUSTOMIZE VSE/ESA WORKSTATION PLATFORM

Specify a unique id for this VSE/ESA system

 VSE/ESA SYSTEM ID ............... ________


Specify execution class(es):  0...9  or  A...Z

     CLASS... A        Class for job submission
     CLASS... P _ _    Class(es) for the Librarian Transaction Server


Specify Libraries visible to the workstation users

     PRIMARY       _______      ________      _______      _______
     _______       _______      ________      _______      _______
     _______       _______      ________      _______      _______
     _______       _______      ________      _______      _______


PF1=HELP                   3=END         4=RETURN    5=PROCESS
```

Figure 25. Customize VSE/ESA Workstation Platform

CLASS for Librarian
The class ids of static partitions or the preferable ones of the dynamic partitions which are to be used for the execution of the Librarian Transaction Server. The Librarian Transaction Server is activated each time a VSE library member is accessed from the workstation. A maximum of three classes can be specified. If the first class is not available in the system, the next class given by the user will be taken for the Librarian Transaction Server. The default class is 'P'.

Dynamic Partition Profiles

Although the Librarian Transaction Server can also run in a static partition, it is recommended to use dynamic partitions to benefit from higher availability.

For **default** partition class **P**, VSE provides profile PWSPROF.PROC in IJSYSRS.SYSLIB. It only contains a minimum set of assignments required by the Librarian Transaction Server. For performance reasons, it is recommended to use this profile for all dynamic partition classes that are to be dedicated to the Librarian Transaction Server. If a profile contains LIBDEF statements defining long library chains, this may add several seconds to each file transfer operation.

Since PWSPROF.PROC does not contain any LIBDEF statements the Librarian Transaction Server program **LIBRTPS.PHASE** always must reside in library IJSYSRS.SYSLIB.

System Console Messages

The job that invokes the Librarian Transaction Server is submitted with the parameter LOG=NO in the VSE/POWER JOB statement. Whenever a library file transfer request is issued, only the messages **1Q47I** and **1QC7I** are displayed on the system console.

For example, the display may look like this:

▸ **Messages 1Q47I and 1QC7I:**

```
1Q47I   P1 PWSLTS 03709 FROM (JONES) , TIME=15:37:12, LOG=NO

1QC7I   P1 PWSLTS 03709 FROM (JONES) , TIME=15:37:42, COMPLETED
```

- **Explanation:**

```
- PWSLTS is the name of the job that invokes the Librarian Transaction Server.

- JONES is the user ID of the user who submitted the file transfer request.
```

Note that there are no other messages pertaining to this job on SYSLOG or SYSLST unless execution of the job fails. Execution could fail, for example, because the Librarian Transaction Server program LIBRTPS cannot be found. In such cases, inspect the system console and the VSE/POWER list (LST) queue for information generated by job PWSLTS.

As shipped with VSE/ESA, VSE/POWER suppresses console messages for jobs that have the parameter LOG=NO in their VSE/POWER job statement. This feature can be suppressed by specifying **SET LOGEQNO=IGN** during VSE/POWER startup. Then parameter LOG=NO is ignored and the complete set of console messages is displayed.

The BUF= Option

For communication and data transfer between the file transfer program and the Librarian Transaction Server, a buffer is obtained from CICS main storage. This buffer has a default size of **4096** bytes.

When transferring large amounts of data, performance can be improved by increasing the buffer size. The maximum is 32767 bytes.

If there is not sufficient CICS main storage available, the buffer size also can be reduced. The minimum is 512 bytes.

Buffer size can be specified with the **BUF=** option in the SEND/RECEIVE command. For example:

▸ **Change the Buffer Size:**

```
SEND LARGE.STR LARGESTR BACKUP (FILE=LIB BINARY BUF=20000
```

- **Explanation:**

```
  Overrides the default of 4096 bytes and increases
  the buffer size to 20000 bytes.
```

Library Access

Generally, workstation users can access all VSE libraries via file transfer. VSE ships a predefined library **PRIMARY** and sublibrary **PRIMARY.$$C**. $$C is meant as a common sublibrary for all workstations users.

In addition, whenever a new user is defined using the *Maintain User Profiles* dialog a new sublibrary PRIMARY.userid can be created for this user by selecting *Maintain Primary Sublibraries* (option **5**). This is referred to as the **user's primary sublibrary**.

A file requested to be transferred to the VSE libraries goes by **default** to the user's primary sublibrary. In order to access other libraries, the library and sublibrary names must be specified in **L=** and **S=** options of the SEND/RECEIVE command.

▸ **User JONES issues the command:**

```
SEND PCFILE HOSTFILE (FILE=LIB
```

- **Explanation:**

```
The file is stored in VSE sublibrary PRIMARY.JONES.
```

```
RECEIVE PCFILE HOSTFILE (FILE=LIB L=PRD2 S=CONFIG
```

- **Explanation:**

```
The file is received from VSE sublibrary PRD2.CONFIG.
```

Note: Security aspects for VSE library access are discussed under "Data Protection" on page 87.

Checkin/Checkout

To prevent simultaneous update of one VSE library member by several users, the VSE Librarian supports a lock function.

When a VSE library member is transferred to a workstation it can be protected against write access by other users by specifying the LOCK option in the RECEIVE command. The member is now checked-out. It cannot be updated or deleted by other workstation users or through batch jobs. Only read-access is allowed.

When the updated member is sent back to the host, it can be checked-in with the UNLOCK option in the SEND command.

A library member can only be replaced if the member is locked by the requesting user. Thus, if a lock is reset by the system service, the requestor gets an error message and the member is not replaced. To replace a member which is not locked, specify the UNCOND option.

It is also possible to lock a member after it has been uploaded or to unlock a member after it has been downloaded. Thus, both the LOCK and UNLOCK options can be specified in the SEND as well as in the RECEIVE command.

For SEND/RECEIVE access of a library member with the FILE=LIB option, the user ID is used as the lock ID.

VSE library members can also be locked and unlocked independent from an upload or download operation by submitting the Librarian commands "LOCK" and "UNLOCK" to the host. For a detailed description of these commands refer to the IBM manual *VSE/ESA System Control Statements*.

▸ **Checkin/Checkout:**

```
RECEIVE  PWSJOB  MYMEM JOB  ( FILE=LIB  LOCK
```

• **Explanation:**

```
- Transfers VSE library member MYMEM.JOB from the default sublibrary
  PRIMARY.userid to workstation file PWSJOB.
  Locks the file in PRIMARY.userid against write-access by other users.

- The member is checked-out.
```

```
SEND  PWSJOB  MYMEM JOB  ( FILE=LIB  UNLOCK
```

• **Explanation:**

```
- The member is checked-in, that is, the updated version of MYMEM.JOB is
  sent back to PRIMARY.userid. The lock on the file is removed.

- Note that option UNLOCK implies option REPLACE.
```

Sending and Receiving Fixed-Format Files

Introduction

For the purpose of file transfer between a workstation and the VSE libraries, fixed-format files are used. They contain logical records that can be read and processed at the host by the VSE system and application programs.

Examples of fixed-format files are:

- Source programs and macros which are to serve as input to a language translator (membertype A-Z, 0-9, $, #, @)
- Cataloged procedures (membertype PROC)
- Program object code (membertype OBJ)
- VTAM B-books
- Any member containing fixed-length 80-byte records

Fixed-format files have the following characteristics:

- They must be sent/received with the CRLF option (default).
- In a VSE library, a fixed-format file must consist of fixed-length 80-byte records. In a sublibrary directory display, such files are identified by the letter **R** in the BYTES/RECORDS column.
- At the workstation, fixed-format files to be stored in a VSE library must consist of logical records delimited by the CRLF (X'0D0A') indicator.
- Workstation-based records smaller than 80 characters are padded with blanks on the right before they are stored in a VSE library.

Sending and Receiving Fixed-Format Files

- Workstation-based records longer than 80 characters are truncated on the right before they are stored in a VSE library. A warning message is issued.
- When receiving fixed-format file records from a VSE library at the workstation, trailing blanks are deleted. Specify the BLANK option to suppress deletion of trailing blanks.
- A fixed-format file can be in ASCII or EBCDIC format. Necessary conversion can be requested through the ASCII/BINARY options.
- Fixed-format files can be in uppercase, lowercase format or mixed case. For upload, uppercase translation may be requested by the UC option.

  Note that VSE Job Control statements must be in uppercase. The default is no uppercase translation (NOUC). There is no translation from uppercase to lowercase.
- If a workstation file is a procedure (membertype PROC) which contains SYSIPT data, the DATA=YES option must be specified for upload. DATA=YES is not required to download a procedure.
- Although object modules (membertype OBJ) are in fixed 80 format, they should not be transferred as fixed-format files. Such members require special consideration described in "Sending and Receiving Fixed-Format Files Containing Binary Data" on page 100.

Examples

About this Section

This section provides various examples of how you may use the SEND/RECEIVE commands with option **FILE=LIB** to transfer fixed-format files.

An example starts with a short heading marked by the '▸' character.

How to Use the Examples

The examples assume that the transferred files are to be read or processed at the target system. This means that they must be converted (ASCII/EBCDIC) and must retain their logical record structure. This assumption requires the ASCII and CRLF options which are the **default**. Therefore, these options are not shown in the SEND/RECEIVE examples following.

For backup or offload purposes, fixed-format files can also be treated like string data. This is discussed in the next section.

The SEND/RECEIVE command formats are described in Figure 26 on page 103.

The following examples show how to:

- Use SEND defaults
- Use RECEIVE defaults
- Specify the VSE library and sublibrary during file transfer
- Perform uppercase translation
- Transfer SYSIPT data
- Keep trailing blanks
- Append a received library member to an existing workstation file
- Start the Librarian Transaction Server in a particular dynamic partition class

▸ **SEND Defaults:**

```
SEND  PWSFILE  MYMEM  ( FILE=LIB
```

• **Explanation:**

- **Default** host filetype **PWS** is assumed.
- Member MYMEM.PWS is stored in **default** sublibrary **PRIMARY.userid**.
- PWSFILE records are stored as 80-byte records in MYMEM.PWS (padded or truncated).
- PWSFILE records are translated from **ASCII** to **EBCDIC**.
- PWSFILE records are **not** translated to uppercase.
- If MYMEM.PWS already exists, file transfer is canceled (**NOREPLACE**).

▸ **RECEIVE Defaults:**

```
RECEIVE  PWSFILE  MYMEM  ( FILE=LIB
```

• **Explanation:**

- **Default** host-filetype **PWS** is assumed.
- Member MYMEM.PWS is received from **default** sublibrary **PRIMARY.userid**.
- Each record from MYMEM.PWS is stored as a logical record in PWSFILE (records separated by CRLF).
- MYMEM.PWS records are translated from **EBCDIC** to **ASCII**.
- Trailing blanks in MYMEM.PWS records are deleted.
- If PWSFILE already exists, it is replaced without warning (**REPLACE**).

▸ **Library/Sublibrary Specification:**

Assuming the user has signed on as JONES, all of the following SEND/RECEIVE commands will access sublibrary PRIMARY.JONES:

```
RECEIVE  PWSFILE  MYMEM  JOB  ( FILE=LIB

SEND  PWSFILE  MYMEM  JOB  ( FILE=LIB  L=PRIMARY

RECEIVE  PWSFILE  MYMEM  JOB  ( FILE=LIB  S=JONES

SEND  PWSFILE  MYMEM  JOB  ( FILE=LIB  L=PRIMARY  S=JONES
```

▸ **Uppercase Translation:**

Assume you have prepared a workstation file VSE.PRC which contains a set of VSE JCL statements in lowercase. These are to be stored as a cataloged procedure in your primary sublibrary.
To be able to execute this procedure on the host, the JCL statements must be in uppercase.

```
SEND  VSE.PRC  MYMEM  PROC  ( FILE=LIB UC
```

• **Explanation:**

- The **option UC** causes the entire contents of file VSE.PRC to be translated to uppercase.

Sending and Receiving Fixed-Format Files

▸ **SYSIPT Data in Cataloged Procedures:**

If the procedure from the previous example contains SYSIPT data (for example, an EXEC statement followed by some data records) this must be specified by the DATA=YES option.

```
SEND  VSE.PRC  MYMEM  PROC  ( FILE=LIB  UC  DATA=YES
```

• **Explanation:**

- Note that the data records are also translated to uppercase. If this is not desired, you must provide the JCL statements in uppercase and omit the UC option.

▸ **Keep Trailing Blanks:**

If you do not wish trailing blanks to be deleted when a member is received, specify the **option BLANK**. Enter for example:

```
RECEIVE  VTMX.B80  VTMX  B  ( FILE=LIB  L=PRD2  S=CONFIG  BLANK
```

▸ **APPEND to Workstation File:**

To receive a fixed-format member and append it to an existing workstation file, specify option APPEND. Enter for example:

```
RECEIVE  ALL.PRC  NEXT  PROC  ( FILE=LIB  APPEND  CRLF
```

• **Explanation:**

- The contents of NEXT.PROC is appended to the end of ALL.PRC.

- Note that the **CRLF option** is specified explicitly. This is because the workstation SEND/RECEIVE command processor checks for CRLF and - if CRLF is found - deletes the end-of-file indicator X'1A' from the target workstation file before new records are appended.
  If **CRLF** is omitted, the end-of-file indicator is **not** deleted.
  The workstation file then contains two separate files, which is probably **not** desired.

▸ **CLASS= option:**

```
SEND  VSE.PRC  MYMEM  PROC  ( FILE=LIB  UC  CLASS=C
```

• **Explanation:**

- Starts the Librarian Transaction Server in a dynamic partition of class C.

Sending and Receiving String Data

Introduction

For the purpose of file transfer between a workstation and the VSE libraries, the term string data defines a view of the data rather than its physical representation. Whenever data is transmitted with the **NOCRLF** option, it is treated as string data, regardless of its physical structure.

This form of file transfer can be used to offload or backup workstation data to a VSE library. It also can be used to distribute workstation data among workstation users by storing the data in a common VSE library. Then it can be retrieved by authorized workstation users.

VSE library members containing string data are identified in the sublibrary directory display by the letter **B** in the BYTES/RECORDS column.

Examples

About this Section

This section provides various examples of how you may use the SEND/RECEIVE commands with option **FILE=LIB** to transfer string data.

An example starts with a short heading marked by the '▸' character.

How to Use the Examples

The following examples assume that the data stored in the VSE libraries is not to be processed at the host. It is only stored for backup purposes or for retrieval by other workstation users.

The SEND/RECEIVE command formats are described in Figure 26 on page 103.

▸ **Backup Workstation-Based File:**

```
SEND  MYPROG.EXE  MYPROG  EXE  ( FILE=LIB  NOCRLF

                                        or

SEND  MYPROG.EXE  MYPROG  EXE  ( FILE=LIB  BINARY

                                        or

SEND  MYPROG.EXE  MYPROG  EXE  ( FILE=LIB  BINARY  NOCRLF
```

- **Explanation:**
  - Option BINARY implies NOCRLF and vice versa.
  - MYPROG.EXE is stored in the **default** sublibrary PRIMARY.userid as a contiguous string of binary ASCII data.

▸ **Restore Host-Based File to the Workstation:**

```
RECEIVE  MYPROG.EXE  MYPROG  EXE  ( FILE=LIB  BINARY
```

- Transfers MYPROG.EXE to a workstation.

Sending and Receiving Fixed-Format Files Containing Binary Data

Introduction

Special care must be taken when transferring fixed-format files that contain binary data. Such files, for example, can be VSE object modules (membertype OBJ) or E-books (membertype E).

Assume you need to download a VSE object module to the workstation in order to restore it to a different library or carry it to another VSE system.

To **preserve the logical record structure of the module** you would have to use the CRLF option for download and upload. However, during upload, the module data is searched for CRLF characters **X’0D0A’** to determine record boundaries. If the data happens to contain the character combination X’0D0A’, this would be interpreted as record delimiter, and the module would be destroyed.

Therefore, object modules and other fixed-format files containing binary data must be transferred as follows:

1. **Download** the file from VSE to the workstation with the **option NOCRLF** or **BINARY**. The file now exists on the workstation as a contiguous string of binary data.
2. For **upload**, use the **option BINARY and LRECL=80**. The transmitted data is broken up into 80-byte segments which are stored as logical records in the library member.

Examples

▸ **Backup/Restore an Object Module:**

```
RECEIVE  MOD.OBJ  MOD OBJ ( FILE=LIB  BINARY


SEND  MOD.OBJ  MOD OBJ ( FILE=LIB  S=$$C  BINARY  LRECL=80
```

• **Explanation:** .

```
- Receives member MOD.OBJ from the user's primary sublibrary and restores
  it to the common sublibrary $$C.
```

The SEND/RECEIVE command formats are described in Figure 26 on page 103.

Sending and Receiving VSE Phases

Introduction

In the past, the only way to put an executable phase in a VSE library was to catalog the phase with the Linkage Editor in a batch job or to restore it from a backup tape.

With VSE/ESA’s Workstation File Transfer Support, it is possible to download a phase from a VSE library to the workstation and transfer it to a library of the same or a different VSE system.

The transfer of VSE phases requires special treatment by the system and special consideration by the user:

- Whenever a phase is transferred between a workstation and a VSE library, it is accompanied by its directory information. On download, the directory information precedes the phase data and is stored as part of the resulting workstation file.

  For **upload**, each file being sent is examined for the presence of a directory header. If a header is found, the file is treated like a phase according to the following rules:

  – A phase being sent to the host must have membertype PHASE or a user-defined type. A phase cannot have the system types:
  PROC
  OBJ
  Any of the one-character types.

  On the other hand, a member which is not a phase cannot have the membertype PHASE.

  – A phase contains string data and must always be transmitted with the NOCRLF and BINARY options.

  If other options such as CRLF, ASCII, LRECL=80, or UC are specified when membertype is PHASE, file transfer is terminated.

  With a user-defined membertype, the member is not recognized as a phase and is stored according to the specified options.

  – A phase must **not** be modified at the workstation. Never try to ’apply fixes’ to a phase you downloaded to the workstation. The results would at best be unpredictable.

Note: Phases are destroyed when transferred with the ASCII/CRLF or LRECL= options and a user-defined membertype.

Examples

About this Section

This section provides various examples of how you may use the SEND/RECEIVE commands with option **FILE=LIB** to transfer VSE phases.

An example starts with a short heading marked by the ’►’ character.

The SEND/RECEIVE command formats are described in Figure 26 on page 103.

► **Backup/Restore a Phase:**

```
RECEIVE  MYMOD.PHS  MYMOD PHASE  ( FILE=LIB  BINARY

SEND  MYMOD.PHS  MYMOD PHASE  ( FILE=LIB  S=$$C  BINARY
```

• **Explanation:**

```
- Receives member MYMOD.PHASE from the user's primary sublibrary
  and restores it to the common sublibrary $$C.
```

► **Browse a Phase:**

Note:

```
Browsing the contents of a phase at the workstation should
be done with the Librarian LIST command. For further
information and examples, refer to
"Executing VSE Librarian Commands" on page 251.
```

VSE Libraries Access Without Signon to the Interactive Interface

To access VSE libraries, the Librarian Transaction Server (LTS) is started in a dynamic or static partition of class P, unless the CLASS= option is specified in the SEND/RECEIVE command. In a system without a class P partition, you must specify the class in the CLASS= option.

In a system with security active (via DTSECTAB), a user must at least be signed on to CICS to access the VSE libraries.

For a description of the general restrictions that come into play when transferring files without being signed on to the Interactive Interface, refer to "File Transfer Without Signon to the VSE/ESA Interactive Interface" on page 85.

SEND/RECEIVE Command Syntax and Options

The following shows the SEND/RECEIVE command formats for VSE library access. A description of the parameters and options is given at the end of this section. In case you are not familiar with the notation of the following syntax diagrams, refer to “How to Read SEND/RECEIVE Syntax Diagrams”, on page 273.

Command Syntax - SEND/RECEIVE (FILE=LIB

```
                           ┌─user_id──PWS──────────────┐
►►──SEND──pws_name──┬─────┬─┼───────────────────────────┼─(──FILE=LIB──┤ Options ├──┬───┬──────────►◄
                    └─id:─┘ │               ┌─PWS─────────┐ │                                └─)─┘
                           └─member_name───┤             ├─┘
                                           └─member_type─┘
```

Options:

```
  ┌─ASCII──┐  ┌─CRLF──────────────────┐  ┌─L=PRIMARY──────┐  ┌─S=user_id──────────┐  ┌─NOUC─┐
├─┼────────┼──┼───────────────────────┼──┼────────────────┼──┼────────────────────┼──┼──────┼──────────►
  └─BINARY─┘  ├─NOCRLF────────────────┤  └─L=library_name─┘  └─S=sublibrary_name──┘  └─UC───┘
              │         ┌─NOCRLF─┐    │
              └─LRECL=80┴────────┴────┘

  ┌─CLASS=P─────────┐  ┌─NOREPLACE─┐  ┌─BUF=4096──┐
►─┼─────────────────┼──┼───────────┼──┼───────────┼──┬─────────┬──┬────────┬──┬──────┬──┬──────────┬──►
  └─CLASS=job_class─┘  └─REPLACE───┘  └─BUF=nnnnn─┘  ├─UNLOCK──┤  └─UNCOND─┘  └─INW1─┘  └─DATA=YES─┘
                                                     └─LOCK────┘

►─┬────────────────┬──┬────────────┬──┬─────────────┬──┬───────────────────┬─────────────────────────┤
  └─DCDF=dcdf_name─┘  └─NLS=suffix─┘  ├─TRACE───────┤  └─USERTRT=user_trt──┘
                                      ├─TRACE=Q─────┤
                                      ├─TRACE=LST───┤
                                      └─TRACECUT────┘
```

Figure 26. Send/Receive Workstation Files (fixed-format, string, phase) to/from a VSE Sublibrary (Part 1 of 2)

SEND/RECEIVE Command Syntax and Options (FILE=LIB

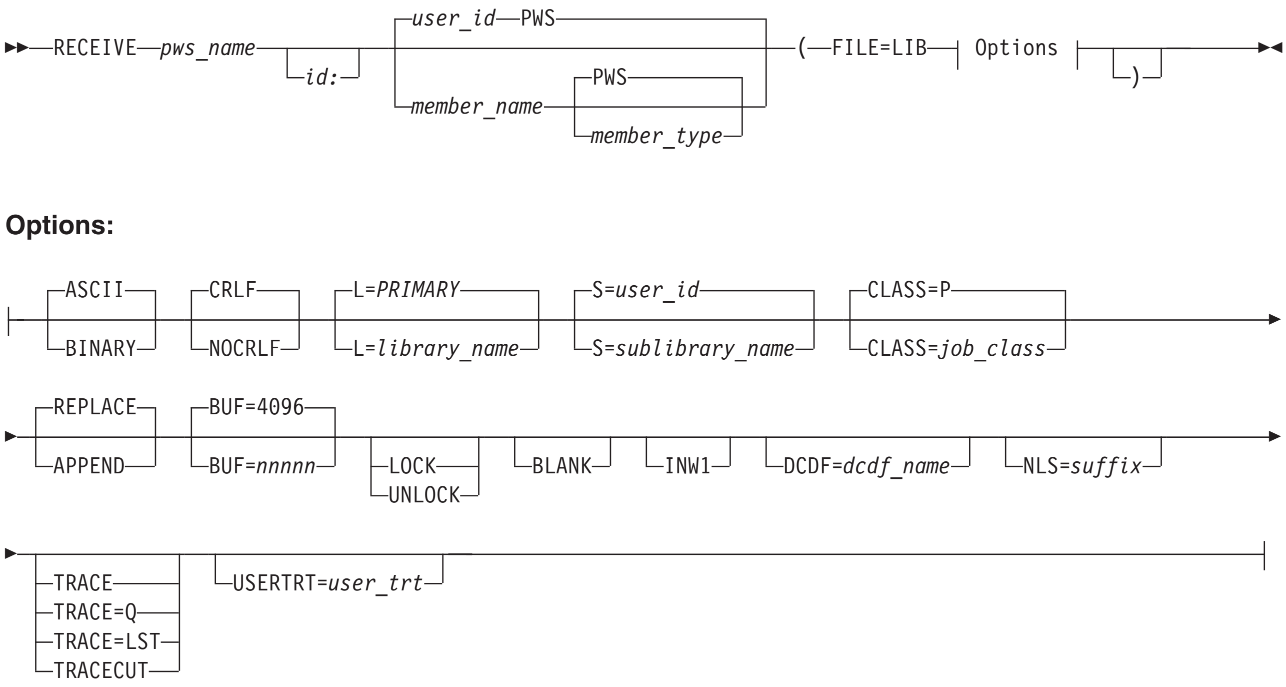


Figure 26. Send/Receive Workstation Files (fixed-format, string, phase) to/from a VSE Sublibrary (Part 2 of 2)

Command Options

The following describes the parameters and options that have specific meaning for file transfer to the VSE libraries.

For a description of the parameters **pws_name** and **id:**, refer to “Common SEND/RECEIVE Command Syntax” on page 70.

For a description of the following **common options**, refer to “Common SEND/RECEIVE Command Options” on page 72.

```
ASCII/BINARY
CRLF/NOCRLFd
APPEND/REPLACE/NOREPLACE
BLANK
INW1
DCDF=
NLS=
TRACE
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=
```

Note: The explanations for the following parameters and options are given separately for SEND and RECEIVE in case there are differences in use or effect. Otherwise, the explanation applies to both SEND and RECEIVE.

member_name
is the name of the member sent or received to/from the VSE sublibrary. Name can be from 1-8 characters long and must be alphameric (A-Z, 0-9, @, $, #). If member_name is omitted, the user ID is used (**default**).

member_type
is the type of the member sent or received to/from the VSE sublibrary. Type can be from 1-8 characters long and must be alphameric (A-Z, 0-9, @, $, #).

member_type is **optional**. However, since a VSE library member must have a membertype, the default type **PWS** is used when this parameter is omitted.

FILE=LIB
defines the library access function as send/receive data files to/from VSE libraries

options
in alphabetic order:

BUF=nnnnn
RECEIVE: defines the size of the receive buffer used to receive data from the host library member. The **default** buffer size is 4096 bytes.

SEND: specifies the size of the send buffer used to pass the workstation data to the VSE Librarian. The **default** buffer size is 4096 bytes.

When transferring very large amounts of data, performance may be improved by increasing the buffer size. **nnnnn** can be a value from 512 to 32767.

A reduction of the buffer size may be necessary when there is not enough main storage available to allocate the default buffer.

CLASS=job_class
specifies the job class for the Librarian Transaction Server. It is strongly recommended to use one of the dynamic job classes. It is, however, also possible to run the Librarian Transaction Server in one of the static partitions.

If the CLASS= option is **omitted**, the server will be started in one of the partition classes defined in the dialog *Customize VSE/ESA Workstation Platform* (see "Defining Partition Classes Via the Interactive Interface" on page 92).

If **no** classes are defined, **default** class P will be used. If a class P partition is not available, the request is denied and a message is returned to the workstation user.

DATA=YES
SEND: must be specified for upload, if the member to be stored is a cataloged procedure (membertype PROC) containing SYSIPT data. SYSIPT data in this context is any data not processed by Job Control (for example, the source code following an EXEC ASSEMBLY statement or any user data processed by a user program).

L=library_name
specifies the name of the VSE library to be accessed (1-7 alphameric characters). If this option is omitted, `PRIMARY` is assumed.

LOCK specifies that the VSE library member is to be locked. The member is protected against write-access by other users.

The **user ID** is used as lock ID.

A locked member can be replaced by the owning user and remains locked. In this case the LOCK option needs not to be specified again.

The member remains locked until it is checked-in with the **option UNLOCK** in a subsequent SEND/RECEIVE command, or the lock is removed with the Librarian UNLOCK command.

LRECL=80

SEND: specify LRECL=80 to upload a fixed-format file that consists of 80-byte records which are **not** delimited by CRLF characters.

The workstation file is treated as a contiguous string of data which is broken up into 80-byte segments. Each segment forms a logical record in the library member.

If the size of the workstation file is **not** a multiple of 80, the last record is padded with blanks. If the ASCII option is active, EBCDIC blanks X'40' are used for padding. If the BINARY option is active, ASCII blanks X'20' are used.

LRECL=80 **overrides the CRLF option**. CRLF is ignored if it is active by default or specified explicitly together with LRECL=80.

NOUC

SEND: no uppercase translation for files sent to a VSE library member. NOUC is the **default**.

S=sublibrary_name

specifies the name of the VSE sublibrary to be accessed (1-8 alphabetic characters). If this option is omitted, the logged-on user ID is taken as sublibrary name.

UC **SEND:** specifies that the contents of the workstation file being sent is to be translated to uppercase. Default is NOUC. Note that VSE accepts job control statements only in uppercase.

UNCOND

SEND: specifies to replace an existing library member which was previously not locked by the requesting user.

UNCOND does not allow to replace a member locked by another user.

UNCOND implies the REPLACE option and is mutually exclusive with the NOREPLACE, LOCK, and UNLOCK options.

UNLOCK

used to unlock a member that was locked with the option LOCK in the SEND/RECEIVE command or with the Librarian LOCK command.

SEND: option UNLOCK implies the option REPLACE.

If the VSE library member does not exist or is not locked, the option UNLOCK is ignored.

Chapter 7. File Transfer To and From VSE/POWER Queues

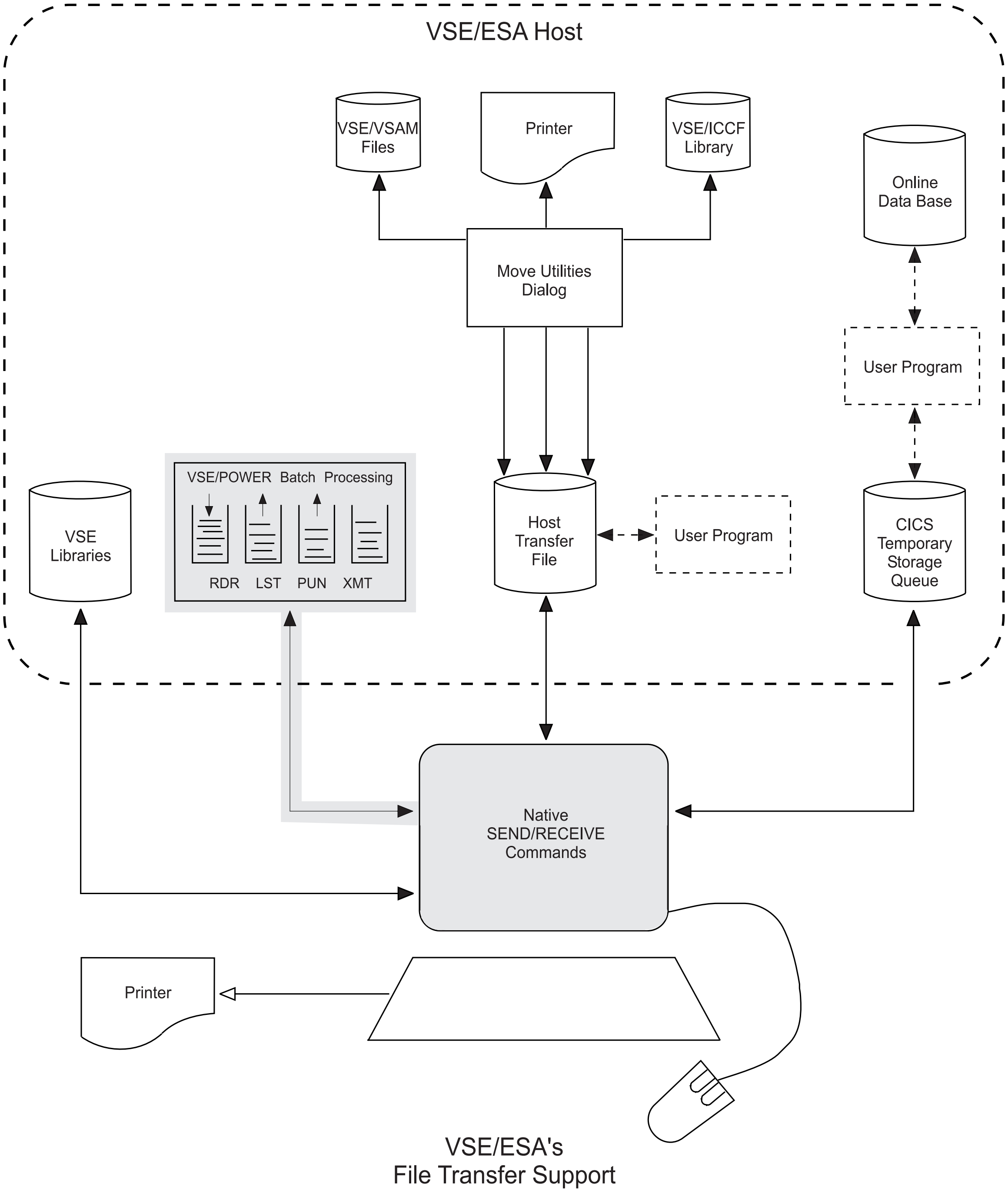


Figure 27. Transferring Data To and From VSE/POWER

Overview

File transfer to/from the VSE/POWER queues can be grouped into the following functional areas:

- Job submission
- Printing workstation files on a VSE/ESA printer
- Offloading workstation files for backup and data distribution purposes

  A summary of the options available for the SEND functions follows after this overview.
- Receiving VSE/POWER queue entries to the workstation.

Additional considerations

VSE/POWER queue access is supported by the *VSE Navigator* application. For details about this application, refer to the *VSE/ESA e-business Connectors User's Guide*.

The following sections describe data transfer to and from VSE/POWER queues by using SEND and RECEIVE commands.

Job Submission

You can send a workstation file that contains a VSE job to the VSE/POWER reader queue for execution in a VSE partition. The command

```
SEND VSE.JOB (FILE=RDR
```

sends the contents of workstation file VSE.JOB to the reader queue. The execution of the job is controlled by the parameters of the * $$ JOB statement within VSE.JOB. Successful execution of the job can be verified with the command

```
RECEIVE VSE.MSG (FILE=PMSG
```

which returns the job completion message including return codes in workstation file VSE.MSG.

For details and examples of job submission, refer to “Job Submission” on page 110.

Printing Workstation Files on a VSE Printer

You can send a workstation file for printing to a VSE system printer or a CICS-controlled 3270 printer and define the output format such as number of copies, number of lines per page, linelength, running heads etc. For example

```
SEND PCLIST HOSTLIST (FILE=LST COPY=3 LINES=55 LINELENGTH=120 NOHEAD
```

Note that this function uses the CICS Report Controller Feature (RCF). If the RCF is not installed or disabled you can still send a workstation file to the VSE/POWER list queue using the FILE=LSTX option (see following section). In this case, however, formatting options such as number of copies, lines per page, etc., are not supported.

For details and examples for printing workstation files, refer to “Printing Workstation Files on VSE Printer” on page 113.

Backup and Data Distribution

You can store (dump) workstation files in the VSE/POWER list or punch queue for backup purposes, or to make these files available to other users on your local or on any remote VSE system. For example, the system administrator could send a workstation file to either the VSE/POWER punch (FILE=PUN) or list (FILE=LSTX) queue for retrieval by all other workstation users:

```
SEND TEST.EXE TEST.EXE (FILE=PUN  BINARY DEST=ANY
                     or
SEND TEST.EXE TEST.EXE (FILE=LSTX BINARY DEST=ANY
```

For backup and data distribution it makes no difference whether you use the list or punch queue.

If you wish to distribute a list queue entry for printing on other VSE systems maintaining the original print format, you must first receive the entry including all print control information. For example:

```
RECEIVE a:VSE.LST VSELIST (FILE=LST CC
```

receives list queue entry VSELIST to diskette file VSE.LST including all print control information via the CC (Control Character) option. At the target system send the file back to the list queue with the command

```
SEND a:VSE.LST VSELIST (FILE=LSTX CC
```

For details and examples of data backup and distribution, refer to “Offloading Workstation Files for Backup and Distribution” on page 117.

Receiving Entries from the VSE/POWER Queues

You can receive any entry from the VSE/POWER reader, list, and punch queue by specifying the entry name, or entry name and entry number:

```
RECEIVE VSE.JOB PAUSEBG (FILE=RDR
RECEIVE VSE.LST CICSICCF 456 (FILE=LST
RECEIVE TEST.EXE TEST.EXE 234 (FILE=PUN BINARY
```

For details and more examples, refer to “Receiving VSE/POWER Queue Entries to the Workstation” on page 120.

Summary of Send Options

Table 4. Send to VSE/POWER Queues - Summary of Options

Function/Specification	Option	Reader Queue FILE=RDR	List Queue FILE=LST	List Queue FILE=LSTX	Punch Queue FILE=PUN
Send print/punch control characters	CC	N/A	Yes	Yes	Yes
Job class	CLASS=	Note	Yes	Yes	Yes
Number of copies	COPY=	N/A	Yes	No	No
Destination user/printer ID	DEST=	Note	Yes	Yes	Yes
Disposition	DISP=	Note	Yes	Yes	Yes
Running header line	HEAD	N/A	Yes	No	N/A
Number of line per page	L=	N/A	Yes	No	N/A
Linelength	LL=	N/A	Yes	No	N/A
Logical record length	LRECL=	Yes	No	Yes	Yes
Destination node	NODE=	Note	Yes	Yes	Yes
Resource security level	RSL=	No	Yes	No	No
Uppercase translation	UC	Yes	No	No	No
Note: For job submission, these values are taken from the * $$ JOB statement.					

The following sections describe data transfer to and from VSE/POWER queues by using SEND and RECEIVE commands.

Job Submission

Introduction

The next pages have various examples of how you may use the SEND command with option **FILE=RDR** to submit jobs from the workstation to the VSE/POWER reader (RDR) queue.

Jobs you send are assumed to contain a valid VSE job stream. There is no checking of control statements by the file transfer function. The job must contain at least a valid VSE job and end-of-job statement, for example like this:

```
// JOB JOBNAME
     .
     .
/&
```

VSE/ESA generates a VSE/POWER job with disposition D and class A, that is, the job will be processed immediately if a partition with class A is or becomes available. If you want to control the job status yourself, you must include VSE/POWER JECL statements.

You may enter, for example:

```
* $$ JOB JNM=NAME,DISP=K,CLASS=B
// JOB JOBNAME
     .
     .
/&
* $$ EOF
```

Rules and Recommendations for Submitting Jobs

1. If errors are detected in the VSE/POWER job statement, the job is not processed but set into hold (H) status.
2. Since VSE job control statements must be in uppercase, the contents of the workstation file being sent is translated to uppercase by default. Translation can be turned off if required.
3. You cannot specify a ″host-filename″ in the SEND command for the VSE/POWER reader queue. If you need to direct this command to a host session other than the default session, specify the session ID preceding the open parenthesis. For example:

   ```
   SEND PCJOB C: ( FILE=RDR
   ```
4. Some 3270 Emulation Programs (for example FTTERM) require a host name to be specified. In that case, specify any name. It will be ignored by the host. Note that for sending to the RDR queue, the host_name (jobname) is taken from the * $$ JOB statement or // JOB statement of the submitted job.
5. If the workstation file being sent contains more than one VSE/POWER job, message **INW0079I** shows information about the last job only. It is, therefore, recommended to have one job per workstation file.
6. The *** $$ EOJ** statement must be the last line in the workstation file. A * $$ EOJ statement followed by one or more blank or non-blank lines will be interpreted by VSE/POWER as the beginning of a new job and result in return code **RC=0002** and a jobname **AUTONAME** in message **INW0079I**.
7. For each submitted job, VSE/POWER collects the job completion message. Job completion messages can be retrieved with the service function FILE=PMSG.

   For details about this service function, refer to “Receiving Job Completion Messages from VSE/POWER” on page 261. The RECEIVE command format is described in Figure 73 on page 265.

Examples

About this Section

This section provides various examples of how you may use the SEND command with option **FILE=RDR** to submit jobs to the VSE/POWER reader queue.

An example starts with a short heading marked by the ′►′ character. The SEND/RECEIVE command formats are described in Figure 28 on page 128. The following examples show how to:

- Submit a job that contains logical records
- Submit a job that exists as binary string
- Retrieve a job completion message
- Suppress job completion messages

Job Submission

▸ **Job Contains Logical Records:** .

Job (in workstation file PWS.JOB):

```
* $$ JOB JNM=pwstest,DISP=d,CLASS=a
     .
     .
* $$ EOJ
```

Command (issued by user JONES):

```
 SEND  PWS.JOB  ( FILE=RDR
```

Message:

```
INW0079I  Job PWSTEST 01234 successfully submitted. RC=0000
```

RDR queue entry:

```
1R46I  PWSTEST   01234  3  D  A    15  FROM=(JONES)
```

• **Explanation:**

- Job records are translated from **ASCII** to **EBCDIC** by default.
- Job records are translated to uppercase (**UC**). If the job contains user data which are not to be translated, the NOUC option must be specified. In this case, job control statements must be provided in uppercase.
- The entry name (PWSTEST) is taken from the *** $$ JOB statement**. It **cannot** be specified in the SEND command.

▸ **Job Exists as Binary String:**

Job:

The example assumes that a job BINJOB was downloaded from a VM/CMS system to workstation file JOB.BIN as a binary string. It originally existed in CMS as an 80-byte record file.

Command (issued by user JONES):

```
 SEND  JOB.BIN  ( FILE=RDR  BINARY  LRECL=80  NOUC
```

Message:

```
INW0079I  Job BINJOB 01235 successfully submitted. RC=0000
```

RDR queue entry:

```
1R46I  BINJOB    01235  3  D  A   234  FROM=JONES
```

• **Explanation:**

- No ASCII/EBCDIC translation is performed (BINARY).
- The job string is broken up at the host into 80-byte segments (LRECL=80).
- No uppercase translation is performed (NOUC).

▸ **Retrieve Job Completion Message:**

For each submitted job, the job completion message is saved by VSE/POWER for later retrieval by the workstation user.

To retrieve a VSE/POWER message use the option FILE=PMSG and enter:

```
    RECEIVE  JOB.MSG BINJOB 1235  ( FILE=PMSG
```

• **Explanation:**

- Requests the completion message for job BINJOB 01235 to be stored in workstation file JOB.MSG.
- When the job has **not** finished execution, the job status is returned in the file transfer message.
- For a detailed description of this service function, refer to "Receiving Job Completion Messages from VSE/POWER" on page 261.

Note:

The RECEIVE command format is described in Figure 73 on page 265.

▸ **Suppress Job Completion Messages:**

For each submitted job, the job completion message is saved by VSE/POWER for later retrieval by the workstation user. Collection of job completion messages can be suppressed with the option **NOMSG**. For this, enter:

```
    SEND  PWS.JOB  ( FILE=RDR  NOMSG
```

Printing Workstation Files on VSE Printer

Introduction

This function allows to send workstation files to the VSE/POWER list (LST) queue for printing. Such files are treated as CICS Report Controller Feature (RCF) reports.

The provided functions are the same as in previous VSE releases but have been enhanced by the options **NOHEAD** and **CC**:

NOHEAD (option)

allows you to suppress the printing of the page header line.

When option NOHEAD is specified there is:

- **no** header,
- **no** time and date,
- **no** page numbering.

Any header text in the comments area of the SEND command is ignored.

CC (option)

allows you to send workstation files to the LST queue whose records are preceded by print control characters.

When CC is specified it is assumed that the first character of each submitted record is a valid machine control character (MCC).

You can also process ASA control characters by using the CC=ASA option. If the file **does not** contain valid control characters in the first position of each record, **message INW0063I** is issued and file transfer is terminated.

Printing Workstation Files on VSE Printer

For the complete list of options that you may use see “Command Options” on page 130.

Printing of workstation-based files is done on an ″as is″ basis. No conversion of workstation print formats into /370 print formats of the host takes place.

Printing can be controlled by the options shown in the table below. These options are one of the following:

- The system **default**.
- Specified in the **SEND** command.
- Set by the CICS Report Controller transactions **CEMS** or **CEOS**.

Table 5. Summary of Print Control Options

PRINT OPTION	DEFAULT	SEND OPTION	CEMS/CEOS
Priority	3	No	Yes
Disposition	L	DISP=	Yes
Class	A	CLASS=	Yes
Number of copies	1	COPY=	Yes
Print destination (user ID)	user ID	DEST=	Yes
Print destination (node)	LOCAL	NODE=	No
Start page - end page	ALL	No	Yes
Separator page (SEP/NOSEP)	NOSEP	No	Yes
Number of lines per page	56	L=	No
Line length	132	LL=	No
Resource security level	PUBLIC	RSL=	No
Running head on each page	HEAD	HEAD/NOHEAD	No
Running foot on each page	none	No	No
Print control characters	NOCC	CC=MCC|ASA	No
Date/Time	Yes	No	No

Examples

About this Section

This section provides various examples of how you may use the SEND command with option **FILE=LST** to send files to the VSE/POWER list queue for printing. These files will be processed as ’reports’ by the CICS RCF.

An example starts with a short heading marked by the ’►’ character.

The SEND/RECEIVE command formats are described in Figure 30 on page 129.

The following examples show how to:

- Use SEND defaults
- Specify job class and disposition
- Specify the number of print copies
- Specify the line length and page depth
- Specify text for the header line
- Suppress printing of the header line

- Send the report to be printed immediately
- Send the report to a remote node

▸ **Defaults:**

```
   User JONES issues the command:
```

SEND PCFILE (FILE=LST

Message:

```
 INW0001I File transfer complete
```

Resulting list queue entry:

```
   1R46I  JONES    00352 3 L A     25   1   TO=(JONES)  FROM=(SYSCICSA)
```

• **Explanation:**

```
   - The contents of PCFILE is translated from ASCII to EBCDIC.

   - Each logical record in PCFILE becomes one line in the list queue entry

   - Since no hostname is specified user ID JONES is the default.

   - Disposition is L (leave), class is A, priority is 3, number of copies is 1.

   - Number of lines per page is 56. Maximum linelength is 132.

   - At the beginning of each page, a header line containing date, time, and page
     number is inserted.

   - For a summary of the defaults and options, refer to
Table 5 on page 114.
```

▸ **Specify Class and Disposition:**

```
   SEND PCFILE HOSTFILE (FILE=LST CLASS=B DISP=D


   Resulting list queue entry:

   1R46I  HOSTFILE 00353 3 D B    25   1   TO=(JONES)  FROM=(SYSCICSA)
```

• **Explanation:**

```
   - Valid classes are A-Z.

   - Disposition can only be L (leave) or D (delete).
```

▸ **Specify Number of Copies:**

```
   SEND PCFILE HOSTFILE (FILE=LST COPY=5

   Resulting list queue entry:

   1R46I  HOSTFILE 00353 3 D B    25   5   TO=(JONES)  FROM=(SYSCICSA)
```

• **Explanation:**

```
   - The number of print copies can be a value between 1 and 255.
```

Printing Workstation Files on VSE Printer

▸ **Specify Line Length and Page Depth:**

```
SEND PCFILE HOSTFILE (FILE=LST LINELENGTH=120 LINES=60
            or
SEND PCFILE HOSTFILE (FILE=LST LL=120 L=60
```

- **Explanation:**

```
- LINELENGTH= or LL= specifies the maximum length of one print line. This can be a value
  between 20 and 205. Records longer than the specified length are truncated.

- LINES= or L= specifies the number of lines per page. A value between 3 and 99 is allowed.
```

▸ **Specify Text for the Header Line:**

```
SEND PCFILE HOSTFILE (FILE=LST) THIS IS THE HEADER TEXT
            or
SEND PCFILE HOSTFILE (FILE=LST HEAD) THIS IS THE HEADER TEXT
```

- **Explanation:**

```
- Each printed page will be preceded by a header line containing the specified text,
  date, time, and page number.

- The length of the header text can be LINELENGTH minus 22, but not longer than 120 characters.

- The header text is translated to uppercase. Extra blank characters are deleted.
```

▸ **Suppress Header Line**

```
SEND PCFILE HOSTFILE (FILE=LST NOHEAD
```

- **Explanation:**

```
- Printing of a header line is suppressed. Thus, there will be no time,
  date, or page numbering in the printout.
```

▸ **Send File for Immediate Printing on System Printer:**

```
SEND PCFILE HOSTFILE (FILE=LST DEST=LOCAL DISP=D CLASS=W
```

- **Explanation:**

```
- The example assumes that the system printer is started with class W.

- Destination LOCAL causes immediate printing.

- Disposition D or K must be specified for immediate printing. (Disposition L is default.)

