IBM z/VSE

Guide for Solving Problems

Version 5

SC34-2615-00

IBM z/VSE

Guide for Solving Problems

Version 5

SC34-2615-00

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

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

This edition replaces SC34-2605-00.

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

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

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

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

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

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

Contents

Figures

Tables

Notices

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

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

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

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

```
IBM Deutschland GmbH
Dept. M358
IBM-Allee 1
71139 Ehningen
Germany
```

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

Trademarks

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

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

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

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

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

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

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

Accessibility

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

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

Using Assistive Technologies

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

Documentation Format

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

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

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

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

About This Book

This manual is intended to help you analyze and solve problems that may occur when you have installed IBM® z/Virtual Storage Extended (z/VSE).

The main subject of this manual is software errors. Basic information on hardware errors is included.

The manual helps you to:

- Recognize a particular type of error.
- Collect and interpret the available information.
- Identify the actions that are necessary to remove the error.

By following the guidelines presented in this manual, you should be able to diagnose and correct most error situations yourself; that is, without calling IBM for help.

Who Should Use This Book

This manual addresses primarily the **system administrator**. Note, however, that any of the following persons may be the first to encounter a problem:

- The system console operator.
- A display station user, including the system administrator.
- An application programmer.
- An application end user.

Most problems, however, will end up with the administrator. Whenever an application program seems to be at fault, the administrator may hand the problem over to the programmer responsible.

How to Use This Book

If you have an error situation and want to do error diagnosis, go directly to “Hardware Errors” on page 18 or “Software Errors” on page 19.

Refer to Chapter 1, “Introduction,” on page 1 for additional details on how to use the manual.

Where to Find More Information

Most problems become apparent through a message. Therefore, the manual *z/VSE Messages and Codes* must be available for error diagnosis. Most of these messages are also available online at a z/VSE user display station.

There are other manuals of the z/VSE library you may have to consult from time to time. Manuals of z/VSE base programs, such as VSE/POWER or CICS®, are referred to in the text when needed for a particular error situation.

For more information on the programs and statements mentioned in this book refer to these publications:

- *z/VSE Planning*

- *z/VSE Guide to System Functions*

z/VSE Home Page

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

You can find the z/VSE home page at

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

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

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

Summary of Changes

For a complete overview of the functions which are new with z/VSE 5.2, refer to the *z/VSE Release Guide*.

Chapter 1. Introduction

If you have an error situation and want to do error diagnosis, go directly to "Hardware Errors" on page 18, or "Software Errors" on page 19.

This chapter first describes users for whom this manual was written. This is followed by a reading chart that shows how to get familiar with the manual. The remainder of the chapter deals with the errors that may occur and what the system provides to assist you in error diagnosis.

Scope of the Manual

This manual discusses mainly z/VSE and its major base programs under the aspect of error diagnosis and problem determination.

The major base programs discussed are:

- VSE/POWER
- CICS
- TCP/IP
- VTAM®
- VSE/ICCF

TCP/IP for VSE/ESA: This manual covers only a small subset of the commands and utilities relating to the z/VSE base program TCP/IP. For further description of the error diagnosis and problem determination aspects of TCP/IP, refer to the manual *z/VSE TCP/IP Support*.

How to Read this Manual

The following reading chart helps you to understand the structure of the manual. It tells you what to read and for which purpose. To become familiar with the manual, follow this chart. The person responsible for error diagnosis should be familiar with the **complete** manual.

YOU ARE LOOKING FOR INTRODUCTORY INFORMATION:

READ:

```
Preface:
Chapter 1. Introduction
Chapter 2. Error Messages
Chapter 3. Starting Error Diagnosis
```

YOU HAVE AN ERROR SITUATION:

START WITH:

```
Chapter 3. Starting Error Diagnosis
```

This leads you to one of the following chapters:

```
Chapter 4. Diagnosis Procedures
Chapter 5. Online Analysis of CICS Transaction
           Server Abends
Chapter 6. Using Dumps for Error Diagnosis
Chapter 7. Handling Resource Constraint Conditions
```

The other chapters and appendixes are referred to when needed:

```
Chapter 8.  Identifying the Problem Area
Chapter 9.  Useful Commands and Utilities
Chapter 10. Using Traces
Chapter 11. Diagnosis Procedures for More Experienced
            Users
Appendix A. Problem Log and Change Log
Appendix B. Problem Symptom String and APAR
            Submission
Appendix C. List of IPL, JCL, and Label Procedures
Appendix D. Error Messages When Using Virtual Tapes
Appendix E. Problems When Using SCSI Disk Devices
```

Figure 1. How to Read this Manual

Recognizing Errors

Following is an overview of how an error situation can be recognized.

A Message is Displayed

Messages reflect the status of your system and most errors are indicated by an error message. Refer to Chapter 2, “Error Messages,” on page 11 for details about messages.

A Program is Canceled

The system may cancel a program before processing is finished. The reason may be either a logic error in the program or an attempt to process incorrect data. When a program is canceled, the system issues an error message.

Incorrect Output is Produced

After an application program has finished its execution, the output may be different from what you expected or intended. The cause could be the incorrect specification of an input file, a logic error in a program, or a defective device.

An Unusual System or Device Status Occurs

A number of errors cause symptoms which are more difficult to interpret. For example:

- Unusual pattern of system control lights.
- No system activity.
- No data can be entered from a terminal.

These symptoms can be caused, for example, by a looping program or a wait condition. Depending on the complexity of a given system, it requires system experience to analyze such symptoms and locate the source of the error.

Error Causes

Hardware Errors

A hardware error can be as simple as a device that has unintentionally been set into a NOT READY status. It could also be a defective I/O device, or a power failure.

This manual is mainly concerned with software errors. It gives some basic information on hardware to enable you to distinguish between hardware and software errors. For hardware errors refer to the hardware documentation of the device in error.

Refer also to the description of the WAIT and LOOP condition in the section “Software Errors” which follows.

Software Errors

Examples of software errors are:

- A program has an error in its logic.
- A program contains incorrect data definitions.
- A program wants to process a file that is not defined to the system.
- An entry is missing in a system table.

Because of a software error a program may come to an abnormal end ('**abend**' for short) and may be canceled. A software error may also cause **incorrect output**, a **loop** or a **wait** condition.

LOOP

A loop in a program is any repetitive execution of a sequence of instructions. Frequently, repetitive execution of a part of a program is intentional. Sometimes, however, a program incorrectly executes a sequence of instructions over and over again. The cause is, in most cases, a logic error; usually in an application program or sometimes in an IBM-supplied program. In exceptional cases a hardware malfunction may also be the cause of a loop.

WAIT

There are several kinds of waits and not all of them are error conditions. When the system (processor) stops processing, the processor is in a wait state. For example, the processor waits for an answer from an I/O device, or there is nothing to process because no program is active. Note, however, that the system wait state does not indicate the wait status of a partition.

Sometimes, an error in an application program, in a control program, or a device causes the processor to stop. Such a condition is a **hard wait** if the processor is unable to recover. Only a new startup (IPL) will reactivate the system. In another error situation the system will resume operation after the cause for the wait (for example, a device was not ready) has been removed. This type of wait is a **soft wait**.

Note: Errors often arise when the system setup is changed; devices are added or deleted, or new programs or applications are installed. Therefore, changes to the system should be carefully planned and documented. Refer to “Change Log” on page 178 for further details on documenting system changes.

Error Handling at Your Installation

Reporting Errors

Successful problem resolution depends above all on documenting all system changes and all problems that occur. A problem should be described in as much detail as possible. For documenting problems and system changes see also Appendix A, “Problem Log and Change Log,” on page 177.

To keep the impact of errors to a minimum, it is necessary to establish communication channels for error reporting. An end user at a terminal usually does not have the knowledge or expertise to solve a problem. He or she should know exactly whom to contact in case of an error or a problem.

User Tasks and Responsibilities

Someone at your installation must be responsible for handling problem situations. Usually, this person will be the system administrator. The system administrator should act as the:

- Focal point for any problems and errors.
- Interface to IBM if help is required.

For example, you (the system administrator) must determine whether an error was caused by hardware or software. If it is a software error, determine the component: a **user program** or **IBM code**. If it is an error in a user program, you may hand over the problem to the owner of the program.

Make it a rule at your installation to save all available information related to a problem. This helps you to solve a problem more quickly. Have this information also available if there is a need to contact IBM. There are a few error situations that require that you contact IBM right away. This manual tells you when to do so.

Diagnostic Information Provided by the System

Besides messages, further information is provided by the system for error diagnosis.

Online Information

If a program (transaction) running under the CICS Transaction Server ends abnormally, error information is saved in a file. You can analyze this data at your terminal via the Interactive Interface of z/VSE. This enables you to determine the cause of the error without shutting down the CICS Transaction Server subsystem. For a detailed description refer to Chapter 5, “Online Analysis of CICS Transaction Abends,” on page 57.

z/VSE saves a number of messages in a message log file. You can access these messages online and use them for error diagnosis. For further details refer to “Reviewing the Message Log File” on page 14.

Whenever a transaction ends abnormally, the CICS Transaction Server creates a transaction dump. Refer to “Handling CICS Transaction Dumps” on page 100 for how to use a transaction dump for error analysis.

Storage Dumps

On certain occasions, such as a program abend, the system writes the contents of storage (a dump) to the dump library.

You can analyze a dump stored in the dump library via the Interactive Interface of z/VSE. For a detailed description refer to Chapter 6, “Using Dumps for Error Diagnosis,” on page 71.

Tools for Collecting Diagnostic Information

Utilities

To collect diagnostic information, various utilities are available. For example:

- Display table of contents (VTOC) of a disk volume.
- Take a stand-alone dump.
- Display VSE/VSAM catalog information.
- Print or display dump information.
- Show file definitions using LSERV.
- Print a VSE/VSAM file.
- Print the system console hardcopy file.
- Print the hardware recorder file.

As a z/VSE user you can access many of these utilities via the Interactive Interface of z/VSE. Refer to Chapter 8, “Identifying the Problem Area,” on page 113 for details.

Traces

Traces are available to support you in error diagnosis. This manual provides information on the following traces:

- Interactive Trace Program
- SDAID Event Traces
- Trace Facility of VSE/POWER RJE/BSC and PNET
- Execution Diagnostic Facility (EDF) of the CICS Transaction Server
- Trace Facility of the CICS Transaction Server
- VTAM Traces.

For further details on these traces refer to Chapter 10, “Using Traces,” on page 141.

System Startup Modes Available

z/VSE provides three **special** system startup modes for handling specific system status or problem situations. These startup modes are:

- COLD
- BASIC
- MINI

The operator may, in the IPL load parameter, request that startup processing is interrupted to allow for selecting a COLD, BASIC, or MINI startup.

z/VSE determines the startup mode of the *CICS Transaction Server* by using the CICS global and local catalogs. However, you can initiate a cold start of the CICS Transaction Server by deleting and then redefining and initializing these global and local catalogs. For details of how to do so, refer to the skeleton SKCICCLD stored in ICCF Library 59.

The special startup modes are described in detail in the following section.

Startup Mode: COLD

If a COLD startup is requested, z/VSE reformats the VSE/POWER queues and restores the label area information. A COLD startup is necessary, for example, if you wish to move the VSE/POWER data file to a different disk.

During a COLD startup, all jobs and job output in the VSE/POWER queues are erased. There are two ways to save and reload these jobs:

- Using the utility program DTRIINIT.

  DTRIINIT loads cataloged jobs into the VSE/POWER reader queue. During a COLD startup, z/VSE activates program DTRIINIT. DTRIINIT uses a load list to reload jobs automatically from a VSE library into the VSE/POWER reader queue. To reload your own jobs, use skeleton SKCOLD to add your job names to the load list (before you perform a COLD startup). The manual *z/VSE Administration* describes skeleton SKCOLD in detail.

  Any job you want DTRIINIT to load into the VSE/POWER reader queue must be cataloged in a VSE library.
- Using the POFFLOAD command.

  The operator can use the POFFLOAD command to save the jobs and job outputs of all VSE/POWER queues on tape. With the same command they can be reloaded after startup has been completed.

The manual *z/VSE Operation* has details on how to use the POFFLOAD command.

Startup Mode: BASIC

The BASIC startup gives you a “basic” system with the original system tables and definitions active. **All your system modifications are ignored**. The following partitions are active after a BASIC startup has been completed:

BG (available)
F1 (VSE/POWER)
F2 (CICS Transaction Server and VSE/ICCF)
F3 (VTAM, if used)
F4 (available)
F5 (available)
F7 (TCP/IP)
FB (Basic Security Manager)

In addition to the system library (IJSYSRS.SYSLIB) and the dump library (SYSDUMP.xx) the following library definitions are active:

- For the F2 partition:

  PRD1.BASE, PRD2.PROD, PRD2.SCEEBASE, PRD2.DBASE
- For all other partitions:

  PRD1. BASE

During a BASIC startup you are requested to specify up to three terminal addresses. These terminals are activated and allow you to correct an error interactively.

In general, a BASIC startup should be selected if the normal startup does not function because of an erroneous system condition.

Such a condition may be caused when adding new terminals to your system, for example. To add new terminals, you use the *Configure Hardware* dialog and the following might happen:

- You enter incorrect CICS definitions.
- While using the dialog you get conditions like VSE/VSAM space or VTAM buffer space exhausted.

The resulting system status would prevent you from starting up the system normally. With startup mode BASIC you get a running system and can correct the error.

As initial terminal, you can also use the TCP/IP *Telnet* program. During basic startup, you are asked if TCP/IP is to be used and you can then specify the initial startup parameters.

Startup Mode: MINI

A MINI startup starts two partitions only, BG and F1 with VSE/POWER running.

In general, a MINI startup should be selected if severe problems inhibit the use of other startup modes. For example, if in the system library (IJSYSRS.SYSLIB) library members are missing or incorrect.

A MINI startup allows you to use the librarian program LIBR in the BG partition and perform recovery actions. You may use program LIBR to:

- Copy backup versions of library members from PRD2.SAVE (or any other VSE library) to IJSYSRS.SYSLIB.
- Restore library members from a backup tape of IJSYSRS.SYSLIB.
- If a library is shown to be defective, perform a LIBR TEST REPAIR=YES.

VSE library maintenance is another task for which you would request a MINI startup. For example, if you want to backup and restore a faulty library or reorganize a library via backup/restore. Deleting or moving a library or sublibrary are other tasks for which a MINI startup should be used. Note that library maintenance is only possible if the particular library object is not presently being used by another partition and is not included in an active LIBDEF chain. These are conditions that usually exist for a “MINI” system.

For library maintenance, you can enter librarian commands at the system console. The preferred method, however, is to have predefined job streams available in the VSE/POWER reader queue. You then simply release these jobs when library maintenance is required.

You may create such job streams for other tasks such as restoring files or libraries.

Dialogs Available for Problem Determination

Through the selection panel *Problem Handling*, z/VSE provides the following dialogs for problem determination:

- **Online Problem Determination**

  Refer to Chapter 5, “Online Analysis of CICS Transaction Abends,” on page 57 for details about this function.
- **Inspect Message Log**

  Refer to “Reviewing the Message Log File” on page 14 for details about this function.
- **Storage Dump Management**

  Refer to Chapter 6, “Using Dumps for Error Diagnosis,” on page 71 for details about this function.
- **Inspect Dump Management Output**

  Refer to Chapter 6, “Using Dumps for Error Diagnosis,” on page 71 for details about this function.
- **Retrace History File**

  Refer to “Printing the System History File” on page 121 for details about this function.
- **Dump Program Utilities**:
  - **Create Stand-Alone Dump Program on Tape/Disk**

Refer to "Creating the Stand-Alone Dump Program on Tape or Disk" on page 98 for details about this function.

- **Remove Stand-Alone Dump Program from Tape/Disk**

  Refer to "Removing the Stand-Alone Dump Program from Disk" on page 98 for details about this function.
- **Scan Dump Files on Tape/Disk**

  Refer to "Scanning the Dump Files on Tape or Disk" on page 99 for details about this function.
- **Format ICCF Data**

  From the *Dump Program Utilities* panel, press PF1 (Help). Then scroll forwards to obtain a brief description of this function.
- **Print SDAID Tape**

  Refer to "Printing and Analyzing SDAID Information" on page 142 for details about this function.

Model User Profile for Problem Determination

As shipped, z/VSE includes a model user profile for accessing a default panel hierarchy for problem determination. The user ID and password is **$SRV**. The password should be changed after initial installation to avoid unauthorized access to system functions.

The panel hierarchy offered is mainly intended for IBM personnel doing remote problem determination for a user site via a data link connecting the user installation with an IBM Support Center, for example. But the $SRV panel hierarchy can also be used for local problem determination.

The initial *z/VSE Function Selection* panel offers the following selections:

Problem Handling

System Console

Manage Batch Queues

Display Active Users/Send Message

Retrieve Message

A $SRV user is a type 2 user (can access VSE/ICCF but has no VSE/ICCF administrative authority) and is not authorized to escape to the CICS Transaction Server. In addition, a $SRV user can:

- Use the dialogs of the *Problem Handling* panel but cannot delete OLPD incidents when using the *Online Problem Determination* dialog.
- Access the z/VSE console but has restricted command authority.
- Look at the VSE/POWER queues and create jobs which produce output for the VSE/POWER list queue. Such jobs must have as destination operand (in the * $$ LST statement) the ID of the user or ANY. ANY indicates that this output is available to any user accessing it.
- Exchange messages with other users and delete messages from the message log.

You can also tailor the model user profile according to the needs of your installation. For details, refer to the chapter "Using the Interactive Interface and Skeletons" in the *z/VSE Administration* manual.

Chapter 2. Error Messages

Most error situations are indicated by a message. Therefore, messages must be read carefully. **Careful reading** of message explanations can save hours of time spent on error diagnosis.

An Error Message is Issued

Take for example the following error message:

```
L019I  INVALID SUBCOMMAND SPECIFICATION
```

This message was issued by the VSE Librarian program. The *z/VSE Messages and Codes* manual tells you that the subcommand specified for an UPDATE command has a syntax error. If necessary, consult the manual *z/VSE System Control Statements* for the correct syntax of the UPDATE subcommand.

You can also get the message explanation on the console by positioning the cursor to the message and using the PF9 (EXPL) key.

Other messages may point to a malfunctioning device. For example:

```
0P73I  I/O ERROR
```

The *z/VSE Messages and Codes* manual tells you why the job was canceled. The manual also suggests how to collect additional information about the device error.

Some I/O error messages show the device number of the failing device. This device number corresponds to the device number known by z/VSE (VSE address). To get the physical number of the failing device, issue a QUERY IO,cuu=<VSE address> command. For a description of the QUERY IO command see "Using the QUERY IO Command" on page 132. Also using PF11 (PCUU) on any 3 character VSE Address will show the 4 character physical device address.

Whenever you are confronted with an error message take the time necessary for a careful interpretation of the message.

On Which Devices Do Messages Appear?

Messages appear at three places:

1. The system console (SYSLOG).

   Messages at the system console reflect the ongoing activities of the system. All system console messages are recorded by the system in the **hardcopy** file. These are the messages documented in the *z/VSE Messages and Codes* manual.

   All messages that are directed to the z/VSE system console appear identically on any *master console*. The master console is established via the user ID that is logged on to the terminal. Through the *Console* dialog of the z/VSE Interactive Interface, a user with the proper authority establishes a master console. For simplicity, this manual uses the term *system console* to mean either the locally-attached system console or any other master console.

   As a display station user, you can use the *Console* dialog and display message explanations online. Refer to "Displaying Messages Online" on page 12 for details.

2. The system printer (SYSLST).

   For each batch job, the system prints start and stop times and, in between, all messages that are issued for the job. For example, if you use COBOL/VSE as programming language, the COBOL diagnostic messages issued during the compile run are printed on SYSLST.
3. A user display station.

   Messages issued to a display station by VSE/ICCF are not recorded. Therefore, if a VSE/ICCF error message (prefix K) appears on your screen you should always write it down, including the partition number and the name of the job or program. This simplifies error diagnosis later.

   When working with the Interactive Interface of z/VSE, you may get error messages that consist of plain text. These messages indicate severe errors like "temporary storage exhausted". For these messages, press the PF1 key to display help text describing the error. For other messages that are preceded by a code, refer to the Interactive Interface codes in the *z/VSE Messages and Codes* manual.

In addition, messages are saved in the:

- z/VSE message log file.

  These are messages that cannot be displayed immediately. If applicable, a display station user is notified that messages have been stored in the message log file. Refer to "Reviewing the Message Log File" on page 14 for further details.
- z/VSE dump library.
- CICS Online Problem Determination (OLPD) file.
- CICS transaction dump files.

For console automation, REXX provides you with the *REXX Console Application Framework* (REXXCO). For further information about REXXCO, refer to the *REXX/VSE Reference* manual.

Accessing Messages from a User Display Station

Displaying Messages Online

In z/VSE, the messages documented in *z/VSE Messages and Codes* can be displayed online at a user display station. You can do this by using the *Console* dialog. Select from the *Operations* panel:

```
1 (Console)
```

As a result, you get a full-screen display of a master console, just as it would appear on the real system console screen. To get a message explanation displayed, either type the message number into the input line (==>) and press the EXPLAIN PFkey (PF9, usually). Or, move the cursor under the message number and press the EXPLAIN PFkey.

You can display **librarian feedback codes**. The following keyword must be used:

```
VSELIBFC - LIBRARIAN FEEDBACK CODES
```

Here is an example of a message with a librarian feedback code:

```
L152I      ENTRY CONDITION FOR MODULE modulename IN PHASE phasename FAILED -
           FEEDBACK CODE =    nn

Explanation:  The indicated Librarian service is called with an
```

```
            incorrect or incomplete operand list. Feedback code nn is primarily
            intended for later problem determination by service personnel. For
            the meaning of the hexadecimal feedback codes, please refer to
            VSELIBFC (via EXPLAIN-key).
```

You can display **VSE/VSAM error and return codes**. The following keywords must be used:

```
VSAMOPEN  (for OPEN macro)
VSAMCLOS  (for CLOSE and TCLOSE macro)
VSAMREQU  (for request macros like GET and PUT)
VSAMXXCB  (for GENCB, MODCB, SHOWCB, and TESTCB macro)
VSAMRESN  (for catalog management requests)
VSAMSHOWCAT (for SHOWCAT return codes)
```

The following sample shows a portion of a message display with a VSE/VSAM error code. The error message may look as follows:

```
4228I   FILE HNS041 OPEN ERROR X'76' (118) CAT= ...
```

You get the following display after pressing the EXPLAIN PFkey:

```
4228I   FILE filename macro ERROR X'nn' (nnn) CAT= ...

        EXPLANATION: The ACB error flag was set to X'nn'
        (given in decimal notation in parentheses) during
        the indicated VSAM operation (execution of OPEN,
        CLOSE, or TCLOSE). See error codes; they are listed
        under "VSAMOPEN/VSAMCLOS/VSAMREQU/VSAMXXCB via EXPLAIN-key".
          .
          .
        If an error was detected within the catalog management
        routines ...
        See "VSAMRESN via EXPLAIN-key".
          .
          .
```

The explanation tells you how to proceed to display the description of the error code. In the example, enter VSAMOPEN and press the EXPLAIN PFkey again. You get the description of the OPEN error code 76 (118).

Once you are in explain mode, you can obtain information about other messages. You need not go back to the system console display. Move the cursor to a referenced message or type the message number into the input line. Then press the ENTER key or the EXPLAIN PFkey.

Reviewing System Console Messages

All messages displayed on the system console are saved in the hardcopy file. You can review them when selecting the Console dialog as shown in the previous paragraphs.

By pressing the REDISPLAY PFkey (PF7, usually), you go into redisplay mode. Use the BACKWARD and FORWARD PFkeys (PF7 and PF8, usually) to scroll through the hardcopy file and look for previously displayed messages. By entering a partition ID (for a static or dynamic partition) or AR in the input line (==>), you can selectively display messages. PF3 brings you back to the current console display.

In case of an error situation, a single message may not be very meaningful. To get a better understanding of the error situation, look also at previous messages. Select those messages that relate to the job or the partition in question. You can get a printout of the hardcopy file to check all the messages that appeared at the system console since the last startup. For details refer to "Printing the Hardcopy File" on page 113.

Reviewing the Message Log File

In z/VSE, the messages of the online part of your system are saved in the message log file. These are mainly CICS Transaction Server messages (prefix DFH) and interactive interface messages issued in case of severe errors (prefix IES). By default, LE/VSE messages are also saved in the message log file.

The message log file is printed at CICS Transaction Server shutdown. You can review that file online for error diagnosis. From the *Problem Handling* panel select:

```
2  (Inspect Message Log)
```

Use the PF-keys as shown on the screen to review the message log.

Outstanding Messages at the System Console

Messages that require an operator response are displayed on the console screen in highlighted form. It may happen that a message is no longer shown on the screen and is still awaiting a reply. This may cause the system to behave unusually. For example:

- A partition does not start processing.
- A job does not complete.
- No input/output activity.

If such symptoms occur, check first for outstanding message replies before starting a time consuming problem analysis. You can check for outstanding message replies by entering:

```
REPLID
```

If no replies are outstanding, the system issues the following message:

```
1I88I NO REPLIES OUTSTANDING
```

If there is a message reply outstanding, the system tells you so by displaying the associated partition number and the ID of the message. You can redisplay messages with the *Console* dialog and use PF7 and PF8 to scan through the display (as was explained above).

Leave the redisplay mode by pressing the END PFkey (PF3, usually).

For further details on redisplaying messages, refer to the manual *z/VSE Operation*.

Chapter 3. Starting Error Diagnosis

What You Should Consider First

Before you start a time consuming analysis consider the following questions. They may give you a clue or pinpoint the error area right away.

- Was an **error-related message** displayed on SYSLOG or SYSLST? If so, first consult the *z/VSE Messages and Codes* manual and follow the instructions given there.
- Were **any changes** applied to the system or to a failing program? For example:
  - A new application or a z/VSE optional program has been installed.
  - A program has been changed (recompiled, recataloged).
  - A program residence or address mode has been changed (using AMODE 31 / RMODE ANY).
  - A job stream (JCL statements) has been changed.
  - Startup procedures and jobs have been changed.
  - The hardware configuration has been changed.
  - A library definition (LIBDEF) has been changed, deleted, or added.
  - The format of the data processed has changed.
  - The location of the data processed has changed.
  - A different supervisor has been used.
  - Settings related to system security have been changed.

Note: Before you change any software, create a backup copy of the successfully running version. This may be the SYSRES, a user file or program, or a job stream. If the changed version fails, you can still use the backup version. The manual *z/VSE Operation* describes the backup and restore dialogs of z/VSE.

How to Start Error Diagnosis

Whenever a problem occurs, perform error diagnosis in the following sequence:

1. Record the problem.
2. Do **problem determination**:

   Is it hardware or software?
3. Do **problem source identification**:

   Is it user code or IBM code?

Recording the Problem

When an error occurs, record the error in the problem log. Do an initial recording first. Later, add as many details as possible. For suggestions for a problem log see Appendix A, “Problem Log and Change Log,” on page 177. The problem log information helps you to evaluate an error more easily. Apart from a specific error description, the following should be recorded for every error situation:

- Messages displayed. Check **SYSLOG** and **SYSLST**.
- Partitions that were running.
- Partition priorities.
- Jobs that were running.

- Devices involved.
- Listings and dumps created.
- Actions taken.