- The active printer class W must be specified. (Class A is default.)
```

▸ **Send Report to a Remote Node:**

```
SEND PCFILE HOSTFILE (FILE=LST DEST=SMITH NODE=REMVSE

Resulting transmit (XMT) queue entry:

1R46I  HOSTFILE 00357 3 D A L   4 TO=REMVSE(SMITH)  FROM=(SYSCICSA)
```

- **Explanation:**

```
- The report will be sent to user SMITH at remote node REMVSE via VSE/POWER PNET.
```

Offloading Workstation Files for Backup and Distribution

Introduction

The previous section shows how workstation data can be sent to the VSE/POWER LST queue and printed.

This section continues by describing how to send workstation files to the LST and PUN queues for the purpose of:
- Data distribution
- Backup/restore

Sending and receiving data to and from the VSE/POWER LST and PUN queues offers system administrator and workstation users a variety of capabilities. These capabilities, for example, may allow:
- The system administrator to send workstation files to the PUN|LST queue for retrieval by workstation users.
- Workstation users to send workstation files to the PUN|LST queue for retrieval by the system administrator.
- Workstation users to send workstation files to the PUN|LST queue for routing to remote nodes in the network.
- Workstation users to back up workstation files on host magnetic tape by sending them to the PUN|LST queue and using the VSE/POWER POFFLOAD command for backup (and later restore).
- Workstation users to import any data on host magnetic tape to the workstation by storing the data in the PUN|LST queue (using DITTO or a user-written program) and subsequently retrieving it at the workstation.

Examples

About this Section

This section provides various examples of how you may use the SEND command with option **FILE=PUN|LSTX**.

An example starts with a short heading marked by the '►' character.

The SEND/RECEIVE command formats are described in Figure 29 on page 128.

The following examples show how to:
- Use SEND defaults
- Specify the entry name
- Specify the destination user and remote VSE node
- Send a workstation file to a VM user
- Make the PUN queue entry accessible by any users (public entries)
- Specify job class and disposition
- Send binary files
- Define the logical record size

Offloading Workstation Files for Backup and Distribution

▸ **Defaults:**

Command (issued by user JONES):

```
    SEND  BOOK.TXT ( FILE=PUN
```

Message:

```
INW0082I  Queue entry JONES 234 created (All other messages indicate an error.)
```

PUN queue entry:

```
1R46I  JONES     00234  3  L  A   215  1    TO=(JONES)  FROM=(JONES)
```

• **Explanation:**

```
-  Workstation file BOOK.TXT is assumed to consist of logical records (CRLF).
   Each workstation file record becomes one record in the VSE/POWER PUN queue entry.

-  Records are translated from ASCII to EBCDIC.

-  Since no host-name is specified the user ID is used as entry-name.

-  The PUN queue entry is stored with disposition L and class A.
```

▸ **Entry Name:**

Command (issued by user JONES):

```
    SEND  BOOK.TXT  BOOK.TXT  ( FILE=LSTX
```

LST queue entry:

```
1R46I  BOOK.TXT  00235  3  L  A   215  1    TO=(JONES)  FROM=(JONES)
```

• **Explanation:**

```
-  Workstation file BOOK.TXT is assumed to consist of logical records (CRLF).
   Each workstation file record becomes one record in the VSE/POWER LST queue entry.

-  Records are translated from ASCII to EBCDIC.

-  The specified entry-name BOOK.TXT is assigned to the list queue entry.

-  The LST queue entry is stored with disposition L and class A.
```

▸ **Destination User and Node:**

To send a file to the punch queue at a remote VSE node for a specific user, specify the DEST= and NODE= options.

Command (issued by user JONES):

```
    SEND  BOOK.TXT  BOOK.TXT  ( FILE=PUN  DEST=SMITH  NODE=VSESP2
```

XMT queue entry:

```
1R46I  BOOK.TXT  00238  3  L A P     215      TO=VSESP2(SMITH)    FROM=(JONES)
```

▸ Send Workstation File to a VM User:

If you have VSE running as a guest under VM you can send a workstation file directly into the reader of any user on the VM host machine.

Command

```
    SEND  BOOK.TXT  BOOK.TXT  ( FILE=PUN  DEST=JONES  DISP=D  CLASS=W
```

• Explanation:

- The workstation file BOOK.TXT is sent to the reader of VM user JONES, provided that the VSE punch queue was properly started for VM and class W. The following VSE/POWER command at the VSE system console would do this:

  ```
  PSTART PUN,00D,W,,VM
  ```

▸ Public Entries:

To make a punch queue entry accessible by any users specify ANY in the DEST= option.

Command (issued by user JONES):

```
    SEND  BOOK.TXT  BOOK.TXT  ( FILE=PUN  DEST=ANY
```

PUN queue entry:

```
   1R46I  BOOK.TXT  00235  3  L  A   215  1    TO=(ANY)    FROM=(JONES)
```

• Explanation:

- Entries with a destination of ANY can be received by any users. They only can be manipulated (alter, delete) by authorized users. These users are allowed to manage all VSE/POWER queues. The authorization is defined via the "Maintain User Profiles" dialog of the Interactive Interface.

▸ Job Class and Disposition:

Command (issued by user JONES):

```
    SEND  BOOK.TXT  BOOK.TXT  ( FILE=LSTX  CLASS=Q  DISP=D
```

LST queue entry:

```
   1R46I  BOOK.TXT  00236  3  D  Q   215  1    TO=(JONES)  FROM=(JONES)
```

• Explanation:

- Valid classes are **A-Z** and **0-9.**

- Valid dispositions are **D,L,H** and **K.**

Offloading Workstation Files for Backup and Distribution

▸ **Binary Files:**

```
Command (issued by user JONES):

    SEND  PROG.EXE  PROGBIN  ( FILE=PUN  BINARY  BUF=20000

PUN queue entry:

   1R46I  PROGBIN   00237  3  L  A    12  1    TO=(JONES)  FROM=(JONES)
```

• **Explanation:**

```
   -  No ASCII/EBCDIC translation is performed.

   -  The unit of transfer is the send buffer (BUF= ). Each record in the PUN queue
      entry has the size of the send buffer, 20000 bytes. The default is 4096 bytes.
      If the workstation file being sent is smaller than the send buffer,
      the resulting PUN queue entry will consist of a single record.
```

▸ **Defining Logical Record Size:**

```
If an application requires the PUN queue entry to have a specific record size,
you can specify the LRECL option:

Command (issued by user JONES):

    SEND  PROG.EXE  PROGBIN  ( FILE=PUN  BINARY  LRECL=255

PUN queue entry:

   1R46I  PROGBIN   00238  3  L  A    48  1    TO=(JONES)  FROM=(JONES)
```

• **Explanation:**

```
   The contents of the workstation file is broken up into 255-byte segments, each of which
   becomes a record in the PUN queue entry.
```

Receiving VSE/POWER Queue Entries to the Workstation

Introduction

Receiving entries from the VSE/POWER queues includes the following functions:

- Retrieve a specific entry by specifying the name and the number of the entry.
- Retrieve only a specified number of lines/records from the top and/or bottom of the entry, rather than the complete entry.
- Specify the HOLD option which will change the disposition of an entry as follows:
  - D will change to disposition H (hold).
  - K will change to disposition L (leave).
- Receive from the VSE/POWER reader queue.

  If you receive a RDR entry, note that VSE/POWER does **not** return * $$ JOB and EOJ statements. These statements are generated by the file transfer program. Therefore, the job statement will only contain the current:
  - Jobname
  - Class
  - Disposition
  - Priority parameters.

Other information contained in the original job statement is **not** returned.

- Receive from the VSE/POWER list and punch queue.

  If you receive a LST or PUN entry, you can:

  – Retrieve records with a length up to 32KB characters.

  – Retrieve individual segments of segmented output.

Rules and Recommendations for RECEIVE

- A user can retrieve any entry from a VSE/POWER queue, unless the VSE/ESA’s Interactive Interface user profile explicitly states that this user is **not** authorized to manage all batch queue entries.

  If this is the case the user can access only those entries that have his user ID or ANY in the TO= or FROM= queue parameters.
- If **no** entry_name is specified, the first entry with disposition D or K and class A is received.
- If entry_name but **no** entry_number is given, and there are several entries with the same name, the first entry in the queue is retrieved, regardless of class or disposition. In case that the first entry is active, nothing is received.
- To receive a specific entry, specify entry_name and entry_number and/or class. Specify the class whenever it is known. This will improve performance.
- To **delete** an entry after it has been received, the DELETE option must be specified. KEEP is the default.
- To receive **segmented** LST or PUN output, entryname, entrynumber, and segment number (S= option) must be specified.
- To receive an entry with a **one-character name**, jobnumber or CLASS= must be specified in addition to the name.
- **Active entries** (identified by an asterisk (*) in the disposition column in the queue display) **cannot be received**.

The format of the RECEIVE command from the RDR, LST, and PUN queue is shown in detail under Figure 31 on page 130.

Examples

About this Section

This section provides various examples of how you may use the RECEIVE command with option **FILE=RDR|LST|PUN**.

An example starts with a short heading marked by the ’►’ character.

How to Use the Examples

Note that the following examples apply to all VSE/POWER queues (RDR, LST, PUN) unless they are explicitly restricted to a specific queue.

The SEND/RECEIVE command formats are described in Figure 31 on page 130.

The following examples show how to:

- Use RECEIVE defaults
- Receive a file with a specified entry name
- Specify the file’s entry name, entry number, and class
- Change the disposition to HOLD after the file has been received
- Change the disposition to DELETE after the file has been received
- Receive output segments (LST and PUN only)

- Handle print control characters
- Receive only a number of records (TOP and BOT options)
- Receive binary (string) data
- Receive a job

▸ **Defaults:**

Sample LST queue:

```
    1R46I    LIST QUEUE   P D C S  PAGES CC FORM
    1R46I  GETLIST  02714 3 D A        5  1   TO=(USER1)   FROM=(USER1)
==> 1R46I  GOODLIST 02715 3 D A        5  1   TO=(JONES)   FROM=(USER1)
    1R46I  HOSTLIST 02712 3 L A       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  HOSTLIST 02710 3 H A       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  SYSALIST 02713 3 D B        5  1   TO=(JONES)   FROM=(JONES)
```

Command (issued by user JONES, who is authorized to access only his own entries):

```
RECEIVE  FIRST.LST  ( FILE=LST
```

Message:

```
INW0075I  Queue entry GOODLIST 2715 received    (All other messages indicate an error.)
```

• **Explanation:**

- Entry GOODLIST is received.
- If **no** entry name is specified, the first entry with disposition D or K and class A owned by user JONES is received.
- Each record from GOODLIST becomes a logical record in FIRST.LST.
- Records are translated from EBCDIC to ASCII.
- GOODLIST is **not** deleted from the LST queue. Option KEEP is **default**.
  Note: This is different from previous VSE/ESA releases where the default was DELETE.
- If JONES were authorized to access all queue entries, GETLIST would have been returned.

▸ **Entry_name:**

Sample LST queue:

```
    1R46I    LIST QUEUE   P D C S  PAGES CC FORM
==> 1R46I  HOSTLIST 02712 3 D B       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  HOSTLIST 02710 3 L B       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  HOSTLIST 02708 3 D C       15  1   TO=(JONES)   FROM=(JONES)
```

Command (issued by user JONES):

```
RECEIVE  HOSTLIST.LST HOSTLIST ( FILE=LST
```

• **Explanation:**

- Entry HOSTLIST 02712 is received.
- If several entries with the same name exist and **no** entry_number or class is given, the **first entry** in the queue is received.

▸ **Entry_name + Entry_number + Class:**

To receive entry HOSTLIST 02708, user JONES would specify either the entry_number or the class in the RECEIVE command.

Sample LST queue:

```
    1R46I    LIST QUEUE   P D C S  PAGES CC FORM
    1R46I  HOSTLIST 02707 3 D A       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  HOSTLIST 02712 3 D A       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  HOSTLIST 02710 3 L A       15  1   TO=(JONES)   FROM=(JONES)
==> 1R46I  HOSTLIST 02708 3 H B       15  1   TO=(JONES)   FROM=(JONES)
```

Command (issued by user JONES):

```
RECEIVE  HOSTLIST.LST HOSTLIST  2708  ( FILE=LST

                                  or

RECEIVE  HOSTLIST.LST HOSTLIST  ( FILE=LST  CLASS=B

                                  or

RECEIVE  HOSTLIST.LST HOSTLIST  2708  ( FILE=LST  CLASS=B
```

- **Explanation:**

  - All commands will receive entry HOSTLIST 2708.
  - When the entry_number is specified, the entry is received regardless of class or disposition.
  - When entry_number is omitted, the first entry with the specified name and class is received.
  - Specifying the class will improve performance. Therefore, the class should be specified whenever it is known.

▸ **HOLD:**

To change the disposition of an entry after it has been received specify the HOLD option:

Sample LST queue:

```
    1R46I    LIST QUEUE   P D C S  PAGES CC FORM
==> 1R46I  HOSTLIST 02707 3 D A       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  HOSTLIST 02712 3 K A       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  HOSTLIST 02710 3 L A       15  1   TO=(JONES)   FROM=(JONES)
    1R46I  HOSTLIST 02708 3 H B       15  1   TO=(JONES)   FROM=(JONES)
```

Command (issued by user JONES):

```
RECEIVE  HOSTLIST.LST  ( FILE=LST  HOLD
```

- **Explanation:**

  - The above command will receive entry HOSTLIST 2707 and change the disposition to H (hold).
  - When the same command is issued again, the next entry HOSTLIST 2712 is received, and the disposition is changed to L (leave).
  - **Without the HOLD** option the same entry (2707) would be received over and over again.

Receiving VSE/POWER Queue Entries to the Workstation

▸ **DELETE:**

Unlike with releases prior to VSE/ESA 1.3, received entries are not automatically deleted. Specify option DELETE to delete a received entry from the queue. For this, you may enter RECEIVE commands in the following ways:

```
RECEIVE  HOSTFILE.PUN  ( FILE=PUN  DELETE

RECEIVE  HOSTJOB.RDR  MYJOB ( FILE=RDR  DELETE

RECEIVE  HOSTLIST.LST  HOSTLIST  2712  ( FILE=LST  DELETE
```

▸ **Receive Output Segments (LST and PUN only):**

VSE/POWER LST and PUN output segments have the same entry name and number and must be identified with the **S= option**. The FULL=YES parameter of the VSE/POWER PDISPLAY command shows the segment numbers (for example, S=001).

Sample LST queue:

```
    1R46I    LIST QUEUE   P D C S  PAGES CC FORM
==> 1R46I  LST1     02710 3 D A      100  1      TO=(JONES)   FROM=(JONES)
           D=07/13/91  S=001  L=06056  DBGP=475
==> 1R46I  LST1     02710 3 D A       85  1      TO=(JONES)   FROM=(JONES)
           D=07/13/91  S=002  L=02789  DBGP=273
```

Command (issued by user JONES):

```
RECEIVE  LST1.LST  LST1  2710  ( FILE=LST  S=1
RECEIVE  LST1.LST  LST1  2710  ( FILE=LST  S=2  CRLF  APPEND
```

• **Explanation:**

- Segment 1 of the listing is received into workstation file LST1.LST.
- Segment 2 of the listing is received and appended to the end of LST1.LST. Note that **CRLF** is specified explicitly to tell the RECEIVE command processor at the workstation to remove the end-of-file mark from LST1.LST before the new file is appended.
- Output segments can be retrieved in any order.

▸ **Handle Print Control Characters:**

Assume that you wish to receive an entry from the VSE/POWER list queue to the workstation to make some small changes to the text and send it back for printing.

In order to maintain the original print format, the entry must be received and sent including the print control characters and commands.

```
RECEIVE MYLIST.MCC MYLIST  (  FILE=LST  CC

SEND MYLIST.MCC MYLIST  (  FILE=LSTX  CC=MCC
```

• **Explanation:**

- The RECEIVE command specifies the option CC which returns the VSE/POWER list queue entry MYLIST including all print control characters and commands to workstation file MYLIST.MCC.

- For this example it is assumed that we have MCC control characters.

- When you edit file MYLIST.MCC you will find the print control characters in position one of each line. Since the ASCII option is active on the RECEIVE command, the contents of the list queue entry including print control characters is translated from EBCDIC to ASCII.Therefore, you will not be able to recognize these characters as valid MCC control characters.

- If you have a need to inspect the control characters in their original form, specify the options BINARY CRLF in the RECEIVE command. Print control characters are hexadecimal characters and can only be viewed with a hex display function on your workstation.

- Be careful not to destroy the control characters when you edit the data.

- To send the file back to the VSE/POWER queue you must specify the CC=MCC or CC option (MCC is default). Specify MCC=ASA if you have ASA print control characters.

- When the option CC is found on the SEND command, the first character of each record is assumed to be a print control character. It is stripped off the record and passed to VSE/POWER in the record prefix.

- The print control characters are not checked for validity. If you send a file which does not contain print control characters with the CC option, the first character of each data record will be lost.

▸ **Top and Bottom:**

Rather than receiving an entire entry, you can specify to get only a number of records from the top or from the bottom of the entry, or both.
For this, you may enter RECEIVE commands in the following ways:

```
RECEIVE  LST1.TOP  LST1  2715  ( FILE=LST  TOP=5

RECEIVE  LST1.BOT  LST1  2715  ( FILE=LST  BOT=25

RECEIVE  LST1.BTH  LST1  2715  ( FILE=LST  TOP=15  BOT=30
```

Receiving VSE/POWER Queue Entries to the Workstation

▸ **Binary (String) Data:**

```
Entries in the PUN queue are very likely to contain binary data or data that was stored
in the queue in binary format:

    RECEIVE  PROG.EXE  PROGBIN  1415  ( FILE=PUN  BINARY
```

• **Explanation:**

```
-  No EBCDIC/ASCII translation is performed.

-  The individual records from entry PROGBIN are concatenated
   to form a contiguous string of binary data in workstation file PROG.EXE.
```

▸ **Receive a Job:**

```
RDR queue entry:

   1R46I  PWSTEST   01234  3  L  B    15  FROM=JONES

Original Job:

       * $$ JOB JNM=PWSTEST,DISP=L,CLASS=B,PRI=9,USER='USER INFO'
       * $$ LST ...
       * $$ PUN ...
       // JOB NAME
           .
           .
       /&
       * $$ EOJ

Command (issued by user JONES):

    RECEIVE  PWSTEST.JOB  PWSTEST  ( FILE=RDR
```

• **Explanation:**

```
      -  The job is received as shown above
         except for the USER= parameter in the JOB statement.
```

Note:

```
VSE/POWER does not return * $$ JOB and EOJ statements.
These statements are generated by the file transfer program.
Therefore, the job statement will only contain the current:

- Jobname
- Class
- Disposition
- Priority parameters.

Other information contained in the original job statement is not returned.
```

VSE/POWER Access without Signon to the Interactive Interface

In a system without the Interactive Interface, all users have unlimited access to the VSE/POWER queues. It is the system administrator’s responsibility to assign the necessary security keys if restricted access to the VSE/POWER queues is desired.

Without signon to the Interactive Interface a user has unlimited access to all VSE/POWER queues and queue entries.

Exception: To submit jobs to the VSE/POWER reader queue in a system with security active (via DTSECTAB) the user must at least be signed on to CICS .

VSE/POWER Access Without Signon to the Interactive Interface

For a description of the general restrictions that come into play when transferring files without being signed on to the Interactive Interface, refer to “File Transfer Without Signon to the VSE/ESA Interactive Interface” on page 85.

Migration Considerations

File transfer to and from VSE/POWER queues as it existed in releases prior to VSE/ESA 1.3 will continue to work with the following modifications:

- The following file transfer messages are either changed or added:

  For SEND (FILE=RDR
  Message *INW0001I FILE TRANSFER COMPLETE* changes to *INW0079I JOB name number SUCCESSFULLY SUBMITTED RC=xxxx*.

  For RECEIVE (FILE=RDR|PUN|LST
  Message *INW0001I FILE TRANSFER COMPLETE* changes to *INW0075I QUEUE ENTRY name number RECEIVED*.

  For SEND (FILE=PUN|LSTX
  Message *INW0082I QUEUE ENTRY name number CREATED* is added.

  To avoid possible confusion with these new message numbers, you may specify option **INW1**. This displays the messages with the prefix INW0001I.
- The source and target **user ID** (TO=, FROM=) of the submitted job and its list and punch output are the actual user ID (not SYSCICSA).
- RECEIVE from the LST and PUN queues now allows specification of jobname and jobnumber as host filename.

  Users of 3270 emulators that require a host filename (for example FTTERM) may have to change existing command procedures accordingly.
- When receiving from VSE/POWER queues, the entries will no longer be automatically deleted. The **default** is changed from DELETE to **KEEP**. This allows to specify DELETE also for entries with a disposition other than D or K.

SEND/RECEIVE Command Syntax and Options

Additional considerations

Please note that file transfer to and from the VSE/POWER queues is also supported by the *VSE Navigator* application. For details about this application, refer to the *VSE/ESA e-business Connectors User’s Guide*.

In case you are not familiar with the notation of the following syntax diagrams, refer to “How to Read SEND/RECEIVE Syntax Diagrams”, on page 273.

Command Syntax - SEND/RECEIVE (FILE=RDR|LST|PUN|LSTX

The following shows the SEND/RECEIVE command formats for the various forms of VSE/POWER queue access. A description of the parameters and options is given at the end of this section.

SEND/RECEIVE Command Syntax and Options (FILE=RDR|LST|PUN|LSTX

```
►►──SEND──pws_name──┬─────┬──┬──────────────────────────┬──(──FILE=RDR──┤ Options ├──┬───┬──────────►◄
                    └─id:─┘  └─dummy_name──┬──────────┬─┘                            └─)─┘
                                           └─dummy_type─┘
```

Options:

```
   ┌─ASCII──┐  ┌─CRLF──────────────────────┐  ┌─UC───┐  ┌─MSG────┐  ┌─BUF=4096──┐
├──┼────────┼──┼───────────────────────────┼──┼──────┼──┼────────┼──┼───────────┼──┬──────┬──────────►
   └─BINARY─┘  ├─NOCRLF────────────────────┤  └─NOUC─┘  ├─NOMSG──┤  └─BUF=nnnnn─┘  └─INW1─┘
               │                ┌─NOCRLF─┐ │            └─ALLMSG─┘
               └─LRECL=nnnnn────┴────────┴─┘

►──┬─────────────────┬──┬────────────┬──┬───────────┬──┬───────────────────┬───────────────────────┤
   └─DCDF=dcdf_name──┘  └─NLS=suffix─┘  ├─TRACE─────┤  └─USERTRT=user_trt──┘
                                        ├─TRACE=Q───┤
                                        ├─TRACE=LST─┤
                                        └─TRACECUT──┘
```

Figure 28. Send a Job to the VSE/POWER RDR Queue - FILE=RDR

```
                             ┌─user_id──────────────────┐
►►──SEND──pws_name──┬─────┬──┼──────────────────────────┼──(──FILE=──┬─PUN──┬──┤ Options ├──┬───┬──────►◄
                    └─id:─┘  └─entry_name──┬──────────┬─┘            └─LSTX─┘                 └─)─┘
                                           └─dummy_type─┘
```

Options:

```
   ┌─ASCII──┐  ┌─CRLF──────────────────────┐  ┌─CLASS=A─────────┐  ┌─DISP=L────────┐  ┌─NOCC──────────┐
├──┼────────┼──┼───────────────────────────┼──┼─────────────────┼──┼───────────────┼──┼───────────────┼──►
   └─BINARY─┘  ├─NOCRLF────────────────────┤  └─CLASS=job_class─┘  └─DISP=──┬─D─┬──┘  │     ┌─MCC─┐   │
               │                ┌─NOCRLF─┐ │                               ├─H─┤      └─CC=─┴─ASA─┴───┘
               └─LRECL=nnnnn────┴────────┴─┘                               └─K─┘

   ┌─DEST=logged_on_user─┐                   ┌─BUF=4096──┐
►──┼─────────────────────┼──┬──────────────┬──┼───────────┼──┬──────┬──┬─────────────────┬──────────►
   └─DEST=destination────┘  └─NODE=node_id─┘  └─BUF=nnnnn─┘  └─INW1─┘  └─DCDF=dcdf_name──┘

►──┬────────────┬──┬───────────┬──┬───────────────────┬─────────────────────────────────────────────┤
   └─NLS=suffix─┘  ├─TRACE─────┤  └─USERTRT=user_trt──┘
                   ├─TRACE=Q───┤
                   ├─TRACE=LST─┤
                   └─TRACECUT──┘
```

Figure 29. Send a Workstation File to the VSE/POWER PUN/LST Queue - FILE=PUN/LSTX

SEND/RECEIVE Command Syntax and Options (FILE=RDR|LST|PUN|LSTX

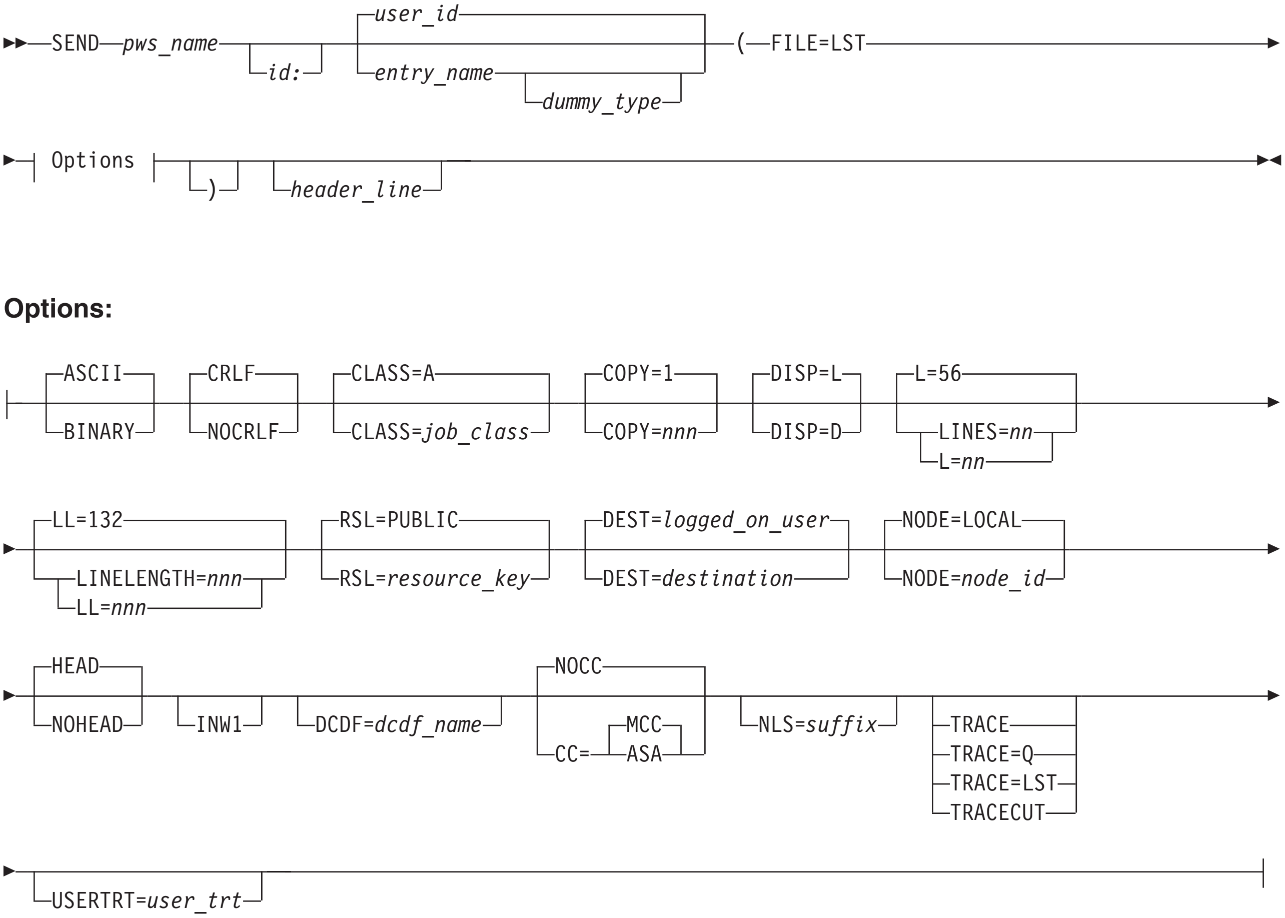


Figure 30. Send a Workstation File to the VSE/POWER LST Queue - FILE=LST

SEND/RECEIVE Command Syntax and Options (FILE=RDR|LST|PUN|LSTX

```
►►──RECEIVE──pws_name──┬─────┬──┬─────────────────────────────┬──(FILE=──┬─RDR─┬──┤ Options ├──┬───┬────►◄
                       └─id:─┘  └─entry_name──┬──────────────┬┘          ├─LST─┤               └─)─┘
                                              └─entry_number─┘           └─PUN─┘
```

Options:

```
                                                  (1)
   ┌─ASCII──┐  ┌─CRLF───┐  ┌─REPLACE─┐  ┌─KEEP───┐  ┌─NOCC─────┐                         ┌─S=0───┐
├──┼────────┼──┼────────┼──┼─────────┼──┼────────┼──┼──────────┼──┬───────────────────┬──┼───────┼──────►
   └─BINARY─┘  └─NOCRLF─┘  └─APPEND──┘  ├─DELETE─┤  │    (2)   │  └─CLASS=job_class───┘  └─S=nnn─┘
                                        └─HOLD───┘  └─CC───────┘

   ┌─BUF=4096──┐
►──┼───────────┼──┬────────────┬──┬────────────┬──┬───────┬──┬──────┬──┬────────────────┬──────────────►
   └─BUF=nnnnn─┘  └─TOP=nnnnnn─┘  └─BOT=nnnnnn─┘  └─BLANK─┘  └─INW1─┘  └─DCDF=dcdf_name─┘

►──┬────────────┬──┬─────────────┬──┬───────────────────┬──────────────────────────────────────────────┤
   └─NLS=suffix─┘  ├─TRACE───────┤  └─USERTRT=user_trt──┘
                   ├─TRACE=Q─────┤
                   ├─TRACE=LST───┤
                   └─TRACECUT────┘
```

Notes:

1 Option NOCC|CC and option S=nnn are **not** valid with option FILE=RDR.

2 Option NOCC|CC and option S=nnn are **not** valid with option FILE=RDR.

Note: Option NOCC|CC and option S=nnn are **not** valid with option FILE=RDR.

Figure 31. Receive Entries from VSE/POWER RDR, LST, and PUN Queue - FILE=RDR|LST|PUN

Command Options

The following describes the parameters and options available.

Note: For RECEIVE, the options NOCC|CC and S= do **not** apply to the VSE/POWER RDR queue. They are used for RECEIVE from the PUN and LST queue only.

For a description of the parameters **pws_name** and **id:**, refer to “Common SEND/RECEIVE Command Syntax” on page 70.

For a description of the following **common options** refer to “Common SEND/RECEIVE Command Options” on page 72.

```
ASCII/BINARY
CRLF/NOCRLF
APPEND/REPLACE/NOREPLACE
INW1
DCDF=
NLS=
TRACE
```

```
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=
```

Note: The explanations for the following parameters and options are given separately for SEND and RECEIVE in case there are differences in use or effect. Otherwise, the explanation applies to both SEND and RECEIVE.

dummy_name

a host specification is not applicable for this function. However, some 3270 Emulation Programs (for example FTTERM) require a host name to be specified. In that case, specify any name. It will be ignored by the host.

Note that for sending to the RDR queue, the host_name (jobname) is taken from the * $$ JOB statement or // JOB statement of the submitted job.

dummy_type

has no meaning and is ignored if specified. (Only included for consistency of command syntax.)

entry_name

SEND FILE=PUN|LST|LSTX: is the name of the PUN or LST queue entry to be created. If the name is omitted, the **user ID** will be used as entry name (**default**).

FILE=PUN|LSTX: The entry name can be from 1-8 characters long.

FILE=LST: The entry name can be from 2-8 characters long. (The CICS Report Controller feature does not accept a one-character name.)

Valid characters in the **entry_name** are:

```
A through Z
0 through 9
- . / @ #
```

Dollar signs ($) are allowed but should be avoided, since they may cause problems when messages containing the name are displayed at the workstation.

Do not use any of the following names, because you will not be able to manipulate them using CEMS, CEOS or VSE/POWER operator commands:

```
ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS
```

RECEIVE (FILE=RDR|PUN|LST: The name of the queue entry to be retrieved.

Default: If an entry_name is not specified, you will receive the first entry found in the queue with **disposition D or K and class A** which you are authorized to access.

entry_number

RECEIVE (FILE=RDR|LST|PUN: Is the VSE/POWER-assigned unique number of the queue entry to be retrieved (one to five digits). Leading zeros may be omitted.

The entry number is only meaningful if an entry_name is also specified.

Default: If no entry_number is given and several entries have the same entry_name, the first entry with the specified name is retrieved.

FILE=RDR | LST | PUN | LSTX
defines the various VSE/POWER queue access functions:

RDR send/receive jobs to/from the reader queue

LST send/receive data to/from the list queue

PUN send/receive data to/from the punch queue

LSTX send (dump) data to the list queue

options
in alphabetic order:

ALLMSG
SEND (FILE=RDR: specifies that all job event messages produced by a submitted job are to be saved by VSE/POWER for later retrieval by the workstation user. MSG is the **default**.

BLANK
RECEIVE: When logical records are downloaded from the host to the workstation with the CRLF option, trailing blanks are automatically deleted from the end of each record to save disk space and download time. Specify the BLANK option if you do **not** wish trailing blanks to be deleted.

Note: When receiving VSE/POWER queue entries that were not created by a file transfer SEND command but, for example, by a VSE batch job, the records of these entries will **not** contain trailing blanks even if option BLANK is specified. This is because VSE/POWER trims off trailing blanks of all records created by batch jobs.

BOT=nnnnnn
RECEIVE (FILE=RDR | LST | PUN: specifies that only the last nnnnnn lines of the queue entry are to be retrieved. 'nnnnnn' can have one to six digits. Maximum value is 999999.

See also the TOP option. TOP and BOT can be specified together to retrieve the beginning and end of the queue entry.

If either TOP or BOT (or both values together) exceed the total number of records of the entry to be received, then all records are received.

For receive from RDR, the * $$ JOB and EOJ statements are not included in the count and are always returned.

BUF=nnnnn
defines the size of the send and receive buffers used to send/receive data to/from the VSE/POWER queues. The **default** buffer size is 4096 bytes.

When transferring very large files, performance may be improved by increasing the buffer size. **nnnnn** can be a value from 512 to 32767.

A reduction of the buffer size may be necessary when there is not enough CICS main storage available to allocate the default buffer.

CC[=MCC|=ASA]

for **RECEIVE (FILE=PUN|LST**: CC specifies that print/punch control characters returned by VSE/POWER are to be inserted in the first position of each retrieved print/punch record.

for **SEND (FILE=LST|LSTX|PUN|**: CC specifies that the first character of each record being sent is a machine control character. To process ASA control characters, CC=ASA must be specified. For FILE=LST, control characters are checked and file transfer is terminated if invalid control characters are found. For FILE=LSTX|PUN, there is no checking for valid control characters.

The default is NOCC.

CLASS=job_class

specifies the VSE/POWER job or output class of the queue entry to be created or received. The job_class must be a character from A through Z or 0 - 9. For **SEND (FILE=LST**, the numeric classes 0 - 9 are **not** supported.

For SEND (and for RECEIVE without an entry_name) the default is CLASS=A.

Whenever existing queue entries are accessed, specification of the class will improve performance.

COPY=nnn

SEND (FILE=LST: the number of copies to be printed. It must be a value between 1 and 255. The **default** is COPY=1.

DELETE

RECEIVE: specifies that the queue entry is to be deleted after it has been received. The default is KEEP.

DISP=L|D

SEND (FILE=LST: The disposition of the LST queue entry.

DISP=L|D|H|K

SEND (FILE=PUN|LSTX: The disposition of the punch or list queue entry.

DEST=destination

SEND (FILE=LST|LSTX|PUN: Specifies the destination of the list or punch queue entry. This is the name that appears in the **TO =** field of the queue entry display. It can be altered with the **USER =** parameter of the VSE/POWER command PALTER.

For example, the destination can be:

- The user ID of another VSE user, to make this entry accessible by this user.
- **ANY** - to make this entry accessible by all VSE users. Entries with a destination of ANY can be received by all users. They can be manipulated (delete, alter) only by authorized users. These users are allowed to manage all VSE/POWER queues. The authorization is defined via the ″Maintain User Profiles″ dialog of the Interactive Interface.
- If VSE is running under VM, a CMS user ID, to send the workstation file directly into the reader of this CMS user.
- **LOCAL** - for immediate printing on the VSE system printer.
- The name of a CICS-controlled 3270 printer.

- ****SYSPRT** - a special name provided by CICS to print large amounts of data on a 3270 printer.

The destination name can be from 1 to 8 characters long.

For **SEND (FILE=LST** (which uses the CICS Report Controller Feature), valid characters are:

```
A through-Z
0 through-9
# $ @

In addition the name can be **SYSPRT.
```

For **SEND (FILE=LSTX|PUN**, valid characters are:

```
A through-Z
0 through-9
# $ @
/ - .

In addition the name can be **SYSPRT.
```

The **default** destination is the ID of the logged-on user.

HEAD
SEND (FILE=LST: is an **optional** header line that will be printed at the top of each page. It shows time, date, and page number. Optionally, heading text can be supplied in the comments area of the SEND command.

HEAD is the **default**. Specify option NOHEAD to suppress the printing of the header line.

The length of the header text can be LINELENGTH minus 22, but no longer than 120 characters. Longer headers are truncated without warning. The header text is translated to uppercase. Extra blank characters are deleted.

HOLD
RECEIVE only: specifies that the disposition of the received VSE/POWER queue entry is to be changed to H (hold) if it was D, and to L (leave) if it was K. (**Default** is KEEP).

KEEP **RECEIVE only:** is the default when receiving from the RDR, LST, and PUN queues. The disposition of the queue entry remains unchanged. Specify DELETE to delete an entry after it has been received, or HOLD to change the disposition to H (hold) or L (leave).

LINES=nn | L=nn
SEND (FILE=LST: the number of lines to be printed per page. **nn** can be a value between 3 and 99. The **default** is L=56.

LINELENGTH=nnn | LL=nnn
SEND (FILE=LST: the length of each line in the print output. **nnn** can be a value between 20 and 205. The **default** is LL=132.

LRECL=nnnnn
SEND (FILE=RDR|LSTX|PUN: specifies a logical record length when workstation files are sent to a VSE/POWER queue. The

workstation file is treated as a contiguous string of data which is broken up into nnnnn-byte segments. Each segment forms a logical record in the queue.

The maximum value for **nnnnn** is the size of the send buffer minus 20 for the record prefix. See also the BUF= option.

When submitting a VSE job that exists at the workstation as a contiguous string rather than of logical records, specify LRECL=80 to break up the string into 80-byte records.

LRECL overrides the CRLF option. CRLF is ignored if it is active by default or specified explicitly together with LRECL.

MSG **SEND (FILE=RDR**: specifies that the job completion message produced by a submitted job is to be saved by VSE/POWER for later retrieval by the workstation user. MSG is the **default**.

NOCC
Not FILE=RDR: specifies no handling of print/punch control characters.

NOCC is the **default**. See also the option CC.

NODE=node_id
SEND (FILE=PUN|LSTX|LST: transmits a workstation file to a remote node connected by VSE/POWER PNET. The node_id must be 1-8 alphameric characters (A-Z 0-9 # $ @). **Default** is NODE=LOCAL.

NOHEAD
SEND (FILE=LST: specifies that no heading line is to be generated at the top of each page. There will be no timestamp and no page numbering. Any heading text in the comments area of the SEND command will be ignored.

HEAD is the **default**.

NOMSG
SEND (FILE=RDR: Suppresses collection of job completion messages. MSG is the **default**.

NOUC
SEND (FILE=RDR: no uppercase translation. Must be specified when jobs are submitted that contain user data which must **not** be translated to uppercase. In that case, the job control statements must be provided in uppercase.

Uppercase translation (UC) is the **default**.

RSL=nn|PUBLIC
SEND (FILE=LST: specifies the security key that will be associated with the print output. The value must be from 1 to 24 or PUBLIC.

Once the print output is stored in the list queue, only terminal users with the same resource level are able to alter its characteristics via the CEMS or CEOS transaction. The output can only be printed on printers with the same RSL value. The default is RSL=PUBLIC.

S=nnn **RECEIVE (FILE=LST|PUN**: specifies the segment (or job suffix) number of the LST or PUN output segment to be received. When LST or PUN output is segmented, the individual segments all have the same entry name and number. To retrieve a segment, it must be identified by entry name, entry number, and the S= option. (Segment numbers are displayed with the `FULL=YES` parameter in the VSE/POWER PDISPLAY command.)

Valid segment numbers are 1-127. If specified for a non-segmented entry, the S= option is ignored.

The S= option is not valid for RDR queue entries.

TOP=nnnnnn

RECEIVE (FILE=RDR|LST|PUN: specifies that only the first nnnnnn lines of the queue entry are to be retrieved. 'nnnnnn' can have one to six digits. Maximum value is 999999.

See also the BOT option. TOP and BOT can be specified together to retrieve the beginning and end of a queue entry.

If either TOP or BOT (or both values together) exceed the total number of records of the entry to be received, then all records are received.

For receive from RDR, the * $$ JOB and EOJ statements are not included in the count and are always returned.

UC **SEND (FILE=RDR**: specifies that the submitted job is to be translated to uppercase. Note that VSE accepts Job Control Statements only in uppercase. UC is the **default**.

If the job contains data which must not be translated to uppercase, specify the NOUC option. In that case, the job control statements must be provided in uppercase.

header_line

Is an **optional** header line that will be printed at the top of each page. It shows time, date, and page number. Optionally, heading text can be supplied in the comments area of the command SEND (FILE=LST.

The length of the header text can be LINELENGTH minus 22, but no longer than 120 characters. Longer headers are truncated without warning. The header text is translated to uppercase. Extra blank characters are deleted.

Chapter 8. File Transfer To and From the Host Transfer File (HTF)

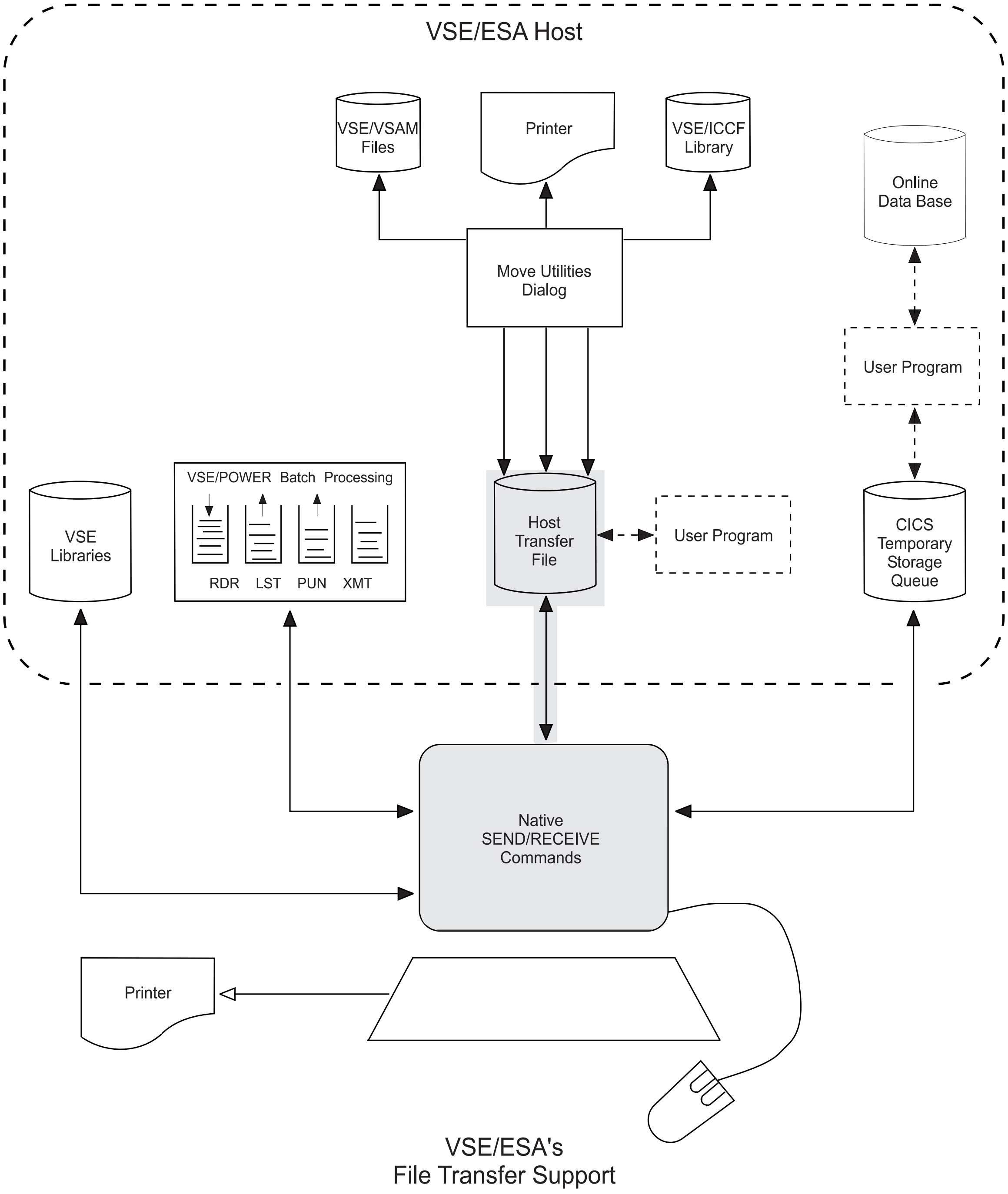


Figure 32. Transferring Data To and From the Host Transfer File

Overview

The **H**ost **T**ransfer **F**ile (**HTF**) is a VSE/VSAM file. It serves as a "holding area" for files that are transferred to and from workstations and can be used to exchange (share) data with other users.

You would send your data to the Host Transfer File if:

- Data is to be moved to a VSE/ICCF library.
- Data is to be moved to VSE/VSAM space.
- You want to keep your data in the Host Transfer File for later retrieval by you or another user.
- You want to distribute data to other users.

You also need the Host Transfer File to store the Data Conversion Descriptor file (DCDF) if you use field-level data conversion as described in Chapter 10, "Data Conversion Support", on page 205.

Data in the HTF can be manipulated at the host with the VSE/ESA *Move Utilities* dialog which provides a list of your files in the HTF and provides functions like view, copy, rename, delete, move to VSAM/ICCF for each file. For details on how to use the *Move Utilities* dialog, refer to "Move Utilities Dialog for Host Access to the HTF" on page 148.

The HTF can contain more than one file at a time, and more than one user can access it at the same time. For this, a file's owner can limit file access in various ways.

The owner may specify a file in the HTF as:

- **PRIVATE** - only the file's owner has access to this file.
- **PUBLIC** - all users who have access to the HTF have also read access to this file in the HTF. They can view, copy, or receive the file; but they cannot delete it or change its contents or name.
- **SHARED** - the owner shares access to this file with a specified number of users. These users have read access to the file. They can view, copy, or receive the file; but they cannot delete it or change its contents or name.

  In addition, these users receive a notification message that a file has been shared with them. When such a file is received or copied the owner of the file receives an acknowledgement message.
- **PUBLIC** and **SHARED** - gives read access to the file to **all** users. Notification messages are sent to selected (sharing) users.

These access rights can be defined with options in the SEND command when the file is transferred to the HTF or with the *Move Utilities* dialog after the file is stored in the HTF.

Additional considerations

▸ **Migration Considerations:**

Note that you cannot migrate an existing HTF from a release prior to VSE/ESA 1.3. **A new HTF must be created!** For details about how to create the HTF, see the next section.

▸ **Additional Support for HTF Access:**

VSE/ESA provides an application interface (**API**) which lets you access the HTF from within CICS application programs. For details see "Application Program Interface (API) for the HTF" on page 173.

The following sections describe:

- Creating the HTF.
- Support in several CICS partitions.
- Examples of SEND/RECEIVE commands to/from the HTF.
- Data protection.
- HTF access without signing on to VSE/ESA's Interactive Interface.
- The *Move Utilities* dialog.
- User exits of the *Move Utilities*.
- The HTF's application program interface (API).
- The SEND/RECEIVE command syntax.

Creating the Host Transfer File (HTF)

Before you can do file transfer to and from the Host Transfer File, you must define the HTF to:

- VSAM and
- CICS

Defining the HTF to VSAM

VSE/ESA provides the skeleton SKIWSTF to help you create the Host Transfer File. The skeleton is shipped in VSE/ICCF library 59.

To create the Host Transfer File, proceed as follows:

1. Copy the skeleton to your VSE/ICCF primary library.
2. Edit skeleton SKIWSTF to complete the job stream. Figure 33 on page 140 shows the skeleton.

Creating Host Transfer File

```
* $$ JOB JNM=IWSTRFL,CLASS=0,DISP=D
// JOB IWSTRFL     CREATE PC HOST TRANSFER FILE
// EXEC IDCAMS,SIZE=AUTO
   DELETE (PC.HOST.TRANSFER.FILE) PURGE CL -
          CATALOG(VSESP.USER.CATALOG)
   DEFINE CLUSTER (NAME(PC.HOST.TRANSFER.FILE) -
          RECORDS (4000 4000) -
          TO (99365) -
          FREESPACE (0 50) -
          SHAREOPTIONS (2) -
          INDEXED -
          KEYS (16 0) -
          RECORDSIZE (100 32758) -
          USECLASS (0) -
          VOLUMES (-V001-)) -
          DATA (NAME(PC.HOST.TRANSFER.FILE.DATA) -
          SPANNED -
          CISZ (2048)) -
          INDEX (NAME(PC.HOST.TRANSFER.FILE.INDEX)) -
          CATALOG (VSESP.USER.CATALOG)
/*
// OPTION STDLABEL=DELETE
INWFILE
/*
// OPTION STDLABEL=ADD
// DLBL INWFILE,'PC.HOST.TRANSFER.FILE',99/365,VSAM,CAT=VSESPUC
/*
// EXEC IESVCLUP,SIZE=AUTO             ADD LABEL TO STDLABUP PROC
D                                              INWFILE
A PC.HOST.TRANSFER.FILE                        INWFILE VSESPUC
/*
// EXEC IESVSMLD,SIZE=AUTO             LOAD DUMMY RECORD INTO INWFILE
80,K,INWFILE
**************************************************
/*
/&
* $$ EOJ
```

Figure 33. Create Host Transfer File (SKIWSTF Skeleton)

- Change the **-V001-** variable. Specify the volume or volumes which have free VSE/VSAM space available for the Host Transfer File. This free space should be part of the VSE/VSAM user catalog (VSESP.USER.CATALOG).
- You may want to access the Host Transfer File from more than one CICS partition concurrently. If so, change SHAREOPTIONS (2) to SHAREOPTIONS (**4**).

  See also "Support in Several CICS Partitions" on page 141.

3. **Submit the completed job stream**.

   If no Host Transfer File exists yet, you get the following messages during DELETE processing:

   ```
   IDC3012I ENTRY PC.HOST.TRANSFER.FILE NOT FOUND
   IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS
            IGG0CLCG-6
   IDC0551I **ENTRY PC.HOST.TRANSFER.FILE NOT DELETED
   IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
   ```

   These messages do not indicate an error but reflect normal processing.

The label of the file is automatically added to the standard labels.

Defining the HTF to CICS

To enable the system to activate the Host Transfer File, you must first **activate** the corresponding **entry** in table **DFHFCTxx**, or define the file using CEDA.

To do this, proceed as follows:

1. Copy table DFHFCTxx (stored in VSE/ICCF library 59) to your primary VSE/ICCF library.
2. Edit DFHFCTxx.

   Remove the asterisk (*) in column 1 for the following entry:

   ```
   COPY IESWFCT
   ```
3. Submit the changed table for processing. Select option **7** (Submit) on the *Primary Library* panel.
4. Migrate the FCT to the CSD file using a Job such as the one below:

   ```
   // EXEC DFHCSDUP,SIZE=600K
     DELETE GROUP(FCTSP)
     MIGRATE TABLE(DFHFCTSP)
     ADD GROUP(FCTC2) LIST(VSELIST)
   /*
   ```
5. Either shut down CICS and bring it up again, or reinstall the group FCTSP.

Note: If you are a user migrating from an earlier release of VSE, make sure that your own entries are included in the new table.

Support in Several CICS Partitions

It is possible to use Workstation File Transfer Support concurrently in more than one CICS partition (in one CPU). This requires a change in skeleton SKIWSTF for the Host Transfer File. Change SHAREOPTIONS (2) to SHAREOPTIONS (**4**). You should not access the Host Transfer File from two CICS partitions concurrently with the *same user ID*.

If you have two partitions with CICS, the notification messages issued by the Workstation File Transfer Support will also appear in the partition without VSE/ICCF. An example of a notification message is:

```
03/06 - 08:48  MSG FROM SYSA
USER SYSA HAS RECEIVED YOUR SHARED FILE TESTDATA
```

Examples

About this Section

This section provides various examples of how you may use SEND/RECEIVE commands for file transfer to and from the HTF.

An example starts with a short heading marked by the '►' character.

The following examples show how to use:

- Use SEND defaults
- Use RECEIVE defaults
- Use the host filename
- Use the host filetype
- Transfer string data with the BINARY option
- Replace an existing file
- Prevent deletion of a file after download
- Make a file accessible to all users with the PUBLIC option
- Share access to a file with selected users
- Combine the 'share' and 'public' attributes

The SEND/RECEIVE command formats are described in Figure 48 on page 181.

► **SEND Defaults:**

```
User JONES issues the command:

SEND PCFILE HOSTFILE

Message:

INW0001I File transfer complete
```

• **Explanation:**

- Workstation file PCFILE is stored in the HTF as HOSTFILE. **FILE=HTF** is the default. The filetype is blank.
- Each logical record in PCFILE becomes one record in HOSTFILE. **CRLF** is the default.
- Records are translated from ASCII to EBCDIC. **ASCII** is the default.
- HOSTFILE is stored as a private file. Only user JONES has access to it. **PRIVATE** is the default.
- If a file HOSTFILE already exists in the HTF, file transfer is canceled. **NOREPRLACE** is default.

Examples of File Transfer To and From the Host Transfer File

▸ **RECEIVE Defaults:**

User JONES issues the command:

```
RECEIVE PCFILE HOSTFILE
```

Message:

```
INW0001I File transfer complete
```

• **Explanation:**

- The Host Transfer File (FILE=HTF is default) is searched for a file named HOSTFILE with a blank filetype, owned by user JONES.

- Each logical record in HOSTFILE becomes one record in PCFILE delimited by CR/LF characters. **CRLF** is the default.

- Records are translated from EBCDIC to ASCII. **ASCII** is the default.

- HOSTFILE is deleted after it has been received.
  DELETE is the default. Specify the KEEP option to prevent deletion.

- If a file PCFILE already exists on the workstation, it is deleted without warning. **REPLACE** is the default.

- If a file HOSTFILE owned by user JONES is not found in the HTF, file transfer is terminated with message:
  INW0012I File/member HOSTFILE not found

▸ **Hostname Defaults:**

User JONES issues the command:

```
SEND PCFILE (FILE=HTF
```

• **Explanation:**

- When no host filename is given, the user ID is taken as filename. The filetype remains blank. In this case, workstation file PCFILE is stored in the HTF under the name JONES. The **user ID** is the default.

- The host filename can only be omitted when at least one option is specified. Otherwise, the command will be rejected by the workstation SEND/RECEIVE command processor.

▸ **Filetype:**

```
SEND PCFILE.TXT HOSTFILE TXT
```

• **Explanation:**

- PCFILE is stored in the HTF with filename HOSTFILE and filetype TXT.

- The filetype can be from one to eight characters long. It must be separated from the filename by one or more blanks.

- For RECEIVE, the exact filename and filetype must be specified.

- If omitted, a filetype of blank is assumed for SEND and RECEIVE.

Examples of File Transfer To and From the Host Transfer File

▸ Transfer String Data (Backup)

To backup a workstation file in the HTF enter:

```
SEND PCFILE.EXE PCFILE EXEBIN (BINARY
```

• Explanation:

- PCFILE.EXE is an executable workstation program. It is sent to the host for backup purposes only. It cannot be viewed or processed at the host.

- The BINARY option suppresses ASCII/EBCDIC translation.

- The BINARY option implies the NOCRLF option. Thus the contents of PCFILE.EXE is treated as a contiguous string of data. If PCFILE.EXE is smaller than the size of the file transfer buffer then PCFILE EXEBIN will consists of a single record. Otherwise, each file transfer buffer will become one record in PCFILE EXEBIN.

- It is good practice to choose a host filetype (such as EXEBIN) that identifies the type of the file so that the correct options can be used on a subsequent download.

▸ Transfer String Data (Restore)

To restore the backup file from the preceding example:

```
RECEIVE PCFILE.EXE PCFILE EXEBIN (BINARY
```

• Explanation:

- The BINARY option suppresses EBCDIC/ASCII translation.

- The BINARY option implies the NOCRLF option. The records of the hostfile PCFILE EXEBIN are concatenated to form a contiguous string of data in workstation file PCFILE.EXE.

- PCFILE EXEBIN is deleted from the HTF.

- For a detailed discussion of the options BINARY and NOCRLF, refer to "Common SEND/RECEIVE Command Options" on page 72.

▸ Replace Existing File in HTF:

User JONES issues the command:

```
SEND PCFILE HOSTFILE (REPLACE
```

• Explanation:

- An existing file HOSTFILE owned by user JONES is replaced with the contents of PCFILE. If a file HOSTFILE does not exist, a new file is created.

- The system first stores PCFILE in a temporary file INW$TEMP in the HTF. Then the existing HOSTFILE is deleted and INW$TEMP is renamed to HOSTFILE. This ensures that the existing file is not destroyed in case of a file transfer failure.

  When you find a file INW$TEMP in the HTF, a previous SEND command was not completed successfully. The INW$TEMP file can be deleted.

▸ **Do Not Delete on Download:**

```
RECEIVE PCFILE HOSTFILE (KEEP
```

• **Explanation:**

```
- File HOSTFILE is not deleted from the HTF. The default is DELETE.
  Make sure to specify KEEP if you do not want the file to be deleted.
```

▸ **Send a PUBLIC File:**

```
User JONES issues the command:

SEND PCFILE.EXE PCFILE EXEBIN (BINARY PUBLIC
```

• **Explanation:**

```
- The PUBLIC option specifies that all users who have access to the HTF have
  read access to file PCFILE EXEBIN owned by user JONES.

- All users can view the file in the HTF, receive it to their workstation, or
  copy it to VSAM or ICCF.

- Only the owner JONES can replace, delete, or rename the file.

- When a file is made PUBLIC in the HTF, no notification message will be sent
  to other users.
```

▸ **Receive a PUBLIC File:**

```
User SMITH receives a PUBLIC file owned by JONES:

RECEIVE PCFILE.EXE PCFILE EXEBIN (BINARY FROM=JONES
```

• **Explanation:**

```
- To receive a PUBLIC file owned by another user you must always
  specify the owner's user ID in the FROM= option.

- Although the KEEP option is not specified, the file is not deleted from
  the HTF. A file can only be deleted by its owner.
```

Examples of File Transfer To and From the Host Transfer File

▸ **Sharing a File with Selected Users:**

User JONES issues the command:

```
SEND PCFILE.EXE PCFILE EXEBIN (BINARY FOR=SMITH,BROWN,MILLER
```

• **Explanation:**

- The **FOR=** options specifies that users SMITH, BROWN, and MILLER have read access to file PCFILE EXEBIN owned by user JONES.

- These users can view the file in the HTF, receive it to their workstation, or copy it to VSAM or ICCF.

- Only the owner JONES can replace, delete, or rename the file.

- Up to eight user IDs (separated by a comma without intervening blanks) can be specified in the FOR= option.

- **Notification:** Each user specified in the FOR= option receives a notification message as follows:

```
07/27-14:25 MSG FROM JONES
USER JONES    HAS GIVEN YOU SHARED ACCESS TO HIS FILE PCFILE  EXEBIN
```

- When file owner JONES replaces file PCFILE EXEBIN with a new version, the users specified in the FOR= option receive another message:

```
07/29-12:33 MSG FROM JONES
USER JONES    HAS PROVIDED NEW VERSION OF SHARED FILE PCFILE  EXEBIN
```

▸ **Receive a SHARED File:**

User SMITH receives a SHARED file owned by JONES:

```
RECEIVE PCFILE.EXE PCFILE EXEBIN (BINARY FROM=JONES
```

• **Explanation:**

- To receive a SHARED file owned by another user, you must always specify the owner's user ID in the **FROM=** option.

- Although the KEEP option is not specified, the file is not deleted from the HTF. A file can only be deleted by its owner.

- **Acknowledgement:** Whenever a shared file is received, the owner of the file (in our case JONES) receives a message as follows:

```
07/28-11:24 MSG FROM SMITH
USER SMITH    HAS RECEIVED YOUR SHARED FILE PCFILE  EXEBIN
```

▸ **Combining PUBLIC and 'shared':**

User JONES issues the command:

```
SEND PCFILE.EXE PCFILE EXEBIN (BINARY PUBLIC FOR=BOSS
```

• **Explanation:**

- All users have read access to the file.

- For user BOSS, notification and acknowledgement messages are sent.

HTF Access Without Signon to the Interactive Interface

With the introduction of the eight-character user ID it is possible to transfer files to and from the HTF without signon to VSE/ESA's Interactive Interface.

Carefully note the following restrictions:

- Since the *Move Utilities* dialog is not available without signon to the Interactive Interface, the HTF can only be used as a repository for workstation data for backup purposes or to share the data among workstation users.
- Without signon to CICS, a workstation file sent to the HTF is stored for user NOIItrmid. This means that
  - If you wish to receive this file back to the workstation, you can do this only with the same terminal ID (the one you used for the send).
  - Any other user who has access to this terminal ID could receive/delete this file.

  It is, therefore, recommended to always sign on to CICS with a unique (up to eight characters) user ID.
- The user IDs specified in the FOR= and FROM= options of the SEND/RECEIVE commands are not checked for validity.
- There are no notification messages when the FOR= or FROM= options are used to inform users that a file has been shared or a shared file has been received.
- The Application Program Interface (API) to access the HTF from a CICS transaction cannot be used. The API extracts the user ID of the requesting user from the VSE/ESA User Status Record. Because the User Status Record is not available without signon to the Interactive Interface, the API returns an error code.
- Unlike the Interactive Interface, CICS allows multiple signon with the same user ID. Try to avoid simultaneous access of the HTF with the same user ID. The results would be unpredictable.

For a description of the general restrictions that come into play when transferring files without being signed on to the Interactive Interface, refer to “File Transfer Without Signon to the VSE/ESA Interactive Interface” on page 85.

Move Utilities Dialog for Host Access to the HTF

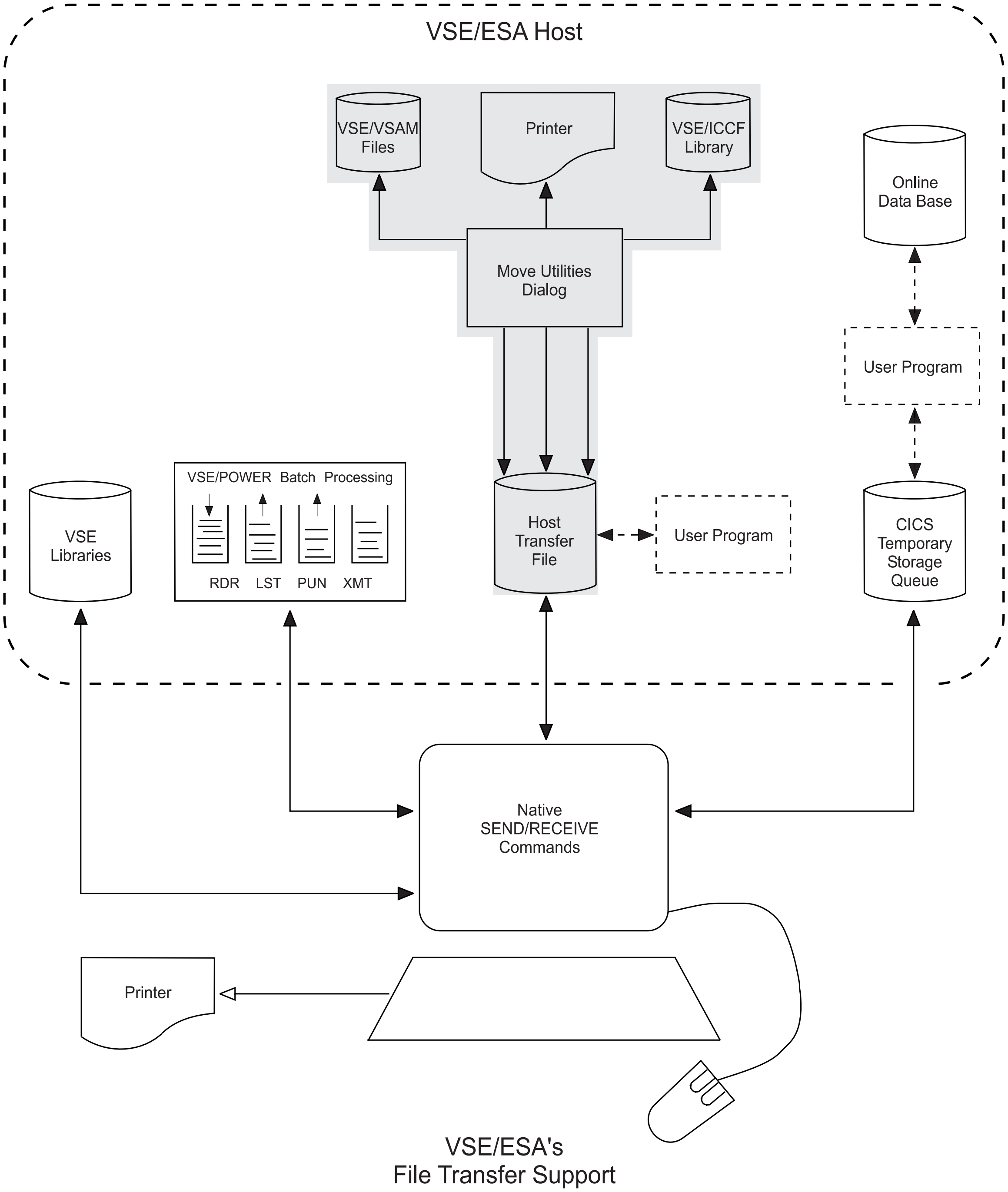


Figure 34. Data Paths of the Move Utilities

The *Move Utilities* dialog of VSE/ESA, as shown in Figure 35 on page 150 moves:

- VSE/VSAM files and VSE/ICCF library members to the Host Transfer File.
- Files in the Host Transfer File to VSE/VSAM files and VSE/ICCF library members.

You can also use the dialog to display files owned by different users and process your own files.

The Host Transfer File (HTF)

The Host Transfer File (HTF) is the master file for all users working with the Move Utilities. It contains user files submitted from or to the programmable workstations and serves as a ′holding area′ for files to be accessed from the workstation.

The Host Transfer File is a VSE/VSAM file that can contain more than one file for each user at the same time. One such file is called an ″entry″ in the HTF.

The File’s Identification

The identification of a file that is stored in the HTF is the file’s filename and its filetype (**optional**).

Both filename and filetype can be up to eight characters long.

For filename and for filetype you can use any characters except:
* : ( ) <>
Trailing underscores (for example, FILE___).
Leading or imbedded blanks.

In case of a move from **VSE/VSAM** or **VSE/ICCF** to the HTF you must **not** use **INW** as the first three characters for the filename.

File Owner

The file owner of an entry in the HTF is identified by the VSE/ESA **user ID** which can be up to 8 characters long.

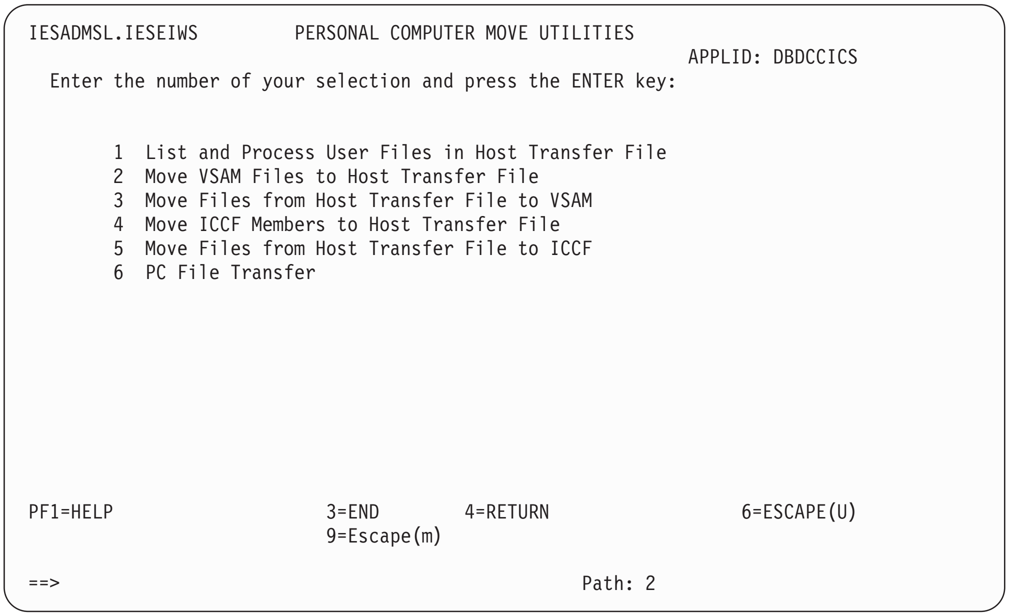
Panel Functions

In the shipped VSE/ESA system, only the administrator can access the Move Utilities. However, your system administrator may give you the authorization to work with the move utilities. In the configuration example used in this manual, you can access the dialog as follows:
In the *VSE/ESA Function Selection* panel shown in Figure 2 on page 8, select:
2 (Personal Computer Move Utilities)

Sample	Synonym
Fast Path: **2**	Default: —— Yours:

This invokes the Move Utilities and displays the *Personal Computer Move Utilities* panel shown in Figure 35 on page 150.

Panel Functions of the Move Utilities



```
IESADMSL.IESEIWS        PERSONAL COMPUTER MOVE UTILITIES
                                                     APPLID: DBDCCICS
  Enter the number of your selection and press the ENTER key:


        1  List and Process User Files in Host Transfer File
        2  Move VSAM Files to Host Transfer File
        3  Move Files from Host Transfer File to VSAM
        4  Move ICCF Members to Host Transfer File
        5  Move Files from Host Transfer File to ICCF
        6  PC File Transfer











PF1=HELP                    3=END       4=RETURN                 6=ESCAPE(U)
                            9=Escape(m)

==>                                               Path: 2
```

Figure 35. Workstation File Transfer Support Panel

Note that PF6=Escape(U) and PF9=Escape(m) are only displayed if you have the authorization for it defined in your user profile.

From this menu you can select the following functions:

1. **List and process user files in Host Transfer File**

   This selection lets you list all files in the Host Transfer File that you own or that users share with you. A set of options is available to process and filter the files to be displayed.

   This selection is described under "List and Process User Files in Host Transfer File" on page 151.

2. **Move VSE/VSAM files to Host Transfer File**

   This selection moves a VSE/VSAM file which is controlled by CICS to the Host Transfer File.

   This selection is described under "Moving VSE/VSAM Files to Host Transfer File" on page 164.

3. **Move files from Host Transfer File to VSE/VSAM**

   This selection moves a file from the Host Transfer File to an existing VSE/VSAM file which is controlled by CICS.

   This selection is described under "Moving Files from Host Transfer File to VSE/VSAM" on page 165. (Note that this function is also available under selection **1** of the dialog.)

4. **Move VSE/ICCF members to Host Transfer File**

   This selection moves a VSE/ICCF member to the Host Transfer File. Note that this function can only be used in a VSE/ICCF environment.

   This selection is described under "Moving VSE/ICCF Members to Host Transfer File" on page 166.

5. **Move files from Host Transfer File to VSE/ICCF**

   This selection moves a file stored in the Host Transfer File to a VSE/ICCF library member.

This selection is described under “Moving Files from Host Transfer File to VSE/ICCF” on page 167. (Note that this function is also available under selection **1** of the dialog.) This function can only be used in a VSE/ICCF environment.

6. **PC file transfer**

   Makes your host session ready to handle file transfer commands from a workstation.

Selections 2 - 5 give you the possibility to invoke user-written exit routines. “Using User Exits with the Move Utilities” on page 169 has more information about exit routines.

If you are authorized, the panel also displays PF6 (Escape(U)) and PF9 (Escape(m)). These have the same effect as selection 6 described above. PF6 and PF9 interrupt dialog processing and set your terminal in a mode to handle file transfer.

Notes:

1. Each person who uses the *Move Utilities* is responsible for the integrity and security of the data associated with the person’s user ID. This is especially important for the selection ′Move files from Host Transfer File to VSAM′.
2. In case you use this *dialog* for the first time, please familiarize yourself with the material contained in Chapter 5, “Getting Ready for File Transfer”, on page 67.

The following sections describe each of the six dialog choices.

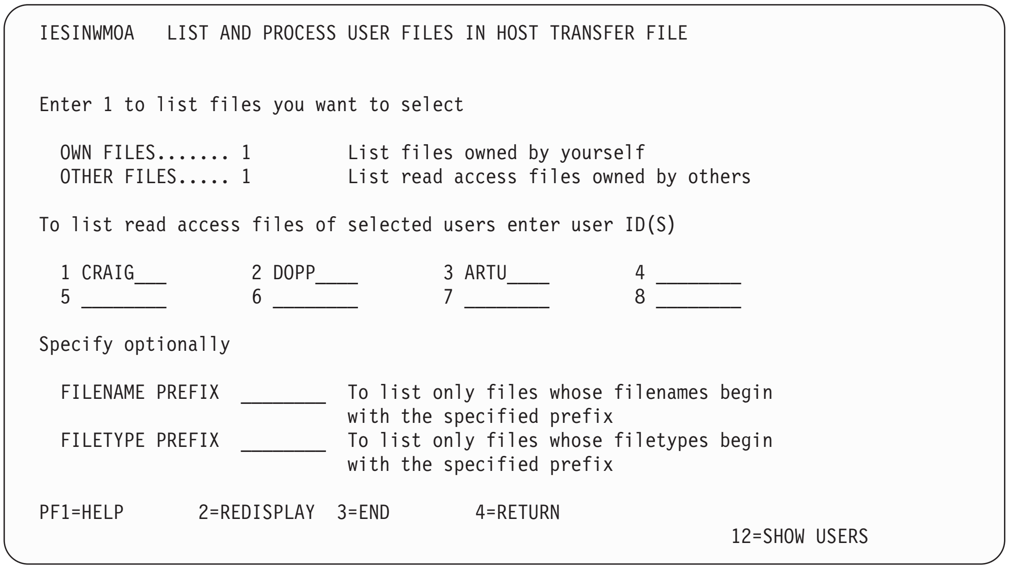
List and Process User Files in Host Transfer File

Option **1** of the *Move Utilities* dialog displays the panel shown below (see Figure 36 on page 152). Use it to control HTF list display. You can filter the list of displayed files according to:

- **File ownership**
  1. Show files which are owned by the requesting user.
  2. Show files of other users, to which the requesting user has read access. This function can be restricted to selected users.
  3. Both of the above.
- **File identification prefix** Show files whose identification matches the given prefixes for filename and filetype.

The displayed file list holds such files which match all specified selection criteria.

Panel Functions of the Move Utilities



```
IESINWMOA   LIST AND PROCESS USER FILES IN HOST TRANSFER FILE


Enter 1 to list files you want to select

  OWN FILES....... 1        List files owned by yourself
  OTHER FILES..... 1        List read access files owned by others

To list read access files of selected users enter user ID(S)

  1 CRAIG___        2 DOPP____        3 ARTU____        4 ________
  5 ________        6 ________        7 ________        8 ________

Specify optionally

  FILENAME PREFIX  ________ To list only files whose filenames begin
                            with the specified prefix
  FILETYPE PREFIX  ________ To list only files whose filetypes begin
                            with the specified prefix

PF1=HELP      2=REDISPLAY  3=END       4=RETURN
                                                          12=SHOW USERS
```

Figure 36. List and Process User Files in the Host Transfer File

OWN FILES
To list all files owned by yourself, enter 1 (default). The file list is sorted by file identification (filename and filetype) in ascending order.

OTHER FILES
To list all files in the HTF to which you have read access, enter 1. All public files of other users, and all files other users have shared with you are displayed.

You may restrict the list of files to selected users. For that, enter 1 to 8 user IDs in the user ID list. Press PF12 to get a list of all user IDs known by the system. The file set is sorted by (1) owner and (2) file identification (Filename and filetype) in ascending order.

USER ID LIST
Restricts the file list to files owned by other users to selected users (1 to 8). This is optional. The files are listed in the sequence (1) owner (as specified in the sequence of user IDs), and (2) file identification (filename and filetype) in ascending order.

FILENAME PREFIX
Restricts the FULIST to those files whose filename begin with a specific prefix. The prefix must consist of one to eight alphanumeric characters without leading or imbedded blanks.

FILETYPE PREFIX
Restricts the FULIST to those files whose filetype begin with a specific prefix.

The program function keys in the panel are:

PF1 (HELP)
displays a help panel. Any input is saved.

PF2 (REDISPLAY)
clears all input fields and sets: OWN FILES = 1. Any input is ignored.

PF3 (END)
leads back to the last Interactive Interface selection panel. Any input is ignored.

PF4 (RETURN)
leads back to the initial *VSE/ESA Function Selection* panel (see Figure 2 on page 8). Any input is ignored.

PF12 (SHOW USERS)
displays a list of all users defined in the system, and allows you to select up to 8 user IDs as input for the fields 1 to 8. Any input is saved.

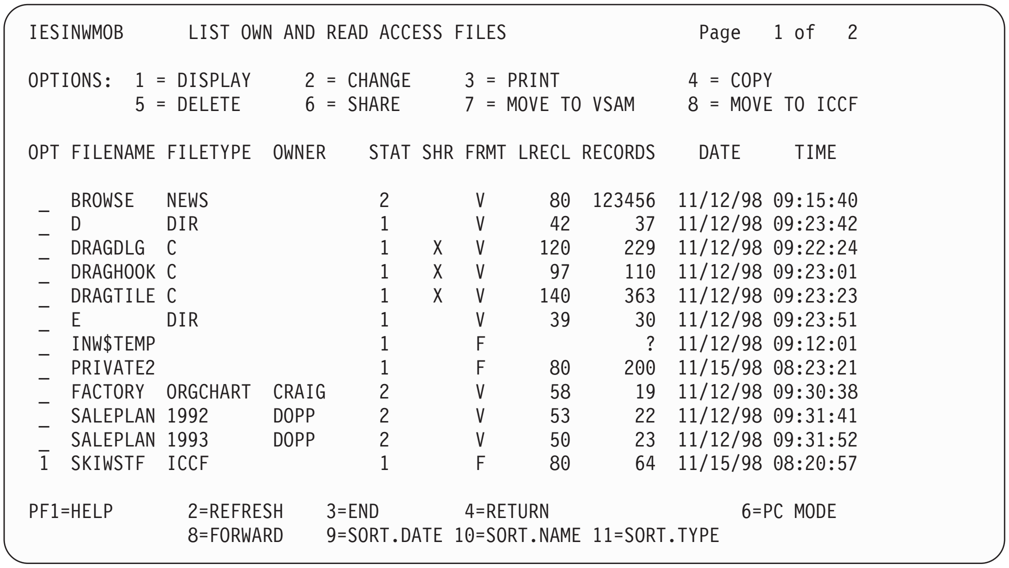
List of Files in Host Transfer File

Figure 37 is an example of a file list. Depending on the parameters you chose in the previous panel (see Figure 36 on page 152), the file list is sorted by:

- **User ID**
  1. Your own files always appear at the top of the list.
  2. Other user's files are listed in one of the following ways:
     a. Alphabetic sequence of the user IDs, when no user list was specified on the previous panel.
     b. The sequence of the user IDs as entered on the previous panel.
- **Filename and filetype**

You also may request the initial sort order from the current panel. An example is given below. To do so, use PF key 2=REFRESH.

Process Files of Host Transfer File: To process the HTF files, select the option number from the OPTIONS list which is displayed in the upper part of the panel and enter the desired option next to the filename.



```
IESINWMOB      LIST OWN AND READ ACCESS FILES                 Page   1 of   2

OPTIONS:  1 = DISPLAY     2 = CHANGE     3 = PRINT           4 = COPY
          5 = DELETE      6 = SHARE      7 = MOVE TO VSAM     8 = MOVE TO ICCF

OPT FILENAME FILETYPE  OWNER    STAT SHR FRMT LRECL RECORDS    DATE     TIME

 _  BROWSE   NEWS                 2        V      80  123456  11/12/98 09:15:40
 _  D        DIR                  1        V      42      37  11/12/98 09:23:42
 _  DRAGDLG  C                    1    X   V     120     229  11/12/98 09:22:24
 _  DRAGHOOK C                    1    X   V      97     110  11/12/98 09:23:01
 _  DRAGTILE C                    1    X   V     140     363  11/12/98 09:23:23
 _  E        DIR                  1        V      39      30  11/12/98 09:23:51
 _  INW$TEMP                      1        F               ?  11/12/98 09:12:01
 _  PRIVATE2                      1        F      80     200  11/15/98 08:23:21
 _  FACTORY  ORGCHART  CRAIG      2        V      58      19  11/12/98 09:30:38
 _  SALEPLAN 1992      DOPP       2        V      53      22  11/12/98 09:31:41
 _  SALEPLAN 1993      DOPP       2        V      50      23  11/12/98 09:31:52
 1  SKIWSTF  ICCF                 1        F      80      64  11/15/98 08:20:57

PF1=HELP       2=REFRESH    3=END        4=RETURN                 6=PC MODE
               8=FORWARD    9=SORT.DATE 10=SORT.NAME 11=SORT.TYPE
```

Figure 37. List Own and Read Access Files

For the files stored in the HTF the options shown in Figure 37 are offered:

1=DISPLAY
Browse the selected file.

This option is described under "Displaying a File" on page 156.

2=CHANGE
Rename and change status of selected file.

This option is described under "Changing a File" on page 159.

3=PRINT
Spool file into VSE/POWER list queue.

This option is described under “Printing a File” on page 159.

4=COPY
Copy file in Host Transfer File.

This option is described under “Copying a File” on page 162.

5=DELETE
Erase file from Host Transfer File if owned by yourself, or drop the shared access to a shared file.

This option is described under “Deleting a File” on page 163.

6=SHARE
Give READ ACCESS for the file to other user(s), and show a list of all users who already have shared access.

This option is described under “Sharing a File” on page 163.

7=MOVE TO VSAM
Move file to VSAM ESDS/KSDS file.

This option is described under “Moving Files from Host Transfer File to VSE/VSAM” on page 165.

8=MOVE TO ICCF
Move file to ICCF library. This option is described under “Moving Files from Host Transfer File to VSE/ICCF” on page 167.

In the panel:

FILENAME
is the name of the file in the Host Transfer File.

FILETYPE
is the type of the file in the Host Transfer File.

OWNER
is the user ID of the owner of the file. If no user is displayed in this column, the file belongs to the user himself.

STAT indicates whether the file is private (1) or public (2).

SHR indicates whether the file is shared (X). If the owner field is blank, the user himself has shared this file with other user(s); otherwise the user was given shared access by the owner.

FRMT shows the record format of the file. Record format is either fixed length (F) or variable length (V).

LRECL
shows the logical record length of the file. For fixed record format (FRMT=F), each record is of the displayed length. For variable record format (FRMT=V), the largest record in the file is of the displayed length.

RECORDS
is the size of the file in numbers of records.

DATE Creation data of the file in the form mm/dd/yy or dd/mm/yy, depending on the format of the system date.

TIME Creation time of the file in hh:mm:ss.

For the program function keys:

PF1 (HELP)
displays a help panel. Any input is saved.

PF2 (REFRESH)
displays a refreshed version of the list in initial sort order. Any input is ignored.

PF3 (END)
leads back to panel *List and Process User Files in Host Transfer File*. Any input is ignored.

PF4 (RETURN)
leads back to the initial *VSE/ESA Function Selection* panel. Any input is ignored.

PF6 (PC MODE)
prepares terminal for file transfer.

PF7 (BACKWARD)
displays previous page.

PF8 (FORWARD)
displays next page. Any input is saved. If any options were specified in the OPT field when PF7, or PF8 is pressed, the options are processed first. Then scrolling is performed.

PF9 (SORT DATE)
displays a refreshed version of the list sorted by (1) creation date, (2) creation time, and (3) user ID. Any input is ignored.

PF10(SORT NAME)
displays a refreshed version of the list sorted by (1) file name, (2) file type and (3) user ID. Any input is ignored.

PF11(SORT TYPE)
displays a refreshed version of the list sorted by (1) file type, (2) file name and (3) user ID. Any input is ignored.

In the sample panel shown by Figure 37 on page 153 the entry INW$TEMP has a question mark (?) in the RECORDS field. This indicates that an error occurred as data was being moved. Because of the error, you should try moving the data again. The file INW$TEMP can be used, but most likely it is *not complete*. It automatically will be deleted when the next file is stored in the Host Transfer File.

Note: When you move data to or from the Host Transfer File, it is first temporarily stored under the internal name INW$TEMP. After it is successfully moved, INW$TEMP is renamed to the file name you specify on the dialog panel or in the workstation SEND command. In case of a system or line breakdown, an existing file with the same name is not destroyed before the new file is written.

Note that the tasks:

- Move file from Host Transfer File to VSE/VSAM
- Move file from Host Transfer File to VSE/ICCF

can be carried out in two ways:

1. Via the selections of the *Personal Computer Move Utilities* panel (see Figure 35 on page 150). In this case, you also have to specify the source file name and the user ID of the owner of the source file.

2. Via options 7 and 8 of the panel *List Own And Read Access Files*. Here you enter either option next to the file you want to move.

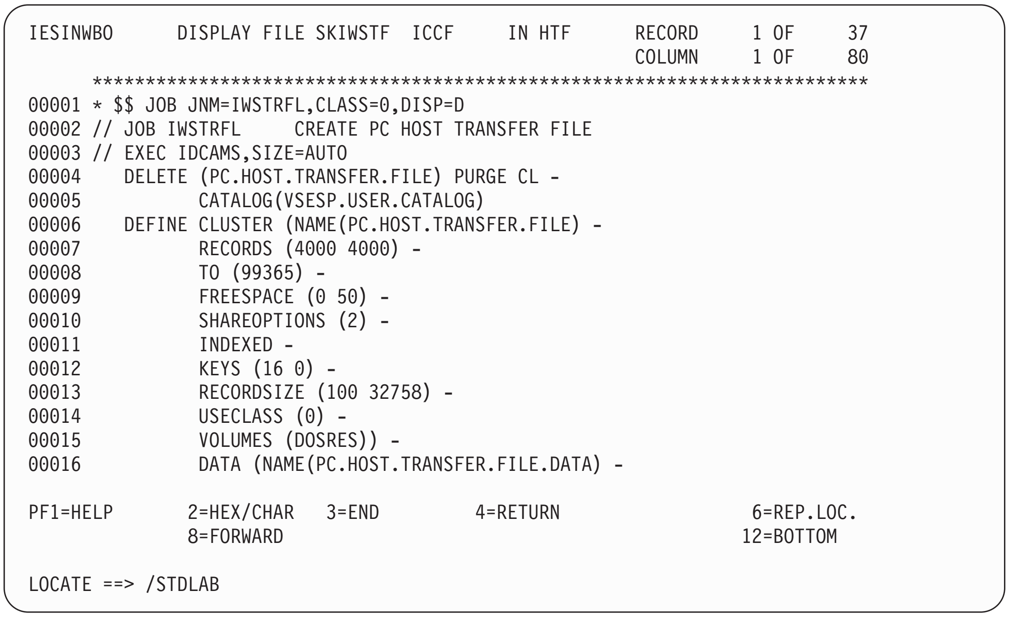
The following sections have details about the **options** available in Figure 37 on page 153.

Displaying a File

This is option 1 in Figure 37 on page 153.

If you select this option for a file, a panel is displayed which shows the beginning of the file selected for browsing. A sample display is in Figure 38.

To browse a file, you can use PF key functions and the Locate function. The RECORD number at the upper right is the one of the second record on your screen (except for the first page, see the figure below).



```
IESINWBO      DISPLAY FILE SKIWSTF  ICCF     IN HTF      RECORD     1 OF     37
                                                         COLUMN     1 OF     80
      ************************************************************************
00001 * $$ JOB JNM=IWSTRFL,CLASS=0,DISP=D
00002 // JOB IWSTRFL     CREATE PC HOST TRANSFER FILE
00003 // EXEC IDCAMS,SIZE=AUTO
00004    DELETE (PC.HOST.TRANSFER.FILE) PURGE CL -
00005           CATALOG(VSESP.USER.CATALOG)
00006    DEFINE CLUSTER (NAME(PC.HOST.TRANSFER.FILE) -
00007           RECORDS (4000 4000) -
00008           TO (99365) -
00009           FREESPACE (0 50) -
00010           SHAREOPTIONS (2) -
00011           INDEXED -
00012           KEYS (16 0) -
00013           RECORDSIZE (100 32758) -
00014           USECLASS (0) -
00015           VOLUMES (DOSRES)) -
00016           DATA (NAME(PC.HOST.TRANSFER.FILE.DATA) -

PF1=HELP       2=HEX/CHAR   3=END         4=RETURN                 6=REP.LOC.
               8=FORWARD                                          12=BOTTOM

LOCATE ==> /STDLAB
```

Figure 38. List and Process User Files in HTF - Display Panel

In the panel:

PF1 displays a help panel. Any input is saved.

PF2 switches between hexadecimal and character display. Character display is the default.

PF3 returns you to the last Interactive Interface selection panel. Any input is ignored.

PF4 ends the display function and returns to the initial *VSE/ESA Function Selection* panel (see Figure 2 on page 8).

PF5 turns DBCS (double byte character set) support on or off. This PF key only is shown if the terminal supports double byte data. Pressing PF5 determines whether the data are to be displayed as double byte characters or as single byte characters.

PF6 repeats the Locate function.

PF7 scrolls backward.

PF8 scrolls forward.

PF9 skips to the top of the file.

PF10 shifts left.

PF11 shifts right.

PF12 skips to the bottom of the file.

The PF keys 5, 7, 8, 9, 10, 11 and 12 only are shown when applicable.

If you are working with a keyboard that does not have all the PF keys available, position the cursor below the PF key description. Press ENTER. This gives you the required function except for PF1.

Use the **Locate function** by entering a search string in the Locate field at the lower left of the panel. Then press ENTER.

DBCS terminal users

If your data are displayed in double byte mode, the part of your file currently shown may contain data which causes a DBCS representation error. In this case, the affected record(s) are marked by a letter (A to H) in front of the record number. A message is issued. Press PF1 for help. To avoid data representation errors, press PF5=*DBCS off*.

Scrolling Forward and Backward

If there is more than one panel of data, use PF7 and PF8 to scroll forward or backward. In addition, you can use the RECORD field. RECORD information and COLUMN information is shown in the top right corner of the panel displayed:

```
RECORD x OF y
COLUMN m OF n

where  x is the record number of the second record on your screen
            (except for the first page).
       y is the total number of records.
       m is the leftmost record position on the display.
       n is the length of the longest record in the file.
```

One record is identical with one line on the screen, but may be longer than 73 characters.

To position to a certain record in the file:

- Replace x with the record number you are looking for.
- Press ENTER.

To position to a certain column of a record:

- Replace m with the column number you are looking for.
- Press ENTER.

Note: You can position to records and columns concurrently.

Using the Locate Function

You can use the locate function to search for the following:

- Single byte character strings.

- Hexadecimal values.
- Double byte character strings (if your terminal supports double byte characters).

The maximum length of a search argument is determined by the length of the command line (60 characters). The search argument format is described below.

If the data you are searching is outside your current window contents, the window will be positioned so that the (first) matching item is displayed about in the middle of the current line. Note that the search starts from the file’s current position which is indicated in the RECORD and COLUMN fields at the upper right of panel. Press PF6 to repeat the search.

Note: The Locate function cannot be used together with RECORD or COLUMN.

Searching for a Single Byte Character (String):

DBCS terminal users

If your terminal supports double byte character representation, switch the input mode to single byte mode. Note that the cursor in the locate field is now only one character long.

Type in the character string in the LOCATE line and then press ENTER. The character string must be enclosed by delimiters. Because character ’x’ indicates hex search, ’x’ should **not** be used as delimiter. For example:

/Result/
will scan for the character string **Result**.

?Corrected /result/ July ?
will scan for a character string **Corrected /result/ July**.

The first example is a 6-character argument. The second one is a 24-character argument. The scan is performed in mixed mode. This means that you can find ’result’, ’RESULT’, or, for example, ’REsuLt’. If your terminal supports Katakana, the search is **not** done in mixed mode.

Searching for a Hexadecimal Value:

Enter the hexadecimal value in the command line and press ENTER. For example:

```
x'C1C2F8F9C3'
```

The search argument always must start with ’x’. The value must be enclosed in apostrophes (’).

DBCS terminal users

Searching for a Double Byte Character (String):

Switch the input mode to double byte mode by pressing *PF5=DBCS ON*, if necessary. Note that the cursor in the locate field is now twice as long. You may only specify a DBCS search argument if the display is in DBCS mode. Type in the double byte character string in the LOCATE line and then press ENTER. The character string need **not** be enclosed by delimiters. The scan is **not** performed in mixed mode. This means that the search will only find the locate string exactly the way it was entered.

Searching for a Mixed Character (String):

If you want to search for a combination of double byte and single byte characters, you must use a single byte delimiter as described in *Searching for a Single Byte Character (String)* above.

When entering the characters in the LOCATE line, proceed as follows:

1. Type in a single byte delimiter.
2. Enter the locate string. Do not forget to set the input mode correctly: Either to double byte mode or to single byte mode, depending on the character currently entered. The cursor then will always show the desired length.
3. Type in a single byte delimiter to terminate the locate string.

Note that you only may specify a DBCS search argument or a combination of double byte and single byte characters if the display is in DBCS mode.

Changing a File

This is option 2 in Figure 37 on page 153.

You can change the name or the status of a file that you own. You cannot use this option to share one of your files with another user. Specify the following:

TARGET FILENAME
Specify the name which the file is to receive. The name must follow the rules listed under "The File's Identification" on page 149.

TARGET FILETYPE
Specify the type which the file is to receive. The type must follow the rules listed under "The File's Identification" on page 149.

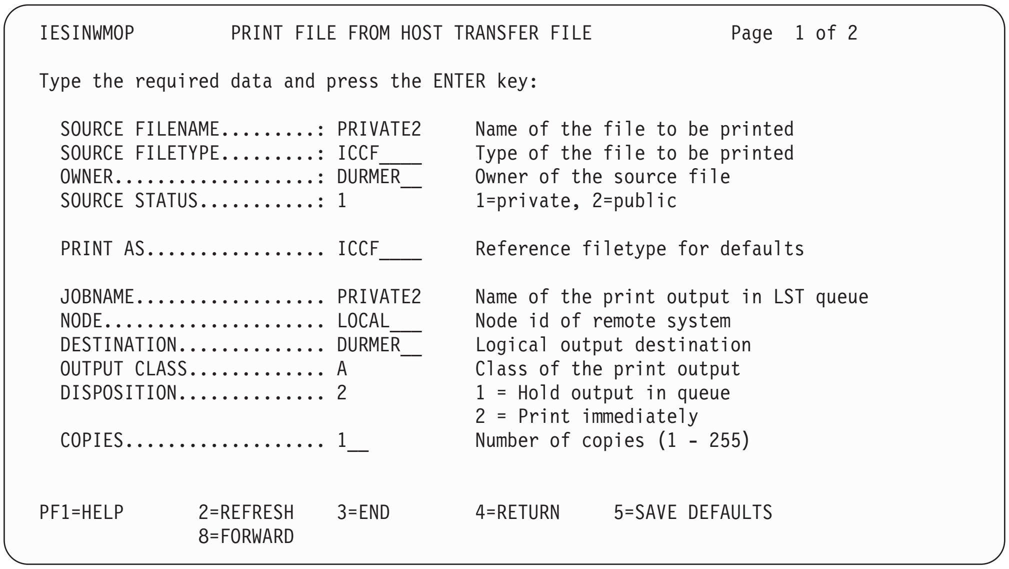
TARGET STATUS
You may change the file's access status from public to private or vice versa.

Printing a File

This function allows the user to print a file from the Host Transfer File. The print option displays additional panels for specification of additional print parameters.

You could make use of print parameters saved previously. The field PRINT AS (reference filetype) indicates which print parameter defaults are (to be) used. Initially the reference filetype is equal to the filetype of the printed file. When a new reference filetype is entered, PF2 must be pressed to load the default values.

Note: This function uses the CICS Report Controller Feature (RCF).