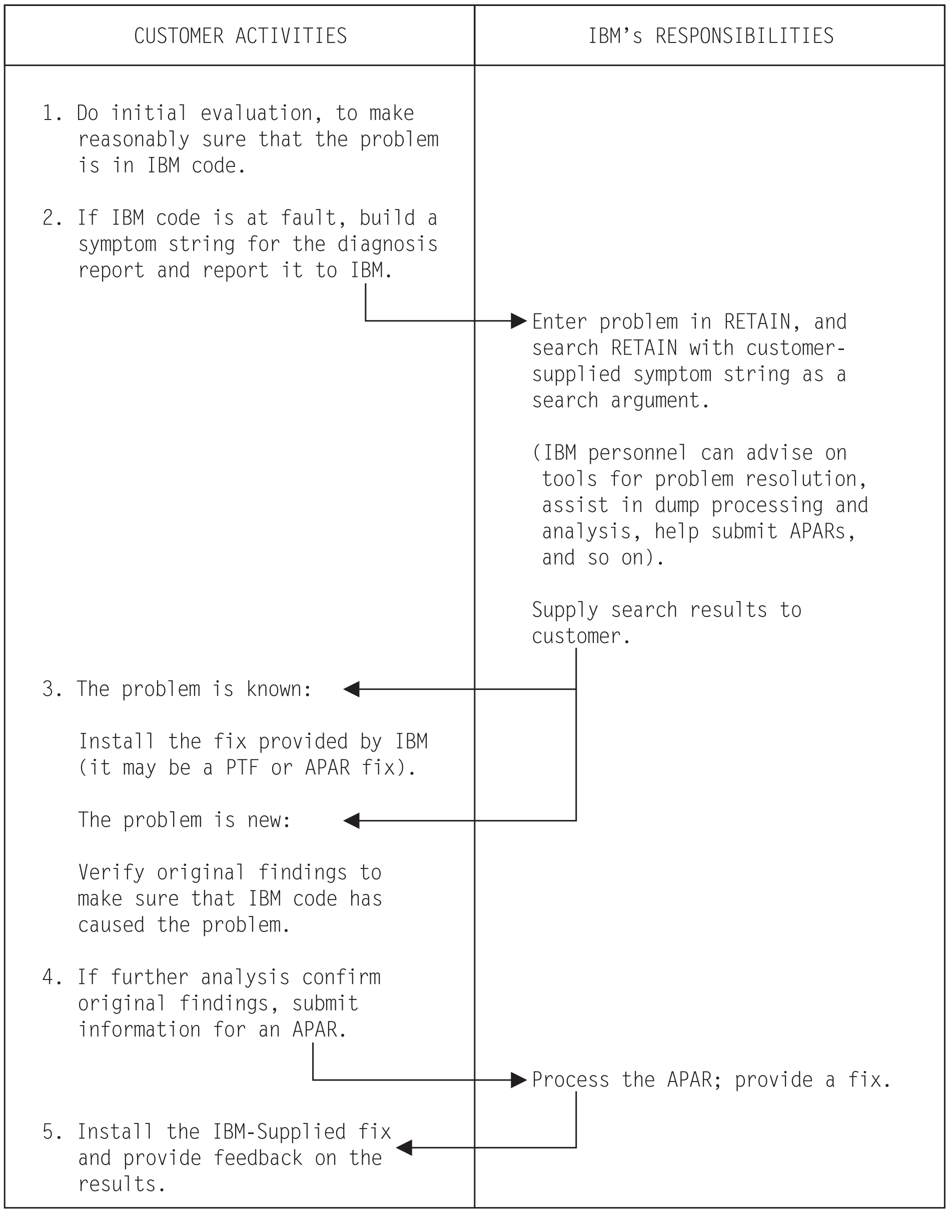
To find the cause of an error, you always start by observing a symptom and then work your way back to whatever was the cause. This section shows you how to proceed when an error symptom is noticed. You begin with problem determination.

Problem Determination: Is it Hardware or Software?

First you have to find out whether the error is caused by hardware or software. Do that by checking for hardware symptoms. A selection of them is listed under "Hardware Errors" on page 18.

Problem Source Identification: Which Software?

If the problem is apparently not caused by hardware, continue your search in the software area. Use the error symptom tables for that purpose. You find them under "Software Errors" on page 19. If you locate the error in IBM code, follow the steps shown in Figure 2 on page 17. The figure shows the diagnosis steps in relation to IBM's responsibilities.



CUSTOMER ACTIVITIES	IBM's RESPONSIBILITIES
1. Do initial evaluation, to make reasonably sure that the problem is in IBM code.	
2. If IBM code is at fault, build a symptom string for the diagnosis report and report it to IBM.	
	Enter problem in RETAIN, and search RETAIN with customer-supplied symptom string as a search argument.
	(IBM personnel can advise on tools for problem resolution, assist in dump processing and analysis, help submit APARs, and so on).
	Supply search results to customer.
3. The problem is known:	
Install the fix provided by IBM (it may be a PTF or APAR fix).	
The problem is new:	
Verify original findings to make sure that IBM code has caused the problem.	
4. If further analysis confirm original findings, submit information for an APAR.	
	Process the APAR; provide a fix.
5. Install the IBM-Supplied fix and provide feedback on the results.	

Figure 2. IBM's Service Concept if IBM Code is at Fault

Note: For details on the diagnosis report mentioned in step 2 see "The Problem Symptom String" on page 179. For details on the APAR document mentioned in step 4 see "Submitting an APAR" on page 183.

If it is an error in a user program or if it is a system handling error, follow the procedures established for your installation.

Hardware Errors

Whenever your system shows a symptom that points to a hardware error, proceed as follows:

1. Consult the appropriate hardware manual first.
2. If you cannot remove the error, contact the IBM Service Representative.

Whenever a hardware error occurs, the system records that error in the recorder file. You can use the **EREP** program to produce reports based on the recorder file information. Such a report provides valuable information for IBM's service personnel. For details on the EREP program refer to “Printing the Recorder File” on page 114.

Following is a selection of hardware error symptoms.

Processor (System) Errors

Consult the appropriate manual of the processor you are using.

Device Errors (General)

Do you observe any of the following?

- Mechanical noise normally not present.
- Enable/disable or read/write switches not correctly set.
- Lamps on or off, which are obviously not normal operating conditions.
- No movement of I/O media which should be moving.

Device Errors (Specific)

Do you observe any of the following?

- **Disk Devices**
  - A disk device always drops its READY status.
  - For the same disk device, certain messages of type 4xxx/0Cxx appear repeatedly (the explanation in the *z/VSE Messages and Codes* manual points to a hardware error).
  - Whenever a particular disk device is accessed, the system starts to write information to the recorder file. This does not apply to FCP-attached SCSI disks (see below).
  - If an error has occurred with an FCP-attached SCSI disk, an additional error message with the prefix 0S4nI is displayed. For explanations of such error messages, refer to the chapter “Configuring Your System to Use SCSI Disks” in the manual *z/VSE Administration*.
- **Tape Devices**
  - The retry's on a particular tape drive seem to be rather frequent.
  - For a particular tape drive, certain messages of type 4xxx/0Cxx appear repeatedly. The explanation in the *z/VSE Messages and Codes* manual points to a hardware error.
- **Virtual Tape Devices**
  - An input/output error (as listed in Appendix D, “Error Messages When Using Virtual Tapes,” on page 187) is shown on the system console.

    Note: Please check the limitations when using virtual tapes. These are described in the relevant section of the manual *z/VSE Administration*.

- **Printers**
  - The spacing or skipping of lines is irregular.
  - The page layout is invalid.
  - Some characters are wrong or out of line.

  Note that such symptoms may also be caused by software errors. For example by a wrong FCB or UCB.

Software Errors

The error symptoms listed in the following table point to software errors. The table helps you find diagnosis information for a particular software error more quickly. In some cases, the error symptom may be caused by a **hardware malfunction**.

Table 1. Error Symptom Reference Table

ERROR SYMPTOM	**For DIAGNOSIS refer to:**
Startup Problem	“Handling Startup Problems” on page 21
Program Abend	“Handling Abend Problems” on page 32
Transaction Abend	“Handling Abend Problems” on page 32
Incorrect Output Problem	“Handling Incorrect Output Problems” on page 34
Performance Problem	“Handling Performance Problems” on page 36
Loop Condition	“Handling Loop Conditions” on page 38
Wait Condition	“Handling Wait Conditions” on page 43
Suspended Online Activities	“Handling Situations Where Online Activities Are Suspended” on page 46
Resource Constraint Condition	Chapter 7, “Handling Resource Constraint Conditions,” on page 103

If your system, or part of it, is in a state that points to a WAIT or LOOP condition, check first for outstanding messages at the system console. This is described in section “Outstanding Messages at the System Console” on page 14. To check for outstanding messages, communication with the system via the system console must still be possible. For details about a WAIT condition refer to “Handling Wait Conditions” on page 43, for details about a LOOP condition refer to “Handling Loop Conditions” on page 38.

Chapter 4. Diagnosis Procedures

This chapter describes the procedures you should follow when various types of error occur. It contains these main sections:

- "Handling Startup Problems"
- "Handling Abend Problems" on page 32
- "Handling Incorrect Output Problems" on page 34
- "Handling Performance Problems" on page 36
- "Handling Loop Conditions" on page 38
- "Handling Wait Conditions" on page 43
- "Handling Situations Where Online Activities Are Suspended" on page 46

Handling Startup Problems

This section describes error situations that may occur during system startup.

Startup begins with turning power on. It lasts through IML (initial microprogram load), IPL (initial program load), and the activation of partitions. During the final stage of startup, message

```
F2 ... DFHSI1517  DBDCCICS Control is being given to CICS
```

is displayed at the system console. This indicates that CICS Transaction Server startup has been successful. A panel for signing-on appears on user display stations.

Areas not covered by this chapter:

1. For power-on and IML problems, refer to the documentation for the processor you are using.
2. For TCP/IP problems, refer to the IBM manual *z/VSE TCP/IP Support*. However, this chapter does also contain a small section describing TCP/IP startup problems ("TCP/IP Startup Problem" on page 25).
3. For DB2® problems, refer to the documentation provided with the DB2 Server for VSE on the *VSE Collection*, SK3T-8348.

What You Should Consider First

Before you follow the diagnosis steps below, consider the following:

1. If startup mode is RECOV or COLD, z/VSE performs time consuming recovery and initialization functions. This may appear as if the system were in a LOOP or WAIT condition. To verify the system status, use the redisplay function at the system console and check for message:

   ```
   IESI0221I  PARTITIONS xx  xx WILL BE INITIALIZED IN
              yyyyyy START MODE
   ```

   If the message shows that a RECOV or COLD startup is being performed, the system is most likely not in an error state but startup will take longer than usually.
2. Has your system setup changed since the last startup? For example, you may have added an IBM-supplied application to your system. **If so, the problem is most likely connected with that change**. Start with error diagnosis in that area.
3. Did you change any of the following since the last system startup:

- ASI IPL procedure.
- ASI JCL procedure.
- Partition startup job.
- A CICS table.
- The CICS System Definition (CSD) file.
- TCP/IP startup configuration.
- A VTAM book.
- Security settings.

If anything was changed, verify the values specified. **Incorrect specifications usually cause startup problems**.

Error Diagnosis Steps for z/VSE Components and Functions

Note: The following information is based on a system which has the characteristics of predefined environment A, B, or C, and was brought up with an initial startup mode of WARM.

Initial Diagnosis Steps

This section describes actions you should perform first to identify the component or function that failed during startup. You are guided to recovery information and from one component or function to the next. You start with the CICS component and proceed, step by step, down to the IPL level.

The tasks to be performed are all related to the **system console**.

CICS: Is the CICS Transaction Server up ?

Check for the following:

- The following message is displayed if the CICS Transaction Server startup was successful:
  ```
  F2 ... DFHSI1517  DBDCCICS Control is being given to CICS
  ```
- Enter the
  ```
  MSG F2  (partition where the CICS Transaction Server is running)
  ```
  command to find out whether the CICS Transaction Server is able to communicate.
- The panel *z/VSE Online* is displayed on user display stations if the CICS Transaction Server startup was successful.

If your actions indicate that the CICS Transaction Server is up and you still have a problem, continue error diagnosis. Use the information given in “Handling Situations Where Online Activities Are Suspended” on page 46.

If your actions indicate that the CICS Transaction Server is not up, find out whether VTAM startup was successful.

VTAM: Is VTAM up ?

Check for the following:

- Is the *Remote Application* panel displayed?
- Check for the VTAM completion message for initialization **IST020I**.

- Enter the following VTAM commands to see whether VTAM responds.

  At a display station:

  ```
  TEST          (to get the "IBM ECHO OK" message)
  ```

  At the system console:

  ```
  D NET,TERMS   (to get a terminal status report)
  ```

If you get a response, VTAM is up and you have a CICS Transaction Server startup problem. Refer to "CICS Transaction Server Startup Problem" on page 24 for recovery information.

If your actions indicate that VTAM is not up, find out whether VSE/POWER startup was successful.

POWER®: Is VSE/POWER up ?

Check for the following:

- Check for the VSE/POWER completion message for initialization **1Q12I**.
- Enter one of the following commands to see whether VSE/POWER responds:

  ```
  D A
  D RDR
  ```

If your actions indicate that VSE/POWER is up, you have a VTAM startup problem. Refer to "VTAM Startup Problem" on page 26 for recovery information.

If your actions indicate that VSE/POWER is not up, find out whether the BG partition was started successfully.

BG: Is BG partition active ?

Check for the following:

- Check whether the BG partition ID is displayed.
- Check whether the following z/VSE startup messages are displayed:

  ```
  IESI0221I  PARTITION F1 WILL BE INITIALIZED IN xxxx START MODE

  IESI0222I  REMAINING PARTITIONS WILL BE INITIALIZED IN xxxx
             START MODE
  ```

If your actions indicate that the BG partition is active, you have a VSE/POWER startup problem. Refer to "VSE/POWER Startup Problem" on page 27 for recovery information.

If your actions indicate that the BG partition is not active, find out whether IPL completed successfully.

IPL: Did IPL complete successfully ?

Check for the following:

- Check for the IPL completion message **0I20I**.
- Enter the

  ```
  MAP
  ```

  command to see whether z/VSE responds.
- Check whether the BG partition ID is displayed.

If your actions indicate that IPL completed successfully, you have a z/VSE startup problem. Refer to "z/VSE Startup Problem" on page 28 for recovery information.

If your actions indicate that IPL did not complete successfully, refer to "IPL Problem" on page 31 for IPL recovery information.

CICS Transaction Server Startup Problem

First check whether the CICS Transaction Server issued a message that indicates an abend situation. If so, perform the actions indicated by the message description in the manual *z/VSE Messages and Codes*.

If a dump was created for the abend, try a BASIC startup and correct the error according to the dump analysis result. For details about BASIC startup, refer to "Startup Mode: BASIC" on page 7. For details about dump analysis, refer to Chapter 6, "Using Dumps for Error Diagnosis," on page 71.

If no abend situation occurred, proceed as follows:

1. Enter the

   ```
   MAP
   ```

   command to check if the CICS Transaction Server startup job was started. Normally, the name of the startup job is CICSICCF.

   If the startup job has not been started, proceed as follows:

   If you use the z/VSE system startup facility, refer to "z/VSE Startup Problem" on page 28 for recovery information. Otherwise, ensure that you did release the CICS Transaction Server startup job (usually CICSICCF) in the VSE/POWER reader queue.
2. Check for CICS (prefix DFH) error messages. Proceed as follows:
   a. Check if any message replies are outstanding by issuing the command

      ```
      REPLID
      ```

      If no reply is outstanding, the system issues the following message:

      ```
      1I88I NO REPLIES OUTSTANDING
      ```

      In case of an outstanding message reply, the system displays the associated partition number and the ID of the message. Go into the redisplay mode by pressing the REDISPLAY PFkey (PF7, usually) and locate the message with outstanding reply. Leave the display mode by pressing the END PFkey (PF3, usually).
   b. Enter

      ```
      xx
      ```

      in the input line (==>) and press PF7 to redisplay and check all the messages of the CICS Transaction Server partition (xx is the partition identifier).

      Press PF3 to leave the display mode.
   c. If message

      ```
      DFHSI1572 APPLID  Unable to OPEN VTAM ACB ...
      ```

      was displayed, it indicates most likely that the definitions for the CICS Transaction Server and VTAM do not agree. It may be caused by an incorrectly or incompletely specified application name (APPLID), for example. If this message occurs, proceed as follows:
      1) Try to force an open of the ACB by entering from the system console:

```
MSG F2
CEMT SET VTAM OPEN
```

2) If this fails to open the ACB, compare the CICS and VTAM definitions:
   - Get a printout of the VTAM definitions by printing the B-books from library PRD2.CONFIG.
   - Display the CICS Transaction Server terminal IDs by entering from the system console:

     ```
     MSG F2
     CEMT INQUIRE TERM
     ```

   Then check the definitions for correctness. For networking definitions, see the introduction to “Handling Situations Where Online Activities Are Suspended” on page 46.

d. If the CICS Transaction Server partition was started and there is no message that indicates a problem, check your VTAM and CICS Transaction Server environment. Enter the following command:

   ```
   D NET,APPLS
   ```

   Use the information displayed to verify the following:

   1) Is there an application entry for the CICS Transaction Server?

      If not, you probably have a problem with VTAM startup books or a VTAM startup failure.

   2) Does the VTAM application name match the CICS Transaction Server APPLID definition?

      If not, you probably changed either the CICS Transaction Server table DFHSTxx, the Terminal Control entries in the CICS System Definition (CSD) file, or the VTAM startup book ATCCONxx.

   3) Did VTAM activate the APPLID for the CICS Transaction Server?

      If not, a VTAM startup problem is indicated. Refer to “VTAM Startup Problem” on page 26 for recovery information.

If there is a need to perform corrections, you may use startup mode BASIC. For details on this startup mode, refer to “Startup Mode: BASIC” on page 7.

You may also refer to the manual *z/VSE Administration* for details about skeleton SKCICS. The skeleton helps you create and load the CICS Transaction Server startup job CICSICCF.

If the CICS Transaction Server startup was successful and the problem still exists, continue error diagnosis with “Handling Situations Where Online Activities Are Suspended” on page 46.

TCP/IP Startup Problem

1. Check that the contents of the IPINIT member is correct. This file is usually contained in PRD2.CONFIG.
2. Check that all devices are defined and operational. The devices defined in IPINIT must also be defined in your IPLPROC.
3. For OSA-Express and HiperSockets™ devices, check that the PFIX limit has been set correctly. TCP/IP using OSA-Express and HiperSockets links require approximately 1 MB PFIX storage (above 16 MB) per link. For further details, refer to the skeleton SKTCPSTR.
4. Check that the partition size in which you are running TCP/IP (default F7) is large enough for your workload. For details, refer to the chapter “TCP/IP, OSA, and HiperSockets Support” in the manual *z/VSE Planning*.

5. Check that you have correctly installed the product key and your customer number which are required before you can use TCP/IP for VSE/ESA in production mode. You do this by running two assembler jobs of which examples are provided in the manual *z/VSE TCP/IP Support*.
6. Check that you have entered a correct TCP/IP system ID. If TCP/IP for VSE/ESA has been started using a system ID, the appropriate `EXEC IPNET` statement will look like this:

   ```
    // EXEC IPNET,SIZE=IPNET,PARM='ID=nn,INIT=... '
   ```

   (where the default ID is `00`).

   If you wish to use TCP/IP services from another partition, this partition has to recognize the system ID. To do so, you must specify:

   ```
    // OPTION SYSPARM='nn'
   ```

   (where *nn* is the system ID).

   For further details, refer to the chapter "BSD/C Sockets - Connecting to TCP/IP" in the manual *TCP/IP for VSE, Programmer' Reference*. You can find this manual in the *TCP/IP for VSE VnRn PDFs* catalog of the VSE Collection kit, SK2T-0060.

VTAM Startup Problem

1. Enter the

   ```
   MAP
   ```

   command to check if the VTAM startup job was started. Normally, the name of the startup job is VTAMSTRT. If the startup job has not been started, proceed as follows:

   If you use the z/VSE system startup facility, refer to "z/VSE Startup Problem" on page 28 for recovery information. Otherwise, ensure that you did release the startup job in the VSE/POWER reader queue.
2. Check for VTAM error messages (prefix IST). Proceed as follows:
   a. Check if any message replies are outstanding by issuing the command

      ```
      REPLID
      ```

      If no reply is outstanding, the system issues the following message:

      ```
      1I88I NO REPLIES OUTSTANDING
      ```

      In case of an outstanding message reply, the system displays the associated partition number and the ID of the message. Go into the redisplay mode by pressing the REDISPLAY PFkey (PF7, usually) and locate the message with outstanding reply. Leave the display mode by pressing the END PFkey (PF3, usually).
   b. Enter

      ```
      xx
      ```

      in the input line (==>) and press PF7 to redisplay and check all the messages of the VTAM partition (xx is the partition identifier).

      Press PF3 to leave the display mode.

   Note that VTAM error messages may have been suppressed depending on the SUPP parameter in ATCSTR00. You can modify the SUPP parameter in two ways:
   - By using the command:

     ```
     F NET,SUPP=NOSUP
     ```
   - By correcting startup book ATCSTR00.

Your startup problem may be caused by one of the following:

- Insufficient VTAM buffer space resulting from newly added terminals or applications.
- Changed VTAM startup books.
- Changed partition sizes.

For details of the above dataspace requirements, refer to the chapter "Installing z/VSE" in the manual *z/VSE Planning*.

In order to display VTAM Startup options, especially IOBUF31, you can use command

```
d net,vtamopts
```

This will show if IO buffer allocation is used. If this is the case, be aware of a higher demand of COPY blocks. The allocated number of COPY blocks is shown in the SIR command output.

If there is a need to perform corrections, you may use startup mode BASIC. For details on this startup mode, refer to "Startup Mode: BASIC" on page 7.

If VTAM startup was successful, continue error diagnosis with "Initial Diagnosis Steps" on page 22.

VSE/POWER Startup Problem

1. Enter the

   ```
   MAP
   ```

   command to check if the VSE/POWER startup job was started. Normally, the name of the startup job is POWSTART.

   If the startup job has not been started and you use the z/VSE system startup facility, refer to "z/VSE Startup Problem" on page 28 for recovery information.
2. Check for VSE/POWER (prefixes 1Q, 1R, 1V) error messages. Proceed as follows:
   a. Check if any message replies are outstanding by issuing the command

      ```
      REPLID
      ```

      If no reply is outstanding, the system issues the following message:

      ```
      1I88I NO REPLIES OUTSTANDING
      ```

      In case of an outstanding message reply, the system displays the associated partition number and the ID of the message. Go into the redisplay mode by pressing the REDISPLAY PFkey (PF7, usually) and locate the message with outstanding reply. Leave the display mode by pressing the END PFkey (PF3, usually).
   b. Enter

      ```
      xx
      ```

      in the input line (==>) and press PF7 to redisplay and check all the messages of the VSE/POWER partition (xx is the partition identifier). Press PF3 to leave the display mode.
3. If you used skeleton SKPWRGEN to regenerate your VSE/POWER phase, verify the values and parameters modified.
4. If you extended the space for the VSE/POWER files, startup of the VSE/POWER partition fails if you specified incorrect values.

For example, you specified a starting block and/or a size such that the added VSE/POWER space overlaps an existing file. Depending on the file which was overlapped, the system will report the error situation with a message. For example, if VSE/VSAM space was overlapped, the messages

```
4n98I  OVLAP UNEXPRD SECRD FILE ...
1Q19I  INVALID DATA FILE EXTENT, RC=nnnn
```

will be issued and startup will be terminated. Other messages require your decision. For example:

```
4n44D  OVERLAP ON UNEXPRD FILE IJDFILE ...
```

If the message refers to a file that (1) you recognize as a **user file** (as opposed to a system file), and (2) you are certain that this file is no longer needed, enter:

```
DELETE
```

But in other cases, enter:

```
CANCEL
```

This terminates the startup. If corrections are necessary, perform a BASIC startup. This gives you a basic z/VSE system that allows you to correct existing errors. For details about this startup mode, refer to “Startup Mode: BASIC” on page 7.

Otherwise, perform a stand-alone restore of the DOSRES volume. This restores the original extents of the VSE/POWER queue files. For the VSE/POWER data and account files, you must restore the SYSWK1 volume.

5. The procedure DTRPOWR is used to assign the VSE/POWER data queue and accounting file to disk. If new extents are used, ensure that DTRPOWR has been updated accordingly.
6. Ensure that you loaded the correct startup jobs into the VSE/POWER reader queue. To perform this task you can use program DTRIINIT (described in the manual *z/VSE System Utilities*) or the POFFLOAD command (described in the manual *VSE/POWER Administration and Operation*). z/VSE also provides these skeletons which you can modify to be used when required:
   - SKCOLD which you can use to load jobs during a cold start of VSE/POWER.
   - SKLOAD which you can use to load a job when required.

If VSE/POWER startup was successful, continue error diagnosis with “Initial Diagnosis Steps” on page 22.

z/VSE Startup Problem

Consider the following first:

- Did you change procedure CPUVAR1 or $COMVAR?
- Did you change any other startup procedure or startup job?
- Are the correct startup jobs available in the reader queue?

If any changes were made, ensure that they are correct and run error-free. Following is a selection of z/VSE startup problems.

Startup Procedure: Startup Procedure Not Available or Incorrect

The following messages may indicate problems with the startup procedure CPUVAR1 or any other startup member:

```
1S59D  CONFLICTING DATA=YES/NO OPTION

IESI0085I  LIBRARIAN MACRO xxxxxxxx TERMINATED WITH RETURN
           CODE yy
```

```
IESI0230I  MEMBER xxxxxxxx DOES NOT EXIST IN LIBRARY
           xxxxxxx.yyyyyyyy ...

IESI0231I  SYNTAX ERROR IN STATEMENT xx...xx. STATEMENT
           WILL BE IGNORED
```

For the following discussion, startup procedure CPUVAR1 is used as an example. If CPUVAR1 is not available during startup, it may be caused by one of the following:

- CPUVAR1 was deleted by mistake in system library IJSYSRS.SYSLIB.

  A copy of CPUVAR1 should still be available in library PRD2.SAVE. If so, perform a MINI startup to get a system with the BG and F1 partition active. At the system console, use librarian program LIBR and copy CPUVAR1 from PRD2.SAVE into IJSYSRS.SYSLIB.

  Otherwise, you must restore system library IJSYSRS.SYSLIB. Use a stand-alone backup tape created with the librarian BACKUP command and RESTORE=STANDALONE.
- You used the utility program DTRSETP to tailor CPUVAR1 or to create a new CPUVARn.

  This may result in conflicting names or syntax errors. A copy of CPUVAR1 should still be available in library PRD2.SAVE. If so, perform a MINI startup to get a system with the BG and F1 partition active. At the system console, use librarian program LIBR and copy CPUVAR1 from PRD2.SAVE into IJSYSRS.SYSLIB.

  Otherwise, you must restore system library IJSYSRS.SYSLIB. Use a stand-alone backup tape created with the librarian BACKUP command and RESTORE=STANDALONE.
- IJSYSRS has been destroyed. To check if errors exist, use the LIBR TEST command. If errors are found, use either:
  - The LIBR TEST command with REPAIR=YES.
  - A LIBR BACKUP/RESTORE. For the RESTORE, you either have to perform a standalone RESTORE for IJSYSRS, or restore using the mini startup process for other systems.

  A sample job showing how to use the TEST command for IJSYSRS is as follows:

  ```
  0 EXEC LIBR
  0 TEST LIB=IJSYSRS
  ```

See “Backup a SYSRES File, Library, Sublibrary, or Member” and “Restore a SYSRES File, Library, Sublibrary, or a Member” in the *z/VSE Guide to System Functions*.

System Library: System Library is Full

If a library full condition exists, CPUVAR1 can be read but the updated version cannot be written back to IJSYSRS.SYSLIB. To remove the condition, proceed as follows:

1. Perform a MINI startup to get a system with the BG and F1 partition active.
2. Invoke the librarian program LIBR from the system console. Use librarian commands to delete members or to release space. The command sequence you enter may look as follows:

   ```
   0 EXEC LIBR
   0 LD LIB=IJSYSRS OUTPUT=STATUS
   0 ACCESS S=IJSYSRS.SYSLIB
   ```

```
0 REL SPACE LIB=IJSYSRS
⋮
0 DEL mn.mt
```

In the above sequence, the LD statement creates a library directory list. This list helps you decide which members can be deleted and shows you how much delayed space there is for release. For details about delayed space, refer to “VSE Libraries” in the topic “Resources that Might Affect Performance” on page 37.

3. Perform IPL again.

If you cannot delete members or if there is no space for release, you must restore library IJSYSRS.SYSLIB. Use a stand-alone backup tape created with the librarian BACKUP command and RESTORE=STANDALONE.

The librarian commands are described in detail under “Librarian Commands” in the manual *z/VSE Guide to System Functions*.

System Library: System Library is Locked

You use the access control function of z/VSE and message

```
0S20I UNAUTHORIZED ACCESS REQUEST FOR:
      yyyyyyyy.zzzzzzzz
0S00I xxx name CANCELED
```

is displayed during IPL. This would indicate that a successful startup cannot be performed because of incorrect access control table definitions that lock the system library. To remove the error, proceed as follows:

1. Perform IPL again and interrupt the IPL process. For details refer to the manual *z/VSE Guide to System Functions*.
2. Enter the IPL and JCL procedure names together with the STOP parameter. For example:

   ```
   IPL=$IPLESA,JCL=$$JCL,STOP=SVA
   ```

   The system responds with message 0J05D and further parameters can be entered.
3. Enter

   ```
   SYS SEC=NO
   ```

   to reset access control temporarily. The system responds with message 0D07D.
4. Press ENTER.

   Startup continues without activating access control. Correct the access control table as soon as the system is up.

If the above steps do not remove the problem, you must restore your system library IJSYSRS.SYSLIB. Use a stand-alone backup tape created with the librarian BACKUP command and RESTORE=STANDALONE. See “Backup a SYSRES File, Library, Sublibrary, or Member” and “Restore SYSRES File, Library, Sublibrary, or a Member” in the *z/VSE Guide to System Functions*.

IPL Problem

IPL is started by entering load information on the *S/390*® *Support Element* panel.

If the load does not complete successfully, follow the directions given on the window, or on any messages that might be displayed, to determine the problem and how to correct it. Refer to the *S/390 Support Element Operations Guide,* GC38-3118 for further information.

Then proceed as follows:

1. Has the correct IPL information been entered?

   Specifying an incorrect IPL device address is the most common cause of an IPL problem.

   After typing the correct IPL information into the *PROGRAM LOAD* panel press ENTER to continue IPL.

2. Does the IPL procedure contain incorrect information? (You may have changed the IPL procedure).

   Proceed as follows:

   Perform IPL again and interrupt IPL processing. Refer to the manual *z/VSE Guide to System Functions*.

   Enter the IPL command with the names of the IPL and JCL procedures appropriate for your environment. For example, for a system with DOSRES on a 3380 disk volume you should enter:

   ```
   IPL=$IPLE80,JCL=$$JCL780,TYPE=SENSE
   ```

   This causes a startup with the original IPL and JCL procedures as shipped with z/VSE. The TYPE=SENSE parameter ensures that all attached devices are sensed and recognized by z/VSE. No ADD commands are required.

   Appendix C, “List of IPL, JCL, and Label Procedures,” on page 185 shows the names of the IPL and JCL procedures shipped with z/VSE. When startup is complete, you can correct your own IPL procedure, if necessary:

   a. From the *Hardware Configuration and IPL* panel select:

      ```
      3  (Tailor IPL Procedure)
      ```

   b. Correct your IPL procedure.

   c. Perform normal shutdown.

   d. Repeat IPL.

3. Did the system issue message:

   ```
   0I94A PHASE phasename NOT FOUND. IPL TERMINATED
   ```

   A library member may have been deleted by mistake.

   To remove the error situation, you must restore system library IJSYSRS.SYSLIB. Use a stand-alone backup tape created with the librarian BACKUP command and RESTORE=STANDALONE. See “Backup a SYSRES File, Library, Sublibrary, or Member” and “Restore a SYSRES File, Library, Sublibrary, or a Member” in the *z/VSE Guide to System Functions*.

4. Did the system issue the following messages:

   ```
   0I04I IPLDEV=X'cuu', VOLSER=number, CPUID=number
   0I03D ENTER SUPERVISOR PARAMETERS [OR ASI PARAMETERS]
   ```

   If YES:

   Enter the supervisor parameters. For details see the description of message 0I03D in the *z/VSE Messages and Codes* manual. Supervisor parameters are described under “The Supervisor Parameters Command” in the *z/VSE System Control Statements*.

If NO:

Display **storage bytes 0 - 3** (low-address storage bytes) with the Display/Alter function. Is an **IPL message code or WAIT code** stored in these bytes? To identify a code, refer to the “VSE/Advanced Functions Codes and SVC Errors” in the *z/VSE Messages and Codes* manual and follow the instructions given there.

If the error is apparently not related to IPL, consider a power-on or IML problem. For such a problem, consult the documentation of the processor you are using.

Data You Should Collect in Case of a Startup Problem

For further analysis, collect the following data:

- System console log (from the hardcopy file).
- Dump or record the contents of low-address storage bytes X'00' to X'17'. Use the Display/Alter function. To interpret the data stored in these bytes refer to the “VSE/Advanced Functions codes and SVC errors” in the *z/VSE Messages and Codes* manual.

  Note: You must do this before you take a stand-alone dump.
- Stand-alone dump. Refer to “Taking a Stand-Alone Dump” on page 97.
- Assembly listings and linkage editor map if you generated your own supervisor.
- Retrace printout of system history file.

Handling Abend Problems

A program or transaction may terminate before processing is finished because of an error situation or because the operator forced termination. The program or transaction comes to an abnormal end (**abend** for short).

Recognizing an Abend

Messages or a notification panel indicate an abend. The **partition identifier** in front of the message text tells you in which partition the abend occurred.

Among the abend messages are the following:

- A message at the system console indicating a program check:
  ```
  0S03I PROGRAM CHECK INTERRUPTION ...
  ```
- A message at the system console indicating that a dump is written:
  ```
  0S30I DUMP STARTED. MEMBER= ... DUMP in SUBLIB= ...
  ```
- A message at a user display station indicating a transaction abend:
  ```
  DFHAP0001 DBDCCICS  An abend (code xxx/yyyy) has occurred ...
  ```
  This abend originates from a transaction running under the CICS Transaction Server. The message is preceded by a notification panel that gives details about the transaction abend. For details about how to analyze such an abend online, refer to Chapter 5, “Online Analysis of CICS Transaction Abends,” on page 57.

What You Should Consider First

Depending on the JCL definitions, the system creates a **partition** or **system dump** of the partition in which the abend occurred.

For a CICS transaction abend, however, only a dump of the transaction related storage (a CICS **transaction dump**) is taken. In case of an exceptional error situation a CICS partition or system dump is created in addition.

Some abend conditions may be related to a user application program, others to one of the IBM-supplied programs.

If the system creates a dump, always record the following in the problem log:

- Dump ID
- Job name
- Partition
- Day and time.

Application Program Abend

Transaction Abend: In case of a CICS transaction abend, the system displays a notification panel on your screen (see Figure 4 on page 58). It gives you details about the abend. You can then activate the Online Problem Determination Program or redisplay your working screen. Your working screen would then show message

```
DFHAP0001 DBDCCICS  An abend (code xxx/yyyy) has occurred ...
```

The Online Problem Determination Program collected and saved all relevant error data in a file. You, or the programmer responsible for the transaction, can access this data right away or later for online error diagnosis. To diagnose a transaction abend, go to **Chapter 5, “Online Analysis of CICS Transaction Abends,” on page 57.**

Batch Program Abend: In case of a batch program abend, the system or partition dump is stored in the dump library. You can analyze such a dump online or in batch mode. Refer to **Chapter 6, “Using Dumps for Error Diagnosis,” on page 71** for details.

The dump analysis gives the programmer initial information for debugging the program. It is advisable first to check whether the setup for running the program was correct.

z/VSE Base Program Abend

One of the z/VSE base programs (such as the subsystems CICS Transaction Server or VTAM) may come to an abnormal end. If this happens, z/VSE creates a partition or system dump and stores it in the dump library.

Analyze the dump as described in Chapter 6, “Using Dumps for Error Diagnosis,” on page 71.

Experienced User: As a more experienced user, you may also refer to “Abend Symptom” on page 167. There you find additional diagnosis information for abend situations.

Data You Should Collect in Case of an Abend

For further analysis, collect the following data:

- System console log (from the hardcopy file).
- Dump output.
- Job stream.
- MAP command output.
- Retrace printout of system history file.

Handling Incorrect Output Problems

Incorrect output problems are usually detected by the user of the output. The application user may notice incorrect print output. A program which processes data of a tape file may detect the incorrect formatting of the data on tape.

On some occasions, the system console operator might discover an error situation such as excessive printer paper consumption. This may be due to an incorrect job setup or erroneous skipping or spacing.

Recognizing Incorrect Output

Incorrect output appears in many forms. Here are a few examples:

- Invalid or unexpected messages.
- Unidentifiable data; in particular, unreadable output on a printer or at a display station.
- Duplicated data.
- Missing data.
- Data out of sequence or in the wrong format.
- Repetitive output (see "Handling Loop Conditions" on page 38 for details).

What You Should Consider First

The cause for incorrect output is of two types:

- Setup errors, such as incorrect specification of an input file or a mismatch between the printer train and the loaded Universal Character Set Buffer (UCB).
- Program errors.

Error Diagnosis Steps

Setup Errors

First make sure that the operational setup was correct. For example, that the correct files were specified and do exist. For error diagnosis, it may be necessary to rerun the job or program with the original setup.

Files:

If you suspect a problem with the files used, gather file information. Utilities are available for:

- Displaying VTOC information.

  For details refer to "Displaying the Table of Contents (VTOC) of a Disk Volume" on page 118.
- Displaying file information.

  For details refer to "Displaying VSE/VSAM File Information Online" on page 115.
- Printing a VSE/VSAM catalog.

  For details refer to "Printing a VSE/VSAM Catalog" on page 117.
- Printing the contents of a VSE/VSAM file.

  For details refer to "Printing the Contents of a VSE/VSAM File" on page 116.

Printer Output:

Proceed as follows:

- If the printer output is not readable or message 0P11D (Data Check) occurred, check whether the correct Universal Character Set Buffer (UCB) is loaded. The system library must contain a UCB that corresponds to the printer chain being used. To create and catalog a new UCB, select

  ```
  4 (Catalog Printer UCB)
  ```

  from the *Hardware Configuration and IPL* panel. Refer to the *z/VSE Administration* manual under "Cataloging Printer UCB" for details.

  Note that simply cataloging does not make the new UCB available. The new UCB must be loaded. It is loaded automatically during IPL. An UCB can also be specified in the VSE/POWER *** $$ LST** control statement, or the operator can enter an LUCB command. To enter an LUCB command, proceed as follows:

  – If a VSE/POWER list task is active on the printer, stop the task by entering:

    ```
    P cuu
    ```

  – Load the correct UCB:

    ```
    LUCB cuu,$$BUCBxx,...
    ```

    See the manual *z/VSE Operation* under "Changing Print Trains" for additional information.

  – When message 0P11D appears, enter:

    ```
    IGNORE
    ```

  – Start the printer again. Enter:

    ```
    S LST,cuu
    and the PGO command, if applicable:
    G cuu
    ```

  At the next IPL, the new UCB will be loaded automatically.
- If the skipping or spacing is incorrect, ensure that the correct Forms Control Buffer (FCB) is loaded. To create and catalog an FCB, select

  ```
  3 (Maintain Printer FCB)
  ```

  from the *Hardware Configuration and IPL* panel. Refer to the *z/VSE Administration* manual for details about the dialog.

Note that loading a new FCB depends on the type of FCB. A standard FCB is loaded automatically during IPL. A standard FCB *can be* and a non-standard FCB *must be* specified in the VSE/POWER *** $$ LST** control statement, or the operator must enter an LFCB command:

```
LFCB cuu,$$BFCBxx,...
```

Always ready the printer before using the LFCB command.

Program Errors

If a program error is the most likely cause of the problem, the programmer responsible should do error diagnosis. Before you contact the programmer, gather the following information:

- Take note of the:
  - Dump ID (if a dump was written).
  - Job name.
  - Partition.
  - Day and time.
- Issue the MAP command. Save the output.
- Save all program output.

If the cause of the error is not obvious, the programmer can use tracing tools. For tracing tools refer to Chapter 10, “Using Traces,” on page 141.

Experienced User: As a more experienced user, you may also refer to “Incorrect Output Symptom” on page 165 for additional diagnosis information.

Data You Should Collect in Case of Incorrect Output

Collect the following data and have it available for error diagnosis:

- System console log (from the hardcopy file).
- Dump ID (if necessary, rerun the job and cancel it to create a dump).
- Job stream.
- A list of all I/O files and volumes used by the particular program.
- EREP output, if possible.
- Program output.

Handling Performance Problems

Recognizing a Performance Problem

A performance problem exists when the performance deviates notably from the established values. The deviation may affect the entire system, only one, or several partitions. Note that a job as such still runs correctly, but needs more time to complete.

What You Should Consider First

Such a situation may be caused *not by an error* but by one of the following conditions:

- A program has to process more data than usual.
- A program has been changed.
- Too many jobs are running at the same time. As a result, the system is overloaded and may deactivate one or more partitions.
- Priority settings have been changed.

But performance degradation can also be caused by real errors. For example:

- An error in a user program.
- An error in a control program.
- Shortage of storage (storage is not released).
- A high number of errors on an I/O device. This can slow down the system considerably.

If one of these conditions or error situations exists, the system may show symptoms such as:

- High paging activity.
- High channel usage.
- High number of busy control units.

Resources that Might Affect Performance

VSE Libraries

Performance problems may be caused by a large amount of “delayed space” in a library. Delayed space means space that is no longer used but has not been released yet. It occurs in libraries that are shared either among partitions or among CPUs. You can find out how much of library space is delayed by creating a directory list with the LISTD command. Refer to “Using the LISTD Librarian Command” on page 128 for details. To free delayed space, use the librarian RELEASE command as described under “Release Space for a Library or Sublibrary” in the *z/VSE Guide to System Functions*.

Make it a rule that non-shared libraries are not placed on shared volumes.

Shared VSAM Files with SHROPT(4,4)

These shared VSAM files may degrade performance if file sharing is performed across VSE systems. For the definition of the lock file, refer to the manual *z/VSE Guide to System Functions*.

Workfiles on virtual disks.

VSE/ICCF DTSFILE

If the VSE/ICCF library owners perform many delete and add operations, the library space may become scattered after some time. This can *increase* considerably the time needed for accessing a library member. You can use message

```
K088I  HI FILE RECORDS=nn (mm%)
```

as an indicator. This message is issued during startup or normal processing and may indicate less space (due to scattered and unused space) than should actually be available. If *mm* shows a relatively low value, consider a reorganization run. This makes the unused space available again and improves performance. To reorganize the DTSFILE, use the VSE/ICCF utility program DTSUTIL.

You might also refer to the VSE Performance manuals, which you can find at this URL:

http://www.ibm.com/systems/z/os/zvse/documentation/performance.html

Error Diagnosis Steps

If a performance problem is indicated, proceed as follows:

1. Cancel one partition after another and check which program causes the performance problem.
2. Find out whether any changes have been applied to the system.

Note that a loop condition may also be the cause of a performance degradation. For details about a loop condition, refer to "Handling Loop Conditions."

Refer also to the *z/VSE Planning* manual which has details about tuning the system under "Storage and Tuning Recommendations".

If you cannot remove the problem, contact IBM for support.

Monitoring the System

To evaluate the performance of your system, you should monitor it over a longer period of time. You can do that online via the Interactive Interface of z/VSE. The following dialogs are available:

- Display System Activity.

  Refer to Figure 3 on page 40 for information provided by the dialog.
- Display Channel and Device Activity.

For a description of these dialogs and how to interpret the information shown, refer to the *z/VSE Operation* manual.

Handling Loop Conditions

In most cases, a loop results from a logic error in a program. In exceptional cases a hardware malfunction may also be the cause of a loop. See description of system loop below.

Recognizing a Loop

Quite often, the symptom gives an indication as to which part of the system is looping.

Repetitive Output

When the system produces the same output over and over, the reason is most likely a looping program. Usually, you can relate such output (a printout, a display, or a tape or disk), to a specific job or program that created it. This tells you where to look for the error.

If the repetitive output consists of messages at the system console, the partition identifier gives you a clue where to start searching for the error.

Other symptoms that point to a loop are more difficult to relate to a particular part of the system.

Online Loop

The following symptoms indicate that either a transaction running under the CICS Transaction Server, or any of the control programs VTAM, the CICS Transaction Server, or VSE/ICCF may be in a loop.

- Extremely slow response time at the display stations, or no response at all.
- Data cannot be entered at a display station.

VSE/POWER Loop

The following symptoms indicate that VSE/POWER may be looping.

- Impossible to communicate with VSE/POWER queues.
- Impossible to flush the batch partitions with the PFLUSH command, or to stop a printout with the PSTOP command.
- VSE/POWER commands are not executed even if they are entered after an RC (Request Communication) command.

Batch Loop

A program in a batch partition may be in a loop if you observe the following:

- A pointless recurrence of input/output activity. For example, the same console messages appear over and over again, or you suspect that the same data is written repeatedly to disk or tape. Also, entering the PDISPLAY A command may show you that a job produces more output segments than expected. VSE/POWER indicates the beginning of a new segment with message 1Q53I.

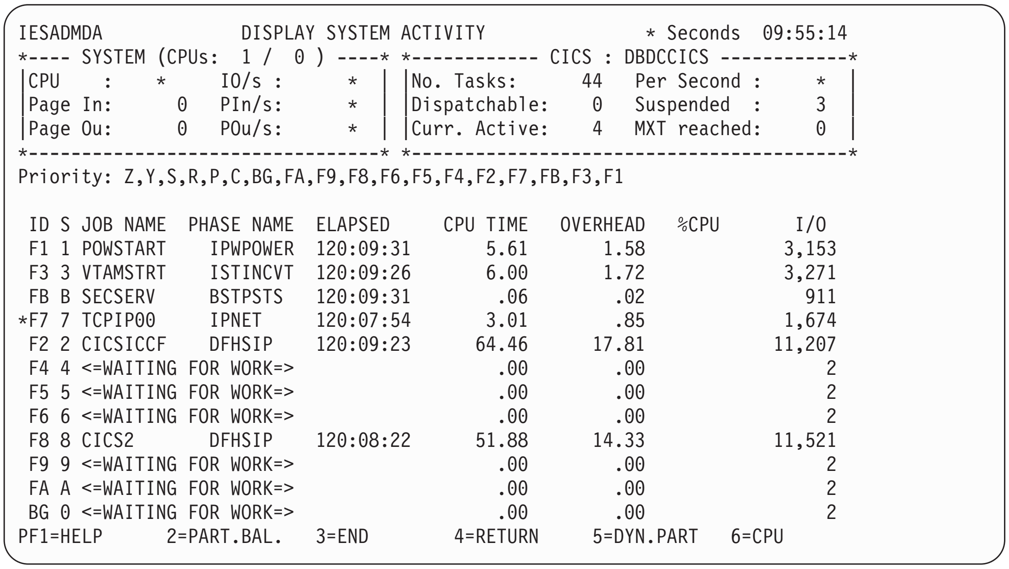
  Also, watch out for message 1Q52I. VSE/POWER issues this message if a job produces more output than specified. You can specify the number of list or punch records expected with the RBM parameter in the * $$ LIST or * $$ PUN statement.
- A program does not change its status for a long time; for example, no input/output activity takes place.

Using the System Activity Display to Recognize a Loop

Select from the *System Status* panel:

1 `(Display System Activity)`

You get a display as shown in Figure 3 on page 40.



```
IESADMDA             DISPLAY SYSTEM ACTIVITY               * Seconds  09:55:14
*---- SYSTEM (CPUs:  1 /  0 ) ----* *------------ CICS : DBDCCICS ------------*
|CPU    :    *     IO/s :      *  | |No. Tasks:      44   Per Second :     *  |
|Page In:      0   PIn/s:      *  | |Dispatchable:    0   Suspended  :     3  |
|Page Ou:      0   POu/s:      *  | |Curr. Active:    4   MXT reached:     0  |
*---------------------------------* *-----------------------------------------*
Priority: Z,Y,S,R,P,C,BG,FA,F9,F8,F6,F5,F4,F2,F7,FB,F3,F1

 ID S JOB NAME  PHASE NAME  ELAPSED     CPU TIME   OVERHEAD   %CPU       I/O
 F1 1 POWSTART    IPWPOWER  120:09:31       5.61       1.58             3,153
 F3 3 VTAMSTRT    ISTINCVT  120:09:26       6.00       1.72             3,271
 FB B SECSERV     BSTPSTS   120:09:31        .06        .02               911
*F7 7 TCPIP00     IPNET     120:07:54       3.01        .85             1,674
 F2 2 CICSICCF    DFHSIP    120:09:23      64.46      17.81            11,207
 F4 4 <=WAITING FOR WORK=>                   .00        .00                 2
 F5 5 <=WAITING FOR WORK=>                   .00        .00                 2
 F6 6 <=WAITING FOR WORK=>                   .00        .00                 2
 F8 8 CICS2       DFHSIP    120:08:22      51.88      14.33            11,521
 F9 9 <=WAITING FOR WORK=>                   .00        .00                 2
 FA A <=WAITING FOR WORK=>                   .00        .00                 2
 BG 0 <=WAITING FOR WORK=>                   .00        .00                 2
PF1=HELP      2=PART.BAL.   3=END        4=RETURN     5=DYN.PART   6=CPU
```

Figure 3. Example of a System Activity Display

A system activity display is updated automatically at fifteen second intervals. The values that are of interest are shown in the columns CPU TIME and I/O. If a program uses up CPU time but does not increase the I/O count, it is likely to be in a loop. Look primarily at the batch partitions. If none of them indicates a problem, check the other partitions too.

Note: Many batch jobs show lots of CPU operations but relatively few I/O operations during most of their processing. These jobs normally should be run with lower priority than those jobs which are less CPU intensive and more I/O intensive.

What You Should Consider First

As a general rule, you should take the actions recommended below. If, **after doing that**, you are still unable to resolve the problem, try to shutdown the system *in an orderly fashion*. This is not always possible because the particular error situation may prevent just that.

After going through some of the actions suggested, you may come to a point where only a re-IPL is left as a last resort. If, after re-IPL, the loop appears again, contact IBM for support.

Error Diagnosis Steps

Note: Many processors include functions such as traces that assist you in diagnosing loops. Refer to the appropriate manual of your processor for such tools.

The following discussion assumes that you do not know which program is looping. In other words, you have to *locate* the loop.

If you are reasonably sure that the problem lies in the **online** part of the system, proceed with step 3 below. Otherwise, find out which jobs are running in the batch partitions by entering the VSE/POWER display command:

```
D A
```

If the D A command is accepted and the active batch jobs are displayed, proceed with step 1 below.

If the system responds with the message "PROCESSING ROUTINE ACTIVE", wait a few seconds and then repeat the command. If you get message "COMMAND IGNORED", issue the command

```
RC  (Request Communication)
```

before you try the D A command (or any other VSE/POWER command) again. If the RC command is not accepted, do instruction stepping via the *MODE SELECTION* panel. Write down 20 to 30 addresses that are displayed. If the loop is rather tight, you may be able to locate it in this way. If the system is up again, use the MAP command to identify the partition in which the addresses are located.

If you are not successful, do a STORE STATUS/MACHINE SAVE and take a stand-alone dump (see "Taking a Stand-Alone Dump" on page 97). Then re-IPL.

1. Cancel (with the PFLUSH command) the active batch partitions **one after another**. Afterwards cancel the CICS Transaction Server, then the VTAM, and then the VSE/POWER partition, if necessary.

   Note: If you want to take a dump, you must do it before entering the PFLUSH command. See step 2 below for the dump command.

   If you are monitoring the system, as described for a batch loop on page 39, and found a job that you suspect to be looping, let the corresponding partition be the **first** that you try to cancel.
2. Observe the behavior of the system. If it seems to be running normally again after you terminated the batch program, you know the loop was caused by that program.

   Make sure that the loop is indeed an error situation. What seems a loop may have been intended: some applications require an immense amount of computing time only, with no input/output activity in between. Therefore, try to contact the person who owns the program.

   The application programmer may need a partition dump for further problem analysis. You have to take the dump when the job is still running. Enter:

   ```
   CANCEL pp,DUMP
   ```

   where *pp* is the batch partition of the program that was looping. Take note of the dump ID which is displayed as part of message 0S30I. Save any printer output.

   Proceed to step 3 if the system remains in a loop after you canceled the batch partitions.
3. **You now know** that the cause of the loop lies in the **online part** of the system, most likely in a transaction running under the CICS Transaction Server.

   Display the tasks that are currently running under the CICS Transaction Server (and VSE/ICCF). From the system console, issue the command:

   ```
   MSG F2
   ```

   The CICS Transaction Server then provides you with a *reply ID* (represented by 'nn' in the following example). You use this reply ID to request CICS master terminal (CEMT) functions.

   Whenever the system asks you for a console command, you must enter the command with the reply-id. Therefore, to display the tasks currently running under the CICS Transaction Server, enter:

   ```
   nn CEMT INQUIRE TASK
   ```

Repeat this command several times, each time waiting a short interval of time. If you discover a transaction that remains endlessly in the system, besides yourself (CEMT) and the transaction IESO, terminate that task by issuing:

```
nn CEMT SET TASK(tt) FORCE
```

where *tt* is the number of the suspicious task. This may cause the problem to disappear. Be sure to inform the application programmer responsible for the transaction.

Give the system some time to terminate the task. Then check whether the task is still running by entering:

```
nn CEMT INQUIRE TASK
```

Repeat the command after some time if the task is still running. If you do not succeed in terminating the task, try the following commands:

```
nn CEMT SET TERM(termid) OUT FORCE
nn CEMT SET TERM(termid) INS ACQ
```

where *termid* is the terminal of the task in error. That should purge the task and put the terminal back into service again. If the commands fail, use VTAM commands to deactivate and activate the terminal in question where xx...xx is the VTAM netname (VTAM ID) of the terminal:

- `V NET,INACT,ID=xx...xx,F,`
- `V NET,ACT,ID=xx...xx`

  Within a few seconds, the display station should be ready for sign-on. **Watch for any error messages** after you entered this command. They may give you a clue to the problem.

If these actions do not help, shut down the system in the proper order as described in steps 4 through 6 below.

4. Try to cancel any program that is running in an interactive partition under VSE/ICCF. First issue the command

   ```
   /MAP
   ```

   to determine which interactive partitions are active. Also, issue the /DISPLAY and the /USERS commands to get an overall picture of the VSE/ICCF environment. Then issue

   ```
   /CANCEL n
   ```

   where *n* is the interactive partition number (which you find in the output of the /MAP command).
5. Terminate the CICS Transaction Server by issuing, from the system console, the following commands:

   ```
   /WARN
   MSG F2
   nn CEMT PERFORM SHUTDOWN
   ```

   The /WARN command sends a warning message to all online users announcing that shutdown will be performed in 10 minutes time. 10 minutes is the default value and the smallest value that can be specified. 30 minutes is the maximum value.

   After some time the system should respond with message:

   ```
   1Q34I F2 WAITING FOR WORK ...
   ```

   In the example, F2 is the CICS Transaction Server partition. If you do not get this message, use a normal display station and check whether the CICS Transaction Server is still up. If yes, force a shutdown of the CICS Transaction Server with the command:

```
CEMT PERFORM SHUTDOWN IMMEDIATE
```

If the CICS Transaction Server terminates, go to step 6.

If the CICS Transaction Server does not terminate, enter the following:

```
CANCEL F2
```

You may have to repeat the command several times. After you entered the CANCEL command, a lot of messages may appear. Ignore them. If the CANCEL command works, continue with step 6.

If the system is unable to execute the CANCEL command, do a STORE STATUS and take a stand-alone dump as described in “Taking a Stand-Alone Dump” on page 97.

6. Restart the CICS Transaction Server.

Experienced User: As a more experienced user, for additional diagnosis information you may also refer to “Loop Symptom” on page 174. A stand-alone dump of a loop condition is shown in Chapter 6, “Using Dumps for Error Diagnosis,” on page 71.

Data You Should Collect in Case of a Loop

You should collect the following data and have it available for error diagnosis:

- System console log (from the hardcopy file).
- The output from the STATUS command.
- Stand-alone dump (refer to “Taking a Stand-Alone Dump” on page 97), or partition dump (DUMP pp,cuu) if the problem can be isolated to a partition. Analysis routines help you interpret the contents of a dump. For details refer to Chapter 6, “Using Dumps for Error Diagnosis,” on page 71.
- Output of the MAP and /MAP command.
- Any output produced by the looping program or task.
- Link map of the looping program (ACTION MAP).
- Source listing of the looping program or task.
- Retrace printout of the system history file.
- Notes from instruction stepping.

Handling Wait Conditions

A wait condition may affect the whole system or only a part of it; a partition or a number of display stations, for example.

Recognizing a Wait

Usually, the symptom gives an indication as to which part of the system is in a wait.

- System Wait:
  - The system is not doing any work at all.
  - No messages appear at the system console.
  - No input/output activity occurs.
  -
- Batch Partition Wait:

  In one or more batch partitions, no work is being done. If a partition is in a wait state, you can use the STATUS command to show the resources that are affected by the wait.

- Suspended Online Activities:

  If one, several, or all display stations do not respond, the online part of your system may cause the problem. Continue your search as described under "Handling Situations Where Online Activities Are Suspended" on page 46.

What You Should Consider First

Make sure that no unusual event or message escaped your attention. Check if there are any messages waiting for a reply. Enter at the system console:

```
REPLID
```

If the system does not accept the command, it is in a severe wait. At the system console, display the low core bytes and write them down. Use the explanations given in Volume 1 of the *z/VSE Messages and Codes* manual to identify the cause of the problem.

You can use the attention routine (AR) command CANCEL to free any attention routine processing that might be active.

If you do not succeed, perform STORE STATUS and take a stand-alone dump. Both steps are described in "Taking a Stand-Alone Dump" on page 97.

If the REPLID command is accepted and no replies are outstanding, the system issues the following message:

```
1I88I NO REPLIES OUTSTANDING
```

If a message is outstanding, the system displays the associated partition number and the ID of the message. In the input line (==>), type

```
xx       (where xx is the partition ID)
```

and press the REDISPLAY PFkey (PF7, usually) to redisplay messages that have disappeared from the screen. Scan through the messages to find the one that is waiting for a reply. Respond to that message as required. Leave the redisplay mode by pressing the END PFkey (PF3, usually).

Diagnosing Steps for a System Wait

Determine Hard or Soft Wait

The processor's monitoring facilities offer a function to display the Program Status Word (PSW). The hexadecimal representation of the PSW is shown below:

02xxxx00 xxxxxxxx
I/O mask bit: if this bit is '1', then the processor is enabled for I/O interruptions.