```
IESINWMOP         PRINT FILE FROM HOST TRANSFER FILE             Page  1 of 2

Type the required data and press the ENTER key:

  SOURCE FILENAME.........: PRIVATE2     Name of the file to be printed
  SOURCE FILETYPE.........: ICCF____     Type of the file to be printed
  OWNER...................: DURMER__     Owner of the source file
  SOURCE STATUS...........: 1            1=private, 2=public

  PRINT AS................ ICCF____     Reference filetype for defaults

  JOBNAME................. PRIVATE2     Name of the print output in LST queue
  NODE.................... LOCAL___     Node id of remote system
  DESTINATION............. DURMER__     Logical output destination
  OUTPUT CLASS............ A            Class of the print output
  DISPOSITION............. 2            1 = Hold output in queue
                                        2 = Print immediately
  COPIES.................. 1__          Number of copies (1 - 255)


PF1=HELP       2=REFRESH    3=END        4=RETURN     5=SAVE DEFAULTS
               8=FORWARD
```

Figure 39. Print File From Host Transfer File (First Page)

In the panel above specify the following parameters:

SOURCE FILENAME
Name of the file to be printed (output only)

SOURCE FILETYPE
Type of the file to be printed (output only)

OWNER
Owner of the source file (output only)

SOURCE STATUS
current file status (output only).

PRINT AS
Reference filetype for defaults. Default input field values are either loaded from (PF2) or saved into (PF5) a HTF file with the reference filetype. The defaults filename is INWPDFLT.

JOBNAME
Name of the print output in the VSE/POWER queue. The default equals the filename as displayed in the field SOURCE FILENAME. If the filename of the HTF file name is only one character long it must be extended by the user to at least two characters.

NODE
Node name of the system receiving the print output. The default is LOCAL which means that the output is spooled to the VSE/POWER list queue of the local system. For remote systems, NODE specifies the name of that system as defined in the VSE/POWER PNET node table.

DESTINATION
Logical output destination. This can be the user ID of a user at the local or remote system, or the logical name of a terminal printer. The default is the user ID.

OUTPUT CLASS
Class of the print output in the VSE/POWER queue. Default is class ’A’. Valid classes are A-Z.

DISPOSITION
Disposition of the print output in the VSE/POWER queue. Valid input is 1 (hold) or 2 (keep); default is 2.

COPIES
Number of copies, must be a number between 1 and 255; default is 1.

In the panel above the program function keys are:

PF1 (HELP)
leads to the display of a help panel, any input is saved.

PF2 (REFRESH)
load the default values for reference filetype as specified in field PRINT AS. A blank JOBNAME in the default file is filled with the filename of the source file.

When no defaults file is found the program sets defaults as shown in the panel. (For DESTINATION the user ID of the requesting user is inserted).

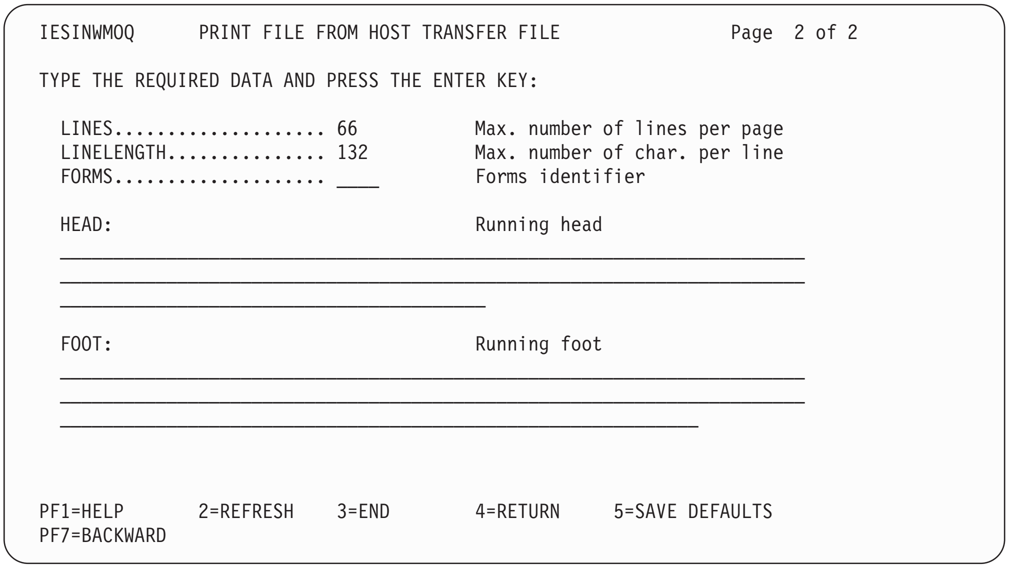
PF3 (END)
leads back to the last Interactive Interface selection panel, any input is ignored.

PF4 (RETURN)
leads back to the initial *VSE/ESA Function Selection* panel (see Figure 2 on page 8), any input is ignored.

PF5 (SAVE DEFAULTS)
saves the values in a default file in the HTF owned by the user, the filename is INWPDFLT, the filetype is as specified in the PRINT AS field.

Field JOBNAME could be saved as empty field; a REFRESH action will then insert the filename of the source file.

PF8 (FORWARD)
Displays the panel shown below in Figure 40. It allows to specify additional print options.



```
IESINWMOQ      PRINT FILE FROM HOST TRANSFER FILE               Page  2 of 2

TYPE THE REQUIRED DATA AND PRESS THE ENTER KEY:

  LINES.................... 66           Max. number of lines per page
  LINELENGTH............... 132          Max. number of char. per line
  FORMS.................... ____         Forms identifier

  HEAD:                                  Running head
  ______________________________________________________________________
  ______________________________________________________________________
  ________________________________________

  FOOT:                                  Running foot
  ______________________________________________________________________
  ______________________________________________________________________
  ____________________________________________________________



PF1=HELP       2=REFRESH    3=END        4=RETURN     5=SAVE DEFAULTS
PF7=BACKWARD
```

Figure 40. Print File From Host Transfer File (Second Page)

In the panel above specify the following parameters:

LINES
Number of lines to be printed per output page; can be a value between 5 and 99. Default is 66.

LINELENGTH
Number of characters to be printed per output line; can be a value between 20 and 205. Default is 132. If the size of the records in the HTF file exceeds the specified linelength, the records are split into segments with the specified linelength.

FORMS
Specifies the type of forms to be used for the output. The operator will be prompted by a message to mount the specified forms before printing is started. When the field is left blank the location's standard form will be used.

HEAD
The running head will be printed as the first line of each output page. The maximum length of the running head is LINELENGTH minus 25. If the text entered by the user exceeds this value, the panel is re-displayed with the text truncated to the allowed maximum length (Warning message is displayed).

FOOT The running foot will be printed as the last line of each output page. The maximum length of the running foot is LINELENGTH minus 5. If the text entered by the user exceeds this value, the panel is re-displayed with the text truncated to the allowed maximum length (Warning message is displayed).

In the panel above the program function keys are:

PF1 (HELP)
leads to the display of a help panel, any input is saved.

PF2 (REFRESH)
load the default values for reference filetype as specified in field PRINT AS on panel IESINWMOP.

PF3 (END)
leads back to the HTF file list.

PF4 (RETURN)
leads back to the initial *VSE/ESA Function Selection* panel (see Figure 2 on page 8), any input is ignored.

PF5 (SAVE DEFAULTS)
saves the values in a default file in the HTF owned by the user, the filename is INWPDFLT, the filetype is as specified in the PRINT AS field.

PF7 (BACKWARD)
Return to the first page of panel *Print File From Host Transfer File*.

Copying a File

This is option 4 in Figure 37 on page 153.

This option allows you to create a copy of the indicated file in the Host Transfer File.

Parameters you must specify are:

TARGET FILENAME
Specify the name which the copied file is to receive in the Host Transfer File. The name must follow the rules listed under “The File’s Identification” on page 149.

TARGET FILETYPE
Specify the type which the copied file is to receive in the Host Transfer File (**optional**). The type must follow the rules listed under “The File’s Identification” on page 149.

TARGET STATUS
Two selections are possible:

- **1** (Private): No other user may have access.
- **2** (Public): All other users may have access.

REPLACE TARGET
Two selections are possible:

- **1**: An existing file with the same file identification as the target file will be replaced.
- **2**: The file is **not** copied if a file with the same file identification as the target file already exists in the Host Transfer File. A warning message will be issued instead.

Deleting a File

This is option 5 in Figure 37 on page 153.

This deletes the file from the Host Transfer File.

If you delete a file that you own, the dialog deletes it from the Host Transfer File. If you delete a file that is shared with you (another user owns it), the dialog only deletes the shared entry so that you can no longer access it. The file itself is not deleted. Only the owner can do that. The system displays an asterisk (*) in the option column as soon as the file has been successfully deleted.

Sharing a File

This is option 6 in Figure 37 on page 153.

A file stored in the Host Transfer File can be shared with other users. That is, if you are the owner of a file, you can give **read access** to one or more other users. For example, a user having read access to a file can use the RECEIVE command to get a copy of the shared file.

After having selected the SHARE option, the system displays all VSE/ESA users defined to the system. An ’X’ in the selection field indicates those users that already have read access to the file you selected. You can then add or delete users. Enter ’1’ for a new user and ’2’ (to overwrite an ’X’) to delete a user. Then press ENTER. For shared files, the following rules are valid:

- A shared file can be replaced by a new version of this file. The previously established access rights remain valid.
- The read access for other users to the file is dropped when the owner either renames or erases the file.
- The owner of a shared file can drop read access for other users by calling the share function and deleting the access right.
- A user who was given read access can drop that access by selecting 5=DELETE for the corresponding file.

Notification Support

VSE/ESA offers notification and acknowledgement support for shared files.

- Any user that is given read access to a file for the first time is notified by a message.
- If the owner of a file replaces a shared file with a new version, all users having read access to this file are notified by a message.
- A user having read access to a file can, depending on the type of file, do one of the following:
  – Copy it into the Host Transfer File.
  – Copy it into the VSE/ICCF library.
  – Copy it into a VSE/VSAM area.
  – Transfer a copy via the RECEIVE command to a workstation.

  If any of these functions is performed, the owner of the file receives an acknowledgement message.

Note: A file may be renamed or erased by its owner. If this happens, the system sends **no** notification message to the users who have read access. Users who had read access to this file will not see this file any longer in their FULIST.

It is the responsibility of the file owner to notify these users if necessary. The *Send Message* dialog can be used for that purpose.

Moving a File to VSE/VSAM

This is option 7 in Figure 37 on page 153. This option is described under "Moving Files from Host Transfer File to VSE/VSAM" on page 165.

Moving a File to VSE/ICCF

This is option 8 in Figure 37 on page 153. This option is only available to users who have access to VSE/ICCF.

This option is described under "Moving Files from Host Transfer File to VSE/ICCF" on page 167.

Moving VSE/VSAM Files to Host Transfer File

Selection **2** of the *Personal Computer Move Utilities* dialog allows you to move entry-sequenced (ESDS) or key-sequenced (KSDS) VSE/VSAM files to the Host Transfer File. Before you do this, make sure the file is:

1. Known to CICS (defined in the CSD using CEDA DEFINE PROGRAM). The service request (SERVREQ) definition should at least allow for read access.
2. Enabled.

Moreover, to move the complete VSAM cluster, you should access VSAM via the full base cluster's file name. This is to ensure unique keys and avoid duplicate key conditions that might occur when using the alternate index, for example.

You need the following information:

SOURCE FILENAME
The name of the file you want to move. You should not use the names of the following system files:

- IESCNTL
- IESTRFL
- INWFILE

TARGET FILENAME
Specify the name for the file in the Host Transfer File. If you do not enter a name, the dialog uses the source file name.

TARGET FILETYPE
Specify the type for the file in the Host Transfer File.

TARGET STATUS
Specify whether the file should be private or public in the Host Transfer File.
1 - Private
2 - Public

REPLACE TARGET
If a file exists in the Host Transfer File with the **same** file identification, specify whether you want it replaced.
1 - YES (Replace)
2 - NO (Do not replace)

EXIT ROUTINE
See “Using User Exits with the Move Utilities” on page 169 for information.

DESCRIPTOR NAME and DESCRIPTOR TYPE
See “Using User Exits with the Move Utilities” on page 169 for information.

Moving Files from Host Transfer File to VSE/VSAM

Selection **3** of the *Personal Computer Move Utilities* dialog allows you to move the selected file from the Host Transfer File to a VSE/VSAM file. The target file (VSE/VSAM file) must meet the following requirements: It has to be

1. an ESDS or KSDS organized file.
2. defined to CICS via the File Control Table. The service request (SERVREQ) definition should at least allow for update access.
3. enabled.

Moreover, to move the complete VSAM cluster, you should access VSAM via the full base cluster’s file name. This is to ensure unique keys and avoid duplicate key conditions that might occur when using the alternate index, for example. In case of problems, contact your system administrator.

You need the following information:

SOURCE FILENAME
The name of the file you want to move.

SOURCE FILETYPE
The type of the file you want to move.

TARGET FILENAME
Specify the name of the existing VSE/VSAM file to which you want to move. You should not use the names of the following system files:

- IESCNTL
- IESTRFL
- INWFILE

OWNER
For public or shared entries which you do **not** own, enter the user ID of the owner. A user ID must be supplied.

DELETE SOURCE FILE
Specify whether the dialog should delete the file from the Host Transfer File after it is moved.
1 - YES (Delete)
2 - NO (Do not delete)

If you enter 1 (YES) for a shared file that another user owns, the dialog only deletes the shared entry so that you can no longer access it. The file itself is not deleted.

EXIT ROUTINE
See "Using User Exits with the Move Utilities" on page 169 for information.

DESCRIPTOR NAME and DESCRIPTOR TYPE
See "Using User Exits with the Move Utilities" on page 169 for information.

After you enter your data, the dialog checks the information. It displays an additional panel. You need to:

1. Enter the name of the target file again.

   This is a security measure. It helps prevent an incorrect update of an existing file.
2. For KSDS files, specify whether the new data should replace or update data already in the target file.
   - Replace

     The dialog first deletes the existing records. It then loads the new records into the file. The CSD file entry for the target file must specify **DELETE** for SERVREQ.
   - Update

     The dialog adds the new records to the existing target file. If it finds a duplicate key, the existing record is replaced with the new record.

   This information is **only** for key-sequenced files. For entry-sequenced files, the dialog ignores any input. For ESDS files, there is no replace. The dialog adds new records to the end of the target file.

Moving VSE/ICCF Members to Host Transfer File

Selection **4** of the *Personal Computer Move Utilities* dialog allows you to move a VSE/ICCF member to the Host Transfer File.

You need the following information:

SOURCE FILENAME
The name of the VSE/ICCF member you want to move.

TARGET FILENAME
Specify the name given to the entry in the Host Transfer File. Follow the rules shown under "The File's Identification" on page 149.

TARGET FILETYPE
Specify the type given to the entry in the Host Transfer File. Follow the rules shown under "The File's Identification" on page 149.

ICCF LIB.-NUMBER
The number of the VSE/ICCF library that contains the member. Note that this entry is optional if the member is in your primary VSE/ICCF library.

FILE STATUS
Specify whether the file should be private or public in the Host Transfer File.
1 - Private
2 - Public

PASSWORD
If the member is password-protected, enter the password.

REPLACE TARGET
If a file exists in the Host Transfer File with the **same** file identification, specify whether you want it replaced.
1 - YES (Replace)
2 - NO (Do not replace)

EXIT ROUTINE
See "Using User Exits with the Move Utilities" on page 169 for information.

DESCRIPTOR NAME and DESCRIPTOR TYPE
See "Using User Exits with the Move Utilities" on page 169 for information.

Notes:

1. The *Move Utilities* dialog stores the records of VSE/ICCF library members in the Host Transfer File as 80 byte fixed-length records. VSE/ICCF print-type members will be truncated at column 80. To move all data of a print-type member, it should be renamed in your ICCF library without using print-type member suffix ″P″ in the new filename.

Moving Files from Host Transfer File to VSE/ICCF

Selection **5** of the *Personal Computer Move Utilities* dialog allows you to move the selected file from the Host Transfer File to VSE/ICCF. Before you do this, make sure the source file contains no source records with more than 80 characters per line. More characters per line will be truncated. Records shorter than 80 characters will be padded with blanks (X'40') on the right.

You need the following information:

SOURCE FILENAME
The name of the file you want to move.

SOURCE FILETYPE
The type of the file you want to move.

TARGET FILENAME
Specify the name which the moved file is to receive in the VSE/ICCF library. Note that the name of a VSE/ICCF member may not contain a slash (/), a comma, or parenthesis.

NEW PASSWORD
You can protect your new VSE/ICCF member against unauthorized access by specifying a password which must be from one to four characters without blanks. Special characters, except commas and parentheses are allowed; the first character must be alphabetic.

ICCF LIB.NUMBER
Specify the VSE/ICCF library number which is to receive your file. If you do not specify one, your primary library will be used.

OWNER
You must enter the user ID of the source file owner if the source file is owned by another user, even if the source file is public or shared. If you

do not know the user ID, use the **PF12=Show Users** function in the panel *List And Process User Files in Host Transfer File* (see Figure 36 on page 152) to get a list of all user IDs defined to the system. Note that you must supply a user ID.

REPLACE ICCF MEMBER

Specify **1** if you want to replace an existing file with the same name as the file in your VSE/ICCF library. If you do not want to replace an existing file, specify **2** (NOREPLACE).

OLD PASSWORD

If you want to replace a target file in the VSE/ICCF library that is password-protected, you must specify this password.

DELETE SOURCE FILE

Two selections are possible:

- **1**: If the file is your file, no matter whether it is private or public, it is deleted from the Host Transfer File after it has been moved to VSE/ICCF. If the file is a shared file owned by another user, the shared access for you will be deleted, but the actual file will be kept.
- **2**: The file is not deleted in the Host Transfer File.

EXIT ROUTINE

See “Using User Exits with the Move Utilities” on page 169 for information.

DESCRIPTOR NAME and DESCRIPTOR TYPE

See “Using User Exits with the Move Utilities” on page 169 for information.

PC File Transfer

Selection **6** of the *Personal Computer Move Utilities* dialog makes the host screen ready to receive file transfer commands from a workstation. Enter this option before you switch to the workstation session.

The system then displays the following:

```
Please switch to PC mode to initiate file transfer or press PF3 to quit.
```

Using User Exits with the Move Utilities

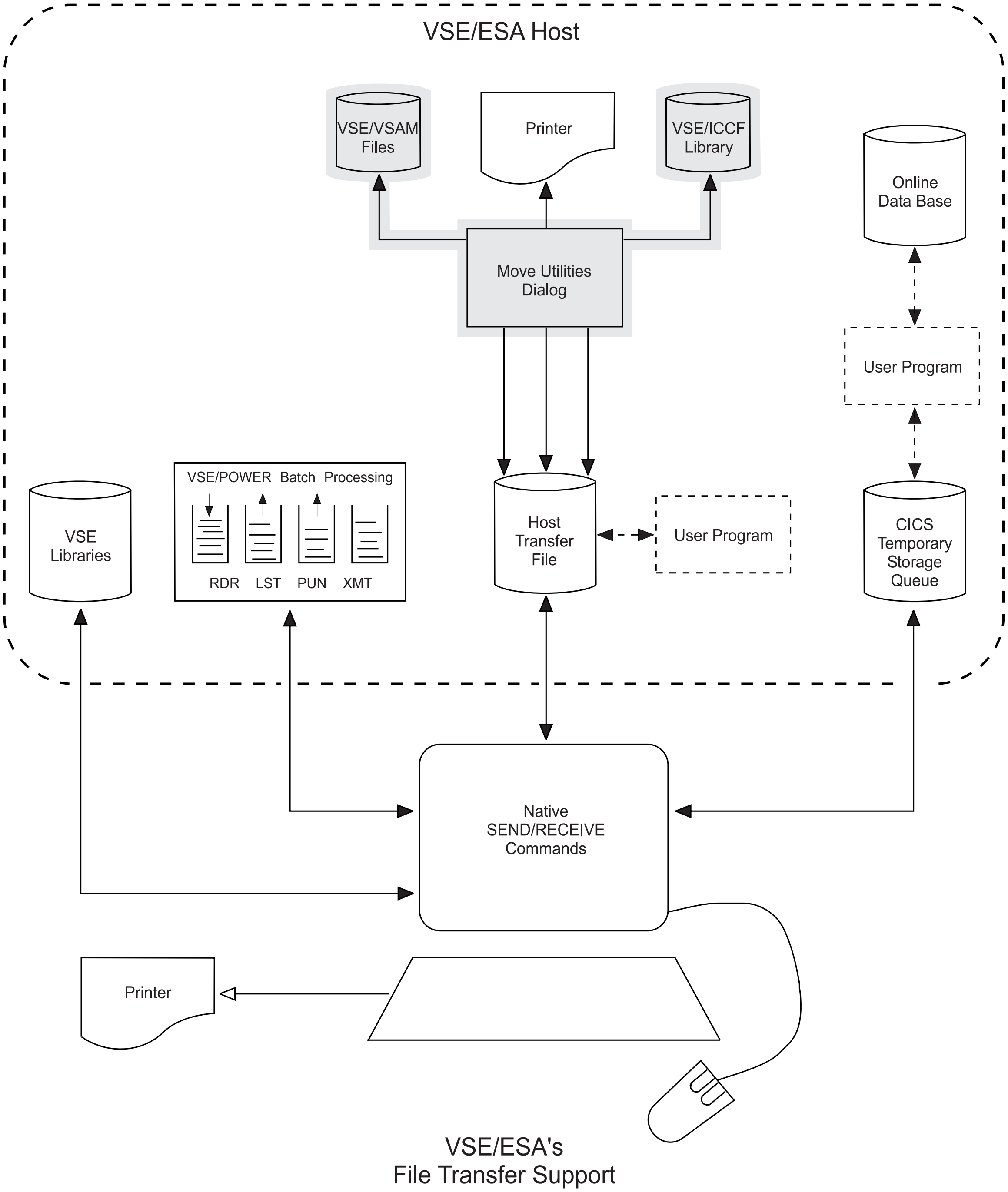


Figure 41. Data Paths of the User Exits of the Move Utilities

The Workstation File Transfer Support provides support for user-written exit programs. Such an exit program allows you to modify individual records when moving files between VSE/VSAM or VSE/ICCF and the Host Transfer File.

User Exits of the Move Utilities

An exit program is a CICS application program that gets control for each record to be moved. Your exit program may convert or modify individual fields or delete an entire record, but you **cannot** insert additional records.

Optionally, you can create a **descriptor record** to tell your exit program how to handle the individual records of a file. For example, to delete or convert specific fields of a record, to add new data to a record, or delete an entire record. You must store the descriptor record in the Host Transfer File before a move operation is initiated.

Note: The descriptor record described here has no relation to the data conversion described under "Creating and Maintaining a Data Conversion Descriptor File" on page 206. A DCDF is used to convert data during file transfer between a workstation and the Host Transfer File. The descriptor record described here can be used to convert data during file transfer between the Host Transfer File and VSE/VSAM or VSE/ICCF.

You activate the user exit by specifying the name of the exit routine and, optionally, a descriptor record (identified by filename and filetype in the HTF) in the corresponding *Move Utility* panel.

VSE/ESA provides samples for user exit programs and descriptor records. They are listed under "Samples of User Exit Programs and Descriptor Records" on page 171.

User Exit Programs

If you specified the name of an exit program for a move operation, it will be invoked with EXEC CICS LINK for each record to be moved. At the same time, the following communication area is supplied:

```
INWCOMM   DS  0CL12   Communication area for user exit program.
DESCADDR  DS  F       Address of descriptor record (hexadecimal
                      zeroes if not used).
RECADDR   DS  F       Address of data record.
DESCLEN   DS  H       Length of descriptor record (hexadecimal
                      zeroes if not used).
RECLEN    DS  H       Length of data record.
EXITAREA  DS  2F      User exit save area (initialized with
                      hexadecimal zeroes).
```

Figure 42. Move Utilities User Exit Communication Area

The **Move Utility** program performs the following steps:

1. Reads the descriptor record (if specified).
2. Reads the first record of the file or member to be moved.
3. Sets up the communication area and links to the user exit routine.
4. Checks the record length (RECLEN) on return from the user exit.
5. Skips the record if RECLEN is zero. Otherwise, moves the record to the target file.

Steps 2 through 5 are repeated until all records are processed. The descriptor record is read only once, but its address is passed with each link. For that reason, the user exit program must not destroy the field containing the descriptor record and address.

Your **exit program** should perform the following steps:

1. Obtain the address of the communication area (DFHEICAP).
2. Check for valid input.

   If the communication area address or the RECADDR or RECLEN fields are zero, return to the calling program.
3. If a record is to be skipped, set RECLEN to zero and return.
4. Otherwise, obtain storage and create a copy of the record. Modify or convert the copy of the record. When finished, move the address of the changed record to RECADDR, move the length of the changed record to RECLEN, and return.

RECADDR and RECLEN may be different from the values received. However, the following rules apply:

- For a move operation from the Host Transfer File to VSE/ICCF, the length of a record must not exceed 80 bytes.
- For a move operation from the Host Transfer File to VSE/VSAM, the maximum record length defined for the VSE/VSAM file cannot be exceeded.
- For a move operation from VSE/VSAM or VSE/ICCF to the Host Transfer File, the maximum record length is 32600.

The user exit save area (EXITAREA) is available to the user exit program to store any information it may want to keep from one linkage to the next.

Descriptor Record

The user exit program must know the record layout in order to convert or delete specific fields, to add new data to a record, or delete an entire record. This information can be passed to the program with a descriptor record.

Descriptor records must reside in the Host Transfer File and must be owned by the requesting user. Public or shared files of other users cannot be used.

You create a descriptor record either on your host system with the VSE/ICCF editor or on a workstation. To store it in the Host Transfer File, use the *Move Utilities* dialog, or a workstation SEND command.

You can freely choose the format and contents of a descriptor record. The file containing the descriptor record must be 80 bytes in length and may consist of any number of physical 80-byte records. They will be concatenated to form one descriptor record. If, for example, the file contains three 80-byte records, DESCADDR will point to the first record and DESCLEN will be 240.

Samples of User Exit Programs and Descriptor Records

Samples of user exit programs, descriptor records, and data files are provided in VSE/ICCF library 59, as shown in Figure 43 on page 172.

MEMBER NAME	LANGUAGE	DESCRIPTOR RECORD	DATA FILE
INWEXTA1	Assembler		INWDAT1
INWEXTA2	Assembler	INWDESC1 or INWDESC2	INWDAT1 INWDAT2
INWEXTC1	COBOL	INWDESC1 or INWDESC2	INWDAT1 INWDAT2

Figure 43. Samples of User Exit Programs

The samples include descriptions on how to install and use them.

Application Program Interface (API) for the HTF

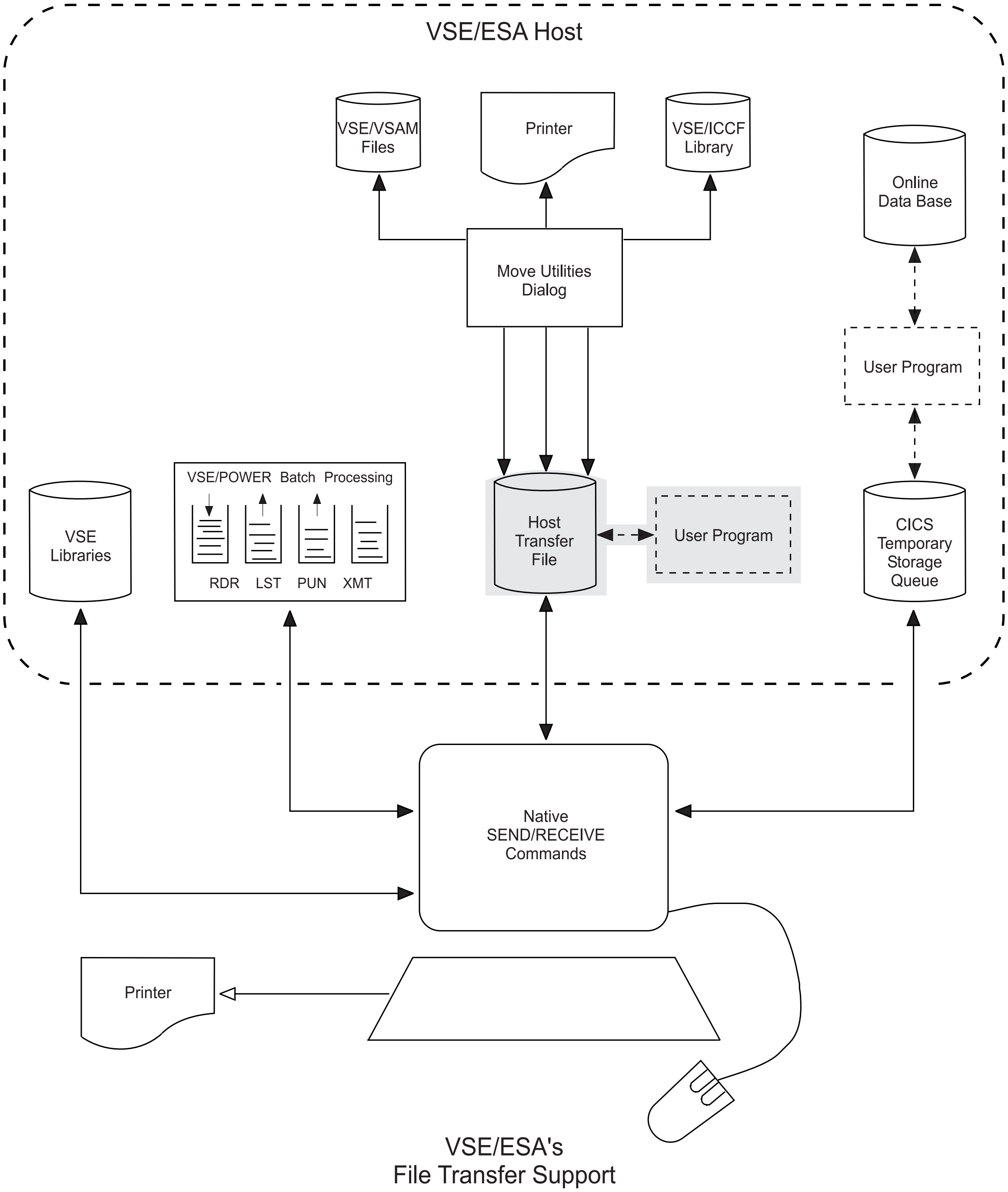


Figure 44. Data Paths of the Application Program Interface of the HTF

Using the Application Interface to the Host Transfer File

Additional considerations

Carefully note the following: For accessing the Host Transfer File (HTF) via the Application Program Interface (API), you must be signed on to VSE/ESA's Interactive Interface.

Overview

VSE/ESA provides an interface that allows CICS application programs to read data from or write data to the Host Transfer File.

In addition, files stored in the Host Transfer File may be deleted or shared with other users. In VSE/ICCF library 59, VSE/ESA provides a sample application program (INWAPIA) written in Assembler language.

The interface provides the following functions for a user-written application program:

- **READ** from the Host Transfer File.

  Reads a record from a file in the Host Transfer File. The maximum record length is 32600. The file can be one of the following:
  - A private file.
  - A file for which the application user has read access.

  In order to read a shared or a public file owned by another user, the application program requires the user ID of the file owner.
- **WRITE** to the Host Transfer File.

  Writes a record to the Host Transfer File. The maximum record length is 32600. With this function the user of the application can create files in the Host Transfer File.
- **SHARE** a file in the Host Transfer File.

  Gives read access to a particular file to other users.
- **DELETE** a file in the Host Transfer File.

  Deletes a file in the Host Transfer File. The file must be owned by the application user.

In addition, the functions **OPEN**x and **CLOSE** are supported. They are required to open and to close a file in connection with the READ and WRITE functions.

User *notification* is optional. If requested, notification messages are issued as listed below.

- When *writing* a file that exists already to the Host Transfer File:

  All users sharing the file are informed that the file has been replaced by a new version.
- When *reading* a file from the Host Transfer File:

  The file owner of a shared file is informed that a copy of the file was requested and has been retrieved.
- When *sharing* a file in the Host Transfer File:

  All specified users are informed that they have read access to the named file.

Invoking the Application Interface

Each time an application program requests one of the above functions, it must link to the interface program INWMAPI. The link must be requested with EXEC CICS LINK together with the address of the *communication area* to be provided by the application program. INWMAPI performs the functions requested by the application program.

The Communication Area

A CICS communication area is needed to exchange information between the application program and the interface program INWMAPI. The application program creates a communication area by obtaining an area of main storage with GETMAIN. The initialization value must be X'00'. In the communication area, the application program must specify the function requested and its related parameters. Next, the application program must link to INWMAPI. On return, the communication area contains the resulting values of the processed function. Figure 45 shows the required layout of the communication area. Its total size is 60 bytes.

```
*---------------------------------------------------------------*
*                  COMMUNICATION AREA FOR                       *
*             APPLICATION AND INTERFACE PROGRAM                 *
*---------------------------------------------------------------*
*  FIELDS GIVEN TO INWMAPI FROM APPLICATION PROGRAM  (INPUT)    *
*---------------------------------------------------------------*
          INWREQU  DS   CL6     Request type
                                (OPENx, READ, WRITE, CLOSE,
                                 SHARE, or DELETE)
          INWNOTI  DS   CL1     Notify indicator
                                X'FF' = YES, X'00' = NO
          INWFNAM  DS   CL8     File name
          INWFTYP  DS   CL8     File type
          INWSTAT  DS   CL1     File status
                                X'F1' = private, X'F2' = public
          INWUPTR  DS    A      Pointer to user IDs
*---------------------------------------------------------------*
*  FIELDS MODIFIED BY INWMAPI AND APPLICATION PROGRAM (MIXED)   *
*---------------------------------------------------------------*
          INWRECA  DS    A      Record address
          INWRECL  DS    F      Record length (maximum 32600)
*---------------------------------------------------------------*
*  FIELDS RETURNED BY INWMAPI TO APPLICATION PROGRAM  (OUTPUT)  *
*---------------------------------------------------------------*
          INWRETC  DS   0CL2    2 byte return code
          INWRC1   DS   CL1     Return code 1
          INWRC2   DS   CL1     Return code 2
          INWUSER  DS   CL8     User ID of file owner for OPENR
                                or user ID if return code X'0418'
          Reserved DS   CL2     Reserved for INWMAPI
          Reserved DS    A      Reserved for INWMAPI
          Reserved DS   CL8     Reserved for INWMAPI
*---------------------------------------------------------------*
*                    END OF USER AREA                           *
*---------------------------------------------------------------*
```

Figure 45. Communication Area Layout

Depending on the request, the application program must set particular fields in the communication area as shown in Figure 46 on page 176. An **x** indicates that the field must be set by the application program. An **r** indicates that a value is returned by the interface program INWMAPI.

Do not change fields other than those marked with an x in Figure 46.

COMMUNICATION AREA ENTRY	OPENW	OPENR	READ	WRITE	SHARE	DELETE	CLOSE
INWREQU	x	x	x	x	x	x	x
INWNOTI	(x)	(x)			(x)		
INWFNAM	x	x			x	x	
INWFTYP	(x)	(x)			(x)	(x)	
INWSTAT	x						
INWUPTR		x*			x		
INWRECA			x,r	x			
INWRECL			x,r	x			
INWRETC	r	r	r	r	r	r	r
INWUSER		r***			r**		

```
Meanings:
:p.
x     =  Required.
(x)   =  Optional.
*     =  Required if user is not file owner but has read access.
**    =  Set by INWMAPI if invalid user ID found in user list.
***   =  Set by INWMAPI if INWUPTR was specified.
```

Figure 46. Field Settings of Communication Area

Return Codes

On return to the application program, the interface program INWMAPI provides return codes in the communication area. The return codes are stored in the fields INWRC1 and INWRC2. Figure 47 on page 177 shows the possible return codes in **hexadecimal** notation and describes their meaning. In the table, Host Transfer File is abbreviated as HTF.

RC1	RC2	MEANING
00	00	Normal return.
00	04	File already exists (OPENW).
00	08	File or record not found.
00	0C	End of file (READ).
00	10	Notification failed.
		USER INTERFACE ERRORS
04	00	Not signed on to interactive interface.
04	04	Facility not a terminal.
04	08	Invalid request.
04	0C	Invalid request sequence.
04	10	Invalid RECORD address.
04	14	Invalid RECORD length.
04	18	Invalid user ID (for share or read public/shared).
04	1C	Invalid interface setup.
04	20	WRITE done by multiple tasks.
04	24	Invalid notify indicator.
04	28	Invalid file name (blank).
04	2C	Invalid status indicator.
04	30	Invalid file type.
		INTERNAL INTERFACE ERRORS
08	04	Link to HTF manager failed.
08	08	HTF not available or disabled.
08	0C	No more VSE/VSAM space in HTF.
08	10	Internal interface error.

Figure 47. Application Interface Return Codes

Application Interfaces

Application Interface for WRITE

For writing a file to the Host Transfer File, the following functions are required:

OPENW
WRITE
CLOSE

The following section shows in which fields of the communication area parameters must be set by the application program.

1. The first request must be an OPENW with the following communication area setup:

```
INWREQU  =  OPENW
INWFNAM  =  name of the file to be written
INWSTAT  =  status of file: X'F1'=private, X'F2'=public

Optionally:

INWFTYP  =  type of the file to be written
INWNOTI  =  X'FF' (if users sharing the file are to be notified)
```

On return, the return codes set in field INWRETC may be:

```
X'0000' = Normal return.  File does not exist.

X'0004' = File exists already.  If the application continues
              with a WRITE request, the file will be replaced.
              To terminate the application, a CLOSE request
              must be issued.
```

For other return codes, refer to Figure 47 on page 177.

2. For each WRITE request, the communication area must have the following setup:
   The maximum record length is 32600.

```
INWREQU  =  WRITE
INWRECA  =  record address
INWRECL  =  record length
```

3. To terminate a single WRITE operation or a WRITE sequence, the application program must finally specify a CLOSE request in field INWREQU.

Application Interface for READ

For reading a file from the Host Transfer File, the following functions are required:

OPENR
READ
CLOSE

The following section shows in which fields of the communication area parameters must be set by the application program.

1. The first request must be an OPENR with the following communication area setup:
   INWUPTR must point to a field of 8 characters in length. It must contain the

```
INWREQU  =  OPENR
INWFNAM  =  name of the file to be read
INWFTYP  =  type of the file to be read

If the file is owned by another user:

INWPUTR  =  pointer to user ID of file owner
INWNOTI  =  X'FF' (if file owner is to be notified)
```

user ID in bytes 0 - 7. The user ID must be left justified and padded with blanks.

On return, the return codes set in field INWRETC may be:

```
X'0000' = Normal return.
X'0008' = File not found.
X'0418' = Undefined user ID.
```

For other return codes, refer to Figure 47 on page 177.

2. For each READ request, the communication area must have the following setup:
   On return, fields INWRECA and INWRECL contain the address and length of

```
INWREQU  =  READ
INWRECA  =  record address: must be X'00000000'
INWRECL  =  record length: must be X'00000000'
```

   the record read. The return codes set in field INWRETC may be:

   ```
   X'0000' = Normal return.
   X'000C' = End of file.
   X'04xx' = INWRECA or INWRECL was not set to X'00000000'.
   ```

   For other return codes, refer to Figure 47 on page 177.
3. To terminate a single READ operation or a READ sequence, the application program must finally specify a CLOSE request in field INWREQU.

Application Interface for SHARE

To give read access for a file to other users, a share request can be issued. The file to be shared must be closed and the application user must be the file owner. The following communication area setup is required:

```
INWREQU  =  SHARE
INWFNAM  =  name of the file to be shared
INWFTYP  =  type of the file to be shared
INWUPTR  =  address of user list

Optionally:

INWNOTI  =  X'FF' (if users sharing the file are to be notified)
```

A maximum of 256 user IDs can be specified in the user list. The list of user IDs must be a string of fields, each 8 characters long. The user ID must be left justified and padded with blanks. The last field must contain hexadecimal zeroes or blanks to terminate the list.

On return, field INWRETC will show return code X'0418' if the user list includes an ID not defined to the system. Field INWUSER will contain the invalid user ID. The sharing request is rejected.

A share request does not require an OPENx or CLOSE request.

Application Interface for DELETE

To delete a file, a delete request must be issued. The application user must be the file owner. A file that is not closed cannot be deleted. The following communication area setup is required:

```
INWREQU  =  DELETE
INWFNAM  =  name of the file to be deleted
INWFTYP  =  type of the file to be deleted
```

A delete request does not require an OPENx or CLOSE request.

Using Different Interface Functions Concurrently

An application program can perform different interface functions concurrently. For example, you may want to copy a file in the Host Transfer File. You can do this by having an OPENR, READ, CLOSE and a OPENW, WRITE, CLOSE sequence processed concurrently. *When using this technique, you must define one communication area for each sequence*.

Note that it is not possible to process two write sequences concurrently.

Additional Considerations

When writing or using an application program, observe the following:

- An application program can only access those files for which the application user has an access right.
- An application program must be initiated from a terminal. It cannot be started as an asynchronous transaction running unattended to a terminal.
- An application program can only be used in a CICS partition using the Interactive Interface of VSE/ESA.
- An application program must not terminate (EXEC CICS RETURN) before:
  - All requested Host Transfer File functions have been processed.
  - A file has been closed (after a write or read request).
- The transaction calling the interface program (INWMAPI) needs a Transaction Work Area (DFHTWA) of at least 128 bytes. If the application program needs its own TWA, the first 128 bytes must be reserved for INWMAPI.

Sample Program

A sample application program (INWAPIA) is shipped in VSE/ICCF library 59. The program displays map INWAMSA which allows you to define Host Transfer File requests and all the related parameters. The program is written in Assembler language. For more details and how to install the sample program, refer to member INWAPIA in VSE/ICCF library 59.

SEND/RECEIVE Command Syntax and Options

The following shows the SEND/RECEIVE command formats for file transfer to/from the Host Transfer File (HTF). The description of parameters and options is given in the next section. In case you are not familiar with the notation of the following syntax diagrams, refer to “How to Read SEND/RECEIVE Syntax Diagrams”, on page 273.

Command Syntax - SEND/RECEIVE (FILE=HTF

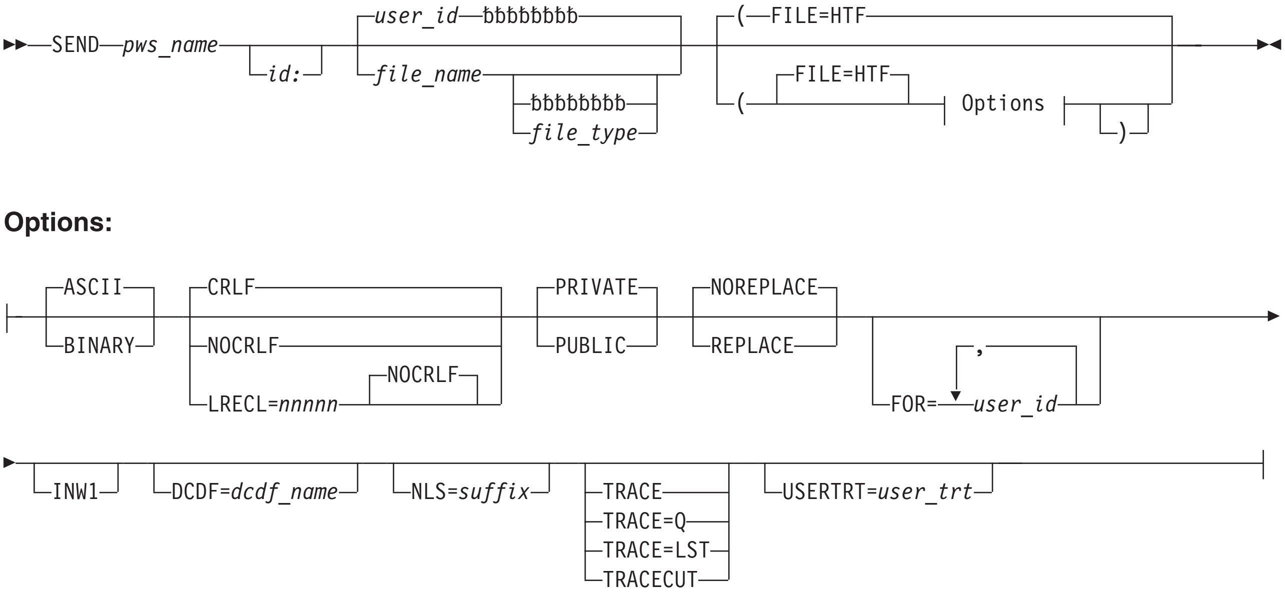


Figure 48. Send/Receive Workstation Files to/from the Host Transfer File - FILE=HTF (Part 1 of 2)

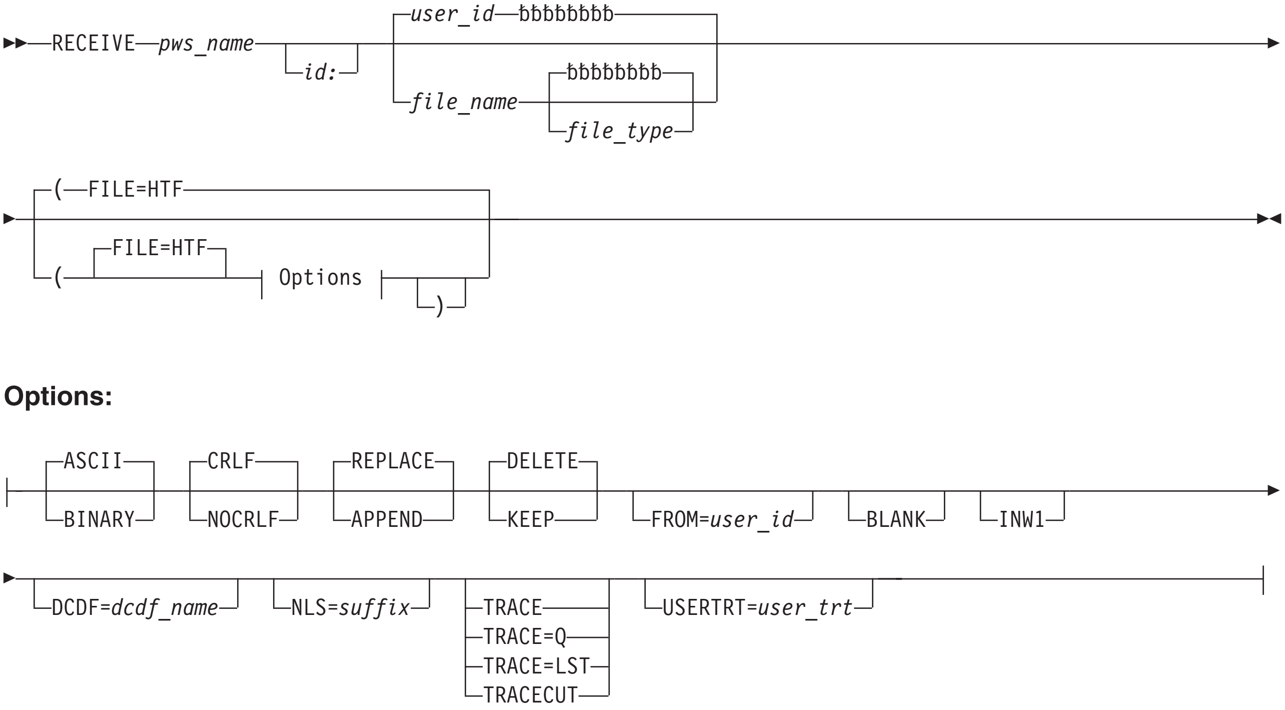


Figure 48. Send/Receive Workstation Files to/from the Host Transfer File - FILE=HTF (Part 2 of 2)

Command Options

For a description of the parameters **pws_name** and **id:**, refer to "Common SEND/RECEIVE Command Syntax" on page 70.

For details about the following **common options**, refer to "Common SEND/RECEIVE Command Options" on page 72.

ASCII/BINARY
CRLF/NOCRLF
REPLACE|APPEND (RECEIVE)
BLANK<
INW1
DCDF=
NLS=
TRACE
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=

Note: The explanations for the following parameters and options are given separately for SEND and RECEIVE in case there are differences in use or effect. Otherwise, the explanation applies to both SEND and RECEIVE.

file_name
allows to specify a name for the file stored in the HTF. To be consistent with the other file transfer environments (VSE/POWER, VSE libraries) the file_name (filename) is now optional. When it is omitted, the user ID (**default**) is used as host filename. The name can be from one to eight characters long and may consist of any characters except for the following:

* : ( ) <>
Trailing underscores (for example, FILE___).
Imbedded blanks.

Dollar signs ($) are allowed but should be avoided since they may cause problems when messages containing the name are displayed at the workstation.

file_type
allows to specify a filetype for the file stored in the HTF. It can be from one to eight characters long and must not contain any of the characters * : ( ) < > or imbedded blanks. Trailing underscores (for example, FILE_), also are not allowed.

The filetype is optional. If it is omitted, a filetype of blank is assumed. Note that the filetype must be DCDF or blank for Data Conversion Descriptor Files (DCDF).

When you use data conversion (DCDF), do **not** use a filetype of INWCMSGS for your data files. This is because the data conversion routines will create a file with filetype INWCMSGS to log conversion errors.

FILE=HTF
defines the file transfer function as send/receive data files to/from the Host Transfer File (default).

options
are one or more of the following optional keyword parameters:

DELETE

RECEIVE: When you receive a file which you own, this file is deleted from the Host Transfer File after is has been successfully copied into the workstation file. Specify KEEP if you do not wish the file to be deleted. DELETE is the **default** for files owned by you. Files you receive from other users with the FROM= option are never deleted.

FOR=user_id

SEND: Allows you to share a file with up to eight users. Such a shared file can only be accessed by a user who logs on to VSE/ESA's Interactive Interface or to CICS with a user ID specified in the **user_id** parameter. **User_id** must be a one to eight character CICS user ID. The user IDs must be separated by commas. Imbedded blanks are not allowed.

If the file is to be shared with more than eight users, you can use the SHARE function of the Move Utilities.

The users specified in the FOR= option are notified by a message (sent to the host session), saying that a file has been stored for them in the Host Transfer File. FOR= together with the PUBLIC option can be used to make a file available to everyone and to send a notification to a selected group of users.

Such a shared file can be updated by its originator by issuing another SEND request for this file with the REPLACE option. The FOR= option need not be specified again. All users that were given access to this file in the previous SEND request will have access to the updated version of the file.

FROM=user_id

RECEIVE: Retrieves a file that was either stored as a PUBLIC file or that another user shared with you. **user_id** must be the user ID of the person who owns the file. The user ID can be up to eight characters long. You can **list and process** files in the HTF in the ***Move Utilities*** dialog to see whether there are any shared files for you, or which PUBLIC files exist from individual users.

If the file you are receiving was sent to you by another user either through the FOR= option on the SEND command or by the SHARE option of the Move Utilities, the owner of the file is notified by a message (sent to the host session) that you have received the file.

KEEP

RECEIVE: When you receive a file which you own, this file, by default , is deleted from the Host Transfer File after is has been successfully copied into the workstation file. Specify KEEP if you do not wish the file to be deleted. Files you receive from other users with the FROM= option are never deleted.

LRECL=nnnnn

SEND: allows to specify a logical record length when workstation files are sent to the HTF. The workstation file is treated as a contiguous string of data which is broken up into nnnnn-byte segments. Each segment forms a logical record in the queue.

The maximum value for **nnnnn** is 32600.

Option LRECL overrides the CRLF option. CRLF is ignored if it is active by default or specified explicitly together with LRECL.

NOREPLACE

SEND: NOREPLACE indicates that if a file with the same name already exists, it is not to be replaced by the new file (**default**). In this case, a message is issued, and the file transfer is terminated.

PRIVATE

SEND: PRIVATE indicates that this is a private file which can only be accessed by its originator (**default**).

PUBLIC

SEND: Specify PUBLIC if you wish to make the file available to everyone in the system.

REPLACE

SEND: Specify REPLACE if you wish to replace an existing file.

Chapter 9. File Transfer To and From CICS Temporary Storage

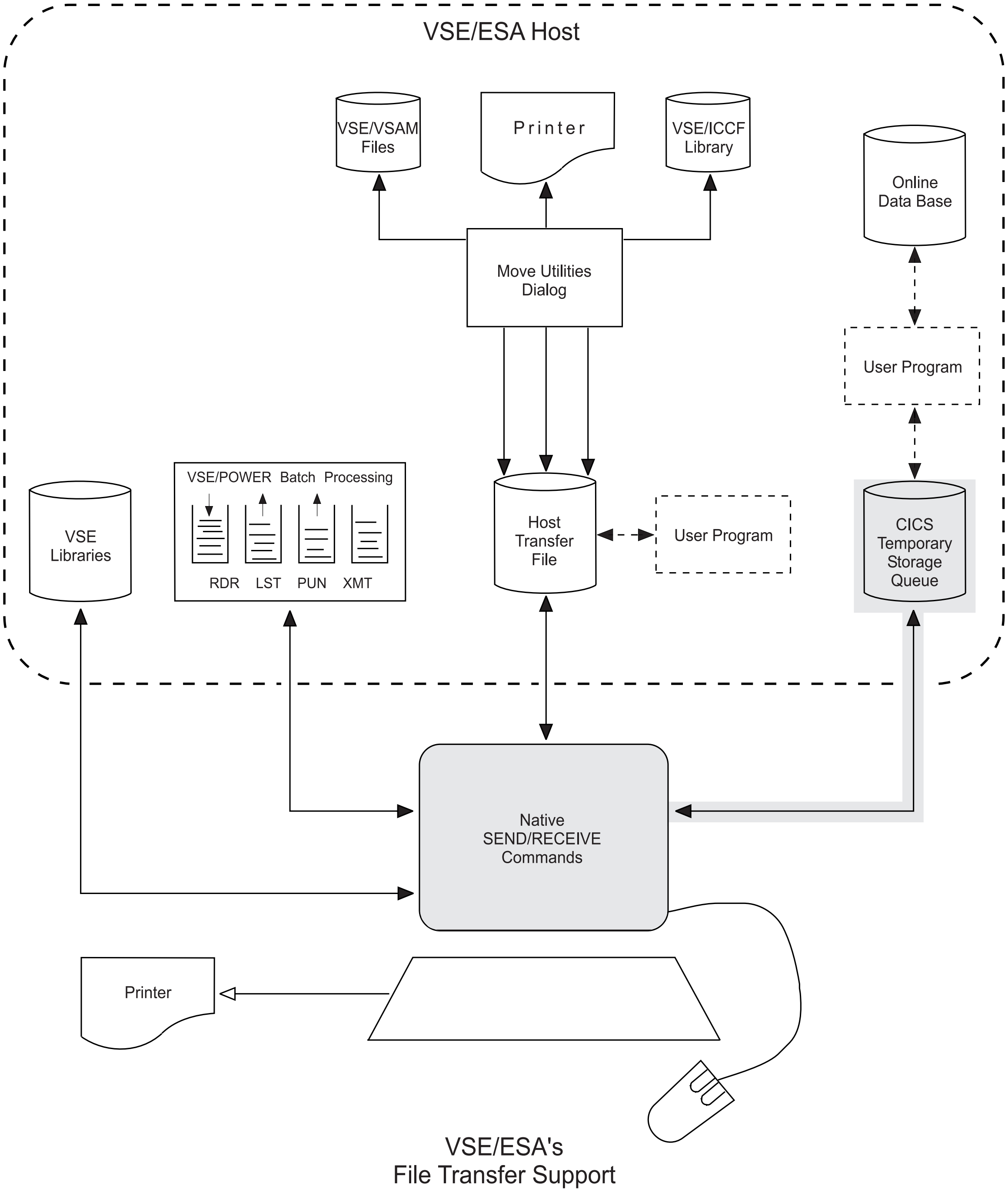


Figure 49. Transferring Data To and From CICS

Overview

Workstation users can send data to or receive data from CICS temporary storage (TS) at the host. Temporary storage is located in VSAM space on disk.

You would send your data to CICS temporary storage if:

- You want to store it for retrieval by another user.
- The data is to be retrieved by an online application.

A TS *queue* is created when the **first file** is stored in temporary storage. Additional files can be **appended** to this same TS *queue*.

Data can be transferred between workstation and temporary storage without further processing. Two workstation users, for example, can exchange data using temporary storage.

You may also want to process data that is sent from a workstation to temporary storage or move data to temporary storage from the host environment. A typical application of the latter would be retrieving of records from an SQL/DS or DL/I data base and storing these records in a temporary storage queue. Workstation users could then retrieve this data from temporary storage.

For such applications, you must provide CICS programs that can retrieve data from temporary storage or store it there. Such programs can be:

- Transactions that are started as a separate task in the host.
- *User exit* programs that the workstation user specifies when sending data to or receiving data from temporary storage. The program must be written according to the specific conventions that are described under "Writing User Exit Programs for CICS Temporary Storage Queues" on page 194. VSE/ICCF library 59 has COBOL, PL/I, and Assembler samples of user exit programs.

The following sections describe data transfer to and from CICS temporary storage (TS) queues by using SEND and RECEIVE commands.

Temporary Storage Conventions

To have a logical separation for the file transfer activities of various users, a unique queue *name* must be defined for each send or receive operation.

If a workstation user does not specify a queue name for a send operation, the system will assign a default:

- The first four characters of the name are **CFTR**.
- The second four characters are the CICS terminal ID (D080, for example).

The **default** for queue name for terminal D080 is thus CFTRD080.

The following rules apply:

- The queue name must always begin with the prefix **CFTR**. The last four characters can be freely chosen if users specify their own queue name.
- The queue name must be at least six characters long. It must not contain the character **$**.
- A queue name can only exist once. If the same queue name is used a second time for a send operation, the system will terminate the transfer and issue an

error message. This is because **NOREPLACE** is the **default** for a send operation. The same queue name can only be used a second time if REPLACE or APPEND is specified.

- By choosing different queue names, the same user can own several queues.

Note: Temporary storage is a common medium used by many transactions and also by CICS itself. Thus, excessive use of temporary storage by one function such as file transfer may impede other functions. In extreme cases, this may even lead to a CICS SOS (short on storage) situation.

Examples

About this Section

This section provides various examples for how you may use SEND/RECEIVE commands for file transfer to and from CICS TS queues.

An example starts with a short heading marked by the '►' character.

How to Use the Examples

Most of the following examples talk about a header record.

The header record is created by the host file transfer program and contains such information as filename and filetype, time and date, user ID, options, and user comments.

For **upload**, the header record is stored as the first record in the TS queue.

For **download**, the first record of the TS queue is assumed to be a header record consisting of at least the filename in positions 1-8. When a user exit program is specified in the RECEIVE command, a complete header record is passed to the program in bytes 29-188 of the communication area.

For a detailed description of the header record, refer to “Layout of the Header Record” on page 197.

In the following examples, only those parts of the header record relevant to the example are shown. The examples explain how to:

- Use SEND default options.
- Use RECEIVE default options.
- View/delete the contents of a TS queue.
- Use host filename and filetype.
- Suppress header record checking (* as filename).
- Specify TS queue names.
- Replace an existing queue.
- Append a file to an existing queue.
- Use correct options for download.
- Invoke a user exit program for upload.
- Invoke a user exit program for download.
- Pass instructions to a user exit program.

File Transfer To and From CICS Temporary Storage (TS)

The SEND/RECEIVE command formats are described in Figure 54 on page 200.

▸ **SEND Defaults:** .