01xxxx00 xxxxxxxx
External mask bit: if this bit is '1', then the processor is enabled for external interruptions.

xxx2xx00 xxxxxxxx
Wait bit: if this bit is '1', then the processor is in the wait state.

In a disastrous error state, the most important bit in the PSW is bit 14, the **Wait** bit. (This is the bit X'02' in the second byte of the PSW). If the **wait bit is zero**, then the processor is active and you should conclude that the system is in a disabled loop.

The second full word in the PSW (the PSW address field) shows the instruction address where the processor has stopped. If you start and stop the processor repeatedly, you may be able to determine the scope of the loop. You could also try to activate the processors instruction trace facility to examine some loop details.

If the **wait bit is one**, then the processor is in the wait state. That is, the processor does not execute any instructions. In this wait state, the second full word of the PSW does not contain an instruction address but may contain useful information that identifies the reason why that wait state has been entered. Before you can draw such a conclusion, you need to examine bits 6 and 7 of the first byte of the PSW. If either of these two bits X'02' (the I/O mask bit) or X'01' (the external mask bit) is ON, then the processor is in a **SOFT WAIT** condition. The processor is in a **HARD WAIT** condition if both these bits are OFF.

Hard Wait Condition

A hard wait is caused by a hardware failure, by program errors like a program check in the supervisor program, or by errors in programs that control the system.

At the system console, display the low address storage bytes and write them down. Use the explanations given in Volume 1 of the *z/VSE Messages and Codes* manual to identify the cause of the problem. If you do not succeed, perform STORE STATUS and take a stand-alone dump. For details, refer to “Taking a Stand-Alone Dump” on page 97.

Re-IPL the system and analyze the dump. For details refer to “Example 4: Analyzing a Stand-Alone Dump” on page 91.

Soft Wait Condition

The system is probably waiting for the completion of an event. If message 0P08 asked you to make a device ready, then try to do so. In case the device cannot be made ready, issue the command RC (Request Communication). Cancel the I/O operation by entering:

```
15 CANCEL cuu
```

where cuu is the address of the device in error and 15 is the reply-id. Be aware that the partition to which the device belongs might be canceled. Report this error to IBM.

You may not be able to relate the soft wait to a device in error. In this case, try to find some error indication through a redisplay of messages. If you do not find a clue, do a STORE STATUS and take a stand-alone dump. For details refer to “Taking a Stand-Alone Dump” on page 97. Then re-IPL.

Experienced User: If you are a more experienced user, you may also refer to “Wait Symptom” on page 170 for additional diagnosis information.

Diagnosing Steps for a Batch Partition Wait

Find out whether a job (partition) is in a wait state. Select from the *System Status* panel:

```
1  (Display System Activity)
```

You get a display as shown in Figure 3 on page 40. This display is updated automatically in fifteen second intervals.

The values that are of interest are shown in the columns:

```
ELAPSED
CPU TIME
IO
```

A job (partition) that is in a wait would show an **increase** in ELAPSED time but **no change** in its CPU time and I/O values.

- To display the status on your system console, issue the STATUS *xx* command (where xx is the partition).
- To save the partition storage, take a dump. In the command input line, enter:

  ```
  DUMP pp,cuu
  ```

  then

  ```
  PFLUSH pp
  ```

  where *pp* is the identifier of the batch partition and *cuu* the address of the output device (tape or printer). Analyze the dump as described in **Chapter 6, "Using Dumps for Error Diagnosis," on page 71.**

Data You Should Collect in Case of a Wait

Collect the following data and have it available for error diagnosis:

- System console log (from the hardcopy file).
- In case of a system wait, create a stand-alone dump (refer to "Taking a Stand-Alone Dump" on page 97). If the problem can be isolated to a partition, create a dump with the attention routine dump command (DUMP pp,cuu).

  Analysis routines help you to interpret the contents of a dump. For details refer to Chapter 6, "Using Dumps for Error Diagnosis," on page 71.
- Any output produced shortly before the WAIT occurred.
- List of files and volumes used.
- EREP output.
- Retrace printout of the system history file.

Handling Situations Where Online Activities Are Suspended

If a diagnosis step does not apply to your installation, consult the documentation of the components you are using.

Introduction

This section describes error situations that affect the online users of your system. The source of such errors is often difficult to pinpoint since many software and hardware components are involved.

You must be able, in case of an error situation, to relate terminal (display station) x at location y to the associated networking definitions in the VTAM books and the CICS System Definition (CSD) file. It is therefore essential for error diagnosis that you have a list that shows the logical and physical structure of your local and/or remote network. Such a list is referred to as **configuration list**.

To create a configuration list for your installation, perform the following steps:

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

Create a new list whenever a change is implemented.

Naming Convention for Terminals

VTAM provides naming conventions for a networking environment that uses a 4-digit subarea number. The manual *z/VSE SNA Networking Support* describes the naming conventions in detail. The following terminal definitions are used for the diagnosis steps shown.

For a local display station:

VTAM netname: **D1000001** (also known as VTAM ID)CICS
terminal ID: **D000**

For a remote display station:
VTAM
netname: **A0208001** (also known as VTAM ID)CICS
terminal ID: **R001**

Terminal definitions are defined in the VTAM books and in the CICS System Definition (CSD) file. Terminal attachments and the VTAM and CICS Transaction Server relationships are shown in the manual *z/VSE SNA Networking Support* under "VTAM and CICS Transaction Server Parameters".

If you use TCP/IP to access VTAM, the Telnet daemon of TCP/IP will interact with the VTAM terminals that are defined as described above.

Recognizing Suspended Online Activities

Usually, a display station user recognizes that communication with the system is no longer possible. This may affect one, several, or all display stations attached.

What You Should Consider First

1. A suspended online system may show the same error symptoms as a **LOOP or WAIT** condition. Therefore, make sure that no such condition exists. If you entered this section ("Suspended Online Activities") directly via the Error Symptom Tables, you did not refer to the LOOP or WAIT section. For a possible LOOP or WAIT condition refer to "Handling Loop Conditions" on page 38 and "Handling Wait Conditions" on page 43.
2. You can use the STATUS command to check the status on your system console: issue **STATUS** *xx* (where xx is the partition). You can obtain other detailed information using the STATUS command: see
   - "Using the STATUS Command" on page 136.
   - Manual *z/VSE System Control Statements*.
3. A display station may show information important for identifying the cause of the problem. Status information may be displayed on the status line at the bottom of the screen. In the bottom left-hand corner, connection indicators may be displayed. Record any error information displayed.
4. A display station user should also use the diagnostic guide of the terminal used, if available. It is usually attached under the cover of the keyboard.
5. Consider the possibility that startup processing was not correct. Check the system console messages that were issued during startup.

Error Diagnosis Steps

If **one or several display stations are not working**, continue with "Some Display Stations not Working" on page 49, otherwise, continue here.

All Display Stations not Working

Such a situation may also be caused by an unusual system status or activity:

- Is the system in the process of writing a dump? Usually, the writing of a dump is indicated by a message. Check the messages on the system console screen. If necessary, re-display previous messages. If a dump is being written, let it finish and the problem will be gone. **The writing of a dump may take several minutes**.
- Is a shutdown of the CICS Transaction Server in progress? A short while after shutdown started, further online activity is inhibited.
- Did a full condition occur for the VSE/POWER queue, data, or account file? Messages like 1Q32I, 1Q38I, or 1QF4I indicate such a condition. Refer to Chapter 7, "Handling Resource Constraint Conditions," on page 103 if such a condition occurs.

If none of the above conditions applies, check whether the CICS Transaction Server partition ever came up:

- Was message

  ```
  F2 ... DFHSI1517  DBDCCICS Control is being given to CICS
  ```

  displayed on the system console? In the input line (==>), type

  ```
  F2
  ```

  to display the messages of the F2 partition at your console.

- Is the *z/VSE Online* panel for signing-on shown on display station screens?

If not, you have a startup problem; proceed as suggested in "CICS Transaction Server Startup Problem" on page 24. Especially watch out for the CICS Transaction Server message

```
DFHSI1572 APPLID  Unable to OPEN VTAM ACB ...
```

which points to conflicting VTAM and CICS definitions.

If the CICS Transaction Server partition did come up, continue with "What About Your VTAM Environment?" on page 52.

Some Display Stations Not Working

Proceed as follows:

1. Use a display station that is working and sign on as the administrator. Access the *Problem Handling* panel. Select:

   ```
   2  (Inspect Message Log)
   ```

   You get a display of the messages stored in the message log file. At the bottom of the last screen you may find a few lines describing why the display station(s) failed.

   For example, if the terminal ID is given as ????, the terminal is not defined in the CICS System Definition (CSD) file. Other explanations may require VTAM expertise.
2. Check the system console messages issued by VTAM during startup. They may reveal the problem.

   If you use TCP/IP to access VTAM, you might also receive TCP/IP error messages on the system console. They may reveal the problem. For explanations of such messages, refer to the chapter "Messages relating to Telnet (TEL)" in the manual *TCP/IP for VSE, Messages and Codes*. This manual is contained in the *TCP/IP for VSE/ESA* shelf (PDFs only) of the VSE Collection kit, SK2T-0060.
3. Find out whether all affected display stations are attached to the same line (control unit).

If (1) The above steps do not provide a clue to the problem, and (2) There are no error messages issued from TCP/IP to the system console (if you use TCP/IP), check the VSE/ICCF and VSE/POWER environment next. Ensure that they work correctly before you continue with the more complex CICS Transaction Server and VTAM environments.

VSE/ICCF: What About Your VSE/ICCF Environment ?

Proceed as follows:

- Check whether all interactive partitions are occupied. At the system console enter the following command:

  ```
  /DISPLAY
  ```

  Repeat the command a few times. If interactive partitions remain "IN USE" indefinitely, you must track down the related display station users. Enter the system console commands /MAP and /USERS. The output gives the IDs of the respective USERs and TERMs (display stations). Get in touch with the other users and ask them to finish their work.
- Your display station may be locked because the program you are running in an interactive partition entered a loop or a wait state. To cancel the program, press

the PA2 key at your display station. You are now able (in most cases) to enter the VSE/ICCF DUMP command. For error diagnosis, you can display general registers, floating point registers, and program storage areas. For further details about the VSE/ICCF DUMP command, refer to the manual *VSE/ICCF User's Guide*.

- A display station may also be locked because the /WARN command was issued for announcing a planned shutdown. To reset the condition, you may issue the /WARN RESET command.

VSE/POWER: What About Your VSE/POWER Environment ?

Check whether the display station in error is logged on to VSE/ICCF and did any work involving VSE/POWER. This could be the submission of a batch job or requesting the display of the VSE/POWER queues. Review the messages at the system console and take proper action, if necessary. Issue the following commands:

```
D A       (to list active VSE/POWER tasks)
D M       (to list VSE/POWER messages)
D Q       (to list VSE/POWER files status)
D STATUS  (to display statistics about the current VSE/POWER session)
```

If the VSE/POWER queue is full, proceed as suggested in section "VSE/POWER File Full Condition" on page 103.

CICS TS: What About Your CICS Transaction Server Environment ?

1. Use the dialog fastpath **364** *Display CICS TS Storage* to check how storage is being used. Especially check if a SOS (short on storage) condition exists.
2. Use the *Inspect Message Log* dialog and check for CICS messages (prefix DFH) related to the terminals in error.
3. Use the following commands to verify the CICS Transaction Server and VTAM environment.
   - For the CICS Transaction Server, use the CICS master terminal function CEMT. Enter the following commands either from the system console or from a display station. From the system console, you must first issue the command:

     ```
     MSG F2
     ```

     The CICS Transaction Server then provides you with a *reply ID* (represented by 'nn' in the following example). Next, enter the command:

     ```
     nn CEMT I TERM
     ```

     This command shows you whether a terminal is in the CICS status *released* or *acquired*.
   - For VTAM, enter at the system console:

     ```
     D NET,TERMS
     ```

     This command shows you whether a terminal is in the VTAM status of ACTIVE or attached to a CICS session (ACT/SES). Compare the data provided by the commands.

   If the data does not provide a clue to the problem, proceed with the next step.
4. Find out whether a **task** is attached to the display station in error; issue the command

   ```
   nn CEMT INQUIRE TASK
   ```

   The system displays all tasks together with the associated display stations. A line within the display might look as follows:

```
Tas(00087) Tra(BLUE) Fac(D080) Run Ter Pri(001)
```

Use this information for the following steps. The two important parameters in this context are:

Tas The number the CICS Transaction Server has assigned to the task.

Fac The device that activated the task:

CNSL = the system console, nnnn = a specific display station (D080, for example).

The reply you get for each task includes the parameters **Run** and **Sus**. **Run** indicates that the task is active and **Sus** that the task is suspended.

a. **If a task is attached to the display station in error**, one of the following conditions might exist:

   1) A task is in a loop.

   2) A task has gone into an indeterminate WAIT.

   If a LOOP is suspected, repeat several times:

   ```
   nn CEMT INQUIRE TASK
   ```

   Exclude transaction **IESO** (required by z/VSE) from the following steps:

   - On each display you get, look for the task with the lowest number and whether its status is active. If that task continues to be the task with the lowest number, it is probably the task that is looping.
   - Try to determine whether this task is running correctly. If the task is not running correctly and if it can be safely purged, enter

     ```
     nn CEMT SET TASK(87) PURGE
     ```

     to purge the task.

     Note: Be sure that the task is in a loop. If you are in doubt, contact the owner or programmer of the application.

   If a WAIT is suspected, do the following:

   Take note of the task number of the TAS parameter. Terminate the task by entering:

   ```
   nn CEMT SET TASK(87) PURGE
   ```

   To ensure that the task did terminate, enter:

   ```
   nn CEMT INQUIRE TASK
   ```

   If the task has not yet terminated, set the display station OUT (out of service) and then INS (into service). Enter:

   ```
   nn CEMT SET TERM(D080) OUTSERVICE PURGE
   nn CEMT SET TERM(D080) INSERVICE
   ```

b.

   If the problem remains or **no task is attached to the display station in error**, the display station itself may be the cause of the problem. Find out whether the display station is INS (in service) by entering:

   ```
   nn CEMT INQUIRE TERM(D080)
   ```

   Verify that the display station is connected to the CICS Transaction Server.

   The system's response to the CEMT INQUIRE TERM command will tell you:

   - Acq (short for Acquired) means connected.
   - Rel (short for Released) means not connected.

   Try to connect a non-connected display station by entering:

   ```
   nn CEMT SET TERM(D080) INSERVICE ACQ
   ```

5. Use DFH0STAT (provided in ICCF library 59) to obtain shutdown statistics. These statistics help to determine how resources such as LSR buffers/files and CICS temporary storage are used.

Note: If you did not succeed in removing the error, check VTAM and CICS definitions as described in "VTAM Startup Problem" on page 26. If the definitions are correct, return to here and continue with "What About Your VTAM Environment?" below.

TCP/IP: What About Your TCP/IP Environment ?

For commands and procedures related to TCP/IP, refer to the IBM manual *z/VSE TCP/IP Support*.

VTAM: What About Your VTAM Environment ?

When you use VTAM commands, you specify the identification of the display station in error via the ID parameter, called the **VTAM ID**. The VTAM ID is identical to the **netname** specified for a terminal. VTAM provides a naming convention for a networking environment that uses a 4-digit subarea number. The manual *z/VSE SNA Networking Support* describes this naming convention in detail.

Use your installations configuration list (refer to "Handling Situations Where Online Activities Are Suspended" on page 46) for the following steps. Proceed as follows:

1. At the system console, enter VTAM commands to determine the status of the display stations. Use the following commands:
   a.
      ```
      F NET,SUPP=NOSUP
      ```
      This command causes all error messages to be displayed on the screen, instead of just those defined as 'serious'.
   b.
      ```
      D NET,TERMS
      ```
      This command displays the status of all display stations, terminal printers, and RJE work stations. You can display the status of a single terminal by adding the ID of the terminal in question. For example:
      ```
      D NET,TERMS,ID=D0800001
      ```
      If the response indicates that a terminal is in normal condition, the following two steps are not necessary.
   c.
      ```
      V NET,INACT,ID=xx...xx,F
      ```
      Where xx...xx is either the VTAM ID (netname) of a single terminal or a node name. Refer also to "Naming Convention for Terminals" on page 47.

      This command sets the display station(s) to inactive.
   d.
      ```
      V NET,ACT,ID=xx...xx
      ```
      This command activates the display station(s) again. Within a few seconds, the display station(s) should be ready for sign-on.

      Watch for any error messages after you entered the command. They may give you a clue to the problem.

When you use a node name for deactivation, you deactivate **all** terminals of your installation that belong to this node. To avoid unnecessary interruption at your installation, use for xx...xx first:

- If local, the VTAM ID (netname) of the terminal in error. D08001, for example.
- If remote, the VTAM ID (netname) of the terminal in error. A0208001, for example.

Refer also to “Naming Convention for Terminals” on page 47.

2. For **local SNA or non-SNA terminals** make sure that the control unit(s) are powered on. Check that the Online/Offline switch of the control unit(s) is set to ONLINE. Enter commands as shown in step 1.
3. For **remote terminals (NCP connected)** make sure the IBM 37xx communications controllers are powered on and check that the NCP is loaded. Issue the commands as shown in step 1. You may get the following message:

```
xxxxx LUB SYS005 NOT AVAILABLE
```

This means that there is not enough space available in the VTAM partition to load the NCP into it before it is loaded into an IBM 37xx. You have two choices:

a. You may try to load the NCP from a sequential file. There are a number of skeletons in library 59 of VSE/ICCF that provide support for this task. The names of these skeletons start with SKN. The skeletons are described in the manual *z/VSE SNA Networking Support* under “NCP Load Module File Skeleton SKNCPCLF”, “ACF/SSP Independent Loader Utility Skeleton SKNCPLD”, and “NCP Generation Workfile Skeleton SKNCPWKF”.

   If you were able to successfully load the NCP, repeat the commands of step 1.

b. Otherwise, use the NCP generation list to find out the size of the NCP used. Increase the partition allocation for the VTAM partition (F3, usually). Change the appropriate startup book accordingly.

If you have a problem in the initial startup of the NCP, check the generation list. If everything looks correct, enter

```
D NET,PATHTAB
```

and print the messages from the hardcopy file for the VTAM partition (F3, usually). Enter:

```
R RDR,PAUSEBG
0 // EXEC PRINTLOG
0 xx,NEW
```

where *xx* is the VTAM partition. With parameter NEW you get all messages since the last IPL.

To restrict the amount of output, you may specify a date instead (MM/DD/YYYY). For details on the parameters of the PRINTLOG program refer to manual *z/VSE System Utilities* under “Printing the Hardcopy File (PRINTLOG)”.

Return the output to the NCP programmer, who should compare the specifications with the NCP source.

CU/CC Setup Correct: If the above steps showed that the Control Unit or Communications Controller setup is correct and you still have no sign-on panel, continue as follows.

1. Set the **TEST** switch of the terminal in error OFF and ON again. The panel with the remote applications should appear on your screen. If you do not get that panel, continue with step 3.
2. **If you get the remote applications panel**, enter

   ```
   A  (DBDCCICS)
   ```

   to get the sign-on panel. DBDCCICS is the IBM-provided application ID of the CICS Transaction Server. **If you get the sign-on panel**, your problem is gone.

   If you do not get the sign-on panel, z/VSE may redisplay the remote applications panel or display a message such as:

   - COMMAND ACCEPTED

     Just wait for completion.
   - LOGON PENDING

     There is already a request outstanding between your display station and z/VSE.
   - APPLICATION NOT ACTIVE

     The application you requested is not available. It may be in the process of being started.
   - LOGON FAILED

     VTAM passed your logon (sign-on) request to the remote application but either the program or VTAM rejected it. Inspect the message log file. From the *Problem Handling* panel select:

     ```
     2  (Inspect Message Log)
     ```
   - UNSUPPORTED FUNCTION

     Treat it as LOGON FAILED.

   If you do not get any of the above messages or no remote application panel, check table **USSTAB** for wrong or missing definitions. If necessary, correct table USSTAB.

   As a short-range solution you might try to force a display of the sign-on panel. Enter the following command:

   ```
   LOGON APPLID(DBDCCICS)
   ```

   where DBDCCICS is the IBM-provided application ID of the CICS Transaction Server.

   If you do not succeed, go to step 4 if it is a remote display station, otherwise to step 5.
3. **If you did not get the remote applications panel**, enter:

   ```
   IBMTEST
   ```

   As a response, message IBMECHO should appear on your screen. If so, it indicates a network definition (USSTAB) error. Check table USSTAB for wrong or missing definitions.

   If message IBMECHO does not appear and if it is a remote display station, go to the next step. Otherwise, continue with step 5.
4. If the display station in error is a **remote** display station connected to a control unit, VTAM may inform you through a message that a higher node is in error. For example:

   ```
   IST608I  VARY ACT FOR ID=A0208001 FAILED  -  HIGHER NODE
            DPA08001 NOT ACTIVE
   ```

   Trying to set this node (the control unit) ACTIVE could lead to yet another such message, this time for the next higher node (the line).

You must search ‘upwards’ until you find the unit (node) that is causing the problem. Then set to ACTIVE the appropriate nodes, proceeding in sequence from the highest to the lowest. Finally, set the display station back to INSERVICE:

```
nn CEMT SET TERM(R001) INSERVICE ACQUIRE
```

R001 is an assumed CICS terminal ID for terminal A0208001. Refer also to “Naming Convention for Terminals” on page 47.

If you were unable to activate any of the higher nodes, check cables, plugs, modems (and the like). If this does not help, issue the command:

```
D NET,ID=A0208001,E
```

Request a listing of the hardcopy file (as described in “Printing the Hardcopy File” on page 113). Continue with the next step.

5. You could not identify the cause of the problem. From the system console, request two partition dumps by entering

```
DUMP pp,cuu   (for the VTAM partition, usually F3)
DUMP pp,cuu   (for the CICS Transaction Server partition, usually F2)
```

where *pp* is the partition ID and *cuu* is the address of the tape unit or printer.

Try to bring the running applications to an orderly completion. Shut down the system and re-IPL. Analyze the dumps as described in Chapter 6, “Using Dumps for Error Diagnosis,” on page 71. If necessary, contact IBM for support.

Data You Should Collect in Case of Suspended Online Activities

Before you contact IBM for support have the following available for error diagnosis:

- System console log (from the hardcopy file).
- Partition dumps.
- Network generation list.
- Assembler listings of programs or applications involved.
- Compiler listings of programs or applications involved.

Chapter 5. Online Analysis of CICS Transaction Abends

A CICS transaction may stop processing because of an error. This is called an **abend** - the short form for 'abnormal end'. When this happens, the **Online Problem Determination (OLPD)** program becomes active. It collects and saves important data related to the error situation.

Diagnostic data is obtained from the:
- CICS TS transaction environment.
- LE/VSE programming environment. If an LE/VSE abnormal termination occurs, the collected diagnostic data is stored in a separate incident report.

OLPD analyzes the data collected and, to the extent possible, describes the cause of the error in plain language. Additional data may be displayed for analysis by the user. The data saved enables you to do online error diagnosis. You can display that data immediately after an abend occurred or later. The data is available until you or the system delete it. **Data that is older than seven days is deleted automatically during system startup**.

System Actions after a Transaction Abend

When z/VSE detects an abend condition it does the following:
1. Collects, saves, and analyzes error data.
2. Writes a transaction dump to the CICS Transaction Server dump file.

   This information can be used in addition to the information provided by the online problem determination program. Moreover, the system may not always be able to collect and save data in case of a transaction abend. The transaction dump is then the only source for a detailed error analysis.
3. Saves the panel in use at the time of the error. This panel is referred to as the **user screen**.
4. Notifies the user about the error by displaying a panel. This panel gives basic information about the transaction abend.
5. Redisplays the user screen together with message DFHAC2206I.

   Note: In exceptional situations, the system may not be able to redisplay the original user screen at all, or perhaps not exactly as it was when the error occurred.

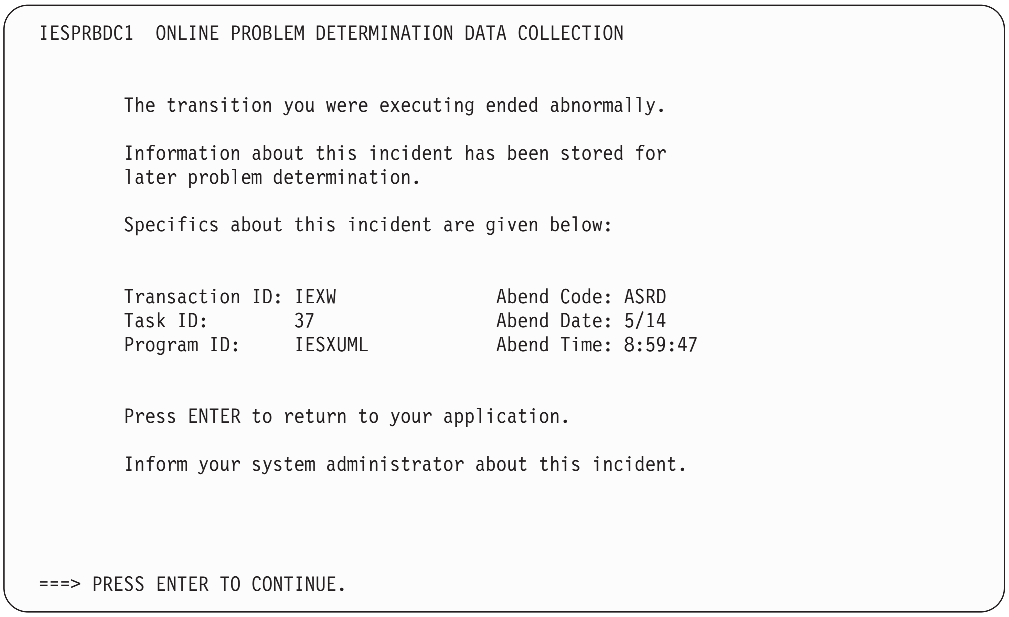
Notifying the User About a Transaction Abend

A transaction abend and its related data is recorded by the system. To notify you about such an incident, the *Online Problem Determination Data Collection* panel is displayed on your screen.

Normally, an abend is identified by an abend code. In the example shown (Figure 4 on page 58), transaction IEXW finished with abend code ASRD. Online problem determination is provided for almost all CICS transaction abend codes and LE/VSE abend codes. CICS transaction abend codes are documented in the *z/VSE Messages and Codes* manual.

Note:

1. For technical reasons, online problem determination is not provided for these abend codes:
   AKCP
   ATPx
   Other abend codes that cause the CICS TS to terminate.
2. LE/VSE may suppress the generation of the standard incident report that is created during transaction abend processing.



```
IESPRBDC1  ONLINE PROBLEM DETERMINATION DATA COLLECTION


        The transition you were executing ended abnormally.

        Information about this incident has been stored for
        later problem determination.

        Specifics about this incident are given below:


        Transaction ID: IEXW              Abend Code: ASRD
        Task ID:        37                Abend Date: 5/14
        Program ID:     IESXUML           Abend Time: 8:59:47


        Press ENTER to return to your application.

        Inform your system administrator about this incident.




===> PRESS ENTER TO CONTINUE.
```

Figure 4. Notification Panel for a Transaction Abend

Whenever such a panel is displayed on your screen, read it carefully and follow the instructions given. In the following discussion, this panel is referred to as notification panel. Information given for **transaction ID, task ID, program ID, abend code, abend date, and abend time,** are highlighted on the screen.

In this example, the redisplayed user screen (after you pressed enter) would show the following message:

```
DFHAC2206 8:59:47 DBDCCICS Transaction IEXW has failed with abend ASRD
```

How to Do Online Problem Determination

To find the cause of an abend, you have to interpret the data collected and stored for a specific incident. This requires CICS skill and experience. As far as possible, the cause of the error is described in plain language. However, it is sometimes necessary to analyze control block data. Usually, the programmer of a failing transaction, or the system administrator interprets and analyzes the incident data.

The incident data as it appears on your screen is referred to as the **incident report**. You can display an incident report in two ways:

- Directly, after an abend occurred (usually if you are a programmer).
- Through the initial panel of the OLPD program (for the administrator or a programmer).

Displaying an Incident Report Directly after an Abend Occurred

Assume that Figure 4 on page 58 represents an abend that occurred when testing a program. To display the incident report right after the abend occurred, proceed as follows:

1. After copying important data from the redisplayed user screen, clear your screen.
2. Enter **OLPD**.

Your screen now shows the first page of the incident report (Figure 5 on page 61). For details on the incident report refer to “Structure and Contents of the Incident Report” on page 60.

Displaying an Incident Report During Normal Processing

Through the initial panel of OLPD you can display any incident report stored. To invoke the initial panel, select from the *Problem Handling* panel:

```
1  (Online Problem Determination)
```

To display an incident report, the initial panel offers two choices:

- Option 1 (Display the user's last incident).
- Options 3 and 4 (List incidents; either of a single user or of all users). From such a list you can select any incident for display.

Follow these steps:

For Option 1:

1. Change the **user ID** (shown on the initial panel), if you want to display an incident report other than your own.
2. Select option **1** (Display the user's last incident).

Your screen now shows the first page (Figure 5 on page 61) of the incident report requested.

For Options 3 and 4:

1. Change the **user ID** if you want to display an incident report other than your own.

   Select option **3** (List the user's incidents), or option **4** (List all incidents for all users).
2. From the *List Display* panel displayed, select the incident you are looking for. Enter:

   ```
   1 (Display)
   ```

   in the option column. Refer to Figure 14 on page 67 for an example of a *List Display* panel.

Your screen now shows the first page (Figure 5 on page 61) of the incident report requested.

Structure and Contents of the Incident Report

An incident report consists of several pages. The text of each page is divided into one or more paragraphs. The number of pages and paragraphs varies from error to error. For each paragraph you can request the display of additional information. This additional information, called ‘level 2 information’, is of a more explanatory nature.

Table 2 shows the structure of the incident report for the example presented in Figure 4 on page 58. It includes 9 pages of level 1 information and 13 pages of level 2 information.

Table 2. Level Structure of Incident Report

LEVEL 1 INFORMATION		
Page No.	**Paragraph No.**	**LEVEL 2 INROFMATION**
1	1	2 pages
	2	
2	1 (only)	1 page
3	1 (only)	1 page
4	1 (only)	5 pages

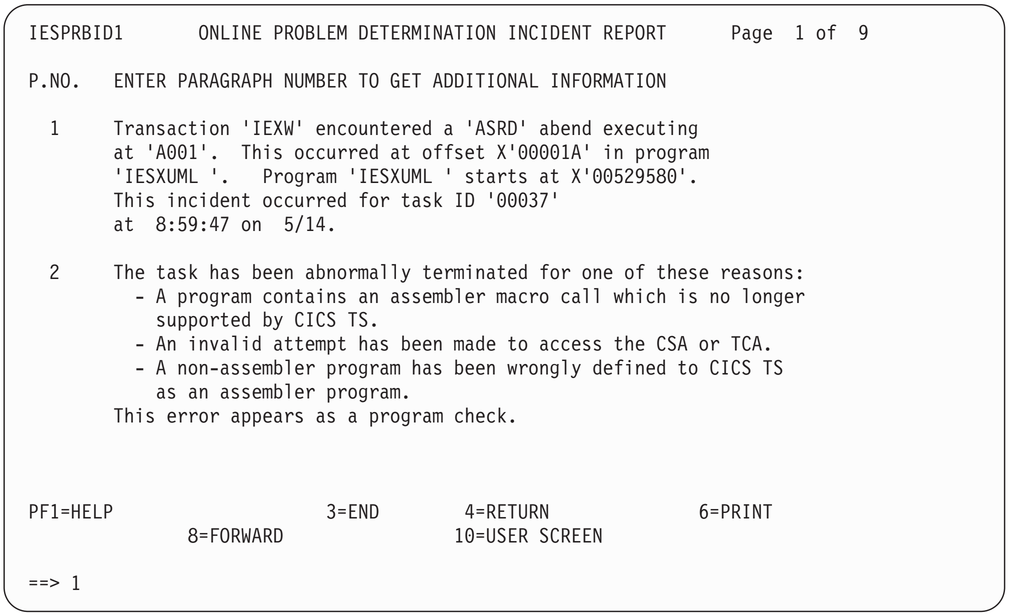
The following paragraphs discuss the first four pages of level 1 information and give one example of level 2 information.

Note:

1. Most panels of an incident report allow you to use PF10 for redisplaying the user screen associated with the particular abend. To get back to the incident display screen, just press any PF key, the CLEAR, or the ENTER key.
2. The panel sequence of an incident report may not be the same for each abend.
3. Incident reports created during LE/VSE abnormal termination processing have a different structure than the standard report. This is described in “Structure and Contents of an LE/VSE Incident Report” on page 63.

First Page of Incident Report

Figure 5 on page 61 shows the first page of the incident report.

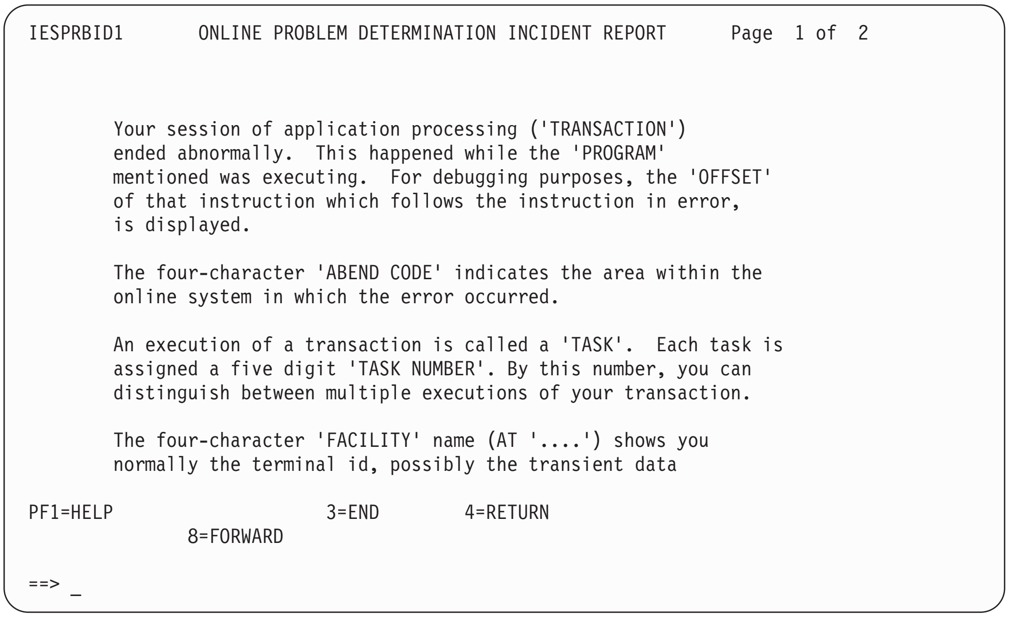


```
IESPRBID1       ONLINE PROBLEM DETERMINATION INCIDENT REPORT      Page  1 of  9

P.NO.   ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

  1     Transaction 'IEXW' encountered a 'ASRD' abend executing
        at 'A001'.  This occurred at offset X'00001A' in program
        'IESXUML '.   Program 'IESXUML ' starts at X'00529580'.
        This incident occurred for task ID '00037'
        at  8:59:47 on  5/14.

  2     The task has been abnormally terminated for one of these reasons:
          - A program contains an assembler macro call which is no longer
            supported by CICS TS.
          - An invalid attempt has been made to access the CSA or TCA.
          - A non-assembler program has been wrongly defined to CICS TS
            as an assembler program.
        This error appears as a program check.



PF1=HELP                    3=END        4=RETURN              6=PRINT
               8=FORWARD                10=USER SCREEN

==> 1
```

Figure 5. Incident Report (Level 1, Page 1 of 9)

Paragraph 1 summarizes the error situation. Part of this information was displayed on the notification panel (Figure 4 on page 58). The other paragraph describes the cause of the error as far as it could be determined.

To display second level information, enter the paragraph number (**1**), as shown in Figure 5. This gives you the first page of second level information for paragraph **1**. This page is shown in Figure 6.

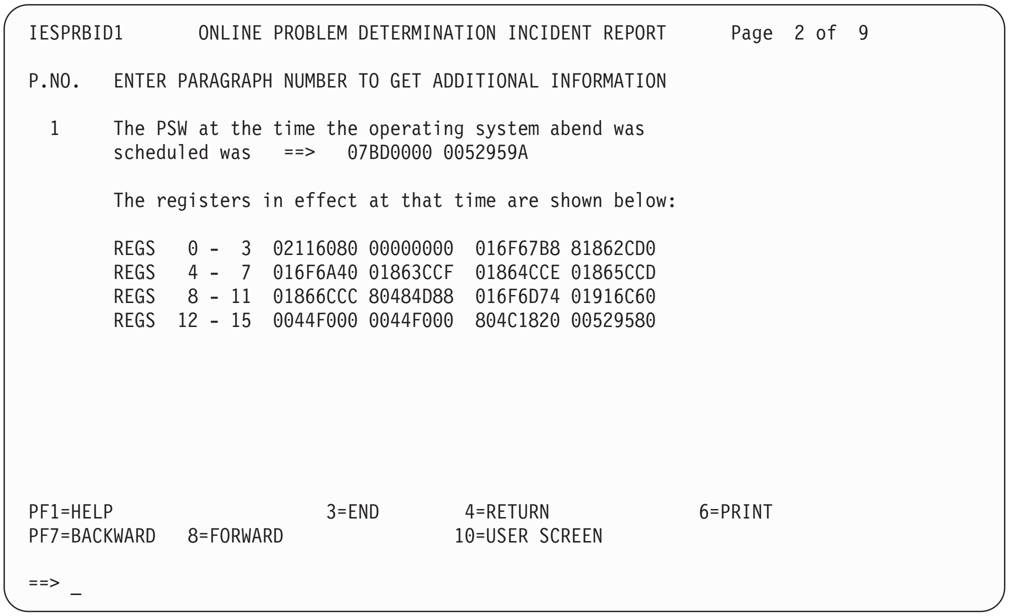


```
IESPRBID1       ONLINE PROBLEM DETERMINATION INCIDENT REPORT      Page  1 of  2



        Your session of application processing ('TRANSACTION')
        ended abnormally.  This happened while the 'PROGRAM'
        mentioned was executing.  For debugging purposes, the 'OFFSET'
        of that instruction which follows the instruction in error,
        is displayed.

        The four-character 'ABEND CODE' indicates the area within the
        online system in which the error occurred.

        An execution of a transaction is called a 'TASK'.  Each task is
        assigned a five digit 'TASK NUMBER'. By this number, you can
        distinguish between multiple executions of your transaction.

        The four-character 'FACILITY' name (AT '....') shows you
        normally the terminal id, possibly the transient data

PF1=HELP                    3=END        4=RETURN
               8=FORWARD

==> _
```

Figure 6. Incident Report (Level 2 Information for Paragraph 1 of Page 1)

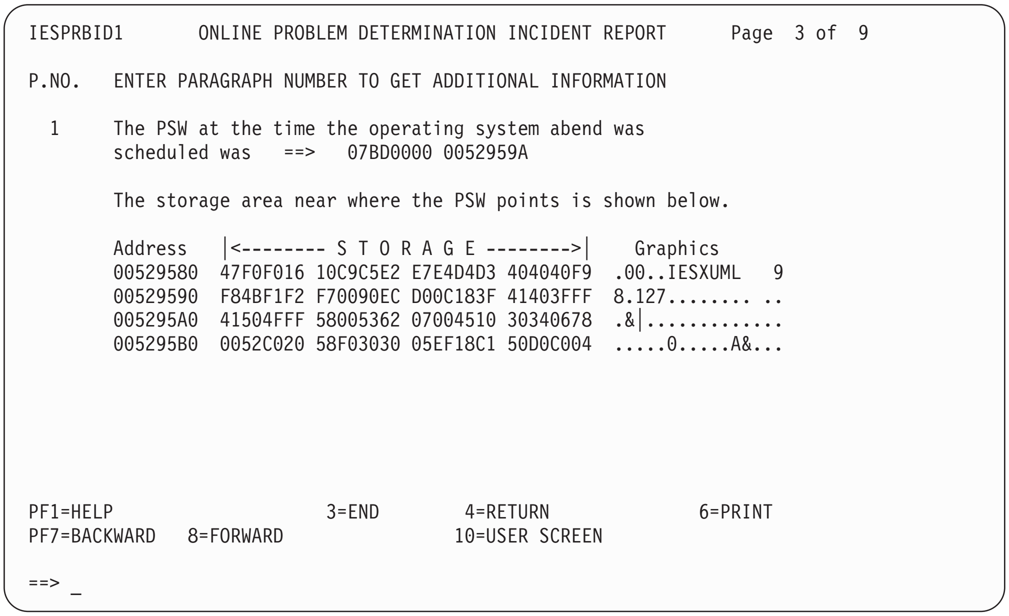
Second and Third Page of Incident Report

For this type of transaction abend, the contents of the registers and selected storage areas are shown in Figure 7 and Figure 8.



```
IESPRBID1       ONLINE PROBLEM DETERMINATION INCIDENT REPORT      Page  2 of  9

P.NO.   ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

  1     The PSW at the time the operating system abend was
        scheduled was   ==>   07BD0000 0052959A

        The registers in effect at that time are shown below:

        REGS   0 -  3  02116080 00000000  016F67B8 81862CD0
        REGS   4 -  7  016F6A40 01863CCF  01864CCE 01865CCD
        REGS   8 - 11  01866CCC 80484D88  016F6D74 01916C60
        REGS  12 - 15  0044F000 0044F000  804C1820 00529580









PF1=HELP                    3=END        4=RETURN              6=PRINT
PF7=BACKWARD   8=FORWARD                10=USER SCREEN

==> _
```

Figure 7. Incident Report (Level 1, Page 2 of 9)



```
IESPRBID1       ONLINE PROBLEM DETERMINATION INCIDENT REPORT      Page  3 of  9

P.NO.   ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

  1     The PSW at the time the operating system abend was
        scheduled was   ==>   07BD0000 0052959A

        The storage area near where the PSW points is shown below.

        Address   |<-------- S T O R A G E -------->|    Graphics
        00529580  47F0F016 10C9C5E2 E7E4D4D3 404040F9  .00..IESXUML   9
        00529590  F84BF1F2 F70090EC D00C183F 41403FFF  8.127........ ..
        005295A0  41504FFF 58005362 07004510 30340678  .&|.............
        005295B0  0052C020 58F03030 05EF18C1 50D0C004  .....0.....A&...







PF1=HELP                    3=END        4=RETURN              6=PRINT
PF7=BACKWARD   8=FORWARD                10=USER SCREEN

==> _
```

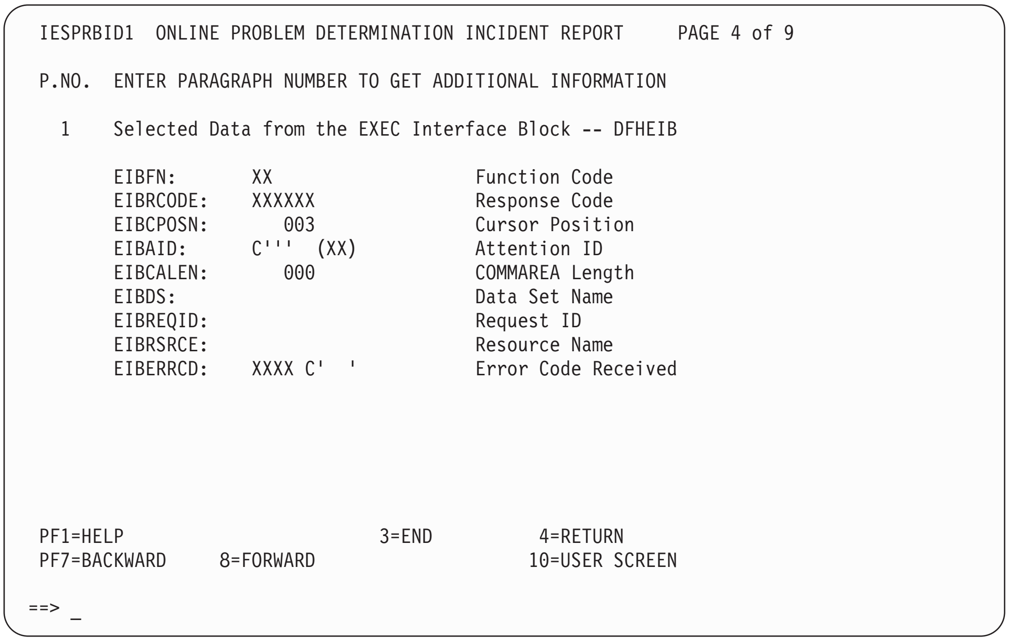
Figure 8. Incident Report (Level 1, Page 3 of 9)

Fourth Page of Incident Report

Depending on the type of the failing transaction, the fourth page of this incident report displays selected EXEC Interface Block information (Figure 9).

The EXEC Interface Block

The EIB contains important information about the current activity of the task at the time of the abend. In the example used, the EIB information is not really important for identifying the cause of the error. But this depends on the type of error.



```
 IESPRBID1  ONLINE PROBLEM DETERMINATION INCIDENT REPORT     PAGE 4 of 9

 P.NO.  ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

   1    Selected Data from the EXEC Interface Block -- DFHEIB

        EIBFN:       XX                  Function Code
        EIBRCODE:    XXXXXX              Response Code
        EIBCPOSN:       003              Cursor Position
        EIBAID:      C'''  (XX)          Attention ID
        EIBCALEN:       000              COMMAREA Length
        EIBDS:                           Data Set Name
        EIBREQID:                        Request ID
        EIBRSRCE:                        Resource Name
        EIBERRCD:    XXXX C'  '          Error Code Received






 PF1=HELP                        3=END          4=RETURN
 PF7=BACKWARD     8=FORWARD                   10=USER SCREEN

==> _
```

Figure 9. Incident Report (Level 1, Page 4 of 9)

Note: Page 5 of the incident report is not shown. It gives additional four lines of EIB information. In this example, the information is zero for all four lines.

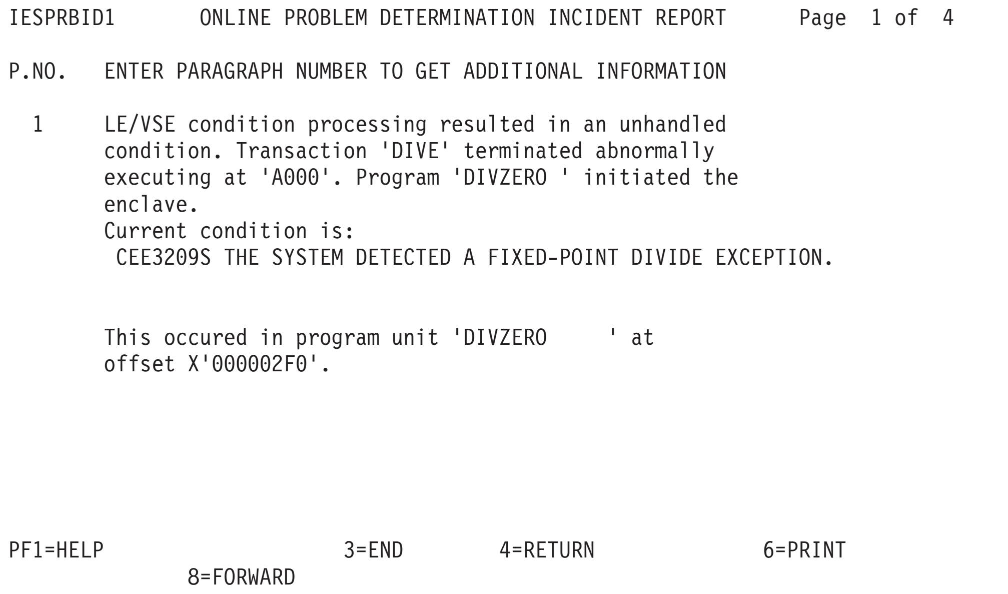
Use the level 2 information provided for details on the EIB.

For a detailed analysis of the EIB information, refer to the manual *CICS Application Programming Reference*. The *CICS User's Handbook* may also be helpful.

Structure and Contents of an LE/VSE Incident Report

An LE/VSE incident report consists of four pages. The text of each page is divided into one or more paragraphs. The number of pages and paragraphs varies from error to error.

Figure 10 on page 64 shows the first page of an LE/VSE incident report.

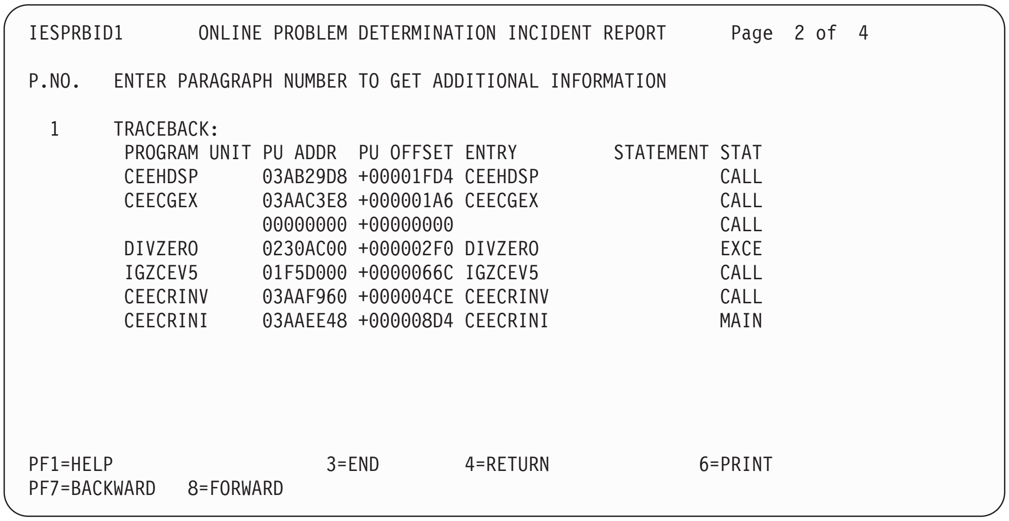


```
IESPRBID1       ONLINE PROBLEM DETERMINATION INCIDENT REPORT      Page  1 of  4

P.NO.   ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

  1     LE/VSE condition processing resulted in an unhandled
        condition. Transaction 'DIVE' terminated abnormally
        executing at 'A000'. Program 'DIVZERO ' initiated the
        enclave.
        Current condition is:
         CEE3209S THE SYSTEM DETECTED A FIXED-POINT DIVIDE EXCEPTION.


        This occured in program unit 'DIVZERO     ' at
        offset X'000002F0'.






PF1=HELP                    3=END        4=RETURN              6=PRINT
               8=FORWARD
```

Figure 10. LE/VSE Incident Report (Page 1 of 4)

In Figure 10, paragraph 1 provides a summary of the error situation. To display the second level information, you would enter the paragraph number (1). The report shown in Figure 11 is then displayed.

Figure 11 shows the second page of the incident report.

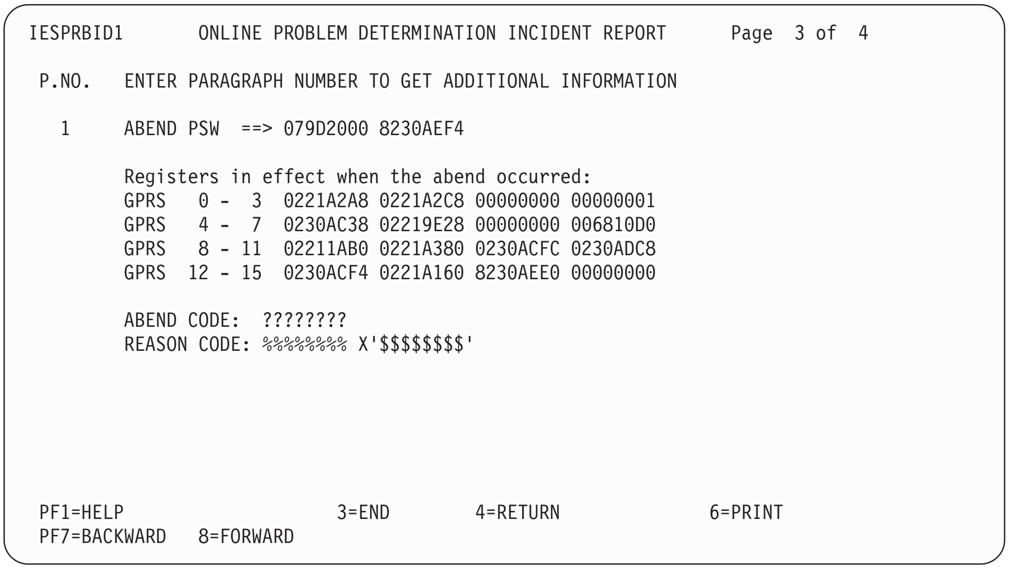


```
IESPRBID1       ONLINE PROBLEM DETERMINATION INCIDENT REPORT      Page  2 of  4

P.NO.   ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

  1     TRACEBACK:
         PROGRAM UNIT PU ADDR  PU OFFSET ENTRY         STATEMENT STAT
         CEEHDSP      03AB29D8 +00001FD4 CEEHDSP               CALL
         CEECGEX      03AAC3E8 +000001A6 CEECGEX               CALL
                      00000000 +00000000                       CALL
         DIVZERO      0230AC00 +000002F0 DIVZERO               EXCE
         IGZCEV5      01F5D000 +0000066C IGZCEV5               CALL
         CEECRINV     03AAF960 +000004CE CEECRINV              CALL
         CEECRINI     03AAEE48 +000008D4 CEECRINI              MAIN





PF1=HELP                    3=END        4=RETURN              6=PRINT
PF7=BACKWARD   8=FORWARD
```

Figure 11. LE/VSE Incident Report (Level Page 2 of 4)

In Figure 11, the traceback section shows a sequential list of all routines that were active when the exception occurred. This second level information describes the use of the traceback in more detail.

Additional diagnostic data might be supplied for certain types of error, as illustrated by Figure 12 on page 65.

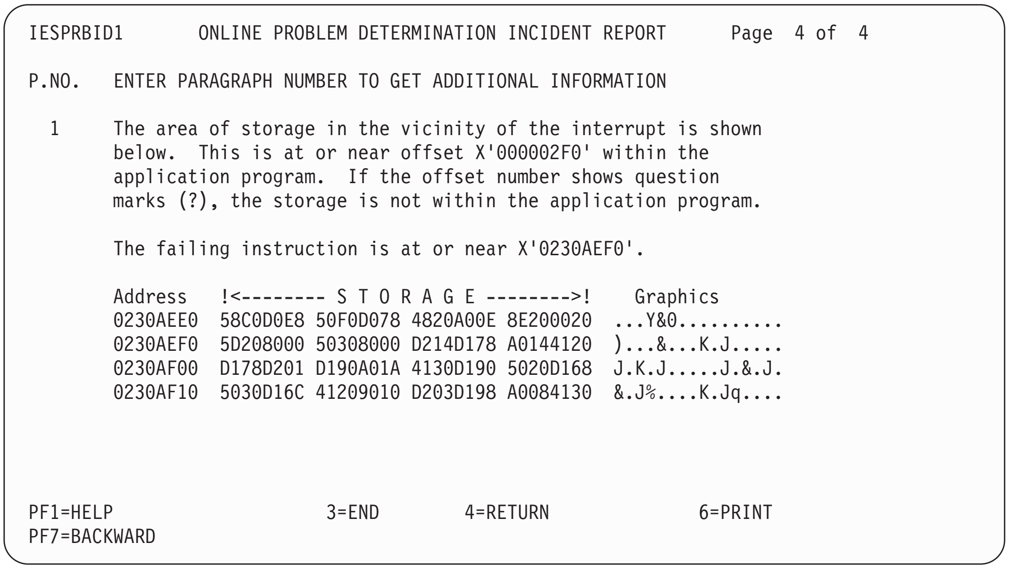


```
IESPRBID1       ONLINE PROBLEM DETERMINATION INCIDENT REPORT      Page  3 of  4

 P.NO.   ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

   1     ABEND PSW  ==> 079D2000 8230AEF4

         Registers in effect when the abend occurred:
         GPRS   0 -  3  0221A2A8 0221A2C8 00000000 00000001
         GPRS   4 -  7  0230AC38 02219E28 00000000 006810D0
         GPRS   8 - 11  02211AB0 0221A380 0230ACFC 0230ADC8
         GPRS  12 - 15  0230ACF4 0221A160 8230AEE0 00000000

         ABEND CODE:  ????????
         REASON CODE: %%%%%%%% X'$$$$$$$$'





 PF1=HELP                   3=END       4=RETURN             6=PRINT
 PF7=BACKWARD   8=FORWARD
```

Figure 12. LE/VSE Incident Report (Level Page 3 of 4)

In Figure 12, “condition information” is displayed for the routine that was active at the time the abend occurred. To interpret the data contained in registers 0 to 15, you require a compiler or assembler listing of your program.

Figure 13 shows the fourth page of an LE/VSE incident report.



```
IESPRBID1       ONLINE PROBLEM DETERMINATION INCIDENT REPORT      Page  4 of  4

P.NO.   ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

  1     The area of storage in the vicinity of the interrupt is shown
        below.  This is at or near offset X'000002F0' within the
        application program.  If the offset number shows question
        marks (?), the storage is not within the application program.

        The failing instruction is at or near X'0230AEF0'.

        Address   !<-------- S T O R A G E -------->!    Graphics
        0230AEE0  58C0D0E8 50F0D078 4820A00E 8E200020  ...Y&0..........
        0230AEF0  5D208000 50308000 D214D178 A0144120  )...&...K.J.....
        0230AF00  D178D201 D190A01A 4130D190 5020D168  J.K.J.....J.&.J.
        0230AF10  5030D16C 41209010 D203D198 A0084130  &.J%....K.Jq....




PF1=HELP                   3=END       4=RETURN             6=PRINT
PF7=BACKWARD
```

Figure 13. LE/VSE Incident Report (Level Page 4 of 4)

Reviewing and Maintaining Incident Data

It is usually the system administrator or the programmer responsible for a transaction who reviews incident data for error diagnosis. In addition, the system administrator is responsible for maintaining the file in which the incidents are stored. The dialogs for these tasks are accessed through the initial panel of Online Problem Determination.