```
User JONES working with CICS terminal address A001 issues the command:

SEND PCFILE  (FILE=TS

Message:

INW0022I File stored in TS queue CFTRA001
```

- **Explanation:**

```
- Workstation file PCFILE is stored in TS queue CFTRA001.
  CFTRA001 is the default queue name which is composed of the characters
  CFTR and the CICS terminal address A001.

- The first record in the queue is the header record. Since no host filename
  was specified in the SEND command, user ID JONES is stored as filename in
  positions 1-8 of the header record.

- Each logical record in PCFILE becomes one record in the queue.
  CRLF is the default.

- Records are translated from ASCII to EBCDIC.
  ASCII is the default.

- The maximum number of records in one TS queue is 32767.

- If a queue CFTRA001 already exists, file transfer is canceled.
  NOREPRLACE is default.
```

▸ **RECEIVE Defaults:**

```
Assume that a TS queue CFTRA001 exists. The first record in the queue is the
header record:
```

JONES 085212 92215 (ASCII CRLF JONES

```
User JONES working with CICS terminal address A001 issues the command:
```

RECEIVE PCFILE (FILE=TS

```
 Message:      INW0025I File received from TS queue CFTRA001
```

- **Explanation:**

```
- The system reads default TS queue CFTRA001.

- Since no host filename is specified, the default name JONES is checked
  against the name in positions 1-8 of the header record.
  A matching name is found, the header record is discarded, and the data records
  are downloaded.

- Each record in the TS queue becomes one record in PCFILE delimited by CR/LF
  characters. CRLF is taken from the header record.

- Records are translated from EBCDIC to ASCII.
  ASCII is taken from the header record.

- TS queue CFTRA001 is not deleted. There is no option to delete
  a queue after download.

- If a file named PCFILE already exists on the workstation it is deleted
  without warning. REPLACE is the default.
```

▸ **Viewing/Deleting the Contents of a TS Queue:**

If you are authorized to work with native CICS (that is, as a VSE/ESA Interactive Interface user), you have the PF6/PF9=ESCAPE keys available. You can view the contents of of a TS queue using the CICS CEBR transaction. In the host session, press either PF key and then type:

```
CEBR CFTRA001
```

• **Explanation:**

- This will let you view the contents of TS queue CFTRA001. You cannot change the contents of the queue.

- To delete the contents of the queue, type PURGE in the command line.

- Press PF3 to return to file transfer mode.

▸ **Host Filename and Filetype:**

User JONES working with CICS terminal address A001 issues the command:

```
SEND PCFILE  HOSTNAME HOSTTYPE (FILE=TS
```

• **Explanation:**

- Host filename and filetype are stored in the header record in CFTRA001:

HOSTNAME 123037 92215 (ASCII CRLF HOSTTYPE JONES

- When the contents of queue CFTRA001 is received, checking is performed on the filename but not on the filetype. Thus any of the following statements will receive the contents of queue CFTRA001:

```
RECEIVE PCFILE  HOSTNAME HOSTTYPE (FILE=TS
        or
RECEIVE PCFILE  HOSTNAME  (FILE=TS
        or
RECEIVE PCFILE  HOSTNAME NEWTYPE (FILE=TS
```

▸ **Suppress Checking for Header Record:**

```
RECEIVE PCFILE  *  (FILE=TS
```

• **Explanation:**

- Specifying an asterisk (*) as host filename causes the entire contents of the queue to be downloaded without any checking.

- If the queue contains a header record, it is downloaded as the first record in the workstation file.

- You should use an asterisk (*) as host filename whenever:

  - You explicitly wish the header record to be downloaded
  - The queue does not contain a header record
  - You don't know whether the queue contains a header record or what the filename in the record is.

- Make sure to delete the header record from the workstation file if it is not wanted.

- A host filetype can be specified, but it is ignored.

- Each record in the TS queue becomes one record in PCFILE delimited by CR/LF characters. **CRLF** is the default.

- Records are translated from EBCDIC to ASCII. **ASCII** is the default.

File Transfer To and From CICS Temporary Storage (TS)

▸ **Specify TS Queue Name:**

```
SEND PCFILE HOSTFILE (FILE=TS QNAME=CFTR01
```

• **Explanation:**

```
- Workstation file PCFILE is stored in TS queue CFTR01.

- Specify the QNAME= option if a queue other than the default queue is to
  be used.

- The queue name must start with the letters CFTR und must be at least six
  characters long.

- The QNAME= option can be specified on the SEND and RECEIVE commands.
```

▸ **Replace an Existing TS Queue:**

```
SEND PCFILE HOSTFILE (FILE=TS QNAME=CFTR01 REPLACE
```

• **Explanation:**

```
- If a queue CFTR01 already exists, it is replaced with the contents of PCFILE

- If a queue CFTR01 does not exist, a new queue is created.
```

▸ **Append to an Existing TS Queue:**

```
SEND PCFILE HOSTFIL1 (FILE=TS QNAME=CFTR01 APPEND
```

• **Explanation:**

```
- The contents of PCFILE (including the new header record) is appended to the
  end of queue CFTR01.

- If a queue CFTR01 does not exist, a new queue is created.

- The number of files that can be appended to one queue is limited only by
  the maximum number of records per queue (32767) and by the amount of VSAM
  space available.

- When a queue containing several appended files is downloaded to the
  workstation, it is treated as if it contained one single file.
  The first header record is processed and discarded. The remaining records
  (including the header records) are downloaded without further checking.
```

▸ **Options in the Header Record:**

When receiving a file from a TS queue, the correct options must be used for logical record handling and EBCDIC/ASCII translation.

If you want to receive the contents of a queue that was stored by someone else, you may not know the format of the queue data and which options to use for download. You should note the following.

The active options ASCII, BINARY, CRLF, or NOCRLF are saved in the header record when a file is stored in a TS queue:

```
SEND PCFILE HOSTFILE (FILE=TS  BINARY
```

creates header record:

HOSTFILE 124237 92215 (BINARY NOCRLF JONES

The options from the header record can then be used for download:

```
RECEIVE PCFILE HOSTFILE (FILE=TS
```

• **Explanation:**

- The SEND operation stores the specified option BINARY and implied option NOCRLF in the header record.
- The RECEIVE command specifies a host filename which triggers header record checking.
- If **none** of the options ASCII, BINARY, CRLF, or NOCRLF are specified in the RECEIVE command and a valid pair of these options is found in the header record, then the header record options are used.
- If **one** of the options ASCII, BINARY, CRLF, or NOCRLF is specified in the RECEIVE command, the options in the header record have no effect.
- Remember that the options from the header record can only be processed when a host filename is specified. An asterisk (*) as filename suppresses header record checking.

▸ **Invoking a User Exit Program for Upload:**

```
SEND PCFILE HOSTFILE (FILE=TS  PROGRAM=CFTRPUT
```

• **Explanation:**

- The data from PCFILE is stored in default TS queue.
- When all data has been stored successfully, control is transferred to user-written program CFTRPUT, which can process the data in the queue as needed.
- Option PROGRAM= can be abbreviated to PROG=.
- For a description of how to write user exit programs, refer to

"Writing User Exit Programs for CICS Temporary Storage Queues" on page 194.

File Transfer To and From CICS Temporary Storage (TS)

▸ **Invoking a User Exit Program for Download:**

```
RECEIVE PCFILE HOSTFILE (FILE=TS  PROGRAM=CFTRGET
```

• **Explanation:**

- First user exit program CFTRGET is called (EXEC CICS LINK) to load data into the TS queue.
- When all data has been stored successfully, CFTRGET returns control to the file transfer program, which downloads the contents of the queue to the workstation.
- Option PROGRAM= can be abbreviated to PROG=.
- For a description of how to write user exit programs, refer to

"Writing User Exit Programs for CICS Temporary Storage Queues" on page 194.

▸ **Passing Information to User Exit Programs:**

```
SEND PCFILE  (FILE=TS  PROG=CFTRPUT) INSTRUCTIONS FOR CFTRPUT

RECEIVE PCFILE  (FILE=TS  PROG=CFTRGET) INSTRUCTIONS FOR CFTRGET
```

• **Explanation:**

- The comment area of the SEND/RECEIVE commands can be used to pass information to the user exit program (for example, to instruct the program how to process the uploaded data or where to get the data for download).
- File transfer options and comments must be separated by a closing parenthesis ')'.
- Sixty characters following the closing parenthesis are treated as comments and are stored in the header record.
- Multiple blank characters in the comments are reduced to one blank by the workstation emulation program before the command is sent to the host.
- Remember: For SEND, the header record is stored as the first record in the TS queue. For RECEIVE, the header record is passed in the communication area.

Rules for Sending and Receiving Files from CICS TS

Rules for Sending Files to CICS Temporary Storage

Carefully note the following:

1. A header record is always the first item built during a send operation. It serves to identify the contents of a queue to a user program. Figure 52 on page 198 shows the layout of the header record.
2. Each logical record in the PC file is stored as one item in the temporary storage queue. When the NOCRLF option is active, the complete contents of one file transfer buffer is stored in one item.
3. If the PC file has an EOF indicator (X'1A'), it will not be stored in the temporary storage queue.

Rules for Receiving Files from CICS Temporary Storage

Carefully note the following:

1. The temporary storage queue to be downloaded to the workstation may or may not contain a header record. If there is **no** header record, an asterisk (*) should be specified as the host file name in the RECEIVE command. This will cause all records in the queue to be downloaded without checking for the header record.

   If there is a **header record**, it must contain the host file name in columns 1 - 8. No other information is needed. The file name specified in the RECEIVE command is checked against the file name in the header record. If the names do not match, the system issues an error message and ends the file transfer. Otherwise, the header record is discarded and the data records are downloaded to the workstation. If an asterisk is specified for the file name, the header record is downloaded together with the data records.
2. There is **no** checking for matching host filetypes.
3. RECEIVE always downloads the complete contents of a TS queue. If a queue contains several appended files, all files are stored in the receiving workstation file, including the header records (except the first one).
4. The options ASCII, BINARY, CRLF, or NOCRLF can be specified in the header record by a user program. If the queue was built with a SEND operation, they are always present. If none of these options is specified in the RECEIVE command, the options in the header record are taken.

   If one or both options in the header record are invalid, the **default** values ASCII/CRLF are used.
5. If the CRLF option is in effect, each item from the temporary storage queue is stored in the workstation file as one logical record, delimited by CR/LF characters. In case of NOCRLF, the temporary storage queue items are concatenated in the PC file to form one contiguous string of data.
6. After a RECEIVE command, the queue is not deleted. This allows issuing more than one RECEIVE command for the same queue.
7. If the contents of a queue is no longer needed, make sure that the queue is deleted to free temporary storage.

Writing User Exit Programs for CICS Temporary Storage Queues

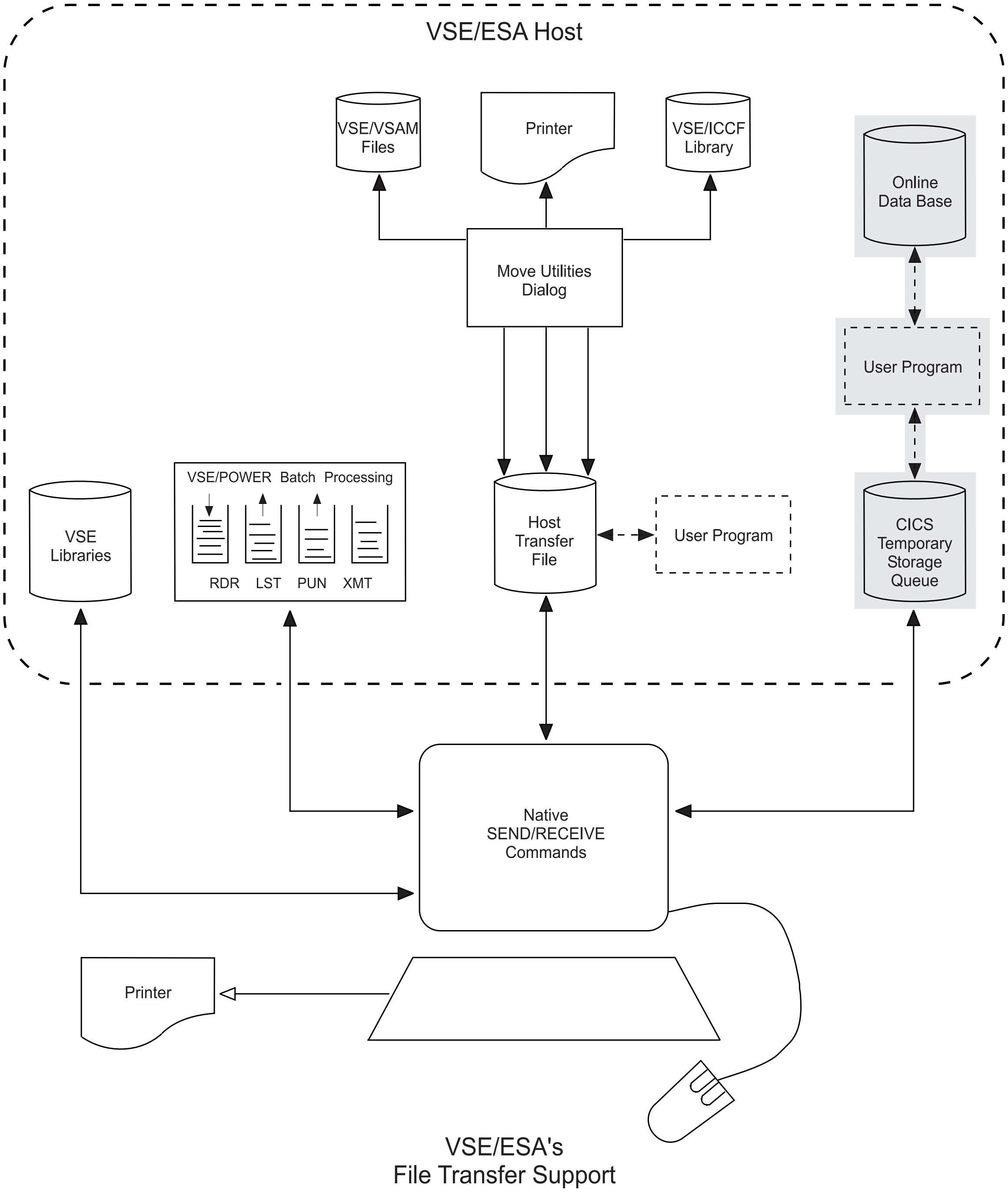


Figure 50. Data Paths of User Exit Programs for CICS Temporary Storage Queues

User applications that process data in temporary storage queues can be:

- Started as separate transactions in host operating mode or
- Invoked from the PC as user exit programs during send or receive operations.

For programs that are started as separate transactions, the only requirements are that they follow the naming conventions for queue names (CFTRxxxx) and can handle the queue's *header* record. This is always the first record in the queue. Figure 52 on page 198 shows its format.

The requirements for application programs that are invoked as user exits are described in the sections Activating a User Exit Program through Layout of the Header Record. Both types of user programs must be defined to CICS using CEDA DEFINE PROGRAM.

Activating a User Exit Program

User exit programs can be invoked by specifying the program name in the SEND and RECEIVE commands. In both cases, the name of the program must start with the characters **CFTR**. The name must be at least six characters long.

For example, if a workstation user with terminal ID **D080** issues the command:

```
SEND PCFILE HOSTFILE (FILE=TS PROGRAM=CFTRGET
```

the file PCFILE will be transferred to temporary storage queue CFRTD080. The queue's header record will contain the name HOSTFILE. Once transfer is complete, the program CFTRGET receives control. It can then read and process the data from the temporary storage queue.

Similarly, the command:

```
RECEIVE PCFILE HOSTFILE (FILE=TS PROGRAM=CFTRGET
```

establishes a link to CFTRGET for loading data into the temporary storage queue CFTRD080. Once the data is loaded, it will be transferred to the workstation and stored in **PCFILE**.

User Exit Communication Area

Communication between the file transfer programs and a user exit program follows CICS interface conventions. Information is exchanged through a COMMAREA. Figure 51 shows the layout of that COMMAREA.

BYTE NUMBER	LENGTH	DATA TYPE	CONTENTS

1	1	CHAR	U = SEND (upload)
			D = RECEIVE (download)

2	1	CHAR	Reserved
3-4	2	BINARY	Return code (initialized) to X'0000')
5-12	8	CHAR	Host file name
13-20	8	CHAR	Queue name
21-28	8	CHAR	SEND: Host file type RECEIVE: If comments are specified, first 8 bytes of comments, else host file type
29-188	160	CHAR	Header Record (RECEIVE only)

Figure 51. Layout of User Exit Communication Area

Conventions for the SEND Operation

Carefully note the following *interface* rules:

1. After the workstation data is stored in the TS queue, control is transferred to the user program via **EXEC CICS LINK** passing the COMMAREA.

   The header record, including any user comments, is stored as first record in the TS queue.
2. The user program is called after file transfer to the TS queue has completed. Thus, when the file transfer completion message **INW0024I** is displayed at the workstation the host session may still be busy for a while executing the user program. It is, therefore, **not** recommended to execute SEND commands which invoke a user program from a command procedure because the host session may not be ready to accept the next SEND.
3. The return code in the communication area has **no** significance.

Conventions for the RECEIVE Operation

Carefully note the following *interface* rules:

1. First, a link to the user program is established (EXEC CICS LINK). The header record is passed in bytes 29-188 of the COMMAREA. The user program must store data into the temporary storage queue for the receive operation. The first record must be a header record.
2. If a queue with the same name already exists, the user program must delete the queue.
3. If resource security level checking is active, an EXEC CICS SYNCPOINT request must be issued following the deletion of the queue.
4. If the queue name passed in the COMMAREA is a default queue name, the user program may overwrite it and use a different queue name.

5. EXEC CICS HANDLE ABEND should not be used by the user program.
6. A non-zero return code passed back by the user program in the COMMAREA indicates that file transfer is to be terminated and an error message is to be issued.

Layout of the Header Record

Figure 52 on page 198 shows the layout of the header record for a temporary storage queue.

The following SEND command example would generate a header record as shown. SYSA is the ID of the user who initiated the operation.

```
SEND PCFILE HOSTNAME HOSTTYPE (FILE=TS REPLACE) THIS IS ...
```

Resulting header record:

```
Byte: 1        10     17     23    31     38           56       95
      |         |      |      |     |      |            |        |

      HOSTNAME  151501 93171 (ASCII CRLF   HOSTTYPE ... SYSA )   THIS IS...
```

User Exit Programs for CICS Temporary Storage Queues

For SEND, the header record is stored as the first item in the CICS temporary storage queue; for RECEIVE, the header record is passed in bytes 29-188 of the COMMAREA.

BYTES	LENGTH	CONTENTS
1–8	8	Host file name
9	1	Blank
10–15	6	Creation time: hhmmss
16	1	Blank
17–21	5	Creation date: yyddd
22	1	Blank
23	1	Left parenthesis
24–29	6	'ASCII' or 'BINARY'
30	1	Blank
31–36	6	'CRLF' or 'NOCRLF'
37	1	Blank
38–45	8	Host file type
46	1	Blank
47–54	8	Reserved
55	1	Blank
56–63	8	User ID
64	1	Blank
65–92	28	Reserved
93	1	Blank
94	1	Right parenthesis
95–160	65	User comments

```
:p.
  All fields have data format CHARACTER.
```

Figure 52. Layout of Header Record

Sample User Exit Programs

To help you write your own user exit programs, VSE/ESA provides the sample programs listed in the following figure. These programs are stored in VSE/ICCF library 59. Each sample program includes a detailed description.

PROGRAM NAME	VSE/ICCF MEMBER	DESCRIPTION
CFTRUPA	INWSAMA1	Assembler program for an upload operation with a SEND command.
CFTRDNA	INWSAMA2	Assembler program for a download operation with a RECEIVE command.
CFTRUPC	INWSAMC1	COBOL program for an upload operation with a SEND command.
CFTRDNC	INWSAMC2	COBOL program for a download operation with a RECEIVE command.
CFTRUPP	INWSAMP1	PL/I program for an upload operation with a SEND command.
CFTRDNP	INWSAMP2	PL/I program for a download operation with a RECEIVE command.

Figure 53. Sample Programs for Temporary Storage Transfer Function

Note: Carefully note that the **length of the header record** (160 bytes) in the CICS temporary queue must **not** be changed in these exit programs. Otherwise, file transfer does not work.

CICS TS Access Without Signon to the Interactive Interface

There are no special restrictions for file transfer to CICS TS. For information about general restrictions, refer to “File Transfer Without Signon to the VSE/ESA Interactive Interface” on page 85.

SEND/RECEIVE Command Syntax and Options

The following shows the SEND/RECEIVE command formats for file transfer to and from CICS TS. The description of the parameters and options is given in the next section. In case you are not familiar with the notation of the following syntax diagrams, refer to “How to Read SEND/RECEIVE Syntax Diagrams”, on page 273.

Command Syntax - SEND/RECEIVE (FILE=TS

```
                              ┌─user_id──ƀƀƀƀƀƀƀƀ──────┐
►►──SEND──pws_name──┬───────┬─┼────────────────────────┼──(──FILE=TS──┤ Options ├──┬──────────────┬──────►◄
                    └─id:───┘ │            ┌─ƀƀƀƀƀƀƀƀ──┐ │                         └─)──comments──┘
                              └─file_name──┼───────────┼─┘
                                           └─file_type─┘
```

Options:

```
   ┌─ASCII──┐  ┌─CRLF─────────────────────┐  ┌─NOREPLACE─┐  ┌─QNAME=CFTRtermID─┐
├──┼────────┼──┼──────────────────────────┼──┼───────────┼──┼──────────────────┼──────────────────────►
   └─BINARY─┘  ├─NOCRLF───────────────────┤  ├─REPLACE───┤  └─QNAME=CFTRxxxx──┘
               │                ┌─NOCRLF─┐│  └─APPEND────┘
               └─LRECL=nnnnn────┴────────┴┘

►──┬──────────────────┬──┬──────┬──┬────────────────┬──┬────────────┬──┬─────────────┬────────────────►
   ├─PROGRAM=CFTRxxxx─┤  └─INW1─┘  └─DCDF=dcdf_name─┘  └─NLS=suffix─┘  ├─TRACE───────┤
   └─PROG=CFTRxxxx────┘                                                ├─TRACE=Q─────┤
                                                                       ├─TRACE=LST───┤
                                                                       └─TRACECUT────┘

►──┬───────────────────┬──────────────────────────────────────────────────────────────────────────────┤
   └─USERTRT=user_trt──┘
```

Figure 54. Send/Receive Workstation Files to/from CICS Temporary Storage Queues - FILE=TS (Part 1 of 2)

```
                                  ┌─user_id──ƀƀƀƀƀƀƀƀ──────┐
►►──RECEIVE──pws_name──┬───────┬──┼────────────────────────┼──(──FILE=TS──────────────────────────────►
                       └─id:───┘  │            ┌─ƀƀƀƀƀƀƀƀ──┐ │
                                  └─file_name──┼───────────┼─┘
                                               └─file_type─┘

►──┤ Options ├──┬──────────────┬─────────────────────────────────────────────────────────────────────►◄
                └─)──comments──┘
```

Options:

```
   ┌─ASCII──┐  ┌─CRLF───┐  ┌─REPLACE─┐  ┌─QNAME=CFTRtermID─┐
├──┼────────┼──┼────────┼──┼─────────┼──┼──────────────────┼──┬──────────────────┬───────────────────►
   └─BINARY─┘  └─NOCRLF─┘  └─APPEND──┘  └─QNAME=CFTRxxxx───┘  ├─PROGRAM=CFTRxxxx─┤
                                                              └─PROG=CFTRxxxx────┘

►──┬───────┬──┬──────┬──┬────────────────┬──┬────────────┬──┬─────────────┬──┬───────────────────┬──┤
   └─BLANK─┘  └─INW1─┘  └─DCDF=dcdf_name─┘  └─NLS=suffix─┘  ├─TRACE───────┤  └─USERTRT=user_trt─┘
                                                            ├─TRACE=Q─────┤
                                                            ├─TRACE=LST───┤
                                                            └─TRACECUT────┘
```

Figure 54. Send/Receive Workstation Files to/from CICS Temporary Storage Queues - FILE=TS (Part 2 of 2)

Command Options

For a description of the parameters **pws_name** and **id:**, refer to “Common SEND/RECEIVE Command Syntax” on page 70.

For details about the following **common options**, refer to “Common SEND/RECEIVE Command Options” on page 72.