Accessing the Initial Panel

To access the initial panel of Online Problem Determination, select from the *Problem Handling* panel:

```
1  (Online Problem Determination)
```

The initial panel provides four options:

- 1 Display the user's last incident
- 2 Delete the user's last incident
- 3 List the user's incidents
- 4 List all incidents for all users

The following paragraphs describe the options in detail.

Using the Display Option

Use the display option for error diagnosis. It allows you to view and analyze the incident report of a user's last incident. For details refer to “Displaying an Incident Report During Normal Processing” on page 59.

The list options (options 3 and 4) also allow you to display an incident report. For details refer to “Displaying an Incident Report During Normal Processing” on page 59.

Using the Delete Option

Use the delete option to erase incidents. Although incident data is stored in wrap-around mode, there may be a need to delete incidents.

Wrap-around mode means, once the file is full, old incidents are overwritten by new ones. However, you may want to keep certain incidents. To prevent them from being overwritten, you have to delete other incidents to free space. During startup the system automatically deletes incidents that are older than seven days.

To delete a user's last incident:

1. Change the **user ID** (shown on the initial panel) if you want to delete an incident other than your own.
2. Select option **2** (Delete the user's last incident).

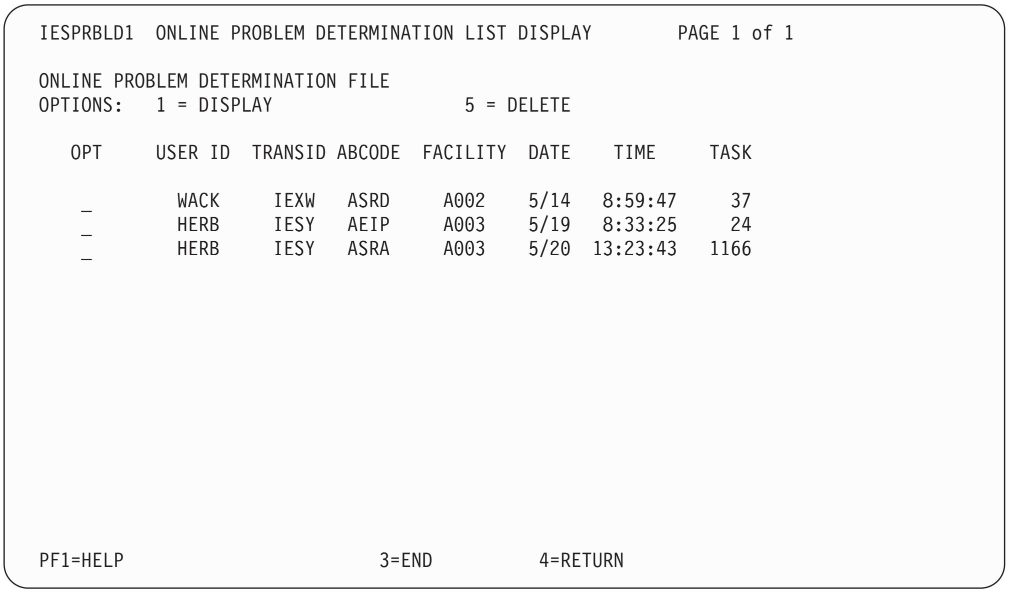
The list options (options 3 and 4) offer an extended delete function. For details refer to the following paragraphs.

Using the List Option

Two list options are available:

- Select option **3** (List the user's incidents) to list all incidents of a single user.
- Select option **4** (List all incidents for all users) to list all incidents for all users.

The panel layout is the same for both list options. Selecting option 4, for example, gives you a List Display panel as shown in Figure 14. The latest incident is always first in the list.



```
IESPRBLD1  ONLINE PROBLEM DETERMINATION LIST DISPLAY       PAGE 1 of 1

ONLINE PROBLEM DETERMINATION FILE
OPTIONS:   1 = DISPLAY                 5 = DELETE

   OPT     USER ID  TRANSID ABCODE  FACILITY  DATE    TIME     TASK

    _        WACK     IEXW   ASRD     A002    5/14   8:59:47     37
    _        HERB     IESY   AEIP     A003    5/19   8:33:25     24
    _        HERB     IESY   ASRA     A003    5/20  13:23:43   1166













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

Figure 14. Example of a List Display Panel

Note: Such a list display may consist of several pages. The example assumes one page.
From this panel you can now select an incident for **display**, or one or more incidents for **deletion**.

- Select option **1** (Display) to display an incident report for error diagnosis.
- Select option **5** (Delete) to delete an incident.

Refer to "Structure and Contents of the Incident Report" on page 60 for a detailed description of an incident report.

Exceptional Error Situations

OLPD may not always be able to analyze a transaction abend. Also, collecting and saving incident data may not always be possible. If this happens, a message is shown on the notification panel (see Figure 4 on page 58). For example:

```
UNABLE TO STORE INFORMATION ABOUT THE INCIDENT.
DIAGNOSTIC INFORMATION HAS BEEN LOGGED.
```

Information related to such an error situation is gathered and stored in the message log file. This file is printed, together with the CICS Transaction Server log, at each CICS Transaction Server shutdown.

For error diagnosis, you can display the file during normal processing. From the *Problem Handling* panel select:

```
2  (Inspect Message Log)
```

You get a display of the messages stored in the message log file. At the bottom of the last screen (if it was the last event) you will find a few lines describing why normal error handling failed. If it was not the last event, review more than just the last few lines.

The CICS Transaction Dump

If an exceptional error situation occurs, the information shown on the notification panel (Figure 4 on page 58) becomes even more important. Write it down and use it to identify the **transaction dump** created by the system. The transaction dump is now the only source available for a detailed error analysis. Refer to “Handling CICS Transaction Dumps” on page 100 for how to print a CICS transaction dump for analysis.

Exceptional errors may also cause a CICS Transaction Server system or partition dump. Refer to Chapter 6, “Using Dumps for Error Diagnosis,” on page 71 for details on how to use such a dump for error diagnosis.

DFHPEP Error Program

z/VSE provides a DFHPEP program for CICS error information. It is available as phase DFHPEP in library IJSYSRS.SYSLIB and as a skeleton in VSE/ICCF library 59. “DFHPEP Error Program” shows the contents of the library skeleton.

You can choose to use DFHPEP for your own error collection and problem handling. If you do, you must decide whether you want your error handling to execute before or after z/VSE collects its error data.

z/VSE does not alter the TCA of the failing transaction. However, because z/VSE uses CICS facilities, it writes entries in the storage chain. *If you do your error handling first and call on CICS facilities, your tracks in the storage chain may affect the accuracy of the incident record*.

- If you want your code to execute first, insert it into DFHPEP before the XCTL.
- If you want your code to execute after z/VSE data collection, change the XCTL to a LINK and place your code after the LINK.

Note: The DFHPEP phase provided by the CICS Transaction Server is called *DFHPEPDY*. It resides in library PRD1.BASE.

```
PEP      TITLE 'CUSTOMER INFORMATION CONTROL SYSTEM    P R O G R A M  E *
               R R O R  P R O G R A M'
         PUNCH ' CATALOG DFHPEP.OBJ REP=YES'
************************************************************************
*                                                                      *
* MODULE NAME = DFHPEP                                                 *
*                                                                      *
* DESCRIPTIVE NAME = CICS TS     PROGRAM ERROR PROGRAM                 *
*                                                                      *
* COPYRIGHT = SEE ABOVE                                                *
*                                                                      *
*                                                                      *
*----------------------------------------------------------------------*
*       THIS MODULE IS UPDATED TO PROVIDE THE HOOK TO THE ONLINE       *
*       PROBLEM DETERMINATION (OLPD) DATA COLLECTOR, IESOPDC.          *
*----------------------------------------------------------------------*
*                                                                      *
DFHEISTG DSECT ,
************************************************************************
* * *          R E G I S T E R   D E F I N I T I O N               * * *
************************************************************************
         DFHREGS ,
         DFHEJECT
***********************************************************************
* * *          D U M M Y   S E C T I O N S                        * * *
***********************************************************************
         DFHPCOM TYPE=DSECT
         DFHEJECT
***********************************************************************
* * * * *              P R O G R A M   E R R O R              * * * * *
* * * * *                    P R O G R A M                    * * * * *
***********************************************************************
DFHPEP   CSECT                     PROGRAM ERROR PROGRAM CSECT
DFHPEP   AMODE 31
DFHPEP   RMODE ANY
         XR    R1,R1
         ICM   R1,B'0011',EIBCALEN GET COMMAREA LENGTH
         BZ    RETURNX             NO COMMAREA; EXIT
         EXEC  CICS ADDRESS COMMAREA(R2) ,
         USING DFHPEP_COMMAREA,R2
         SPACE 3
*---------------------------------------------------------------------*
*       IF THERE IS A NEED TO RESTRICT ACCESS TO NATIVE CICS IN       *
*       CASE OF A TRANSACTION ABEND AND WANT THE VSE/ESA SIGNON       *
*       SCREEN OR THE FUNCTION SELECTION PANEL DISPLAYED INSTEAD,     *
*       CHANGE THE TRANSID BELOW AS FOLLOWS:                          *
*              TRANSID=IEGM    (Z/VSE SIGNON PANEL)                   *
*              TRANSID=IEEP    (FUNCTION SELECTION PANEL)             *
*       AND REMOVE THE ASTERISK IN FRONT OF THE NEXT STATEMENT.       *
*---------------------------------------------------------------------*
*        EXEC  CICS START TRANSID(=C'XXXX') INTERVAL(0)                *
               TERMID(EIBTRMID) NOHANDLE ,
*---------------------------------------------------------------------*
         SPACE 3
*---------------------------------------------------------------------*
*                                                                     *
*       TRANSFER TO THE Z/VSE ONLINE PROBLEM DETERMINATION            *
*                 DATA COLLECTION MODULE -- IESOPDC                   *
*                                                                     *
*---------------------------------------------------------------------*
         EXEC  CICS LINK PROGRAM(=C'IESOPDC ') NOHANDLE,
*                                  IF NOT THERE, FALL INTO RETURN
*---------------------------------------------------------------------*
         SPACE 3
         LA    R1,PEP_COM_RETURN_OK
         B     RETURN
         DFHEJECT
*
RETURNER DS    0H                  RETURN FOR ERROR CASE
         LA    R1,PEP_COM_RETURN_DISABLE
RETURN   DS    0H
         ST    R1,PEP_COM_RETURN_CODE
RETURNX  DS    0H
         EXEC  CICS RETURN ,
         LTORG *
         END   DFHPEP
```

DFHWBEP Error Program

z/VSE provides a DFHWBEP program which you can use to obtain detailed error information for CICS Web Support. It is available as a skeleton in VSE/ICCF library 59.

You can enable this exit program by submitting the skeleton provided in VSE/ICCF library 59.

Chapter 6. Using Dumps for Error Diagnosis

This chapter discusses how to use the saved contents of system storage, called a **dump**, for error diagnosis.

This chapter consists of these main sections:

- "How the SYSDUMP and PRD2.DUMP Libraries Are Used"
- "Dump Types" on page 74
- "Contents of a Dump" on page 75
- "Using the Interactive Interface for Dump Processing" on page 75
- "Selecting the Dumps You Require" on page 76
- "Managing Dumps" on page 77
- "Analyzing Dumps" on page 79
- "Dump Analysis Examples" on page 81
- "Maintaining SYSDUMP Using REXX Procedure DMPMGR" on page 95
- "Extending the SYSDUMP Library" on page 97
- "Taking a Stand-Alone Dump" on page 97
- "Using the Info/Analysis Program" on page 99
- "Using the DOSVSDMP Program" on page 99
- "Taking a CICS Snap Dump" on page 100
- "Handling CICS Transaction Dumps" on page 100
- "Taking a VSE/POWER Disk Dump" on page 102
- "Taking a VSE/VSAM Snap Dump (IKQVEDA)" on page 102

Note:

1. If you want to analyze a dump stored in the dump library, go directly to "Analyzing Dumps" on page 79.
2. z/VSE provides a tool for handling and analyzing a dump, the **Info/Analysis** program. You can use this tool in batch mode and analyze those dumps that are stored in the dump library. z/VSE stores most dumps in the dump library, others you can load from tape or disk into it. The Info/Analysis program is described in the manual *z/VSE Diagnosis Tools*.

How the SYSDUMP and PRD2.DUMP Libraries Are Used

SYSDUMP Library

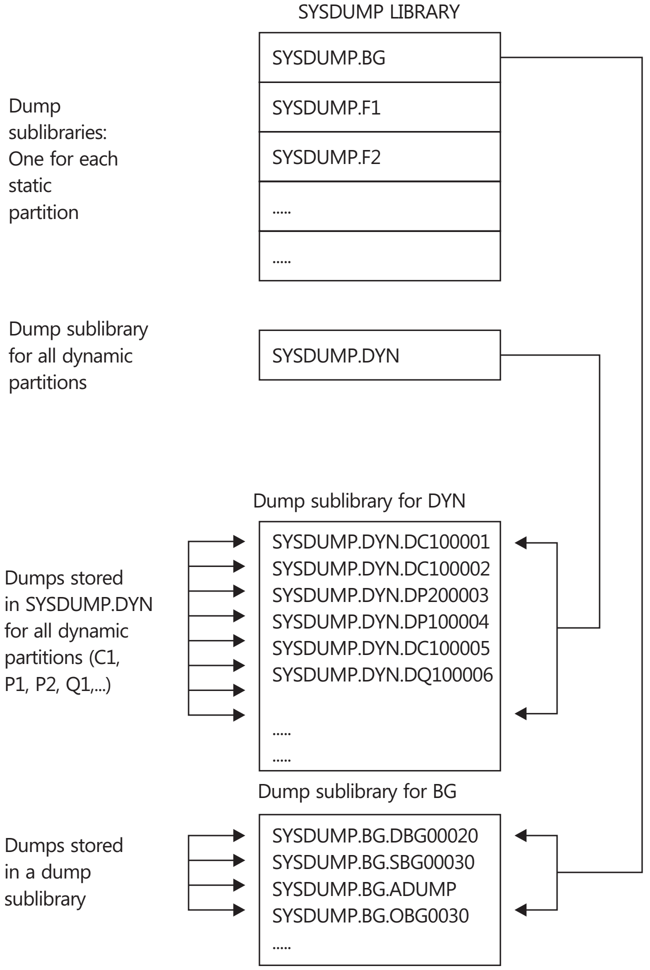
The dump library (SYSDUMP) is defined in VSAM space unless the system was upgraded from a previous release using Fast Service Upgrade. This has the advantage that the library extends up to a certain size depending on the available space in the master catalog. If you have upgraded your system from a previous release and want to move SYSDUMP to VSAM space refer to skeleton SKDMPEXT in ICCF Library 59. The dump library (SYSDUMP) contains 13 sublibraries. 12 sublibraries called SYSDUMP.BG, and SYSDUMP.F1 through SYSDUMP.FB are assigned to the static partitions, and one sublibrary, called SYSDUMP.DYN is assigned to the dynamic partitions. Each sublibrary may contain one or more dumps. Figure 15 on page 73 gives an overview of the SYSDUMP library concept.

The number of the dumps contained in the library SYSDUMP influences the time performance of the Problem Determination dialogs and of the utility program INFOANA. You are therefore recommended to clean up the library from time to time by:

- Deleting the dumps which are no more used. To do so, you can use the REXX/VSE program DMPMGR (see "Maintaining SYSDUMP Using REXX Procedure DMPMGR" on page 95).
- Offloading those dumps to tape which might be used at a later time.
- Moving dumps to the PRD2.DUMP archive library, as explained in "Managing Dumps" on page 77.

Note: In most cases, using the three actions described above you should not need to increase the size of the SYSDUMP library. However, if you decide to increase the size of the SYSDUMP library, you can use the method described in "Extending the SYSDUMP Library" on page 97.

Using the *Storage Dump Management* dialog, you can handle up to 56 dumps. If INFOANA is invoked in batch mode, you can process more than 56 dumps, but with more than that number of dumps you may encounter severe performance problems.



* Note that the 'S' in the dump identifier (SBG00030) indicates a data space dump.
* Note that the 'O' in the dump identifier (OBG00030) indicates a memory object dump.

Figure 15. The SYSDUMP Library Concept

The job control // OPTION SYSDUMP statement or STDOPT SYSDUMP=YES indicates that dumps are to be written to the sublibrary which is active for the partition. Refer to the manual *z/VSE System Control Statements* for further details about these options.

PRD2.DUMP Archive Library

The dump archive library is defined as PRD2.DUMP, which is created in VSAM space. Using the *Storage Dump Management* dialog, you can move dumps from the:

- Dump library to the dump archives.
- Dump archives to the dump library.

This is described in "Managing Dumps" on page 77.

Since a library created in VSAM space will extend itself automatically, you can store dumps with less chance of there being problems caused by a shortage of space.

Dump Types

Following is a description of the various dumps that are stored in the dump library. Either directly by the system or by you via tape.

- Abend dumps
- Macro dumps
- DUMP command dump (via tape)
- Stand-alone dump (via tape and disk)
- VSE/ICCF dump

Abend Dumps

Abend stands for **ab**normal **end**, that is, a program ends before normal completion because of an error. Whenever the **system** detects an abend situation, it terminates the program, creates a dump, and stores the dump in the dump library. The operator can use the CANCEL command to force the same sequence of steps.

Your job control specification for the DUMP operand (in the STDOPT command or // OPTION statement) determines the contents of an abend dump. You can request the following:

- A partition dump, or
- A system dump, and/or
- A data space dump, and/or
- A memory object dump.

Macro Dumps

A dump can also be requested through dump macros (DUMP, JDUMP, PDUMP, SDUMP, SDUMPX) and the CANCEL macro issued by **programs** written in assembler language. The job control specification of the DUMP operand (in the STDOPT command or // OPTION statement) and the resulting dump content is the same as described above for an abend dump.

DUMP Command Dumps

The **operator** can use the DUMP command to request a dump of a static or dynamic partition, or of specified storage areas or data spaces. The dump output device can be a tape or a printer. To be able to load the dump into the dump library for further analysis, a tape must be used.

Refer to “DUMP (Dump Storage Areas)” in the manual *z/VSE System Control Statements* for further details about the DUMP command.

Stand-Alone Dumps

With the stand-alone dump program the **operator** can take a dump of selected parts of virtual storage. The program records the page manager address spaces (PMRAS), the partitions, and the data spaces in separate files on one or more stand-alone dump tapes or on a disk device. The job control specification of the DUMP operand (in the STDOPT command or // OPTION statement) and the resulting dump content is the same as described above for an abend dump.

A stand-alone dump is initiated by the operator if the system has entered a severe error state and system operation is no longer possible. For example, the system entered a hard wait state or is in a continuous loop.

Refer to “Taking a Stand-Alone Dump” on page 97 for details on how to prepare a tape for a stand-alone dump and how to take the dump.

The VSE/ICCF Dump Facility

When running a job in an interactive partition of VSE/ICCF, you may get the following message:

```
K404D  ENTER DUMP COMMAND
```

This message is displayed on the condition that DUMP was specified in the VSE/ICCF job entry statement /OPTION. The message indicates the abnormal termination of a program. As a response you can enter the VSE/ICCF DUMP command to display registers and program storage areas for error diagnosis. To create a dump of the complete interactive partition, enter:

```
SAVE [comment]
```

where “comment” is optional and can be any text used to identify the dump and its contents. Up to 24 characters are allowed. The dump created is stored in the dump library. You can analyze the dump with the Info/Analysis program as shown in this chapter.

Note: You may force the cancelation of a program in an interactive partition if a loop or wait condition is indicated. Press the PA2 key at your display station. In most cases you can then enter DUMP commands to display general registers, floating point registers, and program storage areas for error diagnosis.
For details about the DUMP commands provided by VSE/ICCF refer to the *VSE/ICCF User's Guide*.

Contents of a Dump

All dumps have basically the same layout. Each dump begins with the symptom record followed by data records.

The Symptom Record

The symptom record is a summary of relevant dump data collected by the system. It contains information about the machine environment, the failing component, and information related to the cause of the error.

The symptom record often provides sufficient data for error diagnosis and problem solving. If this is not the case, an analysis of the data records is required.

Data Records

Data records contain the actual dump data. They reflect the storage contents at the time of the error.

Using the Interactive Interface for Dump Processing

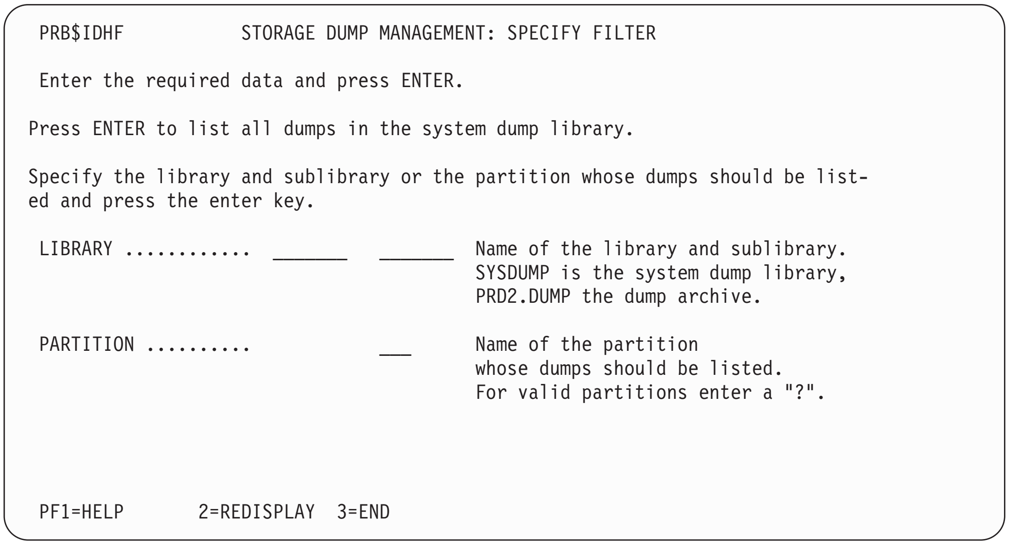
The Interactive Interface provides dialogs for dump processing. You access these dialogs from the *Problem Handling* panel. The panel offers the following dialogs for dump processing:

- **3** (Storage Dump Management)
- **4** (Inspect Dump Management Output)
- **6** (Dump Program Utilities)
  which, in turn, offers:

6 (Format ICCF Dump Data)

Selecting the Dumps You Require

After you have selected **Option 3. Storage Dump Management** from the *Problem Handling* selection menu, the *Storage Dump Management* dialog shown in Figure 16 is displayed.



```
 PRB$IDHF           STORAGE DUMP MANAGEMENT: SPECIFY FILTER

 Enter the required data and press ENTER.

Press ENTER to list all dumps in the system dump library.

Specify the library and sublibrary or the partition whose dumps should be list-
ed and press the enter key.

 LIBRARY ...........  _______  _______  Name of the library and sublibrary.
                                        SYSDUMP is the system dump library,
                                        PRD2.DUMP the dump archive.

 PARTITION ..........          ___      Name of the partition
                                        whose dumps should be listed.
                                        For valid partitions enter a "?".




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

Figure 16. Storage Dump Management: Specify Filter dialog

You can now enter a filter for the library / sublibrary and partition where the dumps are stored. Only those dumps that correspond to the filter will be displayed.

Valid Input for Library / Sublibrary:

SYSDUMP
All dumps in SYSDUMP library will be displayed.

SYSDUMP.nn
The dumps in SYSDUMP library for partition **nn** will be displayed.

PRD2.DUMP
All dumps stored in the PRD2.DUMP archive library will be displayed.

Valid Input for Partition:

nn The dumps in partition **nn** will be displayed.

*** or 'blank'**
All dumps in all partitions will be displayed.

Figure 18 on page 81 illustrates the use of a filter consisting of library SYSDUMP and partition BG.

Managing Dumps

To manage the dumps in the dump library or dump archive, use the *Storage Dump Management* dialog. The dialog can handle

- Up to 13 dump sublibraries.
- The dump archive.
- Up to 56 dumps.

The dialog provides the following options and PF keys for dump management:

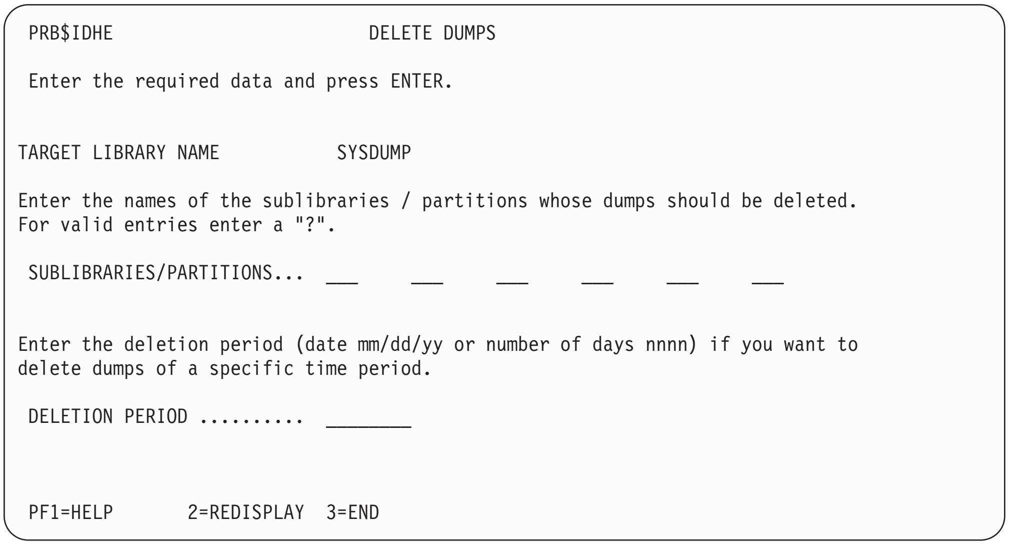
```
5 = DELETE DUMP
(delete a dump from the dump library/archive).

7 = IN/OUT ARCHIVES
(move the dump into / out-of the dump archive library).

8 = ON/OFFLOAD DUMP
(load a dump from the dump library/archive onto tape
or from tape/disk into the dump library/archive).

6 = ADD DUMP
(onload an external dump which is not in the list of
system managed dumps).

9 = DEL ALL
(delete all dumps of a certain partition or delete all
dumps older than a certain date).
```

When you request the dialog you get a FULIST display that shows the names of the dumps that are already stored in the dump library/archive or have been saved on tape. The options you can select are shown at the top of the FULIST display. Enter the option number in the “OPT” column next to the dump you want to process.