```
ASCII/BINARY
CRLF/NOCRLF
BLANK
INW1
DCDF=
NLS=
TRACE
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=
```

Note: The explanations for the following parameters and options are given separately for SEND and RECEIVE in case there are differences in use or effect. Otherwise, the explanation applies to both SEND and RECEIVE.

file_name

SEND: allows to specify a filename for the file stored in the TS queue. To be consistent with the other file transfer environments (VSE/POWER, VSE libraries) the file_name (filename) is **optional**. If it is omitted, the user ID (**default**) is used as host filename. The filename can be from one to eight characters long and may consist of any characters except for the following:

* : ( ) <>
Trailing underscores (for example, FILE___).
Imbedded blanks.

Dollar signs ($) are allowed but should be avoided since they may cause problems when messages containing the name are displayed at the workstation.

The filename is stored in position **1-8** of the header record.

RECEIVE: the file_name (filename) may be a single asterisk (*). If an asterisk is specified the entire contents of the queue (including the header record, if any) is received.

When a filename is specified explicitly, it will be checked against the filename in the header record. If a matching name is found, the header record is discarded, and the data records are downloaded.

If the filenames do **not** match, file transfer is cancelled.

file_type

SEND: allows to specify a file_type (filetype) for the file stored in the TS queue. It can be from one to eight characters long and must not contain any of the characters * : ( ) < > or imbedded blanks. Trailing underscores (for example, FILE_) also are not allowed.

The filetype is optional. If it is omitted, a filetype of **blank** is assumed (**default**).

The filetype is stored in position **38-45** of the header record.

RECEIVE: the filetype can be specified. It is ignored, however.

Note that there is **no** checking for a matching filetype in the header record.

FILE=TS
defines the file transfer function as send/receive data files to/from the CICS TS queues.

options
are one or more of the following optional keyword parameters:

APPEND
SEND: allows to append a workstation file to the end of an existing TS queue.

The number of files that can be stored in one queue is **limited** only by the amount of temporary storage available and by the maximum number of records per queue (**32767**).

Individual files in the queue are separated by the header record.

RECEIVE: see "Common SEND/RECEIVE Command Options" on page 72.

LRECL=nnnnn
SEND: allows to specify a logical record length when workstation files are sent to a CICS TS queue. The workstation file is treated as a contiguous string of data which is broken up into nnnnn-byte segments. Each segment forms a logical record in the queue.

The maximum value for **nnnnn** is 32760.

LRECL overrides the CRLF option. CRLF is ignored if it is active by default or specified explicitly together with LRECL.

NOREPLACE
SEND: if NOREPLACE is active and the queue exists, a message is issued and the file transfer is terminated. NOREPLACE is the **default**.

PROG[RAM]=CFTRxxxx
SEND: specifies to activate a CICS user program after the file has been stored in the temporary storage queue. The program name must start with the characters CFTR and must be from six to eight characters long. It should not contain the character $.

RECEIVE: specifies to activate a CICS user program for loading data into the temporary storage queue before initiating the RECEIVE operation. The user program name must start with the characters CFTR and must be from six to eight characters long. It should not contain the character $.

QNAME=CFTRxxxx
SEND: specifies the name of the temporary storage queue into which the data from the PC is stored. The name must start with the characters CFTR and must be from six to eight characters long. It should **not** contain the character **$**.

If this option is omitted, a **default** queue name is used. The first four characters are always **CFTR**. The last four characters are the terminal ID. For example, if your terminal ID is D080, the default queue name is CFTRD080.

RECEIVE: specifies the name of the temporary storage queue which contains the file to be retrieved. The queue is not deleted by the download (receive) operation.

If this option is omitted and a user exit (PROGRAM=) is **not** specified, a default queue name is used The first four characters are always CFTR. The last four characters are the terminal identification. For example, if your terminal ID is D080, the default queue name is CFTRD080.

If the QNAME= option is omitted and a user exit (PROGRAM=) is specified, the **default** queue name may be overridden by a name supplied by the user program.

Temporary storage queue names, specified in the RECEIVE command or supplied by the user program, must start with the characters CFTR and must be from six to eight characters long. The character **$** should **not** be used.

REPLACE

SEND: specify REPLACE if an existing queue with the same name is to be deleted.

) comments

SEND: allows you to include descriptive information in the header record of the temporary storage queue. Any characters (including blanks) following the closing parenthesis are treated as comments. They are inserted at position 95 of the header record. Comments must not contain *Greater than* (>), *Less than* (<) symbols or any *colons* (:).

RECEIVE: same as for SEND except that the comments are passed to the user exit program as part of the header record in the user communication area (bytes 29 - 188).

Chapter 10. Data Conversion Support

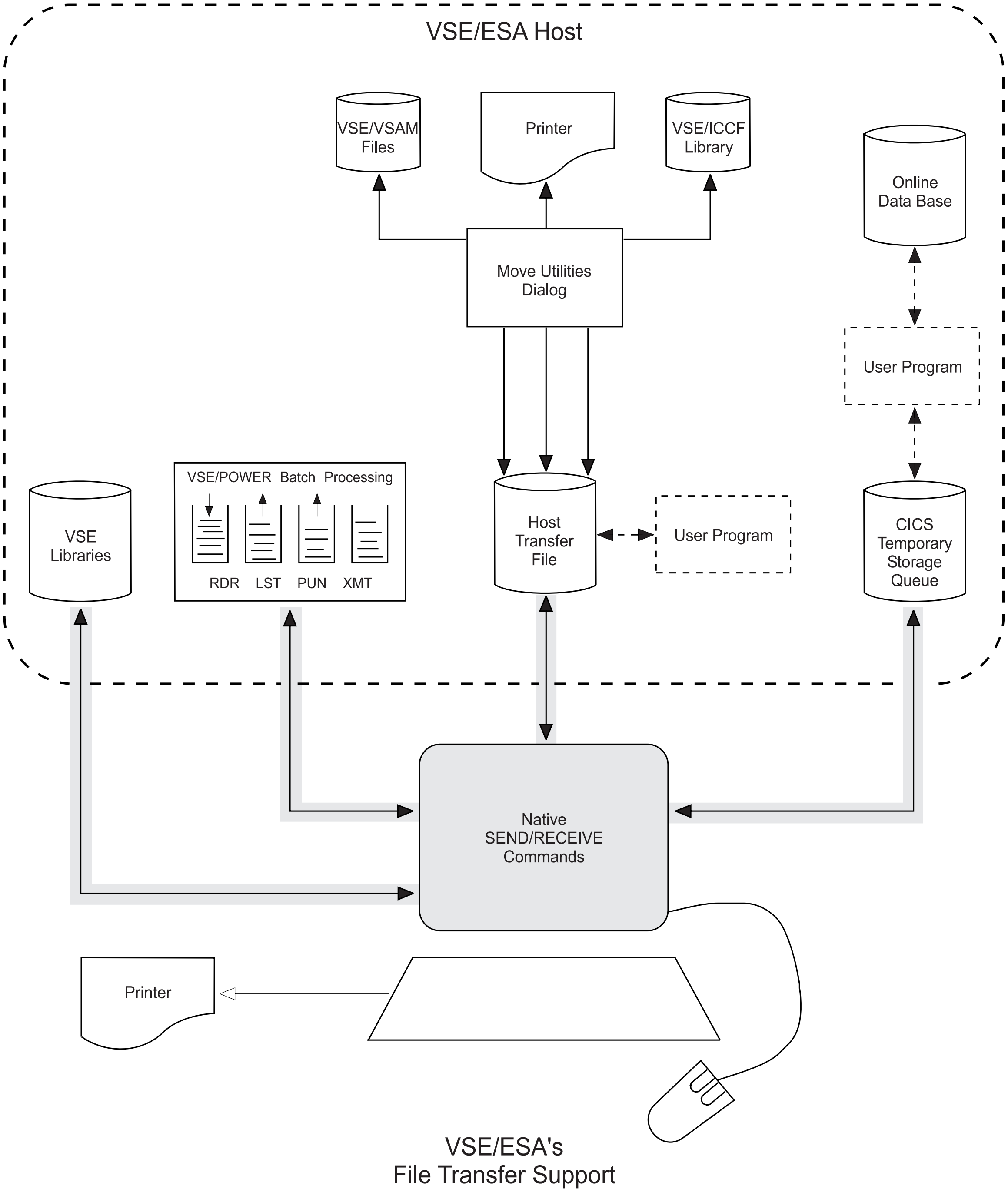


Figure 55. Data Paths of Data Conversion

Overview

In addition to ASCII/EBCDIC translation which applies to the entire file, VSE/ESA file transfer offers data conversion on record and field level. For example, if a host record contains a field with packed decimal numbers, these numbers can be converted on download to ASCII numbers. The format of the records and the type of conversion to be performed on each record field must be defined by the user in a Data Conversion Description File (DCDF).

A DCDF defines the record layout of a data file for both the workstation and the host environment. Each DCDF has:

- A *header record* that defines general conversion rules for the data file to be converted.
- One or more *field descriptor records* that define how the fields of a data file record are to be converted.

VSE/ESA provides dialogs for creating a DCDF at a workstation. Once a DCDF is defined and before you may use it the first time, it must be stored as a permanent member of the Host Transfer File. For this, you must specify either filetype blank or filetype DCDF. Specifying DCDF is strongly recommended.

Notes:

1. As shown below, VSE/ESA supports various data formats and field types for conversion. It is important to remember that all the records of a file to be converted must have the same layout.
2. If you wish, you can use a workstation or host environment editor to create and maintain a DCDF. When using an editor, you must do your own syntax and format checking.
3. When you use data conversion to convert files transferred to/from the Host Transfer File do **not** use a filetype of INWCMSGS for your data files in the HTF because the data conversion routines will create a file with filetype INWCMSGS to log conversion errors.

Using a Data Conversion Descriptor File

When issuing a SEND or RECEIVE command you can specify the name of a Data Conversion Descriptor File. During the data transfer individual fields within a record are converted as directed in the DCDF.

Messages issued during data conversion are stored into a file in the Host Transfer File. The file’s name is the same as of the transferred file and its file type is always **INWCMSGS**.

Creating and Maintaining a Data Conversion Descriptor File

When issuing a SEND or RECEIVE command you can specify the name of a **Data Conversion Descriptor File (DCDF)**. With a DCDF you can convert individual fields within a record during data transfer. VSE/ESA supports various data formats and field types for conversion.

Note that the records of a file you want to convert must all have the same layout.

When planning for conversion, you should also consider the information provided in the sections “Rules for Defining and Converting Fields” on page 221 and “Restrictions for File Conversion” on page 223.

Data Formats and Field Types Supported

VSE/ESA supports various data formats and field types for both the PC and the host environment.

Host Data Formats

For the host environment, the following data formats are supported:

- **FORM/370**

  This is the standard host data format characterized by the following:

  – Fixed length records.

  – Fixed length fields.

  – No record, field, or end-of-file delimiters are inserted during a SEND or RECEIVE operation.

  – **All** host field types are allowed.

- **SQLINPUT**

  This host data format is characterized by the following:

  – During a SEND operation, PC files are converted into the SQL input format. This is the data format expected by the optional program DB2 as input for a DB2 database.

  – The records need not be fixed length.

  – The fields within a record are separated by commas (,). In addition, character fields are enclosed in single quotes (′).

  Note: Character strings to be converted into SQLINPUT data format must not contain the character combination single quote/comma (′,).

  – Only the host field types **HEC** (EBCDIC character string) and **HEN** (EBCDIC numeric data) are supported.

  Note that a DCDF created for host data format SQLINPUT can only be used for upload (SEND) operations.

Host Field Types

Figure 56 on page 208 shows the field types and related lengths supported for the host environment.

HOST FIELD TYPE	LENGTH	DESCRIPTION
HBN	1—4	Binary Number
HDF	8	Hexadecimal Float Data (double precision)
HEC	1—255	EBCDIC Character String
HEN	1—255	EBCDIC Numeric Data
HHC	1—255 or *	Hexadecimal Character String (* = multiple flags in one host field)
HPD	1—16	Packed Decimal Data
HSF	4	Hexadecimal Float Data (single precision)
HVC	3—255	EBCDIC Varying—Length Character String
HZD	1—16	Zoned Decimal Data

Figure 56. Host Field Types

A detailed description of the characteristics of each host field type is given under "Host Field Types" on page 215.

PC Data Formats

For the PC environment, the following data formats are supported:

- ASCII-TX – ASCII Text File
- DOS-RAN – DOS Random File
- BAS-SEQ – BASIC Sequential File
- BAS-RAN – BASIC Random File
- DIF – File in DIF Format
- PASC-ASC – PASCAL ASCII Sequential File
- FORT-FOR – FORTRAN Formatted File
- COB-LSEQ – COBOL Line Sequential File
- User defined data format

Consult your PC documentation for further details on these data formats.

PC Field Types

Figure 57 on page 209 shows the field types and related lengths supported for the PC environment. A more detailed description of the characteristics of each field type is given under "PC Field Types" on page 217.

Figure 58 on page 210 shows which PC field types can be specified for which PC data format.

PC FIELD TYPE	LENGTH	DESCRIPTION
PAC	1-255	ASCII character string.
PAN	1—255	ASCII numeric data. Up to 16 significant decimal digits in character format with the optional characters + - . E and D are allowed. Leading and trailing blanks may be used.
PBI	2	BASIC integer. Represents a 2 byte signed PC binary number.
PBN	1—4	Signed PC binary number.
PCI	2	COBOL integer. Represents a 2 byte signed EBCDIC binary number.
PDF	8	BASIC floating point format (double precision).
PHC	1—255	Hexadecimal character string.
PI2	1—255	Integer represented by decimal digits in character format. Its value is limited by -32768 and +32767. The sign of - or + is optional. Leading and trailing blanks are allowed.
PI4	1—255	Integer represented by decimal digits in character format. Its value is limited by -2,147,483,648 and +2,147,483,647. The sign of - or + is optional. Leading and trailing blanks are allowed.
PPD	1—16	Packed decimal data. The sign depends on the data format.
PPW	1—255	Integer represented by decimal digits in character format. Its value is limited by 0 and 65535. Leading and trailing blanks are allowed.
PSF	4	BASIC floating point format (single precision).
PZD	1—16	Zoned decimal data. Low order byte depends on data format.

Figure 57. PC Field Types

DATA FORMAT:	FIELD TYPES ALLOWED:
ASCII Text	PAC, PAN
BASIC Sequential	PAC, PAN
BASIC Random	PAC, PBI, PSF, PDF
COBOL Line Sequential	PAC, PCI, PPD, PZD
DIF	PAC, PAN
DOS Random	PAC, PBN, PHC, PPD, PZD
FORTRAN Formatted	PAC, PAN, PI2, PI4
PASCAL ASCII	PAC, PAN, PI2, PI4, PPW

Figure 58. PC Data Format and Related Field Types

Conversion Overview

Figure 59 on page 211 gives an overview on which PC field type can be converted into which host field type and vice versa.

In the table, the PC field types are grouped as follows:

- Binary (lines 1 through 3)
- Decimal (lines 4 and 5)
- Formatted decimal (lines 6 through 9)
- Floating point (lines A and B)
- String (lines C and D)

		Host Field Type:								
	PC Field Type	HBN	HZD	HPD	HSF	HDF	HEN	HEC	HHC	HVC
1	PBN	yes	yes	yes	yes	yes	yes	no	no	no
2	PBI	yes	yes	yes	yes	yes	yes	no	no	no
3	PCI	yes	yes	yes	yes	yes	yes	no	no	no
4	PZD	yes	yes	yes	yes	yes	yes	no	no	no
5	PPD	yes	yes	yes	yes	yes	yes	no	no	no
6	PAN	yes	yes	yes	yes	yes	yes	no	Note1	no
7	PI2	yes	yes	yes	yes	yes	yes	no	no	no
8	PI4	yes	yes	yes	yes	yes	yes	no	no	no
9	PPW	yes	yes	yes	yes	yes	yes	no	no	no
A	PSF	yes	yes	yes	yes	yes	yes	no	no	no
B	PDF	yes	yes	yes	yes	yes	yes	no	no	no
C	PAC	no	no	no	no	no	no	yes	Note2	yes
D	PHC	no	no	no	no	no	no	no	yes	no

Figure 59. Possible Field Type Conversions

Notes:

1. No real conversion is performed. A bit in the host hexadecimal field (addressed via the ″displacement″ specification in the DCDF) is mapped into a PC ASCII numeric field and vice versa. For a RECEIVE operation, if the bit is on, the PC field is set to -1. If the bit is off, the PC field is set to 0.
2. No real conversion is performed. A bit in the host hexadecimal field (addressed via the ″displacement″ specification in the DCDF) is mapped into a PC ASCII character field and vice versa. For a RECEIVE operation, if the bit is on, the PC field is set to TRUE, if the bit is off, the PC field is set to FALSE.

Using an Editor to Create a DCDF

You may want to create a DCDF not with the PC dialogs provided, but with a PC or host editor. The file you create must start with a 80-byte header record followed by one or more 80-byte field descriptor records. “Data Conversion Descriptor File Example” on page 212 shows and describes a sample DCDF. Use the DCDF shown in Figure 60 on page 212 as a starting point if you are using an editor for creating a DCDF.

Obviously, when using this technique there is no format and syntax checking performed at DCDF creation time. Only at conversion time does the system recognize any errors. An error causes the message ″**DCDF in Error - No Conversion Possible**″. If you get such a message, you have two ways to correct the error. Either you try to find the error at your desk, or you use the PC dialog. With the PC dialog you can edit the DCDF and the system will display messages about the invalid contents of the DCDF.

For conversion, you must load the DCDF into the Host Transfer File. Use the SEND command or PC dialog if you used a PC editor, or use the *Move Utilities* dialog if you used an editor of your host environment.

Data Conversion Descriptor File Example

Figure 60 shows an example of a DCDF. *This example is intended to help you design your own DCDF when using an editor instead of the dialogs.*

The first line is the header record. All the other lines are field records defining the fields within a record. Columns 1 to 23 of the field records define the fields in the PC environment, columns 24 to 48 in the host environment.

To convert data as specified in this DCDF, you would simply specify the DCDF name for either the SEND or RECEIVE operation. The name to be used is the name assigned to the DCDF when loading it into the Host Transfer File.

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
----+----|----+----|----+----|----+----|----+----|----+----|----+----|----+-----

DCDH PASC-ASC 012     N Y N PACPANPI2PI4PPW                FORM/370 SAMPL-DCDF *
DCDF PAC   8           HEC   8                    Host fld as long as PC fld    *
DCDF PAC  28           HEC  18                    Host fld shorter than PC fld  *
DCDF PAC   3           HEC   7                    Host fld longer than PC fld   *
DCDF PAN  24           HPD  10     2              Host fld with scale 2         *
DCDF PI4  11           HZD   4     3              Host fld with scale 3         *
DCDF PI4  11           HBN   4               Y    Host fld signed bin number    *
DCDF PI2   6           HBN   2               N    Host fld unsigned bin number  *
DCDF PI2   6           HEN  12                 E Host fld with 'E' as exp.-id. *
DCDF PAN  23         E HDF   8                    Host fld is long float.point  *
DCDF PAN  13         E HSF   4                    Host fld is short float.point *
DCDF PAC   4           HHC   4           0        Address 1st bit in host fld   *
DCDF PAC   3           HHC   4          11        Address 12th bit in host fld  *

----+----|----+----|----+----|----+----|----+----|----+----|----+----|----+-----
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

Figure 60. Example of a DCDF

Following is a detailed description of the layout of the DCDF shown in Figure 60.

Layout of Header Record:

Columns
Contents

1 - 4 Header record identifier: Constant 'DCDH'.
5 - 5 Blank or comma.
6 - 13 PC data format.
14 - 14 Blank or comma.
15 - 17 Number of fields in actual data record (decimal format, right adjusted).
18 - 18 Blank or comma.
19 - 19 Blank.
20 - 20 Blank or comma.

21 - 21 Field separator for PC file: ′,′ = Comma as separator. ′ ′ = Blank as separator. ′N′ = No separator.
22 - 22 Blank or comma.
23 - 23 String delimiter for PC file: ′Y′ = Double quote. ′N′ = No string delimiter.
24 - 24 Blank or comma.
25 - 25 Record delimiter for PC file: ′Y′ = Record delimiter (CRLF). ′N′ = No record delimiter.
26 - 26 Blank or comma.
27 - 27 End of file indicator for PC file. ′Y′ = X′1A′ is EOF indicator. ′N′ = No EOF indicator.
28 - 28 Blank or comma.
29 - 58 String of all PC field types used in this DCDF. Each field type is three bytes long.
59 - 59 Blank or comma.
60 - 67 Host data format (FORM/370 or SQLINPUT).
68 - 68 Blank or comma.
69 - 78 Comments.
79 - 79 Blank or comma.
80 - 80 Asterisk (*).

Refer to Figure 61 for the header record parameter values to be entered when using an editor.

PC DATA FORMAT	FIELD SEPARATOR	STRING DELIMITER	RECORD DELIMITER	FILE DELIMITER	FIELD TYPES SUPPORTED
ASCII Text	N	N	Y	Y	PAC, PAN
BASIC Sequential	,	Y	Y	Y	PAC, PAN
BASIC Random	N	N	N	N	PAC, PBI PSF, PDF
COBOL Line Sequential	N	N	Y	Y	PAC, PCI PPD, PZD
DIF		Y	Y	N	PAC, PAN
DOS Random	N	N	N	N	PAC, PBN, PHC PPD, PZD
FORTRAN Formatted	N	N	Y	N	PAC, PAN, PI2, PI4
PASCAL ASCII	blank	N	Y	N	PAC, PAN, PI2 PI4, PPW

Figure 61. Header Record Parameter Values when Using an Editor

Layout of a Field Record:

Columns
Contents

1 - 4 Field record identifier: Constant ’DCDF’.

5 - 5 Blank or comma.

6 - 8 Type of PC field.

9 - 9 Blank or comma.

10 - 12 Length of PC field. Mandatory for all PC data formats which do not have ’comma’ as a field separator. Must be blank for DIF and BAS-SEQ data formats (decimal format, right adjusted).

13 - 13 Blank or comma.

14 - 14 Blank.

15 - 15 Blank or Comma.

16 - 18 Optionally, scale of PC field for PC field types PZD and PPD. Must be blank for all other PC field types (decimal format, right adjusted).

19 - 19 Blank or comma.

20 - 20 Blank.

21 - 21 Blank or comma.

Columns
Contents

22 - 22 Optionally, exponent identifier for PAN field types: ’ ’ = No exponent identifier allowed if target field type is PAN. ’E’ = Use ’E’ in target field. ’D’ = Use ’D’ in target field. Must be blank for all other PC field types.

23 - 23 Blank or comma.

24 - 26 Type of host field.

27 - 27 Blank or comma.

28 - 30 Length of host field. Optional if the host data format is SQLINPUT. Mandatory in all other cases (decimal format, right adjusted).

31 - 31 Blank or comma.

32 - 32 Blank.

33 - 33 Blank or comma.

34 - 36 Optionally, scale of host field for host field types HZD and HPD. Must be blank for all other host field types (decimal format, right adjusted).

37 - 37 Blank or comma.

38 - 38 Blank.

39 - 39 Blank or comma.

40 - 42 Displacement of boolean flag bit in host field. Must be specified, if a boolean PC field is to be converted into a host flag bit or vice versa. The host field type must be HHC. Must be blank if the host field type is not HHC or the PC field type is not boolean.

43 - 43 Blank or comma.

44 - 44 Blank.

45 - 45 Blank or comma.

46 - 46 Indicates for a host field of type HBN whether it is signed or not: ’Y’ = Signed binary number. ’N’ = Unsigned binary number. Must be blank if the host field type is not HBN.

47 - 47 Blank or comma.

48 - 48 Optionally, exponent identifier for field type HEN. ’ ’ = No exponent identifier allowed if target field type is HEN. ’E’ = Use ’E’ in target field. ’D’ = Use ’D’ in target field. Must be blank for all other PC field types.

49 - 49 Blank or Comma.

50 - 78 Optionally, field name or field description.

79 - 79 Blank or comma.

80 - 80 Asterisk (*).

Considerations for Loading a DCDF into the Host Transfer File

Before sending or receiving data that is to be converted, the corresponding Data Conversion Descriptor File (DCDF) must be present in the Host Transfer File.

If you **create a DCDF at your PC**, you can load a DCDF into the Host Transfer File either by entering a SEND command or by using the *Send File: Data Conversion Information* dialog. In both cases, specify ASCII as conversion parameter. If a DCDF is sent to the Host Transfer File, the REPLACE option is in effect. That is, an existing DCDF of the same name is always replaced in the Host Transfer File.

However, there is no need that you load the DCDF into the Host Transfer File in a separate step. You can always specify a DCDF for a normal SEND or RECEIVE (command or dialog) operation. The DCDF is then loaded into the Host Transfer File before the actual data transfer takes place.

If you **create a DCDF with an editor of your host environment**, you can load the DCDF with the *Personal Computer Move Utilities* dialog into the Host Transfer File. Select the following:
4 (Move ICCF members to Host Transfer File)

Field Types Description

Host Field Types

Packed Decimal Data (HPD)

Length:
1 to 16 bytes with a maximum of 16 significant numeric digits.

Format:
Standard FORM/370 packed decimal data format.

Sign: Positive sign digits are 1010, 1100, 1110, and 1111; negative sign digits are 1011 and 1101.

A scale factor can be specified. The value may range from -99 to +99. Positive scale means, that the assumed decimal point is to the left of the rightmost number digit. Negative scale means, that it is to the right of the rightmost digit. For example:

12345 with scale factor +2 becomes 123.45.
12345 with scale factor -3 becomes 12345000.

Zoned Decimal Data (HZD)

Length:
1 to 16 bytes.

Format:
Standard FORM/370 zoned decimal data format. Zone digits are 1111.

Sign: Positive sign digits are 1010, 1100, 1110, and 1111. Negative sign digits are 1011 and 1101.

A scale factor can be specified. The value may range from -99 to +99.

Hexadecimal Float Data (HSF, HDF)

Length:
4 bytes for short float (HSF) or 8 bytes for long float (HDF).

Format:
Standard FORM/370 hexadecimal floating point data format. In byte 0, bit 0 is the sign bit, bits 1 to 7 are the characteristic which represents a signed exponent. It is obtained by adding 64 to the exponent value. Bytes 1 to 3 (HSF) or bytes 1 to 7 (HDF) contain the mantissa of 6 (HSF) or 14 (HDF) hexadecimal digits.

Sign: Byte 0, bit 0 off indicates a positive, bit 0 on indicates a negative value.

The following limits must be observed when converting PC decimal or ASCII numeric data into hexadecimal floating point format:

The maximum value supported is 6.999999999999999E+75.
The minimum value supported is 6.000000000000000E-79.

Since a precise representation of floating point numbers in decimal notation is not possible, rounding occurs frequently. Rounding is not indicated by a message.

Binary Data (HBN)

Length:
1 to 4 bytes.

Format:
Standard FORM/370 format. Signed values: 7, 15, 23, or 31 binary digits preceded by sign bit. Unsigned values: 8, 16, 24, or 32 binary digits.

Sign: Byte 0, bit 0 off indicates a positive, bit 0 on indicates a negative value.

Whether the field is signed or unsigned is to be specified in the field descriptor record of the DCDF. Assignment of a negative value to an unsigned field results in the assignment of the absolute value and a warning message.

EBCDIC Numeric Data (HEN)

Length:
1 to 255 bytes with a maximum of 16 significant numeric mantissa digits. This does not include leading and trailing zeroes. Mantissa stands for the mantissa of a floating point number or otherwise for the number itself.

Format:
Numeric data in EBCDIC character representation. Integer: Optionally signed numeric characters. Fixed Point: Optionally signed numeric characters interspersed, preceded, or followed by a decimal point. Floating Point: Integer or fixed point mantissa followed by an exponent identifier character ’E’ or ’D’ followed by an optionally signed integer for the exponent value. The accepted input value range is 1E-127 <= ABS(X) < 1E+144. A violation leads to an exponent underflow or overflow error message.

Sign: Characters ’+’ or ’-’ preceding the first numeric character, an omitted sign is interpreted as positive. For output, positive or zero values will not get a sign character (and also no preceding blank), negative values will be preceded by a ’-’ character, this is also valid for exponent values.

If more than 16 significant mantissa digits are presented as input, the excess digits will be handled like trailing zeroes and the resulting value will be used. Two different error messages are issued depending on the expected impact on data usability. If the excess digits are fraction digits (following a decimal point) or pertain to a floating point mantissa (followed by an exponent identifier), a truncation warning is given. Otherwise, an overflow error message is issued. Leading and trailing blanks are allowed.

The output format can be controlled by the exponent identifier specified in the field descriptor record of the DCDF. A blank identifier enforces an integer or fixed point representation. If even the integer part of the value does not fit into the target field, an error is indicated and asterisks are written. An identifier of 'E' or 'D' results in integer, fixed point, or floating point output depending on the minimum length required.

Character String Data (HEC, HHC)

Length:
1 to 255 bytes.

Format:
The characters may be either EBCDIC characters (HEC) or hexadecimal characters (HHC) covering the whole character range from X'00' to X'FF'. The difference is that during transfer hexadecimal characters remain unchanged, whereas EBCDIC characters are converted.

Varying-Length Character String Data (HVC)

Length:
3 to 255 bytes.

Format:
The characters are EBCDIC characters covering the whole character range from X'00' to X'FF'. The first two bytes of a HVC field contain the field's data length in binary notation. These two bytes are followed by the actual data. If the data length is smaller than the field length minus 2, unused bytes occur. The highest value that can be accommodated for data length is 253. If a higher value is encountered for download (RECEIVE), the transfer is terminated and an error message is issued. An error message is also issued, if the data length specified is greater than the field length minus 2.

Boolean Data (HHC)

Boolean data is characterized by the host field type HHC in conjunction with one of the following PC field types: PAC or PAN.

Length:
1 to 32 bytes.

Format:
The boolean value is represented by a single flag bit within the specified hexadecimal character string. Its offset must be defined in the field descriptor record of the DCDF. Flag bit on means TRUE, flag bit off means FALSE.

PC Field Types

Packed Decimal Data (PPD)

Length:
1 to 16 bytes with a maximum of 16 significant numeric digits.

Format:
DOS Random and COBOL Line Sequential. The COBOL Line Sequential

format is identical to the host standard format FORM/370. This is also true for the DOS Random PC data format, except for the sign digit.

Sign: The DOS Random and a user-defined PC data format have a positive sign digit of 0011 and a negative sign digit of 1011.

The COBOL Line Sequential format has a positive sign digit of 1100 and a negative sign digit of 1101. For input, the sign is treated in the same way as for the FORM/370 format. For output, the standard sign representations are used.

A scale factor can be specified. The value may range from -99 to +99.

Zoned Decimal Data (PZD)

Length:
1 to 16 bytes.

Format:
DOS Random format and COBOL Line Sequential format. Both formats have zone digits of 0011 and decimal numeric digits of 0000 through 1001. The sign format is different.

Sign: The DOS Random and a user-defined PC data format have a positive sign digit of 0011 and a negative sign digit of 1011.

COBOL Line Sequential uses the values shown in Figure 62 for numeric digits with sign zone (PICTURE with trailing sign). Depending on the rightmost digit and the sign of the number, the rightmost byte of the number will contain one of the following values:

	Rightmost Digit									
Sign	0	1	2	3	4	5	6	7	8	9
+	X'7B' }	X'41' A	X'42' B	X'43' C	X'44' D	X'45' E	X'46' F	X'47' G	X'48' H	X'49' I
_	X'7D' }	X'4A' J	X'4B' K	X'4C' L	X'4D' M	X'4E' N	X'4F' O	X'50' P	X'51' Q	X'52' R

Figure 62. Byte Values for COBOL Line Sequential Format and PZD Field Type

For unsigned numeric digits (PICTURE without sign specification), the following values are used:

0 to 9 = X'30' to X'39'

This is identical with the FORM/370 host format after ASCII to EBCDIC translation. Unsigned numbers are only supported for input. The output has always a trailing sign.

A scale factor can be specified. The value may range from -99 to +99.

Binary Float Data (PDF, PSF)

Length:
4 bytes for short float or 8 bytes for long float.

Format:
Binary floating point. Byte 0 contains the exponent in 128 excess notation. Byte 1 contains the sign and the binary digits 1 to 7. Bytes 2 to 3 or bytes 2 to 7 contain the digits 8 to 23 or 8 to 55.

The byte sequence is reversed. That is, the bytes are stored from left to right in the order 3 to 0 or 7 to 0.

Sign: The sign bit is put into the position of the first mantissa bit. It is not stored in the target field since it is always 1 because only normalized mantissas are allowed. Byte 1, bit 0 off indicates a positive, bit 0 on indicates a negative value.

Due to the sign representation, only normalized mantissas are allowed. Zero must be represented by all zeroes. A zero mantissa with non-zero byte 0 will be interpreted as +0.5 * 2**n.

The maximum value is 1.70141183460469 E+38.
The minimum value is 1.46936811369017 E-39.

Since a precise representation of floating point numbers in decimal notation is not possible, rounding occurs frequently. Rounding is not indicated by a message.

Binary Data (PBN, PBI)

Length:
1 to 4 bytes.

Format:
Signed 7, 15, 23, or 31 binary digits preceded by a sign bit. The byte sequence is reversed. That is, the bytes are stored from left to right in the order 3 to 0, 2 to 0, 1 to 0, or 0.

Sign: Byte 0, bit 0 off indicates a positive, bit 0 on indicates a negative value.

The PC field type BI is identical to type BN. The length is 2. The conversion routine treats these data as type BN.

Binary Data (PCI)

Length:
2 bytes.

Format:
The format is identical to a signed host FORM/370 binary halfword.

Sign: Byte 0, bit 0 off indicates a positive value, bit 0 on indicates a negative value.

ASCII Numeric Data (PAN, PI2, PI4, PPW)

Length:
1 to 255 bytes with a maximum of 16 significant numeric mantissa digits. This does not include leading and trailing zeroes. Mantissa stands for the mantissa of a floating point number or for the number itself.

Format:
Numeric data in ASCII character representation.

Integer: Optionally signed numeric characters. Fixed Point: Optionally signed numeric characters interspersed, preceded, or followed by a decimal point. Floating Point: Integer or fixed point mantissa followed by an exponent identifier character ’E’ or ’D’ followed by an optionally signed integer for the exponent value. The accepted input value range is 1E-127 <= ABS(X) < 1E+144. A violation leads to exponent underflow or overflow error messages.

Sign: Characters ’+’ or ’-’ preceding the first numeric character. An omitted sign is interpreted as positive. For output, positive or zero values will not get a sign character (and also no preceding blank), negative values will be preceded by a ’-’ character. This is also true for exponent values.

If more than 16 significant mantissa digits are presented as input, the excess digits will be handled like trailing zeroes and the resulting value will be used. Two different error messages are issued depending on the expected impact on data usability. If the excess digits are fraction digits (following a decimal point) or pertain to a floating point mantissa (followed by an exponent identifier), then a truncation warning is given. Otherwise, an overflow error is indicated. Interspersed blanks are only allowed preceding or following the exponent identifier character.

- Field type PAN includes types PI2, PI4, and PPW.
- Field type PI2 is restricted to the value range of -32768 to +32767.
- Field type PI4 is restricted to the value range of -2147483648 to +2147483647.
- Field type PPW is restricted to the value range of 0 to +65535 for output.

  For input, all types will be handled as PAN. Output of a negative value to a PPW field results in the assignment of the absolute value and a warning message.

  The output format can be controlled by the exponent identifier specified in the field descriptor record of the DCDF. A blank identifier enforces an integer or fixed point representation. If even the integer part of the value does not fit into the target field, an error is indicated and asterisks are written. An identifier of ’E’ or ’D’ results in integer, fixed point, or floating point output depending on the minimum length required.

  The field types PI2, PI4, and PPW should be specified with a blank exponent identifier. This is to ensure that even if the target fields are too small, they receive the full integer value.

Character String Data (PAC, PHC)

Length:
1 to 255 bytes.

Format:
The characters may be either ASCII characters (PAC) or hexadecimal characters (PHC) covering the whole character range from X’00’ to X’FF’. The difference is that during transfer hexadecimal characters remain unchanged, while ASCII characters are converted.

Boolean Data (PAC, PAN)

Boolean data is characterized by the field types PAC or PAN in conjunction with the FORM/370 host field type HHC.

Length:
Supported field length 1 to 255 bytes.

Format:
The boolean value is represented either by an ASCII character string (PAC) or by an ASCII numeric value (PAN).

PAC: TRUE may be written (.)T(RUE.), FALSE may be written (.)F(ALSE.). Characters within the parenthesis are optional. For input, the string may have leading and/or trailing blanks. For output, no dots are used and the value is aligned to the right. Necessary truncations are performed on the right without warning message.

PAN: TRUE may be specified as value -1. FALSE may be specified as value 0.

Rules for Defining and Converting Fields

The following rules apply for defining and converting fields:

- If the host data format is **FORM/370**, the field type must be specified in the DCDF for every field of the record.
- If the host data format is **SQLINPUT**, a length specification for the host fields is not required. The conversion routines determine the size of the fields based on the contents of the fields.
- All host files for download (RECEIVE) must have fixed-length records of the same format. All host files that have been uploaded (SEND) will contain records of the same format.

  The host record length must be equal to the sum of all host field lengths as defined in the DCDF (if lengths are defined in the DCDF).
- The following PC data formats require field length specifications in the DCDF:
  ACSII-TX
  DOS-RAN
  BAS-RAN
  PASC-ASC
  FORT-FOR
  COB-LSEQ
- For the following PC data formats, no field lengths should be defined in the DCDF:
  BAS-SEQ
  DIF

  For these PC data formats, the field length is determined at conversion time:
  - For upload (SEND), the field separators or other specific data format information is used to determine the length of fields in a PC file.
  - For download (RECEIVE), the length of the PC fields will be determined by the length of the fields in the host file.
- Padding rules:

  If the converted source field is shorter than the target field, the target field will be padded as shown in Figure 63 on page 222.

FIELD TYPE	PADDING CHARACTER	PADDED
PBN,PBI	X'00' or X'FF' (depending on sign)	on the right
HBN,PCI	X'00' or X'FF' (depending on sign)	on the left
HZD	X'F0'	on the left
PZD	X'30'	on the left
PPD,HPD	X'00'	on the left
PAC,	X'20'	on the right
HEC	X'40'	on the right
HEN	X'40'	on the left
PHC,HHC	X'00'	on the right
PAN,PI2,PI4,PPW	X'30' (for ASCII text file PAN) X'20' (for all other file types)	on the left

Figure 63. Padding Rules

If padding occurs, no warning message will be issued.

- Truncation rules:

  If the converted source field is longer than the target field, the following action will be performed:

  – For numeric values:

  If the field does not fit after removal of all leading zeros, the highest possible positive value or the lowest possible negative value will be inserted in the target field. An error message will be issued.

  – For ASCII,EBCDIC, and hexadecimal strings:

  The converted string will be truncated on the right. If at least one of the truncated characters was not

  equal to X’20’ for ASCII, or
  equal to X’40’ for EBCDIC, or
  equal to X’00’ for HEX,

  an error message will be issued.

- Incompatible data types:

  For some data formats (DIF, for example) the information that determines the field type is part of the data file. In this case, only at run time can it be determined whether the field type of the source field is compatible with the field type of the target field. If it is incompatible, the file transfer is terminated with an error message.

- Data not found in PC file:

  A field is defined for the host file but does not exist in the PC file. For example, a BASIC Sequential PC file has only 5 fields in a record, but the DCDF describes 6 host fields. If this happens, the transfer is terminated with an error message.

- Data in PC file is lost:

  A field of a PC file is not defined for the host file. For example, a BASIC Sequential PC file has 6 fields in a record. But the DCDF describes only 5 host fields. The field(s) that exist in the PC file but which are not defined for the host file are lost. If this happens, the transfer is terminated with an error message.

- Data incompatible:

A field of the source file is not compatible with its DCDF description. For example, a non-numeric character byte (X'AF') is found in a field that is defined as packed decimal. The file transfer is terminated with an error message.

Restrictions for File Conversion

The following restrictions apply:

- Only files in which all records have the same layout can be converted. This means that the field descriptions of the DCDF will be applied to all records.
- Only host files with fixed-length records can be converted or are the result of a conversion. An exception is a file of the host data format SQLINPUT which can have varying-length records.

  The length of a host record must be equal to the sum of all host field lengths as defined in the DCDF. Again, a SQLINPUT file is an exception because of varying-length records.
- Only PC files with fixed-length records can be converted or are the result of a conversion. An exception are the PC data formats BAS-SEQ and DIF. The length of the PC record must be equal to the sum of all PC field lengths as defined in the DCDF. If a blank is specified as field separator, the number of fields minus 1 must be added to get the correct length of the PC record.
- When converting a field that can contain a very large number into a field that can only contain a relatively small number, many error messages may be generated because of overflow situations. For example, when converting a host floating point number into a PC zoned decimal number.
- Not all PC data formats for PC supported languages like BASIC, FORTRAN, PASCAL and COBOL are supported for conversion. The data formats supported are those most frequently used. The following restrictions apply:
  - For COBOL, only files with line-sequential organization are supported. For the PICTURE clause, only the numeric form is supported. Zoned decimal fields in these files must have their sign in the low-order byte. For example, if the rightmost digit of a negative number is 5, then the low-order byte has to be X'4D'.

    For that reason, PC COBOL programs should not use the usage clause with COMP or DISPLAY, together with the sign clause with LEADING or SEPARATE, and the leading S in PICTURE. If usage is COMP|DISPLAY, the PC field length is equal to the number of the 9's in PICTURE. If the usage is COMP-0, the PC field length is equal to 2. If the usage is COMP-3, the PC field length is equal to 1 for one 9 in the picture, 2 for two or three 9's in the picture, 3 for four or five 9's in the picture, and so on. The number of digits allowed in COMP and COMP-3 type fields is limited to 16.
  - For FORTRAN, only formatted files are supported. When writing or reading character type fields, the field length should also be used in the FORMAT statement for write or read. The reason is that FORTRAN uses blanks for padding the beginning of fields. The conversion support, however, truncates blanks at the end (if the source field does not fit into the target field).
  - For PASCAL, ASCII files of type TEXT in sequential mode are supported.

    PASCAL ASCII files must have fields with an explicitly specified length (in the DCDF), separated by one blank. That is, the length of a specific field must be the same for all records. This is required since the conversion program uses the length specifications from the DCDF to distinguish the different fields. The separating blanks are required to allow PASCAL programs a correct decoding when downloading (RECEIVE) a file.

The blank character separating the fields *must not* be included in the field length specification in the DCDF. Blanks can be written by PASCAL programs either explicitly or as part of an arithmetic data field. When reading files in PASCAL, the separating blank has to be read explicitly if it is placed in front of a character string. This is true for field type PAC. All other field types (PAN, PI2, PI4, PPW) search for the beginning of the variable representation and will skip the blank.

Please note that in PASCAL real numbers fill up the field by appending zeroes to the mantissa. This can be restricted by an explicit specification of the number of decimal positions in the DCDF.

The conversion program puts boolean variables (TRUE or FALSE) adjusted to the right into the specified field. This is done to get a defined position for the read pointer in PASCAL programs which points to the position directly after TRUE or FALSE.

– The records of BASIC Sequential files for upload (SEND) must be written with the WRITE command instead of the PRINT command. The WRITE command encloses ASCII character strings in double quotes. The conversion routine checks for these double quotes.

– When DIF files are uploaded (SEND), a host record is built out of every tuple. For download (RECEIVE), a tuple is built out of every host record. Therefore, the number of fields in every tuple has to be the same for all tuples. Even the n-th field in every tuple has to be of the same type for every tuple (either ASCII-character or ASCII-numeric). If one of these conditions is not fulfilled, messages are issued and the file transfer may even terminate.

The conversion routines expect that character strings in DIF files are always enclosed in double quotes ("). Keywords like ERROR, TRUE, or FALSE are not supported without double quotes.

Chapter 11. Using the Workstation File Transfer Trace

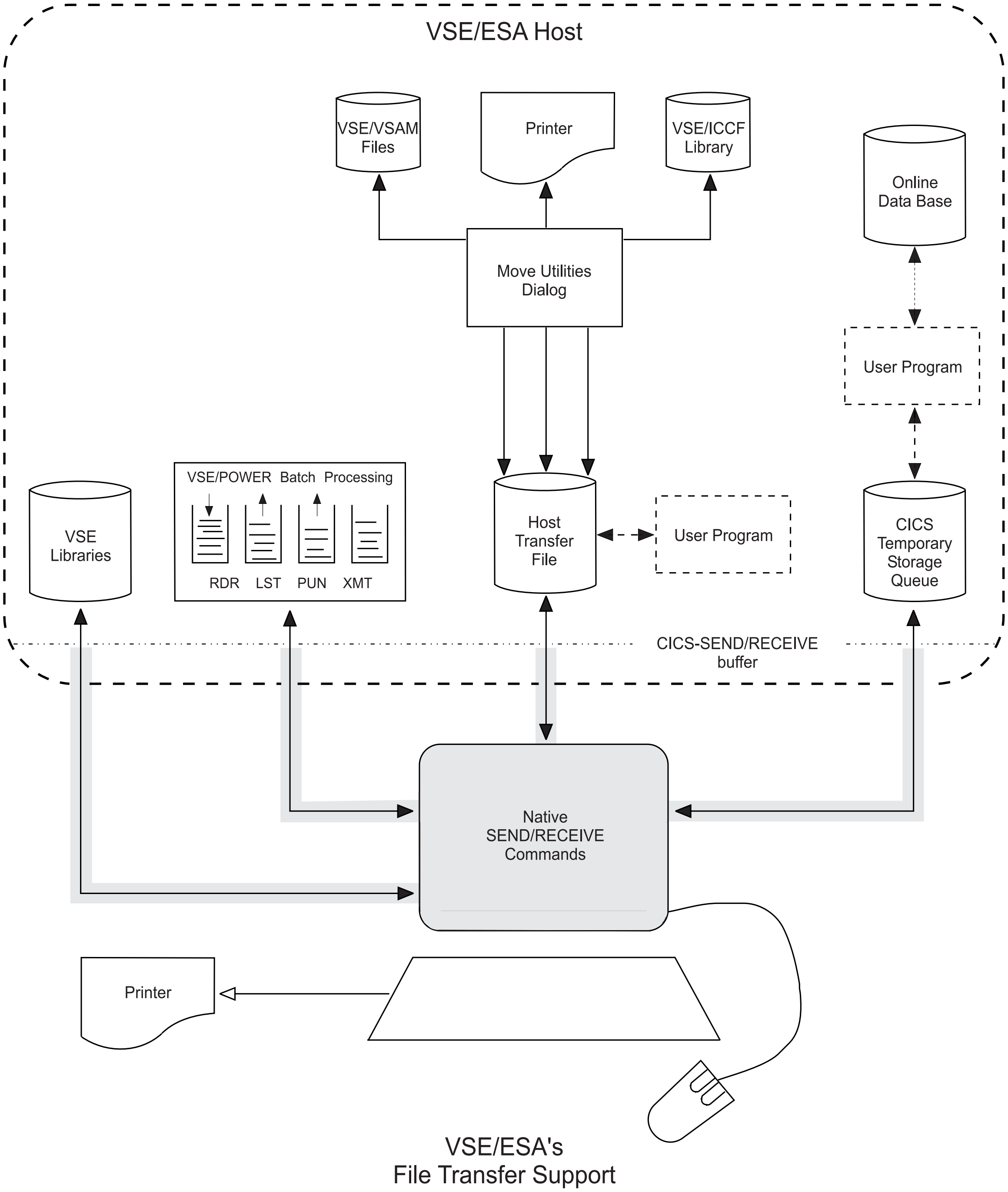


Figure 64. Data Paths of the File Transfer Trace

Overview

When exchanging data between a workstation and a VSE/ESA host system, you can use a trace that is available with VSE/ESA's Workstation File Transfer Support.

You activate the trace via a SEND or a RECEIVE command.

During workstation file transfer a number of problems may occur, such as transmission of incorrect data or, more frequently, a hangup of either the host or the workstation session. Such a hangup is, in many cases, due to an incorrect setup. For example, the Extended Data Stream (EXTDS) feature was not defined for a workstation in DFT mode (Distributed Function Terminal attachment). The user, however, does not get any indication of what went wrong. For such problems, the file transfer trace is provided. It enables a user to easily obtain a trace of the transmitted data and to check for correct setup.

Notes:

1. The trace will not help a user to solve problems like incorrect data transmission or protocol errors. Such errors require support from IBM for trace output analysis.
2. Be aware that this trace covers only the transmission of data between the workstation and the CICS SEND/RECEIVE buffer. It does not show the data paths between the CICS SEND/RECEIVE buffer and the host resource such as the HTF or the VSE libraries.
3. This trace requires the use of the CICS Report Controller Feature. However, you can use the option TRACE=Q without the CICS/RCF.

Starting the Trace

You start the workstation file transfer trace at a workstation by issuing the SEND or RECEIVE command with one of the following options:

TRACE=Q
stores the trace in CICS Temporary Storage (TS)

TRACE=LST
sends the trace to the VSE/POWER list (LST) queue

TRACE
writes the trace to both CICS TS and the VSE/POWER LST queue

TRACECUT
same as TRACE option but for CUT mode only

The TRACE, TRACE=Q, and TRACE=LST options can be used with DFT or CUT mode as long as the controller to which the workstation is attached supports the Extended Data Stream (EXTDS) feature.

The TRACECUT option is reserved for older devices which **do not** support the EXTDS feature.

Note: Use the TRACECUT option **only** if you are absolutely sure that your workstation runs in CUT mode (Control Unit Terminal attachment). Using the TRACECUT option for a workstation in DFT mode will result in a hangup of both the host and the workstation sessions.

To start the trace, use the SEND or RECEIVE command format as shown in the box below:

```
SEND pc-filename host-filename (TRACE|TRACE=Q|TRACE=LST|TRACECUT .. other options ..
RECEIVE pc-filename host-filename (TRACE|TRACE=Q|TRACE=LST|TRACECUT .. other options ..
```

Output Files Created by the Trace

Depending on the trace options specified, each trace record is written to either of two files or both:

- A file in the VSE/POWER list (LST) queue:
  Name of file: IWSTRACE
  Disposition: L
  Class: A
  Destination (TO=): User ID of the requesting user.

  For each trace, a new entry with the name IWSTRACE and a new entry number is created.
- A file in a CICS temporary storage queue named CFTRTRCE.

  The output of several traces is added to the existing queue. If this is not desirable, purge the contents of the queue before you start a new trace. If you are repeatedly running a trace for a very large file, you should purge the queue to avoid using up all temporary storage.

If options TRACE or TRACECUT are specified and your system encounters a problem while writing into one trace file, it stores a corresponding message in the other file and writes the remaining output to that other file only.

Data Collected by the Trace

The trace output includes the following data:

- A general header containing:
  - The user ID and terminal ID of the requesting user.
  - The time and date of the request.
- Information about the setup of the host connection:
  - Whether the EXTDS feature is defined in the terminal definitions.
  - Whether the workstation runs in CUT or DFT mode.
- For each **inbound** and **outbound** transmission:
  - A header record describing the inbound or outbound transmission and showing the name of the host module that issued the request.
  - The data and control information that was transmitted.

For more information about inbound and outbound transmission, refer to “Trace Output Example DFT Mode” on page 229 and “Trace Output Example CUT Mode” on page 235.

Setup Checking During File Transfer

During normal file transfer (without the trace being active), checking for CUT or DFT mode is controlled by the presence of EXTDS in the terminal definition:

- If EXTDS is specified, a QUERY REPLY command is sent to the workstation to determine whether the DFT mode is supported. If so, the DTF protocol is used. Otherwise, CUT mode is assumed.

- If EXTDS is not specified, CUT mode is assumed without further checking (no QUERY REPLY). Note that this will lead to a session hangup if it is a DFT device in reality.

If the file transfer trace is activated with the **TRACE**, **TRACE=Q** or **TRACE=LST** option, a QUERY REPLY command is always sent, regardless of whether or not EXTDS has been specified in the terminal definitions.

Note: If the workstation is attached to a controller that does not have the EXTDS hardware feature, the QUERY REPLY command will result in a command reject. This causes CICS to set the session out of service. In such a case, use the TRACECUT option to activate the trace. Refer also to "Setup Errors (CUT Mode)" on page 241.

Procedures for Using the File Transfer Trace to Solve Problems

Use the trace if you run into one of the following problems:

- Incorrect data

  The file transfer has completed and an INWxxxxI message was issued in the host session as well as in the PC session. However, the data was not transmitted correctly.
- Session hangup

  The file transfer was started at the workstation, but no completion message was issued in the workstation session or in the host session. Your workstation may repeatedly issue the message:

  ```
  Host has not responded within timeout period
  ```

  and the host screen remains blank.

The procedures for these two error situations are described below.

Incorrect-Data Error

For error diagnosis, proceed as follows:

1. Repeat the SEND or RECEIVE command with the same options plus the *trace* option.
2. After file transfer is complete, switch to the host session. You get the message:

   ```
   INW0050I Trace complete - check TS queue CFTRTRCE or list queue IWSTRACE
   ```

   In the workstation session, the normal file transfer completion or error message is displayed.
3. Display the contents of the CICS temporary storage queue (by using the CEBR command) or print the VSE/POWER LST queue entry IWSTRACE to analyze the data portions of the trace output.
   a. If your workstation works in DFT mode, you can do a pre-analysis of the data to narrow down the source of the problem (for final analysis you need to contact IBM for support):
      - If you issued a SEND command, the trace output shows the data as received at the host from the workstation. If the data is still intact at this point, the error occurred at the host. Otherwise, the data was destroyed during transmission from the workstation to the host.
      - If you issued a RECEIVE command, the trace output shows the data before it is sent to the workstation. If the data is still intact at this point,

the error occurred during transmission to the workstation or was caused by the workstation control program. Otherwise, the data was destroyed at the host.

b. If your workstation works in CUT mode, the data is encoded and you need IBM support for final analysis.

Session-Hangup Error

For error diagnosis, proceed as follows:

1. In the host session, press the RESET key and then the PF2 key to end the session. Ignore message INW0002I.
2. Switch back to the workstation session. If necessary, cancel the session by pressing the CTRL and BREAK keys and reissue the SEND or RECEIVE command with the *trace* option.

If the problem was caused by an incorrect setup, message INW0051I or INW0052I is displayed. An incorrect setup may be the missing specification of EXTDS in the terminal definition if your workstation is a DFT device. In some cases, depending on the type of error and the workstation equipment, message INW0051I or INW0052 may not occur. The trace files, however, will contain information about the specific setup problem.

For other transmission errors, if the host does not respond to the workstation (or vice versa), the session will go into a hangup state again when the *trace* is on. At this point you should, if possible, leave the host session at your workstation undisturbed in its current state and look at the trace output in the CICS temporary storage queue from another terminal:

- If the last item in the queue is an **inbound** transmission, then the host did not respond.
- If the item was **outbound**, the workstation failed to respond.

Note: If you cancel the host session that is in the hangup state, the keystrokes that you enter will also be recorded in the trace files.

Trace Output Example DFT Mode

Setup of the Trace

The trace examples given here were produced on an IBM PS/2 running OS/2 EE 1.3. Traces obtained with other workstation equipment may show slightly different output in the QUERY REPLIES returned by the workstation.

Discussion of Output

The trace output shows each transmission from the workstation to the host as INBOUND and from the host to the workstation as OUTBOUND. Each line shows the host module of the file transfer program (INWPCCOM) which received or sent the data. In addition, the trace includes an internal trace item number required by IBM personnel for error diagnosis.

Specifics:

- A line starting with “INFO:” gives information about the setup.
- A line starting with “ERROR:” indicates an invalid setup.

The actual data transmitted is shown in hexadecimal format in the left-hand column, and in edited format in the right-hand column. The data may be in ASCII or in EBCDIC format, depending on the direction of transfer and the specified options. Therefore, editing must be done for either the ASCII or the EBCDIC format. However, since VSE/ESA cannot know the format, the following is assumed:

- SEND command – The data is always in ASCII format.
- RECEIVE command with the ASCII option – The data is in ASCII format.
- RECEIVE command with the BINARY option – The data is in EBCDIC format.

For IBM DBCS workstations, where ASCII-EBCDIC translation is done at the workstation, the following is assumed:

- SEND command with the JISCII or ASCII option – The data is in EBCDIC format.
- SEND command with the (implied) BINARY option – The data is in ASCII format.
- RECEIVE command – The data is in EBCDIC format.
- DBCS characters are treated as single-byte characters.

If these assumptions are not met, the data will not be edited correctly. In this case, the hexadecimal data in the left-hand column of the trace output must be used for error analysis.

For INBOUND transmission, the data is displayed as received from the workstation and before further processing (conversion) is done at the host. For OUTBOUND transmission, the data is displayed before it is sent to the workstation and after any necessary conversion at the host.

The file used for the trace examples included the following data:

```
THIS IS RECORD 0001
THIS IS RECORD 0002
THIS IS RECORD 0003
THIS IS RECORD 0004
THIS IS RECORD 0005
THIS IS RECORD 0006
THIS IS THE LAST RECORD
```

Output Examples

Figure 65 on page 231 shows the example of a trace output for a SEND command issued at a workstation in DFT mode. Figure 66 on page 233 shows the example of a trace output for a RECEIVE command issued at a workstation in DFT mode.

Note that the trace examples reflect error-free file transmissions. The sections following the examples point out the resulting changes in trace data in case of inbound transmission and setup errors.

▸ **SEND Command Used:**

```
SEND PCFILE HOSTFILE (ASCII CRLF TRACE
```

```
FILE XFER TRACE:  USERID=VATR    TERMID=A006 DATE=92126 TIME=151457...01

INBOUND: INPUT COMMAND IND$FILE ................ MODULE=INWPROOT.......09
C9D5C45BC6C9D3C540D7E4E340E3C5E2E3404DE3D9C1C3C5 IND$FILE PUT TEST (TRACE
7ED840                                           =Q

OUTBOUND: ERASE WRITE COMMAND TO CLEAR SCREEN... MODULE=INWPQUER.......20
F5C3                                             5C

INFO: EXTDS FEATURE DEFINED IN TCT ............. MODULE=INWPQUER.......04

OUTBOUND: SEND QUERY REPLY TO CHECK CUT OR DFT.. MODULE=INWPQUER.......21
000501FF02                                       .....

INBOUND: QUERY REPLIES RECEIVED FROM TERMINAL... MODULE=INWPQUER.......06
0017818101000050001801 03E80E2603E80E2608110780   ..aa...&;...Y...Y......
001B818582000811000000007000000002B900250100F103 ..aeb.................1.
C30136                                           C..
001181A600000B01000050001800500018               ..aw......&;..&;.
00168186000800F4F1F1F2F2F3F3F4F4F5F5F6F6F7F7     ..af...411223344556677
000D81870400F0F1F1F2F2F4F4                       ..ag..0112244
00078188000102                                   ..ah...
000681990000                                     ..ar..
000C81950000080008000101                         ..an........
0013819E800003000E00010401810104030003           ..a..........a.....
001D8195000003000E000101040181020902D7C3D3D20110 ..an..........a...PCLK..
8004030003                                       .....
0021819E800003000E000204018103040300030E0202B901 ..a..........a..........
F45C5CC7D8C6C4C5C6                               4**GQFDEF
0021819E800003000E000204018104040300030E0202B901 ..a..........a..........
F4D7D3D6E3E3C5D9F1                               4PLOTTER1
0021819E800003000E000204018105040300030E0202B901 ..a..........a..........
F4D7D9C9D5E3C5D9F1                               4PRINTER1
000F818080818586878895999EA6A8                   ..a..aefghnr.wy

INFO: TERMINAL IS IN DFT MODE .................. MODULE=INWPQUER.......17

OUTBOUND: OPEN REQUEST FOR DATA TRANSFER ....... MODULE=INWPOPN1.......30
00064000F1C20023D000120106010104030A0A0001000000 ..@....#................
00010050055203F0030946543A44415441               ...P.R....FT:DATA

INBOUND: OPEN REPLY (POSITIVE) ................. MODULE=INWPOPN1.......31
0005D00009                                       .....

OUTBOUND: SET CURSOR / GETDATA REQUEST ......... MODULE=INWPPUT1.......41
00064000F1C2000FD0451101050006000905010300 0009D0 ..@......E..............
461101040080                                     F.....

INBOUND: GETDATA ACKNOWLEDGEMENT + DATA ........ MODULE=INWPPUT1.......42
CONTROL INFORMATION
00A8D04605630600000001C08061009D                 ...F.c.......a..
```

Figure 65. Trace Example for a Workstation in DFT Mode (SEND Command) (Part 1 of 2)

▸ **SEND Command Used:**

```
SEND PCFILE HOSTFILE (ASCII CRLF TRACE
```

```
USER DATA
54484953204953205245434F524420303030310D0A544849 THIS IS RECORD 0001..THI
53204953205245434F524420303030320D0A544849532049 S IS RECORD 0002..THIS I
53205245434F524420303030330D0A544849532049532052 S RECORD 0003..THIS IS R
45434F524420303030340D0A5448495320495320524543 4F ECORD 0004..THIS IS RECO
524420303030350D0A5448495320495320524543 4F524420 RD 0005..THIS IS RECORD
303030360D0A5448495320495320544845204C4153542052 0006..THIS IS THE LAST R
45434F52440D0A1A                                 ECORD...

OUTBOUND: SET CURSOR / GETDATA REQUEST ......... MODULE=INWPPUT1.......41
00064000F1C2000FD045110105000600090501030000 09D0 ..@......E..............
461101040080                                     F.....

INBOUND: END OF FILE ........................... MODULE=INWPPUT1.......43
0009D0460869042200                               ...F.i.".

OUTBOUND: CLOSE FILE REQUEST ................... MODULE=INWPCLS1.......34
00064000F1C20005D04112                           ..@......A.

INBOUND: CLOSE FILE REPLY (POSITIVE) ........... MODULE=INWPCLS1.......35
0005D04109                                       ...A.

OUTBOUND: OPEN REQUEST FOR MESSAGE ............. MODULE=INWPOPN1.......33
00064000F1C20023D00012010601010403 0A0A0000000011 ..@....#................
01010050055203F0030946543A4D534720               ...P.R....FT:MSG

INBOUND: OPEN REPLY (POSITIVE) ................. MODULE=INWPOPN1.......31
0005D00009                                       .....

OUTBOUND: INSERT REQUEST FOR MESSAGE ........... MODULE=INWPGET1.......40
00064000F1C2000AD0471101050080000 05AD04704C08061 ..@......G.......Z.G...a
0055494E573030303149202046696C65207472616E736665 .UINW0001I  File transfe
7220636F6D706C6574652E2024000000000000000000000000 r complete. $..........
000000000000000000000000000020202020202020202020 ..............
2020202020202020202020

INBOUND: INSERT REPLY (POSITIVE) ............... MODULE=INWPGET1.......38
000BD04705630600000001                           ...G.c.....

OUTBOUND: SEND MESSAGE TO HOST SCREEN .......... MODULE=INWPMSG .......45
C9D5E6F0F0F4F9C94040E399818385408396949793 85A385 INW0049I  Trace complete
40604083888583924 0A399818385408689938540C3C6E3D9  - check trace file CFTR
E3D9C3C540899540C3C9C3E240A385949796998199A840A2 TRCE in CICS temporary s
A3969981878540404040404040404040404040404040 4040 torage
404040404040404040

E N D   O F   T R A C E                          MODULE=INWPMSG ..... .05
```

Figure 65. Trace Example for a Workstation in DFT Mode (SEND Command) (Part 2 of 2)

▸ **RECEIVE Command Used:**

```
RECEIVE PCFILE HOSTFILE (ASCII CRLF TRACE
```

```
FILE XFER TRACE:  USERID=VATR     TERMID=A006 DATE=92126 TIME=151935...01

INBOUND: INPUT COMMAND IND$FILE ................ MODULE=INWPROOT.......09
C9D5C45BC6C9D3C540C7C5E340E3C5E2E3404DE3D9C1C3C5 IND$FILE GET TEST (TRACE
7ED840                                           =Q

OUTBOUND: ERASE WRITE COMMAND TO CLEAR SCREEN... MODULE=INWPQUER.......20
F5C3                                             5C

INFO: EXTDS FEATURE DEFINED IN TCT ............ MODULE=INWPQUER.......04

OUTBOUND: SEND QUERY REPLY TO CHECK CUT OR DFT.. MODULE=INWPQUER.......21
000501FF02                                       .....

INBOUND: QUERY REPLIES RECEIVED FROM TERMINAL... MODULE=INWPQUER.......06
001781810100005000180103E80E2603E80E2608110780   ..aa...&;...Y...Y......
001B818582000811000000000007000002B900250100F103 ..aeb.................1.
C30136                                           C..
001181A600000B01000050001800500018               ..aw......&;..&;.
00168186000800F4F1F1F2F2F3F3F4F4F5F5F6F6F7F7     ..af...411223344556677
000D81870400F0F1F1F2F2F4F4                       ..ag..0112244
00078188000102                                   ..ah...
000681990000                                     ..ar..
000C81950000080008000101                         ..an........
0013819E800003000E00010401810104030003           ..a..........a.....
001D8195000003000E00010104018102090 2D7C3D3D20110 ..an..........a...PCLK..
8004030003                                       .....
0021819E800003000E0002040181030403000 30E0202B901 ..a..........a..........
F45C5CC7D8C6C4C5C6                               4**GQFDEF
0021819E800003000E00020401810404030003 0E0202B901 ..a..........a..........
F4D7D3D6E3E3C5D9F1                               4PLOTTER1
0021819E800003000E0002040181050403000 30E0202B901 ..a..........a..........
F4D7D9C9D5E3C5D9F1                               4PRINTER1
000F818080818586878895999EA6A8                   ..a..aefghnr.wy

INFO: TERMINAL IS IN DFT MODE ................. MODULE=INWPQUER.......17

OUTBOUND: OPEN REQUEST FOR DATA TRANSFER ....... MODULE=INWPOPN1.......30
00064000F1C20023D000120106010104030A0A0000000011 ..@....#................
01010050055203F0030946543A44415441               ...P.R....FT:DATA

INBOUND: OPEN REPLY (POSITIVE) ................ MODULE=INWPOPN1.......31
0005D00009                                       .....

OUTBOUND: INSERT REQUEST FOR DATA ............. MODULE=INWPGET1.......37
CONTROL INFORMATION
00064000F1C2000AD04711010500800000A2D047         ..@......G.........G
```

Figure 66. Trace Example for a Workstation in DFT Mode (RECEIVE Command) (Part 1 of 2)

▸ **RECEIVE Command Used:**

```
RECEIVE PCFILE HOSTFILE (ASCII CRLF TRACE
```

```
USER DATA
04C08061009D5448495320495320524543 4F524420303030 ...a..THIS IS RECORD 000
```

```
USER DATA
04C08061009D544849532049532052454 34F524420303030 ...a..THIS IS RECORD 000
```

Figure 66. Trace Example for a Workstation in DFT Mode (RECEIVE Command) (Part 2 of 2)

Inbound Transmission Errors (DFT Mode)

Whenever an unexpected reply is received by the host from the workstation, the corresponding INBOUND text line is replaced by its negative counterpart:

```
INBOUND: OPEN REPLY (POSITIVE) ................ MODULE=INWPOPN1.......31
INBOUND: OPEN REPLY (NEGATIVE) ................ MODULE=INWPOPN1.......32

INBOUND: INSERT REPLY (POSITIVE) .............. MODULE=INWPGET1.......38
INBOUND: INSERT REPLY (NEGATIVE) .............. MODULE=INWPGET1.......39

INBOUND: GETDATA ACKNOWLEDGEMENT + DATA ....... MODULE=INWPPUT1.......42
INBOUND: GETDATA REPLY (NEGATIVE) ............. MODULE=INWPPUT1.......44

INBOUND: CLOSE FILE REPLY (POSITIVE) .......... MODULE=INWPCLS1.......35
INBOUND: CLOSE FILE REPLY (NEGATIVE) .......... MODULE=INWPCLS1.......36
```

If negative replies occur during file transfer without any operator intervention, they indicate a malfunction of the host or the workstation file transfer program, or of the hardware. If such an error occurs, contact IBM for support.

However, negative replies are also recorded if, after a session hangup, the operator presses any keys in the host session. It is, therefore, recommended to display the trace file in CICS temporary storage from another terminal to avoid disturbing the host session in the hangup state.

Setup Errors (DFT Mode)

If the EXTDS feature was not specified in the terminal definition for this terminal, the error text

```
ERROR: TERMINAL IS IN DFT MODE - EXTDS REQUIRED. MODULE=INWPQUER.......15
```

is written to the trace file. File transfer is complete with message INW0051I.

Trace Output Example CUT Mode

Setup of the Trace

The following trace examples were produced on an IBM PS/2 with the IBM 3278/79 Emulation Adapter and the IBM PC 3270 Emulation Program, Entry Level Version 2.0.

Discussion of Output

The trace output shows each transmission from the workstation to the host as INBOUND and from the host to the workstation as OUTBOUND. Each line shows the host module of the file transfer program (INWPCCOM) which received or sent the data. In addition, the trace includes an internal trace item number required by IBM personnel for error diagnosis.

The actual data is shown in hexadecimal in the left-hand column, and in edited format in the right-hand column.

The data is transmitted in encoded form, that is, user data will be interspersed with additional control characters. Therefore, do not expect to recognize your data in the original format.

Specifics:

- A line starting with “INFO:” provides information about the setup.

- A line starting with “ERROR:” indicates an invalid setup.
- For INBOUND transmission, the data is displayed as received by the workstation, before further processing (conversion) is done by the host.
- For OUTBOUND transmission, the data is displayed before it is sent to the workstation, after a necessary conversion has been done by the host.
- The file used for the trace examples included the following data:

```
THIS IS RECORD 0001
THIS IS RECORD 0002
THIS IS RECORD 0003
THIS IS RECORD 0004
THIS IS RECORD 0005
THIS IS RECORD 0006
THIS IS THE LAST RECORD
```

Output Examples

Figure 67 on page 237 shows the example of a trace output for a SEND command issued at a workstation in CUT mode. Figure 68 on page 239 shows the example of a trace output for a RECEIVE command issued at a workstation in CUT mode.

Note that the trace examples reflect error-free file transmissions. The sections following the examples point out the resulting changes in the trace data in case of inbound transmission and setup errors.

▸ **SEND Command Used:**

```
SEND PCFILE HOSTFILE (ASCII CRLF TRACE
```

```
FILE XFER TRACE:  USERID=VATR     TERMID=A006 DATE=92126 TIME=152905...01

INBOUND: INPUT COMMAND IND$FILE ................ MODULE=INWPROOT.......09
C9D5C45BC6C9D3C540D7E4E340E3C5E2E3404DD9C5D7D3C1 IND$FILE PUT TEST (REPLA
C3C540E3D9C1C3C57ED8                             CE TRACE=Q

OUTBOUND: ERASE WRITE COMMAND TO CLEAR SCREEN... MODULE=INWPQUER.......20
F5C3                                             5C

INFO: EXTDS FEATURE DEFINED IN TCT ............ MODULE=INWPQUER.......04

OUTBOUND: SEND QUERY REPLY TO CHECK CUT OR DFT.. MODULE=INWPQUER.......21
000501FF02                                       .....

INBOUND: QUERY REPLIES RECEIVED FROM TERMINAL... MODULE=INWPQUER.......06
000B8180808185868788A6                           ..a..aefghw
00178181010000050001800000A02E50002006F090C0780  ..aa...&;.....V...?....
001B8185B2000910600000000700000000006500250100F101 ..ae....-.............1.
7E0136                                           =..
00168186000800F4F1F1F2F2F3F3F4F4F5F5F6F6F7F7     ..af...411223344556677
000D81870400F0F1F1F2F2F4F4                       ..ag..0112244
00078188000102                                   ..ah...
001181A600000B01000050001800500018               ..aw......&;..&;.

INFO: TERMINAL IS IN CUT MODE .................. MODULE=INWPQUER.......19

OUTBOUND: COMMAND ACKNOWLEDGEMENT (OPEN) ....... MODULE=INWPOPN0.......22
114040C3818181115D7A1DC113000000001D7C           .  Caaa.):.A......@

INBOUND: POSITIVE REPLY AID ENTER ............. MODULE=INWPOPN0.......23
115D7B                                           .)#

OUTBOUND: SEND EMPTY FRAME ..................... MODULE=INWPPUT0.......27
114040C21DC1C182115D7F1D7C                       .  B.AAb.)".@

INBOUND: DATA FRAME (DATA ENCODED) ............ MODULE=INWPPUT0.......29
CONTROL INFORMATION
1140C2C182D483C7                                 . BAbMcG
```

Figure 67. Trace Example for a Workstation in CUT Mode (SEND Command) (Part 1 of 2)

▸ **SEND Command Used:**

```
    SEND PCFILE HOSTFILE (ASCII CRLF TRACE

USER DATA
7EE3C8C9E240C9E240D9C5C3D6D9C440F0F0F0F15CD5D27E =THIS IS RECORD 0001*NK=
E3C8C9E240C9E240D9C5C3D6D9C440F0F0F0F25CD5D27EE3 THIS IS RECORD 0002*NK=T
C8C9E240C9E240D9C5C3D6D9C440F0F0F0F35CD5D27EE3C8 HIS IS RECORD 0003*NK=TH
C9E240C9E240D9C5C3D6D9C440F0F0F0F45CD5D27EE3C8C9 IS IS RECORD 0004*NK=THI
E240C9E240D9C5C3D6D9C440F0F0F0F55CD5D27EE3C8C9E2 S IS RECORD 0005*NK=THIS
40C9E240D9C5C3D6D9C440F0F0F0F65CD5D27EE3C8C9E240  IS RECORD 0006*NK=THIS
C9E240E3C8C540D3C1E2E340D9C5C3D6D9C45CD5D24C     IS THE LAST RECORD*NK<

OUTBOUND: SEND EMPTY FRAME ..................... MODULE=INWPPUT0.......27
114040C21DC1C183115D7F1D7C                       .  B.AAc.)".@

INBOUND: DATA FRAME WITH EOF INDICATOR *z ...... MODULE=INWPPUT0.......28
1140C2C183E581835CA9                             . BAcVac*z

OUTBOUND: COMMAND ACKNOWLEDGEMENT WITH MESSAGE.. MODULE=INWPOPN0.......24
114040C3848189C9D5E6F0F0F0F1C94040C689938540A399 .  CdaiINW0001I  File tr
8195A286859940839694979385A3854B405B000000000000 ansfer complete. $......
000000000000000000000000000000000000000040404040 ....................
4040404040404040404040115D7A1DC113000000                        .):.A....
001D7C                                           ..@

INBOUND: POSITIVE REPLY AID ENTER .............. MODULE=INWPOPN0.......23
115D7B                                           .)#

OUTBOUND: SEND MESSAGE TO HOST SCREEN .......... MODULE=INWPMSG .......45
C9D5E6F0F0F4F9C94040E399818385408396949793854A385 INW0049I  Trace complete
40604083888583924 0A39981838540868993854 0C3C6E3D9  - check trace file CFTR
E3D9C3C540899540C3C9C3E240A385949796998199A840A2 TRCE in CICS temporary s
A3969981878540404040404040404040404040404040404040 torage
404040404040404040

E N D   O F   T R A C E                          MODULE=INWPMSG ..... .05
```

Figure 67. Trace Example for a Workstation in CUT Mode (SEND Command) (Part 2 of 2)

▸ **RECEIVE Command Used:**

```
RECEIVE PCFILE HOSTFILE (ASCII CRLF TRACE
```

```
FILE XFER TRACE:  USERID=VATR     TERMID=A006 DATE=92126 TIME=153027...01

INBOUND: INPUT COMMAND IND$FILE ................ MODULE=INWPROOT.......09
C9D5C45BC6C9D3C540C7C5E340E3C5E2E3404DE3D9C1C3C5 IND$FILE GET TEST (TRACE
7ED8                                             =Q

OUTBOUND: ERASE WRITE COMMAND TO CLEAR SCREEN... MODULE=INWPQUER.......20
F5C3                                             5C

INFO: EXTDS FEATURE DEFINED IN TCT ............. MODULE=INWPQUER.......04

OUTBOUND: SEND QUERY REPLY TO CHECK CUT OR DFT.. MODULE=INWPQUER.......21
000501FF02                                       .....

INBOUND: QUERY REPLIES RECEIVED FROM TERMINAL... MODULE=INWPQUER.......06
000B8180808185868788A6                           ..a..aefghw
0017818101000050001800000A02E50002006F090C0780   ..aa...&;.....V...?....
001B8185B2000910600000000700000000650025010 0F101 ..ae....-.............1.
7E0136                                           =..
00168186000800F4F1F1F2F2F3F3F4F4F5F5F6F6F7F7     ..af...411223344556677
000D81870400F0F1F1F2F2F4F4                       ..ag..0112244
00078188000102                                   ..ah...
001181A600000B01000050001800500018               ..aw......&;..&;.

INFO: TERMINAL IS IN CUT MODE .................. MODULE=INWPQUER.......19

OUTBOUND: COMMAND ACKNOWLEDGEMENT (OPEN) ....... MODULE=INWPOPN0.......22
114040C3818181115D7A1DC113000000001D7C           .  Caaa.):.A......@

INBOUND: POSITIVE REPLY AID ENTER .............. MODULE=INWPOPN0.......23
115D7B                                           .)#

OUTBOUND: SEND DATA FRAME (DATA ENCODED) ....... MODULE=INWPGET0.......25
CONTROL INFORMATION
114040C182E283D5                                 .  AbScN
```

Figure 68. Trace Example for a Workstation in CUT Mode (RECEIVE Command) (Part 1 of 2)

▸ **RECEIVE Command Used:**

```
    RECEIVE PCFILE HOSTFILE (ASCII CRLF TRACE

USER DATA
5E7EE3C8C9E240C9E240D9C5C3D6D9C440F0F0F0F15CD5D2 ;=THIS IS RECORD 0001*NK
5E7EE3C8C9E240C9E240D9C5C3D6D9C440F0F0F0F25CD5D2 ;=THIS IS RECORD 0002*NK
5E7EE3C8C9E240C9E240D9C5C3D6D9C440F0F0F0F35CD5D2 ;=THIS IS RECORD 0003*NK
5E7EE3C8C9E240C9E240D9C5C3D6D9C440F0F0F0F45CD5D2 ;=THIS IS RECORD 0004*NK
5E7EE3C8C9E240C9E240D9C5C3D6D9C440F0F0F0F55CD5D2 ;=THIS IS RECORD 0005*NK
5E7EE3C8C9E240C9E240D9C5C3D6D9C440F0F0F0F65CD5D2 ;=THIS IS RECORD 0006*NK
5E7EE3C8C9E240C9E240E3C8C540D3C1E2E340D9C5C3D6D9 ;=THIS IS THE LAST RECOR
C45CD5D24C                                       D*NK<
CONTROL INFORMATION
115D7A1DC113000000001D7C                         .):.A......@

INBOUND: POSITIVE REPLY AID ENTER .............. MODULE=INWPGET0.......23
115D7B                                           .)#

OUTBOUND: SEND DATA FRAME WITH EOF INDICATOR *z  MODULE=INWPCLS0.......26
114040C183E581835CA9115D7A1DC113000000001D7C     .  AcVac*z.):.A......@

INBOUND: POSITIVE REPLY AID ENTER .............. MODULE=INWPCLS0.......23
115D7B                                           .)#

OUTBOUND: COMMAND ACKNOWLEDGEMENT WITH MESSAGE.. MODULE=INWPOPN0.......24
114040C3848189C9D5E6F0F0F0F1C94040C689938540A399 .  CdaiINW0001I  File tr
8195A286859940839694979385A3854B405B000000000000 ansfer complete. $......
000000000000000000000000000000000000000040404040 ....................
404040404040404040404040404040115D7A1DC113000000               .):.A....
001D7C                                           ..@

INBOUND: POSITIVE REPLY AID ENTER .............. MODULE=INWPOPN0.......23
115D7B                                           .)#

OUTBOUND: SEND MESSAGE TO HOST SCREEN .......... MODULE=INWPMSG .......45
C9D5E6F0F0F4F9C94040E399818385408396949793859A385 INW0049I  Trace complete
406040838885839240A39981838540868993854040C3C6E3D9  - check trace file CFTR
E3D9C3C540899540C3C9C3E240A385949796998199A840A2 TRCE in CICS temporary s
A39699818785404040404040404040404040404040404040 torage
404040404040404040

E N D   O F   T R A C E                          MODULE=INWPMSG ..... .05
```

Figure 68. Trace Example for a Workstation in CUT Mode (RECEIVE Command) (Part 2 of 2)

Inbound Transmission Errors (CUT Mode)

If an unexpected reply is received by the host from the workstation, the corresponding INBOUND text line:

```
INBOUND: POSITIVE REPLY AID ENTER .............. MODULE=INWPnnnn.......23
INBOUND: DATA FRAME (DATA ENCODED) ............. MODULE=INWPnnnn.......29
INBOUND: DATA FRAME WITH EOF INDICATOR *z ...... MODULE=INWPnnnn.......28
```

is replaced by:

```
INBOUND: NEGATIVE REPLY AID xx ................. MODULE=INWPnnnn.......18
```

The AID (Attention Identifier) may be one of the following:

F1 = The workstation detected an error in the transmitted data. File transfer is terminated.

F2 = The workstation detected an unrecoverable error. File transfer is terminated.

?? = An unexpected AID (other than F1, F2, or ENTER) was received. File transfer is terminated.

If these negative replies occur repeatedly without any operator intervention, contact IBM for support. However, negative replies are also recorded if, after a session hangup, the operator presses any keys in the host session that generates an AID. It is, therefore, recommended to display the trace file in CICS temporary storage from another terminal to avoid disturbing the host session in the hangup state.

Setup Errors (CUT Mode)

If a setup error is detected, diagnostic information is recorded in the trace files and an attempt is made to display a corresponding message. Depending on the type of error, the system may not be able to display such a message. Check the trace output for diagnostic information.

If the TRACECUT option was specified for a workstation that has the EXTDS feature defined in the terminal definition, message INW0052I is issued and the text shown below is recorded in the trace files:

```
INFO: CUT MODE ASSUMED DUE TO TRACECUT OPTION .. MODULE=INWPQUER.......16
INFO: EXTDS FEATURE DEFINED IN TCT .......... .. MODULE=INWPQUER.......04
ERROR: EXTDS FEATURE INVALID FOR TRACECUT OPTION MODULE=INWPQUER.......08
```

If the TRACE option is used for a workstation connected to a controller that does not support EXTDS, the QUERY REPLY command causes CICS to set the session out of service. If this happens, the session must be reactivated with the CICS command

```
CEMT SET TE(nnnn) INS ACQ
```

where nnnn is the terminal ID. The trace files will contain the following diagnostic information if the EXTDS feature was specified:

```
INFO: EXTDS FEATURE DEFINED IN TCT .......... .. MODULE=INWPQUER.......04
OUTBOUND: SEND QUERY REPLY TO CHECK CUT OR DFT.. MODULE=INWPQUER.......21
ERROR: QUERY REPLY REJECTED .................... MODULE=INWPQUER.......02
ERROR: TERMINAL WILL BE SET OUT OF SERVICE ..... MODULE=INWPQUER.......10
INFO: TERMINAL IS IN CUT MODE ............... .. MODULE=INWPQUER.......19
ERROR: TERMINAL DOES NOT SUPPORT EXTDS FEATURE.. MODULE=INWPQUER.......11
INFO: REMOVE EXTDS FEATURE FROM TCT AND ..... .. MODULE=INWPQUER.......12
INFO: RETRY TRACE WITH TRACECUT OPTION ...... .. MODULE=INWPQUER.......13
```

If EXTDS was not specified, the following diagnostic information will be recorded:

```
INFO: EXTDS FEATURE NOT DEFINED IN TCT ...... .. MODULE=INWPQUER.......03
OUTBOUND: SEND QUERY REPLY TO CHECK CUT OR DFT.. MODULE=INWPQUER.......21
ERROR: QUERY REPLY REJECTED .................... MODULE=INWPQUER.......02
ERROR: TERMINAL WILL BE SET OUT OF SERVICE ..... MODULE=INWPQUER.......10
INFO: TERMINAL IS IN CUT MODE ............... .. MODULE=INWPQUER.......19
INFO: RETRY TRACE WITH TRACECUT OPTION ...... .. MODULE=INWPQUER.......13
```

If this information is shown, repeat the trace with the TRACECUT option.

Chapter 12. Using File Transfer Service Functions

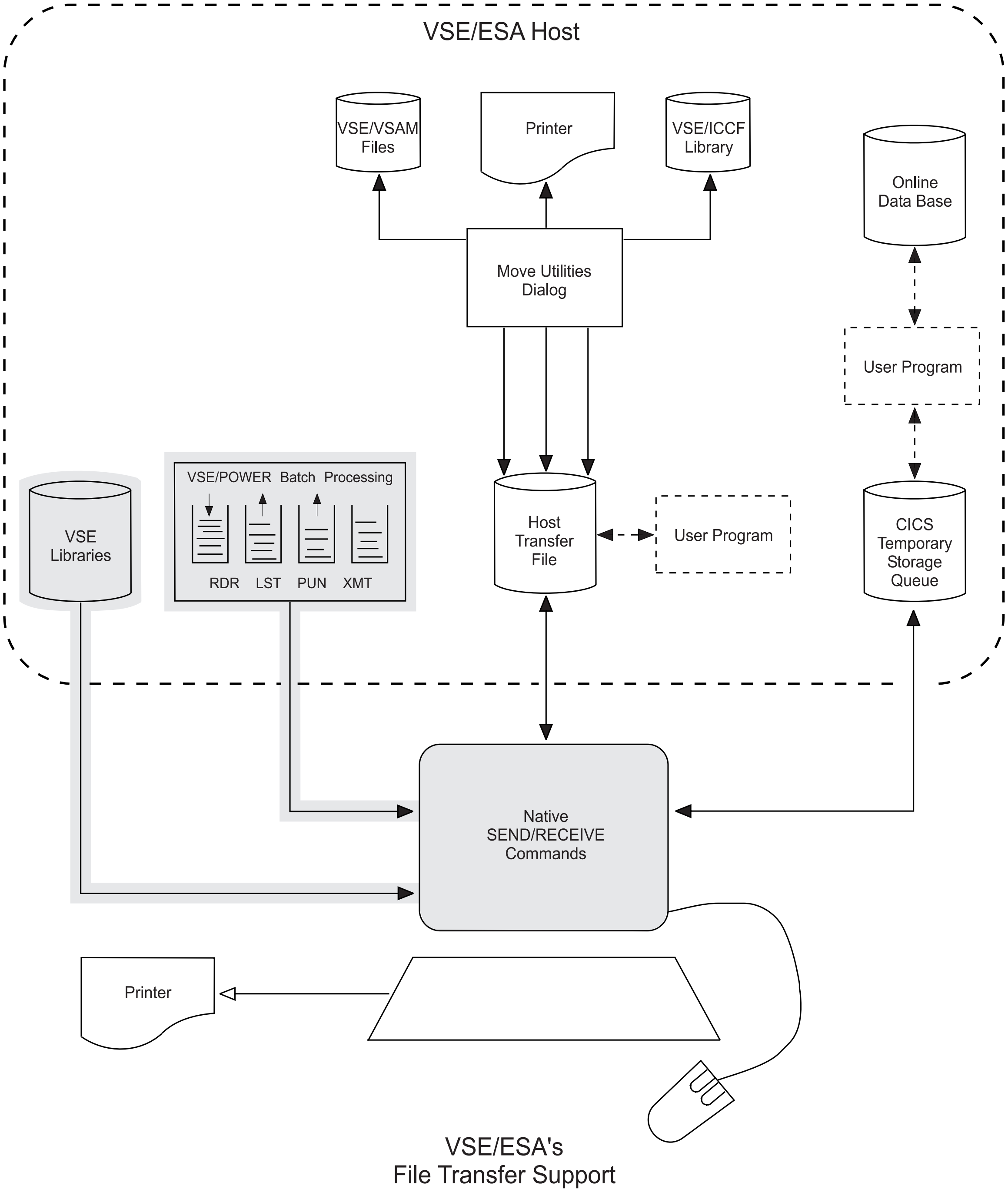


Figure 69. Data Paths of Service Functions

Overview

To support access to VSE/ESA host resources by workstation application programs VSE/ESA provides a set of file transfer service functions.

Service Functions

These service functions allow you to download information about host resources rather than actual host data.

By using these functions, you can:

- Receive VSE library directory information.
- Execute VSE Librarian commands by using services of the Librarian and receive command output.
- Execute VSE/POWER commands and receive command output.
- Receive formatted displays of VSE/POWER queues.
- Receive job completion messages from VSE/POWER.
- Receive a host user profile from the VSE/ESA host.

Using File Transfer Service Functions for VSE Libraries

This section contains information about the following service functions:

- Receiving directory information about VSE libraries.
- Executing VSE Librarian commands.

Receiving Directory Information About VSE Libraries

Overview

This service function transfers information about sublibraries or sublibrary members contained in the specified VSE library to the workstation. It is specifically designed to be used by a workstation front-end program. The same information could also be retrieved with a LISTDIR command specified in a RECEIVE (FILE=LCMD command (see “Executing VSE Librarian Commands” on page 251). However, this command will return list output which is subject to change and should, therefore, not be used as input to a program. The functions described here obtain the library directory information via the Librarian application program interface (API).

Depending on the options specified in the RECEIVE command, this service provides a list of directory entries of

- the sublibraries contained in a specific library, or
- the members contained in a specific sublibrary.

For each sublibrary a fixed-format record is returned. This record contains the following information:

```
library name
sublibrary name
creation date
creation time
number of members
number of used library blocks
number of reclaimed library blocks
space reclamation attribute
number of locked members
```

The layout of the returned sublibrary directory records is shown in Table 7 on page 248.

Directory entries can be displayed either for all sublibraries or for one specific sublibrary in a given library.

For each member a fixed-format record is returned. This record contains the following information:

```
member name
member type
date member was created
time member was created
date of last update
time of last update
number of bytes/records
record format: fixed or string
logical record length
member contains SYSIPT data
member is MSHP controlled/bypassed
number of used library blocks
locked by: userid
```

The layout of the returned member directory records is shown in Table 8 on page 248.

It is also possible to request directory information of a subset of sublibraries **or** members by specifying a generic sublibrary name **or** membername/membertype.

Note: Be aware that it is not possible to request a member display from a generic set of sublibraries. Library, sublibrary, and member information is specified with the following RECEIVE command options:

```
L=library_name
S=sublibrary_name
MN=member_name
MT=member_type
```

For a member directory display the MN= or MT= option (or both) **must** be specified.

Table 6. FILE=LDIR: Sublibrary and Member Directory Display Options. The table shows the possible combinations of the L=, S=, MN=, and MT= options and the resulting directory displays. **O** = option omitted, **S** = option specified.

L=	S=	MN=	MT=	displays the directory entry(ies) of
O	O	O	O	all sublibraries in library PRIMARY
S	O	O	O	all sublibraries in specified library
O	S	O	O	sublibrary PRIMARY.sublib
S	S	O	O	sublibrary lib.sublib
O	O	S	O	all members defined by MN= in sublibrary PRIMARY.userid.
O	O	O	S	all members defined by MT= in sublibrary PRIMARY.userid.
O	O	S	S	all members defined by MN= and MT= in sublibrary PRIMARY.userid.
O	S	S	O	all members defined by MN= in sublibrary PRIMARY.sublib.
O	S	O	S	all members defined by MT= in sublibrary PRIMARY.sublib.
O	S	S	S	all members defined by MN= and MT= in sublibrary PRIMARY.sublib.

Table 6. FILE=LDIR: Sublibrary and Member Directory Display Options (continued). The table shows the possible combinations of the L=, S=, MN=, and MT= options and the resulting directory displays. **O** = option omitted, **S** = option specified.

L=	S=	MN=	MT=	displays the directory entry(ies) of
S	O	S	O	all members defined by MN= in sublibrary lib.userid.
S	O	O	S	all members defined by MT= in sublibrary lib.userid.
S	O	S	S	all members defined by MN= and MT= in sublibrary lib.userid.
S	S	S	O	all members defined by MN= in sublibrary lib.sublib.
S	S	O	S	all members defined by MT= in sublibrary lib.sublib.
S	S	S	S	all members defined by MN= and MT= in sublibrary lib.sublib.

The RECEIVE command format is described in Figure 70 on page 249.

Examples

About this Section: This section provides various examples to show how you may use the RECEIVE command with option FILE=LDIR.

An example starts with a short heading marked by the '►' character.

► **Receive Sublibrary Directory:**

```
RECEIVE  PRIMARY.DIR  ( FILE=LDIR
         or
RECEIVE  PRIMARY.DIR  ( FILE=LDIR  S=*

Receives a list of all sublibraries defined in library PRIMARY to workstation file PRIMARY.DIR.
PRIMARY is the default library used when no library is specified.


RECEIVE  PRD2.DIR  ( FILE=LDIR  L=PRD2

Receives a list of all sublibraries defined in library PRD2 to workstation file PRD2.DIR.


RECEIVE  PRD2CONF.SUB  ( FILE=LDIR  L=PRD2  S=IJB*

Receives a list of all sublibraries whose names start with IJB in library PRD2.


RECEIVE  PRD2CONF.DIR  ( FILE=LDIR  L=PRD2  S=CONFIG

Receives a single directory entry of sublibrary PRD2.CONFIG.
```

▸ **Receive Member Directory:**

```
RECEIVE  PRIMEMEM.DIR  ( FILE=LDIR  MN=*
         or
RECEIVE  PRIMEMEM.DIR  ( FILE=LDIR  MT=*
         or
RECEIVE  PRIMEMEM.DIR  ( FILE=LDIR  MN=*  MT=*
```

Receives a directory list of all members contained in the user's primary library PRIMARY.userid.

```
RECEIVE  TYPEA.DIR  ( FILE=LDIR  MT=A
```

Receives a directory list of all members contained in the user's primary library PRIMARY.userid that have a filetype of A.

```
RECEIVE  STARTIES.DIR  ( FILE=LDIR  MN=IES*
```

Receives a directory list of all members contained in the user's primary library PRIMARY.userid whose names start with the characters IES.

```
RECEIVE  IESBOOKS.DIR  ( FILE=LDIR  L=PRD2  S=SAVE  MN=IES*  MT=B
```

Receives a directory list of all B-books in sublibrary PRD2.SAVE whose name starts with the characters IES.

Command Output Description

The following two tables describe the output of the service functions which may be used as input by a workstation application program.

Table 7. Format of Sublibrary Directory Records

Position	Length	Description	Sample Contents
0-6	7	Library name	PRD2ƀƀƀ
7-14	8	Sublibrary name	CONFIGƀƀ
15-22	8	Creation date YY-MM-DD	93-05-13
23-30	8	Creation time hh:mm:ss	14:20:03
31-40	10	Number of members	0000000132
41-50	10	Number of used library blocks	0000000975
51-60	10	Number of reclaimed library blocks (delayed LB's)	0000000000
61	1	Space reclamation attribute (space reusage): AUTOMATIC (A) or IMMEDIATE (I)	A
62-71	10	Number of locked members	000000032

Table 8. Format of Member Directory Records

Position	Length	Description	Sample Contents
0-7	8	Membername	IESABCDE
8-15	8	Membertype	PROCƀƀƀƀ
16-23	8	Creation date YY-MM-DD	93-05-13
24-31	8	Creation time hh:mm:ss	14:20:03
32-39	8	Last update date YY-MM-DD	93-05-24
40-47	8	Last update time hh:mm:ss	13:34:12
48-57	10	Number of records/bytes	0000000142
58	1	Record format: R=fixed B=string	R
59-68	10	Logical record length	000000080
69	1	MSHP control: 1=no 2=yes 3=bypassed	3
70-79	10	Number of used library blocks	0000000006
80-87	8	Member locked by: userid	USER1ƀƀƀ
88	1	SYSIPT data: YES (Y) or NO (n) (type PROC only)	Y

Command Syntax and Options RECEIVE (FILE=LDIR

Command Syntax

The following shows the RECEIVE command format for receiving a VSE library's directory information. In case you are not familiar with the notation of the following syntax diagram, refer to "How to Read SEND/RECEIVE Syntax Diagrams", on page 273.

```
►►──RECEIVE──pws_name──┬─────┬──┬───────────────────────────┬──(──FILE=LDIR──────────────►
                       └─id:─┘  └─dummy_name──┬────────────┬┘
                                              └─dummy_type─┘

►─┤ Options-1 ├─┤ Options-2 ├──┬───┬──────────────────────────────────────────────────────►◄
                               └─)─┘
```

Options-1:

```
   ┌─L=PRIMARY───────┐ ┌─S=user_id─────────┐ ┌─CLASS=P─────────┐
├──┴─L=library_name──┴─┴─S=sublibrary_name─┴─┴─CLASS=job_class─┴──┬────────────────┬──────►
                                                                  └─MN=member_name─┘
►──┬────────────────┬──────────────────────────────────────────────────────────────────────┤
   └─MT=member_type─┘
```

Options-2:

```
   ┌─ASCII──┐ ┌─CRLF───┐ ┌─REPLACE─┐ ┌─BUF=4096──┐
├──┴─BINARY─┴─┴─NOCRLF─┴─┴─APPEND──┴─┴─BUF=nnnnn─┴──┬───────┬──┬──────┬──┬────────────┬──►
                                                    └─BLANK─┘  └─INW1─┘  └─NLS=suffix─┘
►──┬────────────────┬──┬─────────────┬──┬──────────────────┬──────────────────────────────┤
   └─DCDF=dcdf_name─┘  ├─TRACE───────┤  └─USERTRT=user_trt─┘
                       ├─TRACE=Q─────┤
                       ├─TRACE=LST───┤
                       └─TRACECUT────┘
```

Figure 70. Receive Library Directory List - (List of Sublibraries or Members) - FILE=LDIR

Command Parameters and Options

The following describes parameters and options that are special to using this RECEIVE service function.

For a description of the **pws_name** and **id:** parameters, refer to "Common SEND/RECEIVE Command Syntax" on page 70.

For a description of the following **common options**, refer to "Common SEND/RECEIVE Command Options" on page 72.

Command Syntax and Options RECEIVE (FILE=LDIR

```
ASCII/BINARY
CRLF/NOCRLF
APPEND/REPLACE
BLANK
INW1
NLS=
DCDF=
TRACE
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=
```

dummy_name
a host specification is not applicable for this function. However, some 3270 Emulation Programs (such as FTTERM) require a hostname. In that case, specify any name. It will be ignored by the host.

dummy_type
has no meaning and is ignored if specified (only included for consistency of command syntax).

FILE=LDIR
receive a VSE library directory: a list of all sublibraries of a specified library or of all or selected members in a sublibrary.

options
in alphabetic order:

BUF=nnnnn
defines the size of the receive buffer used to receive the directory information. The **default** buffer size is 4096 bytes.

When transferring very large amounts of data, performance may be improved by increasing the buffer size. **nnnnn** can be a value from **512** to **32767**.

A reduction of the buffer size may be necessary when there is not enough main storage available to allocate the default buffer.

CLASS=job_class
allows to specify the job class for the Librarian Transaction Server. It is strongly recommended to use one of the dynamic job classes. It is, however, also possible to run the Librarian Transaction Server in one of the static partitions.

If no classes are defined default class P will be used. If a class P partition is also not available, the request is denied and a message is returned to the workstation user.

L=library_name
specifies the name of the VSE library to be accessed (1-7 alphameric characters). If this option is omitted, `PRIMARY` is assumed.

MN=member_name
either this or the MT= option **must** be specified to obtain a directory list of sublibrary members. MN=* lists all members. MN=ABC* lists all members whose name starts with ABC. The list can be further refined with the MT= option.

Note: Be aware that this option is invalid if it is preceded by an S= option with a generic (*) sublibrary name.

MT=member_type
either this or the MN= option **must** be specified to obtain a directory list of sublibrary members. MT=* lists members of any type. MT=ABC* lists all members whose type starts with ABC.

Note: Be aware that this option is invalid if it is preceded by an S= option with a generic (*) sublibrary name.

S=sublibrary_name
when neither MN= nor MT= is specified, S= specifies the name of the sublibrary whose directory entry is to be displayed.

S=ABC* displays the directory entry of all sublibraries whose name starts with ABC.

S=* displays directory entries of all sublibraries of the respective library. This is the same when the S= option is omitted.

When the MN= or MT= option is specified, S= defines the sublibrary whose members are to be listed. In this case, generic (*) specification of the sublibrary name is not allowed.

The library to be accessed is defined with the L= option.

Executing VSE Librarian Commands

Overview

This service function allows you to send one or more VSE Librarian commands to the host for execution. It returns any list and/or punch output produced by the command(s) to the workstation. It uses the command interface provided by the Librarian Transaction Server.

The Librarian commands are specified in the comments area of the RECEIVE file transfer command in the same “format” as described in *VSE/ESA System Control Statements*. Commands are separated by semicolons. A semicolon after the **last** command is **optional**. For example, you may enter:

```
RECEIVE PWSFILE ( FILE=LCMD ) ACC S=PRIMARY.HUGO; DEL MY*.TEST;
```

Note that library and sublibrary information must be specified in the comments area. There are **no** defaults as with the LIB and LDIR versions of the RECEIVE command.

The maximum length of the command string is 100 characters.

If more than 100 characters are specified, only the first 100 characters are sent to the Librarian server, and Librarian diagnostics will be returned in the workstation file. If several Librarian commands are specified which alternatively produce list and punch output, the output is returned in a single workstation member in the sequence received from the Librarian Transaction Server. In such a case it may be difficult to relate the output to the corresponding command. It is, therefore, recommended to submit commands which produce list or punch output in separate RECEIVE commands.

If commands are submitted which require external resources (external to the workstation/Librarian Transaction Server environment) such as magnetic tapes for BACKUP or VSAM cluster definitions for DEFINE LIBRARY, it is the user’s responsibility to make these resources available before the commands are submitted.

The RECEIVE command syntax is shown in Figure 71 on page 255.

Examples

About this Section: This section provides various examples of how you may use the RECEIVE command with option FILE=LCMD.

An example starts with a short heading marked by the '►' character.

The following examples show how to:
- Lock/unlock members
- List the library directory
- List the sublibrary directory
- Define a sublibrary
- Delete a sublibrary
- Delete library members
- Rename a sublibrary
- Rename a library member
- Copy a sublibrary
- Copy library members between sublibraries
- Browse a phase

► **Lock/Unlock:**

```
RECEIVE  DUMMY  ( FILE=LCMD )  ACC  S=PRIMARY.$$C; LOCK  MYMEM.JOB  LOCKID=JONES

Locks member MYMEM.JOB in sublibrary PRIMARY.$$C against write-access by other users.
Only user JONES can update the member.

RECEIVE  DUMMY  ( FILE=LCMD )  ACC  S=PRIMARY.$$C; UNLOCK  MYMEM.JOB  LOCKID=JONES

Removes the lock from member MYMEM.JOB.
```

► **List Library Directory:**

```
RECEIVE  PRD2.DIR  ( FILE=LCMD )  LD L=PRD2  OUT=STATUS;

Receives a directory list of all sublibraries in library PRD2 to workstation file PRD2.DIR.
```

► **List Sublibrary Directory:**

```
RECEIVE  CONFIG.DIR  ( FILE=LCMD )  ACC S=PRD2.CONFIG; LD *.*;

Receives a directory list of all members in sublibrary CONFIG.DIR to workstation file PRD2.DIR.

RECEIVE  CONFIG.DIR  ( FILE=LCMD )  LD S=PRD2.CONFIG;

Same as above; in addition, sublibrary status information is returned.
```

► **Define Sublibrary:**

```
RECEIVE  TEST.DEF  ( FILE=LCMD )  DEFINE S=PRIMARY.TEST;

Defines a new sublibrary TEST in the PRIMARY library.
```

- **Delete Sublibrary:**

```
RECEIVE  TEST.DEL  ( FILE=LCMD )  DEL S=PRIMARY.TEST;
```

Deletes sublibrary TEST and all its members from the PRIMARY library.

- **Delete Members:**

```
RECEIVE  MEM.DEL  ( FILE=LCMD )  ACC S=PRIMARY.TEST; DEL *.TEST1;
```

Deletes all members with membertype TEST1 from PRIMARY.TEST.

```
RECEIVE  MEM.DEL  ( FILE=LCMD )  ACC S=PRIMARY.TEST; DEL TEST.A;
                                                       DEL ABCD.PROC; DEL HUGO.PHASE;
```

Deletes selected members from PRIMARY.TEST.

- **Rename Sublibrary:**

```
RECEIVE  SLIB.REN  ( FILE=LCMD )  REN S=PRIMARY.OLD   PRIMARY.NEW;
```

Renames sublibrary PRIMARY.OLD to PRIMARY.NEW

- **Rename Member:**

```
RECEIVE  MEM.REN  ( FILE=LCMD )  ACC S=MY.LIB; REN MEM.OLD   MEM.NEW;
```

Renames member MEM.OLD to MEM.NEW in sublibrary MY.LIB.

- **Copy Sublibrary:**

```
RECEIVE  SLIB.COP  ( FILE=LCMD )  COPY S=PRIMARY.TEST   PRIMARY.TEST1
```

Creates a new sublibrary PRIMARY.TEST1 and copies the contents from PRIMARY.TEST to it.

- **Copy Member between Sublibraries:**

```
RECEIVE  MEM.COP  ( FILE=LCMD )  CONN S=PRIMARY.TEST   PRIMARY.TEST1; COPY MEMBER.A
```

Copies member MEMBER.A from sublibrary PRIMARY.TEST to PRIMARY.TEST1.

- **Browse a Phase:**

```
RECEIVE  PHASE.BRS ( FILE=LCMD ) ACC S=MY.LIB; LIST TEST.PHASE
```

Lists the contents of phase TEST in hex and character string format.

This command should be used by the workstation front-end when the BROWSE function is selected for a library member which contains string data (record format B).

Command Output Description

The following table describes the output of this service function which may be used as input by a workstation application program.

Table 9. FILE=LCMD: Send a Librarian Command for Execution to the Host - Command Output

Whenever one or more Librarian commands are submitted to the host with a RECEIVE command, feedback from the Librarian is returned to the workstation: (1) in the file transfer message and (2) in the workstation file. **Message**: INW0085I Librarian command(s) successfully submitted - MAXRC=nnnn where MAXRC is the maximum return code from the Librarian. A return code of zero indicates that all commands were executed successfully. A non-zero return code means that there was a problem with at least one of the submitted commands. **Workstation file**: Contains the list output and/or messages produced by each command in exactly the same format as it is written on SYSLST when the commands are submitted in a batch job. Examples follow.
Command (define sublibrary): RECEIVE TEST.DEF (FILE=LCMD INW1) DEFINE S=PRIMARY.TST **Message**: INW0001I Librarian command(s) successfully submitted - MAXRC=0000 **Workstation file contents**: DEFINE S=PRIMARY.TST L113I RETURN CODE OF DEFINE IS 0
Command (delete members): RECEIVE DEL (FILE=LCMD INW1) ACC S=MY.LIB; DEL XXX.A; DEL YYY.A **Message**: INW0085I Librarian command(s) successfully submitted - MAXRC=0004 **Workstation file contents**: ACC S=MY.LIB L113I RETURN CODE OF ACCESS IS 0 DEL XXX.A L113I RETURN CODE OF DELETE IS 0 DEL YYY.A L082I MEMBER YYY.A NOT FOUND IN SUBLIBRARY MY.LIB L113I RETURN CODE OF DELETE IS 4

Command Syntax and Options RECEIVE (FILE=LCMD

Command Syntax

The following shows the RECEIVE command format for sending Librarian commands to the host for execution. In case you are not familiar with the notation of the following syntax diagram, refer to “How to Read SEND/RECEIVE Syntax Diagrams”, on page 273.

Command Syntax and Options RECEIVE (FILE=LCMD

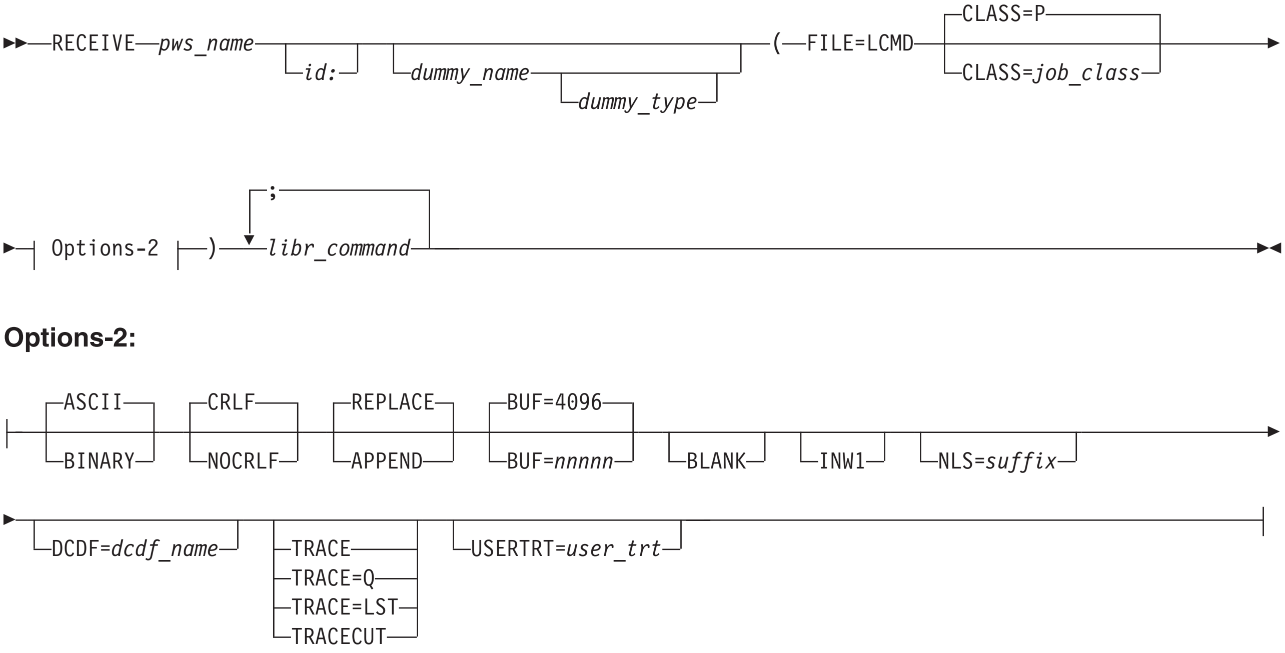


Figure 71. Send a Librarian Command for Execution to the Host - FILE=LCMD:

Command Options

The following describes the parameters and options that have specific meaning for sending Librarian commands to the host for execution.

For a description of the **pws_name** and **id:** parameters, refer to “Common SEND/RECEIVE Command Syntax” on page 70.

For a description of the following **common options**, refer to “Common SEND/RECEIVE Command Options” on page 72.