Delete Dumps

You can either delete specific dumps by selecting option **5**, or you can delete all dumps if you press **PF9**:

- If you select option **5**, you can delete dumps in the library you no longer need. You should frequently delete such dumps to free space in the library.
- If you press **PF9** (9=DEL ALL), the following panel is displayed in which you can define the dumps you wish to delete:



```
 PRB$IDHE                        DELETE DUMPS

 Enter the required data and press ENTER.


TARGET LIBRARY NAME           SYSDUMP

Enter the names of the sublibraries / partitions whose dumps should be deleted.
For valid entries enter a "?".

 SUBLIBRARIES/PARTITIONS...  ___    ___    ___    ___    ___    ___


Enter the deletion period (date mm/dd/yy or number of days nnnn) if you want to
delete dumps of a specific time period.

 DELETION PERIOD ..........  ________



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

Figure 17. Selecting Dumps to be Deleted

After the above panel has been displayed, you can enter one or both selection criteria. All the dumps matching the criteria entered will be deleted. If you enter a date for *Deletion Period*, all the dumps of this date and older will be deleted.

When you leave the *Storage Dump Management* dialog using **PF5** (Process), a delete job will be created to perform the deletions you have chosen.

Archive Out/In

Select option **7** if you want to move a dump:
- From the SYSDUMP library into the PRD2.DUMP archive library. This is only possible if you are currently displaying dumps contained in SYSDUMP or SYSDUMP.nn (where **nn** is a partition name).
- From the PRD2.DUMP archive library back into the SYSDUMP.nn library. This is only possible if you are currently displaying dumps contained in PRD2.DUMP).

Offload Dump

Select option **8** if you want to offload a dump from the dump library/archive to tape. This may be necessary if you want to free space in the dump library/archive. Dumps saved on tape can be later reloaded (ONLOAD DUMP) for analysis. When you offload a dump you are asked for the following information:
- Tape address
- Volume serial number

Onload Dump

Select option **8** if you want to ONLOAD a dump, written to tape with OFFLOAD DUMP, into the dump library/archive. When you onload a dump you are asked for the following information:
- Tape label
- Tape address
- File number (optional)

Onload External Dump

The *Storage Dump Management* dialog offers in addition the *Onload External Dump* function. Use this function to load the following dumps into the dump library/archive:

- Stand-alone dumps created either on tape or on disk.
- DUMP command dumps created on tape.

Note: You can also use PF6 from the *Storage Dump Management* dialog to perform these functions.

When onloading an external dump, you are asked for the following information:

- Sublibrary name
- Dump name
- Input device (tape or disk)

Analyzing Dumps

Dump analysis usually requires a skilled person. That person must have a general system skill as well as being familiar with subsystems like the CICS Transaction Server or VSE/POWER. To interpret a symptom record, might be relatively easy. But to locate and interpret error related dump data is usually a complex task. The Interactive Interface of z/VSE makes this task easier.

Storage Dump Management Dialog

With the *Storage Dump Management* dialog you retrieve selected dump information. The job stream created by z/VSE places the output in the VSE/POWER List Queue. The dialog provides the following options:

2 = PRINT SYMPTOMS
(Print a symptom record:
see "Example 1: Displaying a Symptom Record" on page 81).

3 = PRINT DUMP
(Print a formatted dump or selected dump data:
see "Example 2: Displaying a Formatted Dump" on page 84).

4 = ANALYZE SA DUMP
(Analyze a stand-alone main dump file and print the data).

9 = ANALYZE CICS DUMP
(Analyze a CICS dump. If you have upgraded from a previous z/VSE®

release and you still have a CICS coexistence environment,
a panel will be displayed where you must specify if the dump originates
from CICS/VSE or CICS TS. For CICS TS, further panels are shown in
which you specify the dump component identifiers you wish to analyze).

When you request the dialog you get a FULIST display that shows the names of the dumps that are stored in the dump library/archive or have been saved on tape.

Inspect Dump Management Output Dialog

With the *Inspect Dump Management Output* dialog you access the output in the VSE/POWER List Queue created by the *Storage Dump Management* dialog. You can either display or print the output information.

The *Inspect Dump Management Output* dialog provides the following options:

1 = DISPLAY
(display output on your terminal).

2 = CHANGE
(change parameters of list queue entries).

3 = PRINT
(print a list queue entry).

5 = DELETE
(delete output in list queue).

How to Analyze a Dump

The following steps are suggested:

1. First, display or print and analyze the **symptom record** contents.
2. If the symptom record information is not sufficient to solve the problem, create a **formatted dump**. A formatted dump gives you all the important supervisor control blocks, their contents and their addresses. For example:
   - PSW (Program Status Word)
   - COMREG (Partition Communication Region)
   - SYSCOM (System Communication Region)
   - TCBSAVE (Task Control Block Save Area)
3. In a third step you may use addresses retrieved from the symptom record and/or formatted dump to display/print **selected dump areas**. For example, areas of an application program in error.

When you select option 4 from the *Storage Dump Management* panel to analyze a **stand-alone** main dump file, z/VSE activates an analysis program. This program analyzes the stand-alone dump for you. In addition, you can treat a stand-alone dump as any other dump. That is, create a formatted dump or display/print selected areas.

Note: You can also create **unformatted** printouts from dumps on tape created by the stand-alone dump program or via the DUMP command. This may be necessary, for example:

- If none of your dump sublibraries is big enough to hold a stand-alone dump.
- If the dump was taken with the DUMP BUFFER,cuu command.

You can create an unformatted printout by using the DOSVSDMP utility program. For details, refer to chapter “The DOSVSDMP Utility” in the *z/VSE Diagnosis Tools*.

Dump Analysis Examples

In the following sections examples are used to show you how to perform dump analysis with the Interactive Interface. The following examples are used:

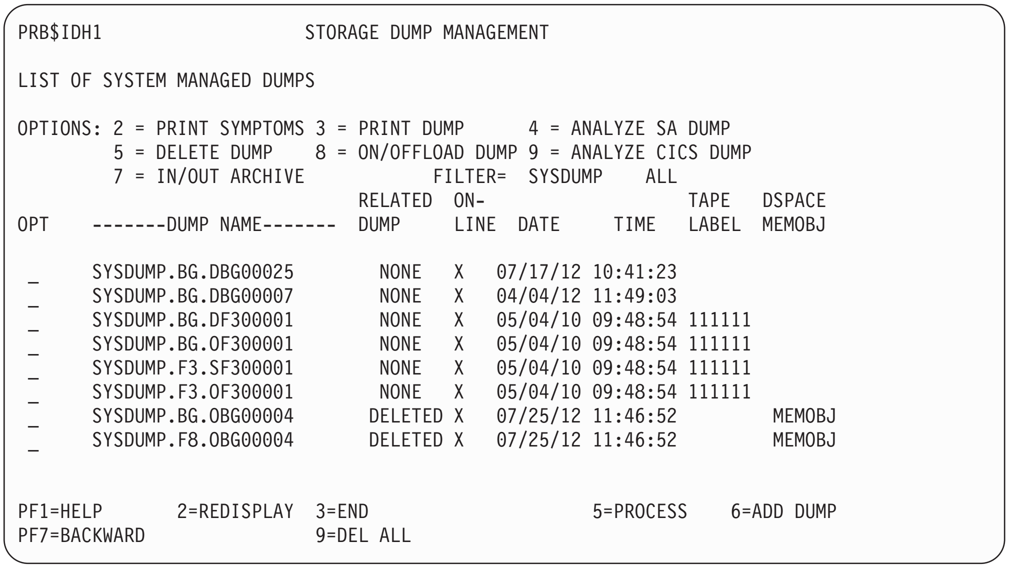
- Example 1: Displaying a Symptom Record
- Example 2: Displaying a Formatted Dump
- Example 3: Displaying Selected Dump Areas
- Example 4: Analyzing a Stand-Alone Dump

No example is shown for printing dump data on your local system printer. The sequence of steps is almost identical as shown for displaying dump data. The dialog for printing dump data will guide you. In addition, use the HELP text offered via the PF1 key.

If you want to analyze a stand-alone dump stored in the dump library/archive, go directly to “Example 4: Analyzing a Stand-Alone Dump” on page 91.

Example 1: Displaying a Symptom Record

The panels for *Storage Dump Management* and the *List Queue* are shown for the first example. For the other examples it is assumed that you are familiar with these panels.



```
PRB$IDH1                   STORAGE DUMP MANAGEMENT

LIST OF SYSTEM MANAGED DUMPS

OPTIONS: 2 = PRINT SYMPTOMS 3 = PRINT DUMP      4 = ANALYZE SA DUMP
         5 = DELETE DUMP    8 = ON/OFFLOAD DUMP 9 = ANALYZE CICS DUMP
         7 = IN/OUT ARCHIVE            FILTER=  SYSDUMP    ALL
                                RELATED  ON-                   TAPE   DSPACE
OPT    -------DUMP NAME-------  DUMP     LINE  DATE     TIME   LABEL  MEMOBJ

 _     SYSDUMP.BG.DBG00025        NONE   X   07/17/12 10:41:23
 _     SYSDUMP.BG.DBG00007        NONE   X   04/04/12 11:49:03
 _     SYSDUMP.BG.DF300001        NONE   X   05/04/10 09:48:54 111111
 _     SYSDUMP.BG.OF300001        NONE   X   05/04/10 09:48:54 111111
 _     SYSDUMP.F3.SF300001        NONE   X   05/04/10 09:48:54 111111
 _     SYSDUMP.F3.OF300001        NONE   X   05/04/10 09:48:54 111111
 _     SYSDUMP.BG.OBG00004       DELETED X   07/25/12 11:46:52         MEMOBJ
 _     SYSDUMP.F8.OBG00004       DELETED X   07/25/12 11:46:52         MEMOBJ


PF1=HELP       2=REDISPLAY  3=END                    5=PROCESS    6=ADD DUMP
PF7=BACKWARD                9=DEL ALL
```

Figure 18. Storage Dump Management Panel

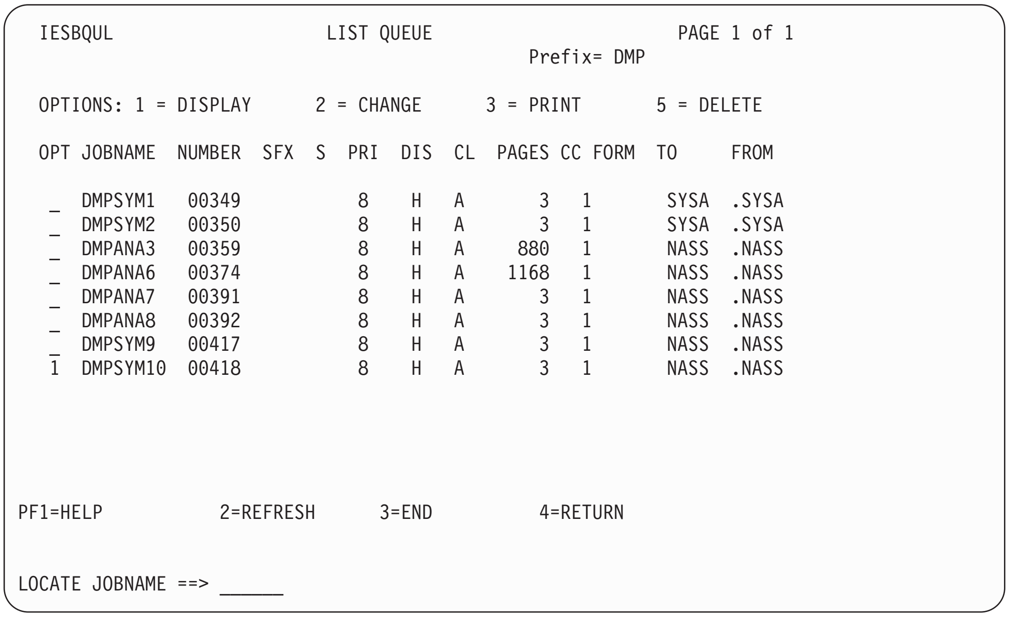
To display a symptom record, proceed as follows:

1. Access the *Problem Handling* panel and select the *Storage Dump Management* dialog.
2. Enter

   ```
   2  (Print Symptoms)
   ```

   in the option column for the dump you want to analyze and press **PF5** (Process). See Figure 18.
3. From the *Problem Handling* panel select the *Inspect Dump Management Output* dialog to access the List Queue. A *List Queue* sample panel is shown in Figure 19 on page 82.

Press PF1 to display HELP text. It gives you the PF-key settings for the display function you are going to use. For example, during display you can enter a character string and press PF6 for a search of that string.



```
 IESBQUL                    LIST QUEUE                        PAGE 1 of 1
                                              Prefix= DMP

 OPTIONS: 1 = DISPLAY      2 = CHANGE      3 = PRINT       5 = DELETE

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

  _  DMPSYM1   00349           8    H   A       3   1       SYSA  .SYSA
  _  DMPSYM2   00350           8    H   A       3   1       SYSA  .SYSA
  _  DMPANA3   00359           8    H   A     880   1       NASS  .NASS
  _  DMPANA6   00374           8    H   A    1168   1       NASS  .NASS
  _  DMPANA7   00391           8    H   A       3   1       NASS  .NASS
  _  DMPANA8   00392           8    H   A       3   1       NASS  .NASS
  _  DMPSYM9   00417           8    H   A       3   1       NASS  .NASS
  1  DMPSYM10  00418           8    H   A       3   1       NASS  .NASS




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


LOCATE JOBNAME ==> ______
```

Figure 19. List Queue Panel

4. Enter

   1 `(Display)`

   in the option column for the related jobname and press **ENTER**.

As a result, the symptom record is displayed on your screen. The symptom record display is shown in Figure 20 on page 83.

Symptom Record Description

See the manual *z/VSE Diagnosis Tools* for a description of the symptom record.

Analyzing Symptom Record Information (Sample)

To demonstrate the analysis of symptom record information, the information shown in Figure 20 on page 83 is used.

```
          // JOB DMPSYM10 PRINT DUMP SYMPTOMS
⋮
                                          DUMP SYMPTOMS
          SYSDUMP.F2.DF200000
          ENVIRONMENT:
              CPU MODEL ....... 2818
              CPU SERIAL ...... 019814
              TIME ............ 17:51:54:00
              DATE ............ 04/06/14
              SYSTEM ID ....... 5686CF706
              RELEASE ......... 8
              FEATURE ......... 1C
              DUMPTYPE ........ SADUMP
              PROBLEM NUMBER .. 99999999


          REQUIRED SYMPTOMS:

          OPTIONAL SYMPTOMS (SDB):

          OPTIONAL SYMPTOMS (NON-SDB):
              DATE_NOT_AVAILABLE
              MACHINE=ESA
              MODE=PAGING
              ACTIVE_SPACE_ID=0
              DUMPED_DATA_FROM_SPACE_ID=0
              PMR_ADDRESS_SPACE_ID=00
              DUMPED_DATA=SUPERVISOR+SVA
⋮
```

Figure 20. Symptom Record Example

When displayed or printed the symptom record information is enclosed by job control information. Look for the heading DUMP SYMPTOMS. Below that heading you find the information important for error diagnosis. In the example used, the following diagnostic information is supplied:

SYSDUMP.F2.DF200000

Name of the dump. SYSDUMP is the name of the dump library. F2 is the name of the dump sublibrary (for partition F2). DF200000 is the number of the dump assigned by the system. Note that in the case of a data space dump, the letter D in the system-assigned dump number would be an S (SF200000).

DUMPTYPE .... SCPREQ

SCPREQ indicates that the dump resulted from an abend.

AB/S2100

The first cancel code is 21, the second 00 (no meaning). Use the *z/VSE Messages and Codes* manual to look up the meaning of cancel code 21. You will find the following entry for cancel code 21:

```
21  0S04  ILLEGAL SVC - HEX LOCATION nnnnnnnn -
          SVC CODE HEX nn
```

REGS/01000
REGS/06978

Registers 1 and 6 seem to be related to the error. The value of 978 is the difference between the contents of register 6 minus the PSW instruction address. You can use this address to locate the suspected error area (location) in a formatted dump, or to display data selectively.

MS/0S04I

0S04 is the error message issued by the system to indicate the error situation. This message number is also shown for cancel code 21 (see AB/S2100 above). Refer to the *z/VSE Messages and Codes* manual for an explanation of the message.

RIDS/DTSCICS

DTSCICS is the name of the program or phase that caused the error.

ADRS/00000A76

A76 is the absolute address of the instruction following the failing one (outside SVA, LTA, and partition). You can use this address to locate the suspected error area (location) in a formatted dump, or to display data selectively. Use the MAP command to determine where address A76 is located.

OPCS/SVC02

An SVC (error) code of 02 was issued. This code also appears in the message text of message 0S04. Use the *z/VSE Messages and Codes* manual to look up the explanation for SVC code 02.

Analysis Summary for Symptom Record

You now know the name of the program or phase (DTSCICS). Together with the description of message 0S04 and SVC code 02 in the *z/VSE Messages and Codes* manual you have definite clues where to look for the error. If this information is not sufficient to solve the problem, you can use the addresses given in REGS and ADRS for further analysis. Create a formatted dump for that purpose. How to do this is shown in “Example 2: Displaying a Formatted Dump.”

Example 2: Displaying a Formatted Dump

A formatted dump can be created from any dump stored in the dump library/archive, including a stand-alone dump. The following example shows how to create a formatted dump. In a formatted dump the supervisor control block information is presented in an easy to read way. Input is the same dump as for the symptom record display in Example 1.

Proceed as follows:

1. Access the *Problem Handling* panel and select the *Storage Dump Management* dialog. (Figure 18 on page 81 shows the *Storage Dump Management* panel.)
2. Enter

   ```
   3  (Print Dump)
   ```

   in the option column for the dump you want to analyze and press **PF5** (Process). A selection panel is displayed on your screen.
3. From the *Print Dump: Select Function* panel select

   ```
   1  (Print complete formatted dump)
   ```

and press **ENTER**.

4. From the *Problem Handling* panel select the *Inspect Dump Management Output* dialog to access the List Queue.

   The *List Queue* panel is displayed on your screen.

   Press PF1 for the display of HELP text. It gives you the PF-key settings for the display function you are going to use. For example, during display you can enter a character string and press PF6 for a search of that string.

5. Enter

   ```
   1  (Display)
   ```

   in the option column for the related jobname and press **ENTER**. For the example chosen, the entry line looks as follows:

   ```
   1  DMPANA3  00359    8   H    A    880   1    .NASS   .NASS
   ```

   The complete *List Queue* panel is shown in Figure 19 on page 82.

As a result, the first page of the formatted dump is displayed on your screen. The following pages show parts of the formatted dump display. "Analyzing a Formatted Dump" on page 89 gives a description of the formatted dump displayed.

Note: In the following figure on some pages the two rightmost columns are not shown.

```
PRINT FORMAT
SYSDUMP.BG.TEST

NAME =         COMPONENT ID =         TYPE = OVERVIEW LIST


CANCLMSG    BG           TEXT DATA
CREGS       BG           HEXADECIMAL DATA
FREGS       BG           HEXADECIMAL DATA
FL CTREG    BG           HEXADECIMAL DATA
GREGS       BG           HEXADECIMAL DATA
AREGS       BG           HEXADECIMAL DATA
PSW         BG           HEXADECIMAL DATA
BEAR        BG           HEXADECIMAL DATA
HARDCOPY    MESSAGES     TEXT DATA
LOCATORS                 CONTROL BLOCK LIST
SYSDUMP.BG.DBG01182                                                                                     PAGE 000002

NAME = CANCLMSG COMPONENT ID = BG      TYPE = TEXT DATA


0S03I PROGRAM CHECK INTERRUPTION  - HEX LOCATION 0050034E - INTERRUPTION CODE 01 - OPERATION EXCEPTION
0S00I JOB PAUSEBG  CANCELED
0S07I PROBLEM PROGRAM  PSW = 071D2001 80500350
SYSDUMP.BG.DBG01182                                                                                     PAGE 000003

NAME = CREGS    COMPONENT ID = BG      TYPE = HEXADECIMAL DATA


   CTL REGS    0-3    0D04EE43 01000004 00060100 40400001
               4-7    00000001 01137040 10000000 01000004
               8-B    00000000 00000000 00000000 00000000
               C-F    00000000 01000004 C0081134 0005F330
SYSDUMP.BG.DBG01182                                                                                     PAGE 000004

NAME = FREGS    COMPONENT ID = BG      TYPE = HEXADECIMAL DATA


    FL REGS    0-1    4040404040404040 0000000000000000
               2-3    4040404040404040 0000000000000000
               4-5    4040404040404040 0000000000000000
               6-7    4040404040404040 0000000000000000
               8-9    0000000000000000 0000000000000000
               A-B    0000000000000000 0000000000000000
               C-D    0000000000000000 0000000000000000
```

```
               E-F    0000000000000000 0000000000000000
SYSDUMP.BG.DBG01182                                                                                        PAGE 000005

NAME = FL CTREG COMPONENT ID = BG       TYPE = HEXADECIMAL DATA


               FCR    00000000
SYSDUMP.BG.DBG01182                                                                                        PAGE 000006

NAME = GREGS    COMPONENT ID = BG       TYPE = HEXADECIMAL DATA


    GP REGS    0-1    0000000000000010 0000000000000002
               2-3    0000000180302000 00000000FFFFC2C7
               4-5    000000000063FFFF 0000000000000000
               6-7    0000000000500634 00000000000005D8
               8-9    0000000000500659 000000000050007A
               A-B    0000000000000000 000000000063FFFF
               C-D    0000000000500078 00000000003E8000
               E-F    00000000805002E8 0000000000000000
SYSDUMP.BG.DBG01182                                                                                        PAGE 000007

NAME = AREGS    COMPONENT ID = BG       TYPE = HEXADECIMAL DATA


    AC REGS    0-3    00000000 00000000 00000000 00000000
               4-7    00000000 00000000 00000000 00000000
               8-B    00000000 00000000 00000000 00000000
               C-F    00000000 00000000 00000000 00000000
SYSDUMP.BG.DBG01182                                                                                        PAGE 000008

NAME = PSW      COMPONENT ID = BG       TYPE = HEXADECIMAL DATA


       PSW     071D2001 80500350
SYSDUMP.BG.DBG01182                                                                                        PAGE 000009

NAME = BEAR     COMPONENT ID = BG       TYPE = HEXADECIMAL DATA


      BEAR     00000000 00089AEC
SYSDUMP.BG.DBG01182                                                                                        PAGE 000010

NAME = HARDCOPY COMPONENT ID = MESSAGES TYPE = TEXT DATA
.
.
.
NAME = LOCATORS COMPONENT ID =          TYPE = LOCATOR DATA
 BLOCK        COMP      BASE(ADDR)        KEYFIELD


BG            BG         00500000
LOWCORE       SYSTEM     00000000
SYSCOM        SYSTEM     00000420
COMREG        BG         000005D8
PIB2TAB       BG         00002010
PIBTAB        BG         000020E0
PUBTABLE      SYSTEM     00005418
PUBOWN        SYSTEM     0000D418
PUB2TAB       SYSTEM     00090E44
LUBTAB        BG         000021BC
LUBEXT        BG         00222B30
DIBTAB        BG         00004E90
PCB           BG         00063170
AF-TIB        0021       00063F50
AF-TCB        0021       0006B4F0
LOADLIST      BG         00218C20
ICCFVT        SYSTEM     00011808
LPT           SYSTEM     0022F028
LIB_ANC       BG         0022F348
LDT           SYSTEM     0022FCB0
SDT           SYSTEM     00230AC8
EDT           SYSTEM     00233508
DDT           SYSTEM     00235140
L-TASK-R      BG         00235188
LOTPOOL       SYSTEM     0023AB98

NAME=SYSCOM   COMPONENT ID=SYSTEM   BASE=00000420
 00000420  00041068 00000F00 000162C0 00005470 00000000 000008B0 0006DE10 0000FAB8 00 *................................*
 00000440  00000000 222AF2E0 0081003C 000CE67F 00600000 0005D634 00000000 00075370 00 *......2..a....W".-....O.........*
 00000460  C0601CFC 00200098 07802A80 0009626A 00038878 0006F120 00000021 0027C000 00 *.-.....q..........h...1.........*
 00000480  800118EC 00012078 00029FE2 000163A4 00027EC8 000EEB50 0000D418 000009F0 00 *...........S...u..=H...&..M....0*
 000004A0  00000000 FFFF0000 00015608 00000000 0002B948 00010660 0039001A 00A000C0 00 *.......................-........*
 000004C0  00B00000 00000178 00000201 02000200 00000000 00014638 00074FE0 00000000 00 *................................*
 000004E0  00000000 00008000 FFFF8001 7FF9000C A05A8848 00000000 0BF400FE 00215BF4 00 *............"9....h......4....$4*
 00000500  0001488C 00090067 00040000 03E74FFF 00000000 C00C0000 00000000 0022ECF0 00 *.............X.................0*
 00000520  80028780 00011660 00000000 003E5000 00085A28 000D0000 00000000 00011808 00 *..g....-......&.................*
 00000540  00011898 00014C98 00000000 0022F028 10000B94 00024F08 5B5BD1C3 D3404040 00 *...q.. q......0....m....$$JCL   *
 00000560  00000AC2 00000000 0004FE88 0004F9A8 0028C000 0006E1BC 00000000 002161B8 00 *...B.......h..9y............../.*
 00000580  000E0035 00229F10 20596E70 0006F064 A0C82AE0 00000000 0FFFC500 00000000 00 *..........>...0..H........E.....*
 000005A0  00000000 00260080 0004DB80 2220F000 00046620 A0CC6F00 22295800 00000000 00 *..............0.......?.........*
 000005C0  00000000 2211D000 00000400 E5E2C5D3 C9E44040                            00 *............VSELIU              *

NAME=COMREG   COMPONENT ID=BG       BASE=000005D8
 000005C0                                                        F1F061F0 F261F1F3 00 *                        10/02/13*
SYSDUMP.BG.DBG01182                                                                                        PAGE 000012

 000005E0  61F2F000 00000000 00000000 00000000 D7C1E4E2 C5C2C740 0063FFFF 0050087B 00 */20.............PAUSEBG .....&.#*
 00000600  0050087B 00000010 222FFFFF FF7E44D3 000044D0 19000000 5418CC0C EBF504A7 00 *.&.#.........=.L.............5.x*
 00000620  21A021AD 21BC38F1 F0F0F2F1 F3F2F7F5 000020D0 00000000 4E780000 F2F00000 00 *.......100213275........+...20..*
 00000640  00000010 F0400000 000117FD 00218A50 00000A04 20000000 00000000 05D811E1 00 *....0 .........&.............Q..*
 00000660  003E8000 68680F40 40404040 40404000 40404040 40404000 0028302B C4000000 00 *.......        .       .....D...*
 00000680  00222B30 020480C0 00000000 00000000 00218C20 00022F00 0E02000E 00000000 00 *................................*
 000006A0  024D7000 00000000 C9D1C2C6 C2C70000 D4D6E3C5 E2E34040 00000000 00000000 00 *.(......IJBFBG..MOTEST  ........*
 000006C0  00000000 005000E0 00063338 0050087B E2E8E240 40404040 A2000044 44000000 00 *.....&.......&.#SYS     s.......*
 000006E0  00000000 00000000 40404040 40404040 00000000 00000000 00000000 00000000 00 *........        ................*
 00000700  00000000 C6D6D9E2 C5C3C2C7 00000000 00000000 00000000 00000000 00000000 00 *....FORSECBG....................*
 00000720  00000000 00000000 00000000 00000000                                     00 *................                *

NAME=PIB2TAB  COMPONENT ID=BG       BASE=00002010
 00002000                                      05D80000 00210000 00063170 001080FF 00 *                .Q..............*

NAME=PIBTAB   COMPONENT ID=BG       BASE=000020E0
 000020E0  0000C2C7 80500000 00000000 0032FF00                                     00 *..BG.&..........                *

NAME=PUBTABLE COMPONENT ID=SYSTEM   BASE=00005418
 00005400                                                        00098000 B00080F8 00 *                        .......8*
 00005420  000CFFFF 110000F8 000DFFFF 210000F8 000EFFFF 400000F8 00600200 B00080F8 00 *.......8.......8.... ..8.-.....8*
 00005440  00610900 B00080F8 00620100 B00080F8 0190FFFF 6EFF0204 0191FFFF 6EFF0204 00 *./.....8.......8....>....j..>...*
 00005460  019DFFFF 6EFF0204 019EFFFF 6EFF0204 0200FFFF 6EFF02FC 0201FFFF 6EFF02FC 00 *....>.......>.......>.......>...*
 00005480  0202FFFF 6EFF02FC 0480FFFF 54C302C0 0481FFFF 54C302C0 0482FFFF 54C302C0 00 *....>........C...a...C...b...C..*
 000054A0  0483FFFF 54C302C0 0488FFFF 54C302C0 0489FFFF 54C302C0 048AFFFF 54C302C0 00 *.c...C...h...C...i...C.......C..*
 000054C0  048BFFFF 54C302C0 0660FFFF 56C302C0 0661FFFF 56C302C0 0662FFFF 56C302C0 00 *.....C...-...C.../...C.......C..*
 000054E0  0663FFFF 56C302C0 0690FFFF 56C302C0 0691FFFF 56C302C0 0692FFFF 56C302C0 00 *.....C.......C...j...C...k...C..*
 00005500  0693FFFF 56C302C0 0700FFFF E00002F8 0701FFFF E00002F8 0702FFFF E00002F8 00 *.l...C.........8.......8.......8*
 00005520  0A34FFFF 56C302C0 0A35FFFF 56C302C0 0A36FFFF 56C302C0 0A37FFFF 56C302C0 00 *.....C.......C.......C.......C..*
 00005540  0A38FFFF 56C302C0 0A39FFFF 56C302C0 0A58FFFF 56C302C0 0A59FFFF 56C302C0 00 *.....C.......C.......C.......C..*
 00005560  0A5AFFFF 56C302C0 0A5BFFFF 56C302C0 0A5CFFFF 56C302C0 0A5DFFFF 56C302C0 00 *.....C...$...C...*...C...)...C..*
 00005580  0A5EFFFF 56C302C0 0A5FFFFF 56C302C0 0FDFFFFF 90FF02F8 0FECFF00 120000F8 00 *.;...C.......C.........8.......8*
 000055A0  0FEDFF00 200000F8 0FEEFFFF 430000F8 0FEFFFFF 430000F8 0FFAFF00 120000F8 00 *.......8.......8.......8.......8*
 000055C0  0FFCFF00 120000F8 0FFDFF00 200000F8 0FFEFFFF 430000F8 0FFFFF00 B0800200 00 *.......8.......8.......8........*