```
ASCII/BINARY
CRLF/NOCRLF
APPEND/REPLACE
BLANK
INW1
NLS=
DCDF=
TRACE
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=
```

dummy_name
a host specification is not applicable for this function. However, some 3270 Emulation Programs (such as FTTERM) require a hostname. In that case, specify any name. It will be ignored by the host.

dummy_type
has no meaning and is ignored (only included for consistency of command syntax).

Command Syntax and Options RECEIVE (FILE=LCMD

FILE=LCMD
send any VSE Librarian command(s) from the workstation to the host for execution and return the command output, if any, to the workstation.

options
in alphabetic order:

BUF=nnnnn
defines the size of the receive buffer used to receive the command output. The **default** buffer size is 4096 bytes.

When transferring very large amounts of data, performance may be improved by increasing the buffer size. **nnnnn** can be a value from **512** to **32767**.

A reduction of the buffer size may be necessary when there is not enough main storage available to allocate the default buffer.

CLASS=job_class
allows to specify the job class for the Librarian Transaction Server. It is strongly recommended to use one of the dynamic job classes. It is, however, also possible to run the Librarian Transaction Server in one of the static partitions.

If the CLASS= option is omitted, the server will be started in one of the partition classes defined in the dialog *Customize VSE/ESA Workstation Platform* (see "Defining Partition Classes Via the Interactive Interface" on page 92). If no classes are defined default class P will be used. If a class P partition is also not available, the request is denied and a message is returned to the workstation user.

libr_command
enter Librarian commands separated by semicolons (**;**). For the commands, you can use a **maximum size** of **100** characters.

The commands must not contain any colons (:). Use blanks instead of colons.

Using File Transfer Service Functions for VSE/POWER

This section contains information about the following service functions:

- Executing a VSE/POWER command at the host.
- Receiving job completion messages from VSE/POWER.

Executing a VSE/POWER Command

Overview

This service function allows you to send a VSE/POWER command to the host for execution and receive any output produced by the command to the workstation. Only one command can be submitted at a time.

The VSE/POWER command is specified in the comments area of the RECEIVE file transfer command in the same "format" as described in *VSE/POWER Administration and Operation*.

For example, you may enter:

```
RECEIVE PWSFILE ( FILE=PCMD ) PDISPLAY LST
```

After successful execution of the RECEIVE command, PWSFILE contains a display of the VSE/POWER LST queue in exactly the same format as the display would appear on the console.

The commands accepted by VSE/POWER via this function are shown in the box below:

```
  PALTER jobname
  PBRDCST
  PCANCEL jobname
  PDELETE jobname | MSG
  PDISPLAY jobname | MSG | A | T | PNET | DYNC
* PDISPLAY TAPE | VIO
* PFLUSH DEV
* PGO DEV
  PHOLD jobname
  PINQUIRE
  PLOAD DYNC
  PRELEASE jobname
* PRESTART DEV
* PSETUP DEV
* PSTART DEV
* PSTOP  DEV
* PVARY  DYNC
  PXMIT node-id
* PXMIT DEV
```

Note: Commands marked with an asterisk (*) only can be issued by a user with administrator authority who has unlimited access to all VSE/POWER queue entries.

The returned command output is described in Table 10 on page 259.

When this function is used to display the contents of the VSE/POWER queues, the queue entries are normally returned in variable format, in the same way as they would appear at the system console. If required (for example, if the queue entries are to be redisplayed by a program at the workstation), a formatted display of the queue entries can be requested by specifying the option **FORMAT**.

The format of formatted queue display records is shown in Table 11 on page 259.

The format of the RECEIVE command is shown in Figure 72 on page 260.

Examples

About this Section: This section provides various examples of how you may use the RECEIVE command with option FILE=PCMD.

An example starts with a short heading marked by the '►' character.

Executing a VSE/POWER Command

▸ **Release a Job:**

To release a job for execution which is held in the RDR queue with disposition H or L:

```
RECEIVE  POW.MSG  ( FILE=PCMD )  PRELEASE  RDR,MYJOB,2356
         or
RECEIVE  POW.MSG  ( FILE=PCMD )  R  RDR,MYJOB,2356
```

• **Explanation:**

If the command is successfully executed by VSE/POWER, message **INW0084I OK** is issued and workstation file POW.MSG is empty.

If the command is executed by VSE/POWER but produces an error, message **INW0001I File transfer complete** is issued, and POW.MSG contains the error message returned by VSE/POWER.

Any other message indicates an error during file transfer or VSE/POWER access.

▸ **Display Queue Entries (Variable Format):**

To display VSE/POWER RDR queue entries:

```
RECEIVE  POW.RDR  ( FILE=PCMD )  PDISPLAY  RDR
         or
RECEIVE  POW.RDR  ( FILE=PCMD )  D  RDR
```

• **Explanation:**

If the command is successfully executed by VSE/POWER, message **INW0001I File transfer complete** is issued and workstation file POW.RDR contains the queue display records:

```
1R46I  READER QUEUE   P D C S  CARDS
1R46I  INSABEND 00021 9 L 0       18 FROM=(SYSA)
1R46I  DTRMIGR  00093 5 D A      354 FROM=(VATR)
1R46I  PTR      00034 3 K Q        6 FROM=(SYSA)
```

If the command is executed by VSE/POWER but produces an error, message **INW0001I File transfer complete** is issued, and POW.RDR contains the error message returned by VSE/POWER.

If the queue is empty or no entries matching the specified parameters are found, message **INW0076I No entries found in VSE/POWER queue** is issued, and POW.RDR is empty.

Entries whose names start with $SPL are VSE/POWER internal entries and can be ignored. For example:

```
1R46I  $SPL1672 01672 3 *    TO=(USER1) FROM=(USER1)
```

▸ **Display Queue Entries (Formatted):**

To display RDR queue entries to be processed by a workstation program:

```
RECEIVE  POW.RDR  ( FILE=PCMD  FORMAT )  PDISPLAY  RDR
         or
RECEIVE  POW.RDR  ( FILE=PCMD  FORMAT )  D  RDR
```

Formatted output is described in Table 11 on page 259.

Command Output Description

The following table describes the output of this service function which may be used as input by a workstation application program.

Table 10. Service Function - Execute a VSE/POWER Command - FILE=PCMD - Command Output. When a VSE/POWER command is submitted for execution at the host, feedback from VSE/POWER is returned in the file transfer message and/or the receiving workstation file, as follows:

Command	Message	Workstation File Contents
PDISPLAY	INW0001I	*If display was successful:* VSE/POWER queue display records
	INW0001I	*If syntax error in VSE/POWER command:* 1R98I xxxx INVALID VSE POWER COMMAND or 1R52I D OPERAND xx MISSING OR INVALID
	INW0076I	*No matching entries found:* empty (byte count zero)
	all others	*File transfer or VSE/POWER error:* empty (byte count zero)
PALTER, PDELETE or others	INW0084I	*If function executed successfully:* empty (byte count zero)
	INW0001I	*If entry not found:* 1R88I NOTHING TO DELETE | ALTER
	INW0001I	*If syntax error or restricted command:* 1R98I xxxx INVALID VSE POWER COMMAND or 1R52I D OPERAND xx MISSING OR INVALID or 1R85I xx COMMAND NOT ALLOWED ...
	all others	*File transfer or VSE/POWER error:* empty (byte count zero)
Note: The returned VSE/POWER messages are shown for the PDISPLAY, PALTER, and PDELETE commands. Other commands may return other messages.		

Table 11. Formatted Display of VSE/POWER Queue Entries - FILE=PCMD FORMAT:

Position	Description	Format
1	Queue identifier: R L P	Fixed-1
2-9	Jobname	Var-8
10-14	Jobnumber	Var-5
15	Priority	Fixed-1
16	Disposition	Fixed-1
17	Class	Fixed-1
18	System ID (shared spooling)	Fixed-1
19-21	Number of copies	Var-3
22-31	Number of records spooled	Var-10
32-41	Number of pages spooled	Var-10
42-51	Number of lines/cards spooled	Var-10
52-59	Target destination node: TO=NODE	Var-8
60-67	Target destination user: TO=(USER)	Var-8
68-75	Originating node name: FROM=NODE	Var-8
76-83	Originating user: FROM=(USER)	Var-8
84-86	Job suffix (segment) number	Var-3
87-102	User information	Var-16
103-110	Creation date: MM/DD/YY	Fixed-8

Table 11. Formatted Display of VSE/POWER Queue Entries - FILE=PCMD FORMAT: (continued)

Position	Description	Format
111-115	Next due date: MM/DD	Fixed-5
116-120	Next due time: hh:mm	Fixed-5
121-124	Record format: ASA, BMS, 3270, etc.	Var-4
125-128	Flash identifier (3800)	Var-4
129-136	Forms identifier	Var-8
137-139	Number of separator pages/cards	Var-3
140-143	Compaction table name	Var-4

Command Syntax and Options RECEIVE (FILE=PCMD

RECEIVE Command Syntax

The following shows the RECEIVE command format for sending a VSE/POWER command to the host and receiving command output (if any). In case you are not familiar with the notation of the following syntax diagram, refer to “How to Read SEND/RECEIVE Syntax Diagrams”, on page 273.

```
►►──RECEIVE──pws_name──┬─────┬─┬───────────────────────────────┬──────────────────────────►
                       └─id:─┘ └─dummy_name──┬────────────┬──(──FILE=PCMD──┬────────┬─┘
                                             └─dummy_type─┘                └─FORMAT─┘

►──┤ Options-2 ├──)──power_command──────────────────────────────────────────────────────►◄
```

Options-2:

```
   ┌─ASCII──┐  ┌─CRLF───┐  ┌─REPLACE─┐  ┌─BUF=4096──┐
├──┼────────┼──┼────────┼──┼─────────┼──┼───────────┼──┬───────┬──┬──────┬──┬────────────┬──►
   └─BINARY─┘  └─NOCRLF─┘  └─APPEND──┘  └─BUF=nnnnn─┘  └─BLANK─┘  └─INW1─┘  └─NLS=suffix─┘

►──┬────────────────┬──┬───────────┬──┬────────────────────┬─────────────────────────────┤
   └─DCDF=dcdf_name─┘  ├─TRACE─────┤  └─USERTRT=user_trt─┘
                       ├─TRACE=Q───┤
                       ├─TRACE=LST─┤
                       └─TRACECUT──┘
```

Figure 72. Send a VSE/POWER Command for Execution to the Host - FILE=PCMD

RECEIVE Command Options

The following describes RECEIVE parameters and options for VSE/POWER command execution.

For a description of the **pws_name** and **id:** parameters, refer to “Common SEND/RECEIVE Command Syntax” on page 70.

For a description of the following **common options**, refer to “Common SEND/RECEIVE Command Options” on page 72.

ASCII/BINARY
CRLF/NOCRLF
APPEND/REPLACE

BLANK
INW1
NLS=
DCDF=
TRACE
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=

dummy_name
a host specification is not applicable for this function. However, some 3270 Emulation Programs (such as FTTERM) require a hostname to be specified. In that case, specify any name. It will be ignored by the host.

dummy_type
has no meaning and is ignored (only included for consistency of command syntax).

FILE=PCMD
send a VSE/POWER command to the host and receive command output

options
in alphabetic order:

BUF=nnnnn
defines the size of the receive buffer used to receive data from VSE/POWER. The **default** buffer size is **4096 bytes**.

When transferring very large amounts of data, performance may be improved by increasing the buffer size. **nnnnn** can be a value from **512** to **32767**.

A reduction of the buffer size may be necessary when there is not enough main storage available to allocate the default buffer.

FORMAT
if specified together with a 'PDISPLAY queue' command, returns a fixed-format display of the contents of a VSE/POWER queue.

If FORMAT is **not** specified, queue displays are returned in the same format as they are shown on the system console.

When FORMAT is specified together with a command other than 'PDISPLAY queue' it is ignored.

power_command
a VSE/POWER command.

Receiving Job Completion Messages from VSE/POWER

Overview

When jobs are submitted from the workstation to the VSE host, feedback is required on the job status. Since the SEND/RECEIVE protocol does not allow to send messages originating at the host to the workstation, a mechanism is provided by VSE/POWER which allows workstation users to retrieve job completion messages from the host to the workstation.

Message Collection

Job Completion Messages from VSE/POWER

Whenever a job is submitted by a workstation user with the MSG option (default) in the SEND command, the resulting job completion message is saved by VSE/POWER for this user.

- Messages are saved for all jobs executed on the local system, as well as for jobs executed on a remote node or on another system in a shared spooling complex.
- Messages are also saved for jobs generated by a ’mother’ job by means of the DISP=I parameter. The job completion messages of the ’child’ jobs and any ’grandchild’ jobs will contain the original job number of the ’mother’ job.
- By default, a maximum of 20 messages will be queued for each user in wrap-around mode. That is, when the queue is full, old messages are discarded to save new messages. The fact that messages were discarded is recorded in the VSE/POWER Statistics Status Report.
- The maximum number of messages to be saved for each user can be specified in the JCMQ=nn parameter of the SET autostart statement.
- Messages not retrieved are lost when VSE/POWER is shut down.

Message Retrieval

When a job is submitted, the jobname and jobnumber are returned in the file transfer message, for example:

```
INW0079I  Job TEST 1234 successfully submitted. RC=0000
```

The resulting job completion message can be retrieved with the command:

```
RECEIVE  TESTJOB.MSG TEST 1234 ( FILE=PMSG
```

After it has been received, the message is deleted from the VSE/POWER message queue.

Message retrieval can specify:

- Retrieve all completion messages for all jobs submitted by the requesting user (not supported for 3270 emulators that require a host filename to be specified).
- Retrieve the completion messages for all jobs with a specific jobname (same jobname, different jobnumbers).
- Retrieve the completion message(s) for a specific job identified by jobname and jobnumber. If no message exists for this job, the system will check the status of the job in the RDR queue and return job status information in the file transfer message.

Job completion messages are returned in a fixed format for easy processing by a workstation application program.

The returned messages may look like this:

```
JOB TEST     01234 EXECUTED NODE=LOCAL    DATE=07/25/91 TIME=11:32:01 MAXRC=0000 LASTRC=0000

JOB TEST     01235 EXECUTED NODE=LOCAL    DATE=07/25/91 TIME=11:32:55 MAXRC=0000 LASTRC=0000 ORG=01234

JOB TEST0012 01236 CANCELED NODE=VSEESA02 DATE=07/25/91 TIME=13:41:15 MAXRC=0016 LASTRC=0004
```

The message layout is shown Table 13 on page 264.

The RECEIVE command format is described in Figure 73 on page 265.

Examples

About this Section: This section provides various examples of how you may use the RECEIVE command with option FILE=PMSG.

An example starts with a short heading marked by the '►' character.

► **Retrieve Messages Using Command Defaults:**

```
RECEIVE  ALL.MSG  ( FILE=PMSG
```

receives all job completion messages created by the jobs submitted by the requesting user to workstation file ALL.MSG.

```
Message: INW0001I File transfer complete
```

```
If no messages found: INW0080I No job completion messages found
```

Note: This function is **not** supported for 3270 emulators that require a host filename to be specified.

► **Retrieve All Messages for Specific Jobname:**

Assume you have submitted a job TEST which used DISP=I parameter to generate a new job. The generated job also has the name TEST but a different jobnumber.

```
RECEIVE  TEST.MSG  TEST  ( FILE=PMSG
```

receives the job completion messages for the jobs named TEST:

```
JOB TEST     01234 EXECUTED NODE=LOCAL    DATE= ... MAXRC=0000 LASTRC=0000
JOB TEST     01235 EXECUTED NODE=LOCAL    DATE= ... MAXRC=0000 LASTRC=0000 ORG=01234
```

Note that the message for the generated job TEST 01235 contains the original jobnumber 01234.

► **Retrieve Message for Specific Job:**

```
RECEIVE  TEST.MSG  TEST  1235  ( FILE=PMSG
```

receives the job completion message for job TEST 1235.

If no message is found for the specified job, the system will check the RDR queue to determine the job status and return message:

```
   INW0081I  No job completion message found. Job status = xxxx
```

The Job Status will indicate whether the job is executing, dispatchable, held, not found, etc. (Refer to Table 12 on page 264.)

► **Retrieve Messages of Another User:**

Only the system administrator (type-1 user) can retrieve job completion messages created by another user. User SYSA, for example, can issue the command:

```
RECEIVE  JONES.MSG  ( FILE=PMSG  USER=JONES
```

to receive all messages created by jobs submitted by user JONES. In this case the messages are **not** deleted.

Command Output Description

The following table describes the output of this service function which may be used as input by a workstation application program.

Job Completion Messages from VSE/POWER

Table 12. Service Function - Receive Job Completion Messages - FILE=PMSG - File Transfer Messages

When **messages** are received: INW0001I File transfer complete When **no messages** found for user or jobname: INW0080I No job completion messages found When **no messages** found for specific job: INW0081I No job completion messages found. Job status = code where ***code*** is:

```
0002 - job with specified name and number and owned
       by requesting user (or USER= ) not found
0004 - job is running
0006 - job not dispatchable (not disposition D or K)
0008 - job class disabled or not defined
0010 - job class busy (no free partition)
0012 - job not running although dispatchable
       and job class available
0014 - job is 'in preparation' and should be
       running shortly
```

All **other messages** indicate an error during file transfer or VSE/POWER access.

Table 13. Formatted Job Completion Messages - FILE=PMSG:

Position	Description	Format
1-3	JOB	Fixed-3
4	blank	
5-12	Jobname	Var-8
13	blank	
14-18	Jobnumber	Var-5
19	blank	
20-27	EXECUTED or CANCELED	Fixed-8
28	blank	
29-33	NODE=	Fixed-5
34-41	Nodename where job was executed; default is LOCAL	Var-8
42	blank	
43-47	DATE=	Fixed-5
48-55	Job completion date: mm/dd/yy	Fixed-8
56	blank	
57-61	TIME=	Fixed-5
62-69	Job completion time: hh:mm:ss	Fixed-8
70	blank	
71-76	MAXRC=	Fixed-6
77-80	Maximum return code	Fixed-4
81	blank	
82-88	LASTRC=	Fixed-7
89-92	Last return code	Fixed-4
93	blank	
94-97	ORG= (blank if not applicable)	Fixed-4
98-102	Original job number, if job was generated with DISP=I. Otherwise blank.	Var-5

Command Syntax and Options RECEIVE (FILE=PMSG

RECEIVE Command Syntax

The following shows the RECEIVE command format for ’job completed’ message retrieval from VSE/POWER. In case you are not familiar with the notation of the following syntax diagram, refer to “How to Read SEND/RECEIVE Syntax Diagrams”, on page 273.

```
►►──RECEIVE──pws_name──┬─────┬──┬──────────────────────────────┬──(──FILE=PMSG──────────────►
                       └─id:─┘  └─entry_name──┬──────────────┬─┘
                                              └─entry_number─┘

   ┌─USER=logged_on_user─┐
►──┼─────────────────────┼──┤ Options-2 ├──┬───┬──────────────────────────────────────────►◄
   └─USER=user_id────────┘                 └─)─┘
```

Options-2:

```
   ┌─ASCII──┐   ┌─CRLF───┐   ┌─REPLACE─┐   ┌─BUF=4096──┐
├──┼────────┼───┼────────┼───┼─────────┼───┼───────────┼──┬───────┬──┬──────┬──┬────────────┬──►
   └─BINARY─┘   └─NOCRLF─┘   └─APPEND──┘   └─BUF=nnnnn─┘  └─BLANK─┘  └─INW1─┘  └─NLS=suffix─┘

►──┬───────────────┬──┬─────────────┬──┬──────────────────┬───────────────────────────────────┤
   └─DCDF=dcdf_name┘  ├─TRACE───────┤  └─USERTRT=user_trt─┘
                      ├─TRACE=Q─────┤
                      ├─TRACE=LST───┤
                      └─TRACECUT────┘
```

Figure 73. Receive Job Completion Messages - FILE=PMSG

RECEIVE Command Options

The following describes RECEIVE parameters and options for ’job completed’ message retrieval from VSE/POWER

For a description of the **pws_name** and **id:** parameters, refer to “Common SEND/RECEIVE Command Syntax” on page 70.

For a description of the following **common options**, refer to “Common SEND/RECEIVE Command Options” on page 72.

ASCII/BINARY
CRLF/NOCRLF
APPEND/REPLACE
BLANK
INW1
NLS=
DCDF=
TRACE
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=

entry_name

the name of the job for which job completion messages are to be retrieved. All messages saved for jobs with the specified name are returned. If you specify '*', all messages saved for the requesting user are returned.

Default: If no entry_name is specified, all messages saved for the requesting user are returned (not supported for 3270 emulators that require a host filename to be specified).

entry_number

specifies the number of the job for which the job completion message is to be retrieved. If you specified '*' as entry name, the entry number is ignored.

FILE=PMSG

receive job completion messages

options

in alphabetic order:

BUF=nnnnn

defines the size of the receive buffer used to receive data from VSE/POWER. The **default** buffer size is 4096 bytes.

USER=user_id

job completion messages are saved for the user who submitted the jobs and can only be retrieved by the same user signed-on with the same user ID. Only a system administrator (type-1 user) can retrieve messages created by another user by specifying the USER= option. In this case, the messages will **not** be deleted after they have been received.

Receiving Host User Profile

To enable a workstation front-end program to control a user's access to VSE machines and resources, profile information can be downloaded to the workstation. Do **not** confuse this host user profile with the user profile that is defined under VSE/ESA Interactive Interface.

The host user profile includes some data from the VSE/ESA Interactive Interface user profile. In addition, the host user profile provides other information about host items. The entire information of the host user profile is shown in the figure below.

Has user **signed** on to II **?**
Terminal **ID**
User **ID**
Is **security** active **?**
Does user have general **access to VSE/POWER?**
Does user have general **access to the VSE libraries ?**
Can user **manage all VSE/POWER queues ?**
Can user **submit jobs ?**
VSE **system ID** (user defined**,** to identify VSE machine to workstation)
User type (1,2 or 3)
Dynamic/static **partition classes** for submission of jobs generated by the workstation interface
Dynamic partition classes for the Librarian Transaction Server
Host language
Names of VSE libraries that can be accessed from a workstation
version of VSE/ESA system
CPU ID
Is Host Transfer File **(HTF)** defined **?**

Note: Be aware that this profile information does **not** serve any security purposes at the workstation. It simply enables a workstation front-end program to offer a user only those host objects to which he has access.

Examples

About this Section

The following section provide examples of how you may use the RECEIVE command with option FILE=PROF.

An example starts with a short heading marked by the '►' character.

► **Download a Host User Profile:**

```
During the signon procedure, the workstation may issue a RECEIVE command
to download the profile information, for example:
   RECEIVE JONES.PRF (FILE=PROF
```

• **Explanation:**

```
- The file transfer program will collect the information at the host and return
  (for each item of information) one logical record in the specified workstation file.

- Each record has an identifier starting in column one and the information itself
  (where applicable) starting in column ten of the record.

- If a record is not present, the corresponding function is not supported or
  the corresponding information was not supplied by the system administrator.
```

An example of the profile records is shown in the box below:

```
column 1  column 10
▼         ▼

IUI
TRMID     A005
UID       JONES
SEC
LIBACC
POWACC
POWALL
SUBMIT
SYSID     VSE100
UTYP      1
SCLASS    PC
JCLASS    D
HLANG     E
LIB       IJSYSRSPRD1   PRD2   PRIMARY
VER       01.03.00
CPUID     FFAF11F03090
HTF
END
```

For users not signed on to VSE/ESA's Interactive Interface, only a subset of this information will be available.

An example of such a profile record provided is shown in the box below:

Receiving Host User Profile

```
column 1  column 10
▼         ▼

TRMID     A005
UID       NOIIA005
SEC
LIBACC
POWACC
POWALL
SUBMIT
SCLASS    P
JCLASS    A
VER       01.03..00
CPUID     FFAF11F03090
HTF
END
```

A detailed description of the returned profile information is given in Table 14.

The format of the RECEIVE command to receive profile information is shown in Figure 74 on page 270.

Command Output Description

The following table describes the output of this service function which may be used as input by a workstation application program.

Table 14. Format of Host User Profile Records - FILE=PROF. Each profile item is returned as one logical record with an identifier starting in column 1 and the actual information starting in column 10. **Status** indicates: A = always present, O = optional, IUI = only when signon to IUI.

Column 1	Column 10	Format	If present, means	Status
IUI	blank	--	User has signed on to IUI.	IUI
APPC	User ID	Var-8	Connected via LU6.2.	O
TRMID	CICS terminal ID	Fixed-4	see Column 10	A
UID	User ID of signed-on user	Var-8	see Column 10	A
SEC	blank	--	Security (SEC=YES) is active.	O
LIBACC	blank	--	User has general access authority to VSE libraries.	O
POWACC	blank	--	User has general access authority to VSE/POWER queues.	O
POWALL	blank	--	User can manage all batch queues (defined by the dialog *Maintain User Profile*).	O
SUBMIT	blank	--	User can submit jobs to VSE/POWER (defined by the dialog *Maintain User Profile*).	O
SYSID	Unique identification of the VSE system.	Var-8	see Column 10	IUI O
UTYP	User type (1,2, or 3) as defined by the dialog *Maintain User Profile*.	Fixed-1	see Column 10	IUI A

Table 14. Format of Host User Profile Records - FILE=PROF (continued). Each profile item is returned as one logical record with an identifier starting in column 1 and the actual information starting in column 10. **Status** indicates: A = always present, O = optional, IUI = only when signon to IUI.

Column 1	Column 10	Format	If present, means	Status
SCLASS	Job class(es) for the Librarian Transaction Server (defined by the dialog *Customize VSE/ESA Workstation Platform*). For example, A or AB or ABC. When no job classes were defined, default class P is used.	Var-4	see Column 10	A
JCLASS	Job class for user job submission. When no job class is defined, default class A is used.	Fixed-1	see Column 10	A
HLANG	Host language. E=English G=German, etc.	Fixed-1	see Column 10	IUI A
LIB	Library names. Up to twenty libraries can exist. When no libraries are defined, default library PRIMARY is inserted if under IUI. Otherwise there is no library record.	Fix-7[...]	see column 10	IUI O
VER	Version of the VSE/ESA system, such as 01.03.00	Fixed-8	see Column 10	A
CPUID	the CPU ID	Fixed-12	see Column 10	A
HTF	blank	--	The Host Transfer File is defined	O
END	blank	--	Indicates the end of profile records	A

Command Syntax and Options RECEIVE (FILE=PROF

In case you are not familiar with the notation of the following syntax diagram, refer to "How to Read SEND/RECEIVE Syntax Diagrams", on page 273.

```
►►──RECEIVE──pws_name──┬─────┬──┬───────────────────────────────────────────┬──┤ Options-2 ├──┬───┬──────►◄
                       └─id:─┘  └─dummy_name──┬────────────┬──(──FILE=PROF──┘                 └─)─┘
                                              └─dummy_type─┘
```

Options-2:

```
   ┌─ASCII──┐   ┌─CRLF───┐   ┌─REPLACE─┐
├──┼────────┼───┼────────┼───┼─────────┼──┬───────┬──┬──────┬──┬────────────────┬──┬────────────┬──────►
   └─BINARY─┘   └─NOCRLF─┘   └─APPEND──┘  └─BLANK─┘  └─INW1─┘  └─DCDF=dcdf_name─┘  └─NLS=suffix─┘

►──┬────────────┬──┬──────────────────┬──────────────────────────────────────────────────────────────┤
   ├─TRACE──────┤  └─USERTRT=user_trt─┘
   ├─TRACE=Q────┤
   ├─TRACE=LST──┤
   └─TRACECUT───┘
```

Figure 74. Service Function - Receive Host User Profile - FILE=PROF

For a description of the **pws_name** and **id:** parameters, refer to "Common SEND/RECEIVE Command Syntax" on page 70.

For a description of the following **common options**, refer to "Common SEND/RECEIVE Command Options" on page 72.

ASCII/BINARY
CRLF/NOCRLF
APPEND/REPLACE
BLANK
INW1
DCDF=
NLS=
TRACE
TRACE=Q
TRACE=LST
TRACECUT
USERTRT=

Part 3. Appendixes

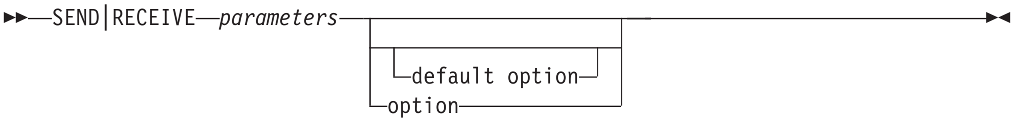
Appendix. How to Read SEND/RECEIVE Syntax Diagrams

SEND/RECEIVE syntax diagrams show you how to enter a SEND or RECEIVE command from the workstation’s command interface for performing any kind of file transfer between workstation and host.

SEND/RECEIVE syntax diagrams are written in a special notation, known as railroad tracks. The following paragraphs teach you how to read this notation.

Basic Reading Rules

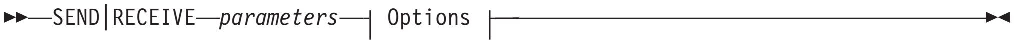
Read the syntax diagram from left to right and from top to bottom. Do this by following the horizontal line, that is, the main path.



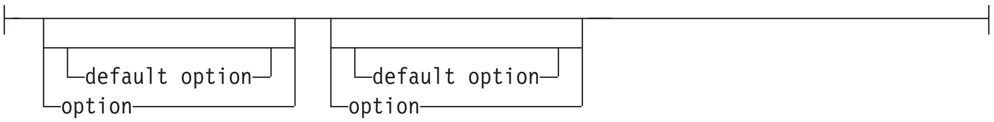
The start of the command syntax is indicated by ’►►’; the end of the command syntax is marked by ’►◄’.

An arrowhead ’►’ means that the command syntax is continued. A line starting with an ’►’ means that the syntax is continued from the previous line.

The main path may contain a reference that leads to additional syntax. The referenced item appears on the main path like this: ’| reference name |’. The **options** that may be specified in SEND/RECEIVE commands are referenced this way. Their syntax is continued separately enclosed by two ’|’.



Options:



The SEND/RECEIVE parameters appear **on**, **below** and **above** the main path. A parameter’s place determines its significance:

- **Required** parameters are shown exactly **on** the main path.
- **Optional** parameters are indicated through a by-pass **below** the current main path.
- **Default** values are shown **above** the current main path.

Parameters: Keywords and Variables

Parameters are classified as keywords or variables:

- Keywords are displayed in uppercase letters and must be entered exactly as shown. For example, the SEND or RECEIVE command name is a keyword.
- Variables appear in lowercase letters and stand for names or values that you supply. For example, the workstation’s file to be sent to or received from the host (pws_name) is a variable.

Note that a parameter can be a combination of a keyword and a variable. Enter all punctuation exactly as shown in the diagram, such as colons ′:′, semicolons ′;′, commas ′,′, quotation marks, equal signs ′=′ and open and closed parenthesis ′( )′.

Entering Parameters in the Proper Sequence

As basic rule, enter the SEND/RECEIVE parameters in the same sequence as shown in the diagrams. The **options**, however, may be specified in any order.

Specifying a Required Choice

A stack of parameters that starts **on** the main path and extends downward means a set of alternatives one of which you must specify. For an example see the receive function from VSE/POWER (Figure 31 on page 130) which shows the required keyword stack consisting of RDR, LST, PUN.

```
►►──RECEIVE──parameters──FILE=──┬─RDR─┬──────────────────────────────────────────►◄
                                ├─LST─┤
                                └─PUN─┘
```

Specifying an Optional Choice

A stack of parameters **below** the main path indicates that one parameter **may** be specified. For an example see the optional keyword stack consisting of TRACE, TRACE=Q, TRACE=LST, TRACECUT in the send/receive functions.

```
►►────┬─────────┬──┬────────────┬────────────────────────────────────────────────►◄
      │         │  │            │
      ├─option1─┤  ├─TRACE──────┤
      ├─option2─┤  ├─TRACE=Q────┤
      └─option3─┘  ├─TRACE=LST──┤
                   └─TRACECUT───┘
```

Specifying a Sequence More Than Once

An arrow below or above a path that returns to a previous point means the sequence of items included by the arrow can be specified more than once.

In the Host Transfer File syntax diagram, this is the case for the variable *user_id* belonging to the FOR= keyword. Every newly added *user_id* has to be separated from the next one by a comma (′,′) as indicated on the return arrow (see Figure 48 on page 181).

Options:

```
├──┬──────────────────┬──────────────────────────────────────────────────┤
   │      ┌─,───────┐ │
   │      ▼         │ │
   └─FOR=───user_id─┴─┘
```

Glossary

If you do not find the term you are looking for, refer to the index of this book or to the IBM *Dictionary of Computing* , New York: McGraw-Hill, 1994.

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 symbol (A) after 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 sections of the vocabularies are identified by symbol (I) after definition; definitions from draft international standards, draft proposals, and working papers in development by the ISO/TC97/SC1 vocabulary subcommittee are identified by symbol (T) after definition, indicating final agreement has not yet been reached among participating members.

*** abend.** 1. Abnormal end of task. 2. Synonym for abnormal termination.

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 Control Logging and Reporting. An IBM licensed program to log all attempts of access to protected data and to print selected formatted reports on such attempts.

access method. A program, that is, a set of commands (macros), to define files or addresses and to move data to and from them; for example VSE/VSAM or VTAM.

application program. A program written for or by a user that applies directly to the user's work. See also batch program and online application program.

*** ASCII (American National Standard Code for Information Interchange).** The standard code, using a coded character set consisting of 7 bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.(A)

assemble. To translate a program from assembler language into object code.

assembler. A computer program used to assemble. Synonymous with assembly program.

assembler language. A programming language whose instructions are usually in one-to-one correspondence with machine instructions and allows to write macros.

*** background partition.** An area of virtual storage in which programs are executed under control of the system. By default, the partition has a processing priority lower than any of the existing foreground partitions.

*** backup copy.** A copy, usually of a file or a library member, that is kept in case the original file or library member is unintentionally changed or destroyed.

batch processing. 1. Serial processing of computer programs. 2. Pertaining to the technique of processing a set of computer programs in such a way that each is completed before the next program of the set is started. (A)

batch program. A program that is processed in series with other programs and therefore normally processes data without user interaction.

byte. Eight adjacent binary digits that are operated upon as a unit and that constitute the smallest addressable unit of information within a computer system. Normally, it represents a stored character.

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 VSE/VSAM catalog.

*** cataloged procedure.** A set of control statements placed in a library and retrievable by name.

Glossary

central processing unit (CPU). The hardware component that interprets and executes instructions. Synonym for processor.

CICS Transaction Server for VSE/ESA. This is the successor system to CICS/VSE.

*** CICS/VSE (Customer Information Control System).** An IBM program that lets transactions entered at remote terminals be processed concurrently by application programs. It includes facilities for building, using, and maintaining data bases. Its latest version is called CICS Transaction Server for VSE/ESA.

*** CKD (Count-Key-Data) disk device.** A disk device that stores data in the format: count field, usually followed by a key field, followed by the actual data of a record. The count field contains, among others, the disk address of the stored record. Storage is allocated by tracks and cylinders. Contrast with FBA disk device. See also extended-count-key data device.

COBOL (Common Business-Oriented Language). A high-level programming language based on English used primarily for business application programs.

Common User Access (CUA) architecture. Guidelines for the dialog between a human and a workstation or terminal. One of the three SAA architectural areas.

compile. To translate a source program into an executable program (object program). See also assembler.

compiler. A program used to compile.

*** component.** 1. Hardware or software that is part of a computer system. 2. A functional part of an operating system, for example: job control program, VSE/POWER.

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.

count-key-data (CKD) device. A disk device that stores data in the record format: count field, key field, data field. The count field contains, among others, the address of the record in the format: cylinder, head (track), record number and the length of the data field. The key field, if present, contains the record's key or search argument. CKD disk space is allocated by tracks and cylinders. Contrast with FBA disk device. See also extended-count-key data device.

CPU (Central Processing Unit). The hardware component that interprets and executes instructions. Synonym for processor.

CUA (Common User Access) architecture. Guidelines for the dialog between a human and a workstation or terminal. One of the three SAA architectural areas.

Customer Information Control System (CICS). An IBM program that controls online communication between terminal users and a data base; transactions entered at remote terminals are processed concurrently by user-written application programs. The product includes facilities for building, using, and servicing data bases. Its latest version is called CICS Transaction Server for VSE/ESA.

CUT mode (Control Unit Terminal mode). IBM PCs operating in CUT mode make not use of the extended data stream facility. Contrast with DFT mode.

DASD. Direct access storage device, normally a disk device.

data base. A set of data available online that is organized by a common system and used for a common purpose.

data communication. The transmission and reception of data.(A)

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 utility (DITTO).** An IBM program that provides file to file services for card I/O, tape, and disk devices.

Data Language/I (DL/I). A data base access language used with CICS/VSE.

data management. A major function of the operating system. It involves organizing, storing, locating, and retrieving data.

data protection. See access control.

data set. See file.

DBCS (Double Byte Character Set). A character set which allows Korean, Japanese, and Chinese languages to be internally represented by two bytes per character.

DCDF (Data Conversion Descriptor File). With a DCDF you can convert individual fields within a record during data transfer between a PC and its host. The DCDF defines the record fields of a particular file for both, the PC and the host environment.

DFT mode (Distributed Function Terminal mode). IBM PCs operating in DFT mode make use of the extended data stream facility allowing simultaneous host sessions by using screen windows.

dialog. 1. In an interactive system, a series of related inquiries and responses similar to a conversation between two people. 2. For VSE/ESA, a set of panels that can be used to complete a specific task, defining a file, for example.

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.

* **distribution tape.** A magnetic tape that contains, for example, a preconfigured operating system like VSE/ESA. This tape is shipped to the customer for system installation.

* **DITTO/ESA for VSE (Data Interfile Transfer, Testing and Operations) utility.** An IBM program that provides file to file services for card I/O, tape, and disk devices.

DL/I (Data Language/I). A data base access language used with CICS/VSE.

DOSRES (disk operating system residence volume). The disk volume on which the system sublibrary IJSYSRS.SYSLIB is located including the programs and procedures required for system startup.

double-byte character set (DBCS). A character set which allows Korean, Japanese, and Chinese languages to be internally represented by two bytes per character.

dump. 1. Data that has been dumped.(I) (A) 2. To write at a particular moment some contents of storage to another data medium for the purpose of safeguarding or debugging the data.(T)

dynamic partition. A partition created and activated on an ’as needed’ basis. After processing, the occupied space is released. Contrast with static partition.

EBCDIC (Extended Binary-Coded Decimal Interchange Code). A coded character set consisting of 8-bit coded characters.

ECKD (extended count-key-data) device. A disk storage device that has a data transfer rate faster than some processors can utilize. A specialized channel program is needed to convert ordinary CKD channel programs for use with an ECKD device.

EHLLAPI (Emulator High Level Language Application Program Interface). An application program interface available with most IBM 3270 emulation programs for IBM personal computers.

emulation. The use of programming techniques and special machine features that permit a computer system to execute programs written for another system or for the use of I/O devices different from those that are available.

end user. 1. A person who makes use of an application program. 2. In SNA, the ultimate source or destination of user data flowing through an SNA network. May be an application program or a terminal operator.

ESDS (entry sequenced data set). An entry-sequenced file under VSE/VSAM. Its records are loaded without respect to their contents and whose relative byte addresses cannot change. Records are retrieved and stored by addressed access, and new records are added to the end of the file. See also SAM ESDS.

extended count-key-data (ECKD) device. A disk storage device that has a data transfer rate faster than some processors can utilize. A specialized channel program is needed to convert ordinary CKD channel programs for use with an ECKD device.

extended recovery facility (XRF). In VSE/ESA, a feature of CICS/VSE that provides for enhanced availability of CICS by offering one CICS system as a backup of another.

Fast Service Upgrade (FSU). A service function of VSE/ESA for the installation of a refresh release without regenerating control information such as library control tables.

FBA (Fixed Block Architecture) disk device. A disk device that stores data in blocks of fixed size. These blocks are addressed by block number relative to the beginning of the file.

file. A named set of records stored or processed as a unit.(T) Synonymous with data set.

* **foreground partition.** A space of virtual storage in which programs are executed under control of the system. By default, a foreground partition has a higher processing priority than the background partition.

FULIST (FUnction LIST). A type of selection panel that displays a set of files and/or functions for the choice of the user.

GB. Gigabyte.

generate. To produce a computer program by selecting subsets of standardized code under the control of parameters.(A)

gigabyte (GB). 1024 MB of storage (see MB). One gigabyte equals 1,073,741,824 bytes.

help panel. Information displayed by a system in response to a user’s help request.

host mode. In this operating mode, a PC can access a VSE host and the Move Utilities of VSE/ESA can be used.

host processor. 1. In a network, the processor in which the access method for the network resides. 2. In an SNA network, the processor that contains an SSCP.

host transfer file (HTF). Used by the Workstation File Transfer Support of VSE/ESA as an intermediate storage area for files that are sent to and from IBM Personal Computers.

ICCF. See VSE/ICCF.

index. In data management, a table used to locate the records of a file.

initial program load (IPL). The process of loading system programs and preparing the system to run jobs.

intelligent workstation. See programmable workstation.

interactive. A characteristic of a program or system which 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.

IPL (Initial Program Load). The process of loading system programs and preparing the system to run jobs.

JCL (Job Control Language). A language that serves to prepare a job or each job step of a job to be run. Some of its functions are: to determine the I/O devices to be used, set switches for program use, log (or print) its own statements, fetch the first phase of each job step.

JECL (Job Entry Control Language). A control language that allows the programmer to specify how VSE/POWER should handle a job.

job. One program or a group of related programs called job steps complete with the JCL statements necessary for a particular run. A job is identified in the job stream by a JOB statement followed by one EXEC statement for each of the programs or job steps.

job control statement. A particular statement of JCL.

job stream. The sequence of jobs as submitted to an operating system.

*** KB (kilobyte).** 1024 bytes of storage.

Kanji. A set of symbols used in Japanese ideographic printing. Every symbol is represented by two bytes.

key. In VSE/VSAM, one or several characters taken from a certain field (key field) in data records for identification and sequence of index entries or of the records themselves.

key sequence. The collating sequence either of records themselves or of their keys in the index or both. The key-sequence is alphanumeric.

KSDS (Key-Sequenced Data Set). A VSE/VSAM file whose records are loaded in key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in the file in key sequence.

label. 1. An identification record for a tape, disk, or diskette volume or for a file on such a volume. 2. In assembler programming, a named instruction generally used for branching.

label information area. An area on a disk to store label information read from job control statements or commands. Synonymous with label area.

language translator. A general term for any assembler, compiler, or other routine that accepts statements in one language and produces equivalent statements in another language.

*** librarian.** The set of programs that maintains, services, and organizes the system and private libraries.

library. See VSE library and VSE/ICCF library.

*** library directory.** The index that enables the system to locate a certain sublibrary of the accessed library.

*** library member.** The smallest unit of data to be stored in and retrieved from a sublibrary.

linkage editor. A program to build a phase (executable code) from one or several independently translated object modules or existing phases or both. In creating the phase, the program resolves cross references among the modules and phases available as input. The program can catalog the newly built phases.

link-edit. To create a loadable computer program by having the linkage editor process compiled (assembled) source programs.

*** MB (megabyte).** One megabyte equals 1,024 KB or 1,048,576 bytes

*** member.** The smallest unit of data that can be stored in and retrieved from a sublibrary.

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.

* **migrate.** To move to a changed operating environment, usually to a new release or version of a system.

mode. A method of operation. See also ESA mode.

MSHP (Maintain System History Program). A program used for automating and controlling various installation, tailoring, and service activities for a VSE system.

network. 1. An arrangement of nodes (data stations) and connecting branches. 2. The assembly of equipment through which connections are made between data stations.

networking. Making use of the services of a network environment.

node. 1. In SNA, an end point of a link or junction common to several 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, 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 code.** Output from a compiler or assembler which is itself executable machine code or is suitable for processing to produce executable machine code.(A)

object module (program). A program unit that is the output of an assembler or compiler and is input to a linkage editor.

online application program. An interactive program used at display stations. When active, it waits for data. Once input arrives, it processes it and sends a response to the display station or to another device.

online processing. Processing by which the input data enters the computer directly from a display station and the output data is transmitted directly to the display station.

* **operating system.** Software that controls the running of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management.(I) (A)

* **operator command.** A statement to a control program, issued via a console or terminal. It causes the control program to provide requested information, alter normal operations, initiate new operations, or end existing operations.

* **operator console.** A display console used for communication between the operator and the system.

optional program. An IBM program that a user can install under VSE/ESA by way of available installation-assist support.

OS/2 (Operating System/2). An IBM operating system that runs on IBM personal computers with appropriate micro processors.

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 static partition and dynamic partition.

* **password.** In computer security, a string of characters known to the computer system and a user. He must specify it to gain full or limited access to the system and to the data stored in it.

PC (personal computer). A microcomputer for individuals or small businesses.

* **phase.** The smallest unit of executable code that can be loaded into virtual storage.

PL/I. A programming language designed for use in a wide range of commercial and scientific computer applications.

PNET. Networking support available with VSE/POWER; it provides for the transmission of selected jobs, operator commands, messages, and program output between the nodes of a network.

POWER. See VSE/POWER.

* **pregenerated operating system.** An operating system such as VSE/ESA that is shipped by IBM mainly in object code. IBM defines such key characteristics as the size of the main control program, the organization and size of libraries, and required system areas on disk. The customer does not have to generate an operating system.

Print Services Facility/VSE. An access method that provides support for the advanced function printers.

* **private library.** A user-owned library that is separate and distinct from the system library.

procedure. See cataloged procedure

Glossary

* **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. The hardware component that interprets and executes instructions. (I) (A)

profile. A description of the characteristics of a user or a computer resource.

program. 1. To design, write, and test computer programs. (I) (A) 2. A sequence of instructions suitable for processing by a computer.

programmable workstation (PWS). A workstation that has some degree of processing capability and that allows the user to change its functions. Contrast with nonprogrammable workstation.

PTF (program temporary fix). A solution or by-pass of one or more problems documented in APARs. PTFs are distributed to IBM customers for preventive service to a current release of a program.

PWS (programmable workstation). A workstation that has some degree of processing capability and that allows the user to change its functions. Contrast with nonprogrammable workstation.

* **queue.** 1. A line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a network. 2. To arrange in, or form, a queue.

queue file. A disk file maintained by VSE/POWER that holds control information for the spooling of job input and job output.

reader. An input device that reads data that is punched into cards or is written or printed on paper.

real storage. See processor storage.

* **record.** A collection of related data or words, treated as a unit. See logical record, physical record.

refresh release. An upgraded VSE/ESA system with the latest level of maintenance for a release.

restore. To write back on disk data that was previously written from disk to an intermediate storage medium such as tape.

RPG II. A commercially oriented programming language suitable for writing application programs that meet common business data processing requirements.

RRDS (Relative Record Data Set). A VSE/VSAM file whose records are loaded into fixed-length slots and represented by the relative-record numbers of the slots they occupy.

* **run.** 1. A performance of one or more jobs. 2. A performance of one or more programs. 3. To cause a program or job to be performed.

SAA (Systems Application Architecture). A set of IBM software interfaces, conventions, and protocols that provide a framework for designing and developing applications that are consistent across systems.

SAA (Systems Application Architecture) architectures. The Common User Access architecture, the Common Programming Interface, and the Common Communications Support.

SAM (sequential access method). A data access method that writes to and reads from an I/O device record after record (or block after block). On request, the support performs device control operations such as line spacing or page ejects on a printer or skip a certain number of tape marks on a tape drive.

SAM ESDS file. A SAM file managed in VSE/VSAM space, so it can be accessed by both SAM and VSE/VSAM macros.

SDLC (Synchronous Data Link Control). A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or non-switched links. The configuration of the link connection may be point-to-point, multipoint, or loop.

security. See access control.

* **selection panel.** A displayed list of items from which a user can make a selection. Synonymous with menu.

* **shared virtual area (SVA).** In VSE, a high address area that contains the System Directory List (SDL) of frequently used phases, resident programs shared between partitions, and an area for system support.

* **shutdown.** The process of ending operation of a system or a subsystem, following a defined procedure.

* **sign-on.** 1. The procedure to be followed at a terminal or a workstation to establish a link to a computer. 2. To begin a session at a workstation.

skeleton. A set of control statements and/or instructions that requires user-specific information to be inserted before it can be submitted for processing as a job.

source member. A library member containing source statements in any of the programming languages supported by VSE.

* **source program.** A computer program expressed in a source language.(I) (A) Contrast with object module.

source statement. A statement written in symbols of a programming language.

spooling. The use of disk storage as buffer storage to reduce processing delays when transferring data between peripheral equipment and the processor of a computer. In VSE, this is done under the control of VSE/POWER.

stand-alone program. A program that runs independently of (not controlled by) the VSE system.

* **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.

start up. 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. A device, or part of a device, that can retain data. See also processor storage and virtual storage.

storage dump. See dump.

sublibrary. In VSE, 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.

* **subsystem.** A secondary or subordinate system or program, usually capable of operating independently of, or asynchronously with, the operating system.

SVA (shared virtual area). In VSE, a high address area that contains a list (SDL) of frequently used phases, resident programs shard between partitions, and an area for system support.

Synchronous Data Link Control (SDLC). A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or non-switched links. The configuration of the link connection may be point-to-point, multipoint, or loop.

Systems Application Architecture (SAA). A set of IBM software interfaces, conventions, and protocols that provide a framework for designing and developing applications that are consistent across systems.

system sublibrary. The sublibrary 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 station with a keyboard.

trace. 1. To record a series of events as they occur. 2. A record of specified events during the run of a program. 3. A program to produce such a record.

transaction. 1. In a batch or remote batch entry, a job or job step. 2. In CICS/VSE, an application program (or 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.

* **utility program.** 1. A 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 that performs an everyday task such as copying data from one storage device to another.

volume. A data carrier that is mounted and demounted as a unit, for example, a reel of tape or a disk pack. Some disk units have no demountable packs. In that case, a volume is the portion available to one read/write mechanism.

VRDS (Variable-length Relative-record Data Set). A VSE/VSAM relative record data set with variable-length records. See also RRDS.

VSAM. See VSE/VSAM.

VSE (Virtual Storage Extended). A system that consists of a basic operating system and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware it controls form a complete computing system. Its current version is called VSE/ESA.

VSE/Advanced Functions. Part of VSE Central Functions a base program of VSE/ESA. for a VSE-controlled installation.

VSE/ESA (Virtual Storage Extended/Enterprise Systems Architecture). The most advanced VSE system currently available.

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/SP Unique Code. Part of VSE Control Functions a base program of VSE/ESA.

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.

VSE/VSAM managed space. User-defined space on disk under the control of VSE/VSAM.

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).

VTOC (volume table of contents). A table on a disk volume identifying and describing the files stored on it.

workstation. See programmable and nonprogrammable workstation.

Workstation File Transfer Support. Enables the exchange of data between IBM Personal Computers (PCs) linked to a VSE/ESA host system where the data is kept in intermediate storage. PC users can retrieve that data and work with it independently of VSE/ESA.

* **X.21.** In data communication, a recommendation of the CCITT that defines the interface between data terminal equipment and public data networks for digital leased and circuit switched synchronous services.

* **X.25.** In data communication, a recommendation of the CCITT that defines the interface between data terminal equipment and packet switching networks.

Index

File Number: S370/S390-34
Program Number: 5690-VSE

Printed in U.S.A.

SC33-6709-01

Spine information:

IBM VSE/ESA Programming and Workstation Guide Version 2 Release 7 SC33-6709-01