NAME=PUBOWN   COMPONENT ID=SYSTEM   BASE=0000D418
 0000D400                                                        10000800 00000800 00 *                        ........*
 0000D420  02000200 02000000 00000000 00005FFF 5FFF0000 00000000 00000000 00000000 00 *................................*
 0000D440  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 0000D460  00000000 00000000 00000000 00000000 00000000 00000001 00000000 00000000 00 *................................*
 0000D480  00000000 00000000 0000                                                  00 *..........                      *

NAME=PUB2TAB  COMPONENT ID=SYSTEM   BASE=00090E44
SYSDUMP.BG.DBG01182                                                                                        PAGE 000013

 00090E40           0002C500 00000000 00000000 00000000 00000600 00000000 00000000 00 *    ..E.........................*
 00090E60  00000000 00000200 00000000 00000000 00000000 00005C00 00000000 00000000 00 *......................*.........*
 00090E80  00000000 00017300 00000200 00000000 00000000 00000300 00000100 00000000 00 *................................*
 00090EA0  00000000 00000300 00000100 00000000 00000000 00000E00 00000000 00000000 00 *................................*
 00090EC0  00000000 00000E00 00000000 00000000 00000000 00000E00 00000000 00000000 00 *................................*
 00090EE0  00000000 00000E00 00000000 00000000 00000000 00326D00 00000000 0000C4D6 00 *......................_.......DO*
 00090F00  E2D9C5E2 00D44D00 00000000 0000E2E8 E2E6D2F1 000B7000 00000000 0000E2C1 00 *SRES.M(.......SYSWK1..........SA*
 00090F20  C4E4D4D7 00000110 0000C1E4 00000000 00000000 00000000 00000000 00000000 00 *DUMP......AU....................*
 00090F40  00011000 00C1E400 00000000 00000000 00000000 00000000 00000000 01100000 00 *.....AU.........................*
 00090F60  C1E40000 00000000 00000000 00000000 00000000 00000001 100000C1 E4000000 00 *AU.........................AU...*
 00090F80  00000000 00000000 00000000 00000000 00000110 0000C1E4 00000000 00000000 00 *......................AU........*
 00090FA0  00000000 00000000 00000000 00011000 00C1E400 00000000 00000000 00000000 00 *.................AU.............*
 00090FC0  00000000 00000000 01100000 C1E40000 00000000 00000000 00000000 00000000 00 *............AU..................*
 00090FE0  00000001 100000C1 E4000000 00000000 00000000 00000000 00000000 00000110 00 *.......AU.......................*
 00091000  0000C1E4 00000000 00000000 00000000 00000000 00000000 00011000 00C1E400 00 *..AU.........................AU.*
 00091020  00000000 00000000 00000000 00000000 00000000 01100000 C1E40000 00000000 00 *........................AU......*
 00091040  00000000 00000000 00000000 00000001 100000C1 E4000000 00000000 00000000 00 *...................AU...........*
 00091060  00000000 00000000 00000110 0000C1E4 00000000 00000000 00000000 00000000 00 *..............AU................*
 00091080  00000000 00011000 00C1E400 00000000 00000000 00000000 00000000 00000000 00 *.........AU.....................*
 000910A0  01100000 C1E40000 00000000 00000000 00000000 00000000 00000001 100000C1 00 *....AU.........................A*
 000910C0  E4000000 00000000 00000000 00000000 00000000 00000400 00000000 00000000 00 *U...............................*
 000910E0  00000400 00000000 00000000 00000400 00000000 00000000 00000110 0000C1E4 00 *..............................AU*
 00091100  00000000 00000000 00000000 00000000 00000000 00011000 00C1E400 00000000 00 *.........................AU.....*
 00091120  00000000 00000000 00000000 00000000 01100000 C1E40000 00000000 00000000 00 *....................AU..........*
 00091140  00000000 00000000 00000001 100000C1 E4000000 00000000 00000000 00000000 00 *...............AU...............*
 00091160  00000000 00000110 0000C1E4 00000000 00000000 00000000 00000000 00000000 00 *..........AU....................*
 00091180  00011000 00C1E400 00000000 00000000 00000000 00000000 00000000 01100000 00 *.....AU.........................*
 000911A0  C1E40000 00000000 00000000 00000000 00000000 00000001 100000C1 E4000000 00 *AU.........................AU...*
 000911C0  00000000 00000000 00000000 00000000 00000110 0000C1E4 00000000 00000000 00 *......................AU........*
 000911E0  00000000 00000000 00000000 00011000 00C1E400 00000000 00000000 00000000 00 *.................AU.............*
 00091200  00000000 00000000 01100000 C1E40000 00000000 00000000 00000000 00000000 00 *............AU..................*
 00091220  00000001 100000C1 E4000000 00000000 00000000 00000000 00000000 00000110 00 *.......AU.......................*
 00091240  0000C1E4 00000000 00000000 00000000 00000000 00000000 00011000 00C1E400 00 *..AU.........................AU.*
 00091260  00000000 00000000 00000000 00000000 00000000 03000000 00000000 E5C4C9C4 00 *............................VDID*
 00091280  D3C10000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *LA..............................*
 000912A0  00000000 01000000 00000000 00000000 01000000 00000000 00000000 00000000 00 *................................*
 000912C0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 000912E0  00000000 00000000 00000000 01000000 00000000 00000000 00000000 00000000 00 *................................*
 00091300  00000000 0000                                                          00 *......                          *

NAME=LUBTAB   COMPONENT ID=BG       BASE=000021BC
 000021A0                                                                 30003000 00 *                            ....*
 000021C0  31003200 00000B00 0B00FFFF FFFFFFFF 0C00FFFF FFFF0B00 FFFFFFFF FFFFFFFF 00 *................................*
 000021E0  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00 *................................*
 00002200  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFF2F00 00 *................................*
 00002220  FFFF0C00 0C000C00 0C00FFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00 *................................*
 00002240  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00 *................................*
 00002260  TO NEXT LINE ADDRESS SAME AS ABOVE

NAME=LUBEXT   COMPONENT ID=BG       BASE=00222B30
SYSDUMP.BG.DBG01182                                                                                PAGE 000014

 00222B20                                      00000000 00000000 00000000 00000000 00 *                ................*
 00222B40  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 00222B60  TO NEXT LINE ADDRESS SAME AS ABOVE
 00222B80  00000000 00261010 00000000 00000000 00000000 00000000 00000000 00261000 00 *................................*
 00222BA0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 00222BC0  TO NEXT LINE ADDRESS SAME AS ABOVE
 00222C40  0000FFFF 00000000 0000FFFF 00000000 0000FFFF 00000000 0000FFFF 00000000 00 *................................*
 00222C60  TO NEXT LINE ADDRESS SAME AS ABOVE
 00222CA0  0000FFFF 00000000 0000FFFF 00000000 0000FFFF 00000000 00000000 00000000 00 *................................*
 00222CC0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000110 00 *................................*
 00222CE0  TO NEXT LINE ADDRESS SAME AS ABOVE
 00222D00  00000000 00000000 0000FFFF 00000000 00000000 00000000 00000000 00000000 00 *................................*
 00222D20  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 00222D40  0000FFFF 00000000 0000FFFF 00000000 00000000 00000000 00000000 00000000 00 *................................*
 00222D60  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 00222D80  TO NEXT LINE ADDRESS SAME AS ABOVE
 00222DE0  00000000 00000000 00000000 00000000 0000FFFF 00000000 0000FFFF 00000000 00 *................................*
 00222E00  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 00222E20  TO NEXT LINE ADDRESS SAME AS ABOVE
 002234A0  00000000 00000000 00000000 00000000 0000FFFF 00000000 00000000 00000000 00 *................................*

NAME=DIBTAB   COMPONENT ID=BG       BASE=00004E90
 00004E80                                      00000000 00000000 00500000 00000000 00 *                .........&......*
 00004EA0  00000019 00000000 00000000 00000000 00510000 00000000 00000019 03E80000 00 *.............................Y..*
 00004EC0  00000000 00000000 00790000 00000000 00000013 03E80000 00000000 00000000 00 *.....................Y..........*
 00004EE0  00500000 00000000 00000019 00000000                                    00 *.&..............                *

NAME=PCB      COMPONENT ID=BG       BASE=00063170
 00063160                                      02000000 00080000 00000000 0001001F 00 *                ................*
 00063180  00000000 000000C1 00000000 00063170 00100000 00063338 80000000 21000000 00 *.......A........................*
 000631A0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 221F8D80 00 *................................*
 000631C0  00000000 00061190 00061190 00000000 00000000 00000000 00000000 00500000 00 *.............................&..*
 000631E0  00640000 00500000 02500000 00000000 00000000 024D7000 00000000 00000000 00 *.....&...&............(.........*
 00063200  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 00063220  00000000 00000000 00000000 00000000 00000000 00000000 80063238 90408021 00 *............................. ..*
 00063240  80063240 91408021 80063248 92408000 80063250 93418000 80063258 94408000 00 *... j ......k ..... l.......m ..*
 00063260  80063260 95400021 0022A0E0 00063528 00000000 00000000 00000000 00000000 00 *...-n ..........................*
 00063280  00000000 00000000 C9D1C2D7 C2C70000 222A9040 01137040 00063170 00000000 00 *........IJBPBG..... ... ........*
 000632A0  00000000 00800000 00000000 00000000 00000000 00006FCC 00006F8F 00000000 00 *......................?...?.....*
 000632C0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 000632E0  0022D320 00000000 00000000 00000001 00000000 00000000 00000000 00000000 00 *..L.............................*
 00063300  00000000 00000000 00000000 0008EA30 0006E2E0 00000000 00000000 00000000 00 *..................S.............*
 00063320  00000000 00000000 00000000 00000000 00000000 00000000 00388002 10000000 00 *................................*
 00063340  0010C2C7 000005D8 000021BC 00004E78 000020E0 00002010 00000000 0006DE5C 00 *..BG...Q......+................**
 00063360  00000000 00000000 00000000 00000000                                     00 *................                *

NAME=AF-TIB   COMPONENT ID=0021     BASE=00063F50
SYSDUMP.BG.DBG01182                                                                                         PAGE 000015

 00063F40                                      00064050 00640130 0006B4F0 00020000 00 *                .. &.......0....*
 00063F60  00000000 00210010 00063170 80000000 00061190 00000000 00012000 83000000 00 *............................c...*
 00063F80  CC0BE522 487AD4E8 00061190 00000000 00000000 00000000 00000000 00000000 00 *..V..:MY........................*
 00063FA0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 0008EB20 00 *................................*
 00063FC0  00000000 00000000 00000000                                              00 *............                    *

NAME=AF-TCB   COMPONENT ID=0021     BASE=0006B4F0
 0006B4E0                                      02040008 00000000 00063F50 0022D3B8 00 *                ...........&..L.*
 0006B500  00020079 00000000 00000000 0006B530 0006DB10 0006B794 20D68000 10000000 00 *.......................m.O......*
 0006B520  00000000 22298340 00000000 00000000 00000000 00000000 040C0000 8001D192 00 *......c ......................Jk*
 0006B540  0001C10A 0006B4F0 0001B1A0 0003E840 0001B23C 00000021 00000008 00000048 00 *..A....0......Y ................*
 0006B560  000C12E4 000C0FC8 0001EC24 00230AC8 0006E61C 8001C828 8001C3E4 8001C3D6 00 *...U...H.......H..W...H...CU..CO*
 0006B580  00000000 0002177C 00021824 00000000 00000000 00085E18 0000000C 800207DA 00 *.......§..............;.........*
 0006B5A0  00086130 800207DA 00086130 00086140 00000000 00000000 00000000 00000000 00 *../......./.../ ................*
 0006B5C0  00000000 00000000 00085E48 000020E0 00005478 00000090 10000A00 00015A28 00 *..........;.....................*
 0006B5E0  80029A36 80029A56 0002014C 0006B4F0 000020E0 00261010 00029228 0230000E 00 *...........<...0..........k.....*
 0006B600  00000000 C8C4E4D7 00200000 0000114F 00000000 00641000 00001A48 00400078 00 *....HDUP..................... ..*
 0006B620  00400078 00400078 00200320 00001155 00000000 00000000 000D88F8 00000482 00 *. ... ....................h8...b*
 0006B640  0022FCB0 00230AC8 000F0AD0 000005D8 024D7000 00000000 D4D6E5C5 D4D6C4C5 00 *.......H.......Q.(......MOVEMODE*
 0006B660  5B5BC2C3 D3D6E2F5 00000000 00000000 00000000 0006F2F8 00000000 00000000 00 *$$BCLOS5..............28........*
 0006B680  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00 *................................*
 0006B6A0  00000000 00000000 00000000 00000000 0006B6F8 00020023 00000000 00000000 00 *...................8............*
 0006B6C0  00000000 00000000 00000000 00000000 D3C9C2D9 C9D7E3E7 00000000 00000000 00 *................LIBRIPTX........*
 0006B6E0  00000000 00000000 00000000 00000000 00000000                            00 *....................            *
```

Analyzing a Formatted Dump

For the following description refer to “Example 2: Displaying a Formatted Dump” on page 84. The figure does not show the complete dump. Only selected parts are shown.

A formatted dump is divided into three parts.

The **first part** contains diagnostic information associated with the error, such as:

- Control registers (CREGS)
- Floating point registers (FREGS)
- General purpose registers (GREGS)
- Access registers (AREGS)
- Program Status Word (PSW)
- Hardcopy file messages (MESSAGE)

Start diagnosis with the sections listed above. Ignore the job control information that precedes the dump.

The **second part** of a formatted dump (identified by NAME = LOCATORS) lists system control blocks. For each control block the start address is given; see column BASE (ADDR). You can thus identify the address of a control block and look for its contents in part three of the formatted dump. Also, you can use such an address or the address found in a control block to display dump data selectively. This is shown in Example 3.

The **third part** of a formatted dump shows the contents of the control blocks listed in part two. It starts with the lower area of processor storage (LOWCORE), followed by the system communication region (SYSCOM). It requires good system knowledge to analyze and interpret the contents of control blocks. But if a difficult problem arises you may contact IBM for support and you may then be asked for data that a formatted dump provides.

Example 3: Displaying Selected Dump Areas

In Example 1 and Example 2 several addresses came up that are related with the error. You can use these addresses to display the related areas. You would do this if you cannot solve a problem with the information provided in the symptom record and the formatted dump. But you should realize that additional expertise is required to analyze unformatted dump data.

To display selected dump areas proceed as follows:

1. Access the *Problem Handling* panel and select the *Storage Dump Management* dialog. (Figure 18 on page 81 shows the *Storage Dump Management* panel.)
2. Enter

   ```
   3  (Print Dump)
   ```

   in the option column for the dump you want to analyze and press **PF5** (Process). A selection panel is displayed on your screen.
3. From the *Print Dump: Select Function* panel select

   ```
   2  (Print selective parts of the dump)
   ```

   and press **ENTER**. You get a panel to define the dump area.
4. In the *Print Selected Dump* panel define the area you want displayed. Define either

   ```
   FROM ADDRESS  and  TO ADDRESS
   ```

   or

   ```
   FROM ADDRESS  and  LENGTH
   ```

   The system places the output of the submitted job in the VSE/POWER List Queue.
5. From the *Problem Handling* panel select the *Inspect Dump Management Output* dialog to access the List Queue.

   The *List Queue* panel is displayed on your screen.

   Press PF1 to display HELP text. It gives you the PF-key settings for the display function you are going to use. For example, during display you can enter a character string and press PF6 for a search of that string.
6. Enter

   ```
   1  (Display)
   ```

   in the option column for the related jobname and press **ENTER**. For the example chosen, the entry line looks as follows:

```
1  DMPANA8  00392      8   H    A     3   1    .NASS   .NASS
```

   The complete *List Queue* panel is shown in Figure 19 on page 82.

As a result, the selected area is displayed on your screen. Following is the example of a selected area display. The FROM ADDRESS was 00000580.

Note: In the following figure, the rightmost portion of the display is not shown. This portion shows for each line the hex values translated into the corresponding characters.

```
       // JOB DMPANA8 PRINT SELECTIVE PARTS OF A DUMP
                  .
                  .
 00000580                                     F0F961F0 F761F9F2 61F1F900 00000000 00 *            03/03/04/19.....*
 000005A0  00000000 00000000 C9D5C6D6 C1D5C140 00495FFF 004006D1 004006D1 00000010 00 *......INFOANA .... .J. .J....*
 000005C0  014FFFFF FF5FCCD3 1000CCD0 19000800 30F8A642 3FD7BE7B 1440144D 145C38F0 00 *.......L......8...P.#. .(.*.0*
 000005E0  F9F0F7F9 F2F2F5F1 00001370 00000078 46400000 F1F90000 00000010 F0400000 00 *90792251...... ..19......0 ..*
 00000600  00008A59 00171D30 00000940 12A00000 00000000 059011E1 00248000 40600F40 00 *......... ............... -. *
 00000620  40404040 40404000 40404040 40404000 00000000 00000000 0016E000 02140042 00 *       .     ................*
 00000640  00000000 00000000 00173000 00000000 00000000 00000000 004FE000 00000000 00 *.............................*
 00000660  C9D1C2C6 C2C70000 C4C1E3C1 E3E2E3F1 00000000 00000000 00000000 004000E0 00 *IJBFBG..DATATST1.......... ..*
                  .
                  .
```

Figure 21. Selected Dump Area Example

Example 4: Analyzing a Stand-Alone Dump

A stand-alone dump is a dump of selected parts of virtual storage of your system. The stand-alone dump program produces a main dump file of the system areas, dump files containing page manager data and one additional dump file for each partition and/or data space to be dumped.

To process a stand-alone dump stored on tape or disk, you must first unload the dump from tape or disk to the dump library/archive.

The onload function of the *Dump Management* dialog asks you to specify the file number to be onloaded:

- The order in which the partition dumps appear on the tape is determined by the priority that you defined using the PRTY command when the dump was created.
- The dump of the various partitions always begins at File 4 on the tape.
- Partitions dumps are always preceded by a dump of the Supervisor and the SVA, which is contained in File 3 of the dump tape.
- The Supervisor and SVA dump is **always** present.

Format of the Stand-Alone Dump Tape:

File 1: Stand-alone dump program

File 2: Work file

File 3: Main dump file

The main dump file includes the symptom record, the dump data, control block locators (LBD entries) for supervisor control blocks, and the last 200 messages of the hardcopy file.

File 4 to n: Page manager address spaces

Each file includes a symptom record and a storage dump of the page manager address space.

File n+1 to m: Dumps of partitions and/or data spaces.

Each file includes a symptom record and a storage dump of the partition or data space, respectively.

The order of the dumped partitions and/or data spaces depends on the specification of the // OPTION SADUMP job control statement.

A stand-alone dump is created by the operator when a severe error occurred. For example, a WAIT or a LOOP. In such an error situation most often a new startup is necessary to return to normal processing. Before the new startup is initiated a stand-alone dump should be taken. Refer to “Taking a Stand-Alone Dump” on page 97 for details on how to take a stand-alone dump.

To load the dump from tape or disk into the dump library/archive, select from the *Storage Dump Management* panel ADD DUMP (PF6).

If the stand-alone dump has been (off)loaded to tape before, select ON/OFFLOAD DUMP (option 8).

If a stand-alone dump is too big to be loaded into a dump sublibrary, you can create an unformatted printout directly from tape by using the DOSVSDMP utility. For details, refer to “The DOSVSDMP Utility” in the manual *z/VSE Diagnosis Tools*.

When you request analysis of a stand-alone dump, programs IJBXDBUG and IJBXSDA are activated. The programs analyze the stand-alone main dump file for you and create an analysis report that you can display or print.

To request an analysis report, proceed as follows:

1. Access the *Problem Handling* panel and select the *Storage Dump Management* dialog. (Figure 18 on page 81 shows a *Storage Dump Management* panel.)
2. Enter
   ```
   4  (Analyze SA Dump)
   ```
   in the option column for the stand-alone dump you want to analyze and press **PF5** (Process). You can then specify the file number of the file on the stand-alone dump tape you want to analyze. No specification causes the system to analyze the main dump file (File 3).

   z/VSE places the output of the job created and submitted in the VSE/POWER List Queue.
3. From the *Problem Handling* panel select the *Inspect Dump Management Output* dialog to access the List Queue. The *List Queue* panel is displayed on your screen.
4. Enter
   ```
   1  (Display)
   ```
   in the option column for the related jobname and press **ENTER**.

As a result, the analysis report is displayed on your screen. To interpret the analysis report, system skill is required. The following section describes the contents of an analysis report.

Analysis Report Contents

The analysis report contains general and specific information. For details refer to the manual *z/VSE Diagnosis Tools*.

Examples of Analysis Reports: On the following pages, examples of analysis reports are shown (first page only):

1. Example of Hard Wait Condition.
2. Example of Soft Wait Condition.
3. Example of Running System (Loop) Condition.

Example of Hard Wait Condition

```
SYSTEM STATUS:  HARD WAIT                           HARD WAIT CODE:  FFF
CURRENT TASK:  AR TASK


HARD WAIT REASON CODE:  24 - PROGRAM CHECK IN SUPERVISOR

PROGRAM OLD PSW INDICATES 31 BIT ADDRESSING MODE.

PROGRAM CHECK TYPE:  0010 SEGMENT TRANSLATION EXCEPTION
ADDRESS OF PROGRAM CHECK:  A0D1E4E8
PROGRAM CHECK INSTRUCTION:  58600014

SYSTEM STATUS:  HARD WAIT
CURRENT TASK:  AR TASK




DEVICE ANALYSIS FOR ACTIVE NON TP DEVICES ONLY:

DEV  TYPE  TSK  I/O REQUEST STATUS AND INFORMATION
---  ----  ---  --------------------------------------------------------------
009  3277  N/A  LAST I/O INTERRUPT WAS FROM THIS DEVICE
(NO BUSY DEVICES AND NO DEVICES WITH I/O QUEUED)



TASK ANALYSIS FOR ACTIVE TASKS ONLY:

TASK NAME     STATUS    TASK INFORMATION
------------  --------  ------------------------------------------------------
CMT TASK      WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
DSP TASK      WAITING   ON DISPATCHER SERVICE
CST TASK      WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
HCF TASK      WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 0005D608
FCP TASK      WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
AR TASK       READY TO   RUN
F1 MAIN TASK  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 0027C040
                        SUB SYSTEMS IN THIS PARTITION:  POWER
                        LIBRARIAN SERVICE ACTIVE
F2 MAIN TASK  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SUB SYSTEMS IN THIS PARTITION:  ICCF CICS IUI
                        LIBRARIAN SERVICE ACTIVE
SYSDUMP.BG.SAHW0001

                        SVC RETRY INDICATOR ON      SVC: 84 (HEX)
F3 MAIN TASK  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 0028FBB8
                        SUB SYSTEMS IN THIS PARTITION:  VTAM
                        LIBRARIAN SERVICE ACTIVE
FB MAIN TASK  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T0099 FB SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00503CA0
T009A FB SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T009B F1 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00509DF8
                        LIBRARIAN SERVICE ACTIVE
T009C F3 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T009D F3 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
```

```
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T009E F3 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 002985D4
T009F F3 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00555698
T00A0 F2 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T00A1 F2 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 84 (HEX)
T00A2 F2 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        LIBRARIAN SERVICE ACTIVE
                        SVC RETRY INDICATOR ON      SVC: 84 (HEX)
T00A3 F2 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 84 (HEX)
T00A4 F2 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 84 (HEX)
T00A5 F2 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T00A6 F2 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T00A7 F2 SUB  WAITING   FOR I/O, ECB OR TECB        CCB/ECB ADDRESS: 00A83538
```

Example of Soft Wait Condition

```
DATE DUMP WAS TAKEN:  03/19/04                      DUMP TYPE:  SADUMP
SUPERVISOR ID:  Y-ESA10/27-16.20                    SUPERVISOR NAME:  $$A$SUPX
SYSTEM STATUS:  SOFT WAIT
CURRENT TASK:  BG MAIN TASK                         BASE PHASE:  NO NAME



DEVICE ANALYSIS FOR ACTIVE NON TP DEVICES ONLY:

DEV  TYPE  TSK  I/O REQUEST STATUS AND INFORMATION
---  ----  ---  ---------------------------------------------------------------
(NO BUSY DEVICES AND NO DEVICES WITH I/O QUEUED)



TASK ANALYSIS FOR ACTIVE TASKS ONLY:

TASK NAME     STATUS    TASK INFORMATION
------------  --------  -------------------------------------------------------
CST TASK      WAITBND   WAIT FOR I/O OR ECB POST    CCB/ECB ADDRESS: 000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T13 TASK      WAITBND   WAIT FOR I/O OR ECB POST    CCB/ECB ADDRESS: 03F1F0
AR TASK       WAITBND   WAIT FOR I/O OR ECB POST    CCB/ECB ADDRESS: 000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
BG MAIN TASK  WAITBND   WAIT FOR I/O OR ECB POST    CCB/ECB ADDRESS: 403504
                        JOB CONTROL ACTIVE IN THIS PARTITION
                        OCCF SERVICE REQUEST PENDING

SYSDUMP.BG.TEST

NAME = LOCATORS COMPONENT ID =          TYPE = LOCATOR DATA
⋮
```

Example of Running System (Loop) Condition

```
DATE DUMP WAS TAKEN:  02/11/04                 DUMP TYPE:  SADUMP
SUPERVISOR ID:  Y-ESA12/15-09.11               SUPERVISOR NAME:  $$A$SUPX
SYSTEM STATUS:  RUNNING                        PSW:  040C0000 0001AB0C
CURRENT TASK:  BG MAIN TASK                    BASE PHASE:  CANCTEST

AREA POINTED TO BY PSW:  SUPERVISOR

DEVICE ANALYSIS FOR ACTIVE NON TP DEVICES ONLY:

DEV  TYPE  TSK  I/O REQUEST STATUS AND INFORMATION
---  ----  ---  ---------------------------------------------------------------
00E  1403  BG   I/O NOT STARTED, REASON UNKNOWN

TASK ANALYSIS FOR ACTIVE TASKS ONLY:

TASK NAME     STATUS    TASK INFORMATION
------------  --------  -------------------------------------------------------
CST TASK      WAITBND   WAIT FOR I/O OR ECB POST    CCB/ECB ADDRESS: 000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
T13 TASK      WAITBND   WAIT FOR I/O OR ECB POST    CCB/ECB ADDRESS: 03EBC0
AR TASK       WAITBND   WAIT FOR I/O OR ECB POST    CCB/ECB ADDRESS: 000000
                        SVC RETRY INDICATOR ON      SVC: 1D (HEX)
BG MAIN TASK  READY     READY TO RUN
                        TERMINATOR ACTIVE FOR TASK

.
.
.
.
.
.
```

Maintaining SYSDUMP Using REXX Procedure DMPMGR

The REXX program DMPMGR is stored as skeleton SKDMPMGR in ICCF 59. You can use DMPMGR to:

1. Check the current usage of dump space.
2. Automatically take appropriate actions if the dump library space is becoming full.

DMPMGR determines the current size of the dump library. If a certain limit is exceeded, certain actions are taken:

- A console message is written
- Dumps of certain partitions are deleted
- Dumps of certain partitions are printed
- Dumps of certain partitions are offloaded

You can specify:

- A date, to handle dumps only that are created during this date or earlier.
- An age, to handle dumps only that are older than the given age.

The printing and offloading of dumps are done using INFOANA–invocations in separate POWER jobs. Deletion of dumps is done within one POWER job.

The parameters that you use together with DMPMGR are:

LIMIT=nn
The percentage of used library space that must be exceeded to trigger actions. Default is 90%.

MSG To issue the highlighted console message `!!! Dump Library is almost full !!!` if the SYSUMP library is becoming full. This parameter is optional.

DELETE=(p1,...,pn) or DELETE=p1
The list of partitions whose dumps are to be deleted if dump library runs full. This parameter is optional.

PRINT=(p1,...,pn) or PRINT=p1
The list of partitions whose dumps are to be printed if dump library runs full. This parameter is optional.

OFFLOAD=(p1,...,pn) or OFFLOAD=p1
The list of partitions whose dumps are to be saved on tape. This parameter is optional.

TAPE_UNIT=ttt
A tape unit to be used for offloading dumps

DATE=yy-mm-dd
A date-limit for dump processing. This parameter is optional. Default is the current date.

AGE=n
An age-limit (in days) for dump processing. This parameter is optional. Default is 0 days.

CLASS=p
The VSE/POWER job class of the generated jobs. This parameter is optional. Default is CLASS Y.

DISP=D or DISP=H
The disposition of the generated jobs. This parameter is optional. Default is DISP=H.

LIB=*lib.sublib*
The name of the library where the dump resides. This parameter is optional. Valid values are:
- SYSDUMP (the dump resides in the system dump library).
- SYSDUMP.*partition* (the dump resides in the partition dump library).
- PRD2.DUMP (the dump resides in the dump archive).

Default is LIB=SYSDUMP.

Here are two examples of the use of DMPMGR.

```
// EXEC REXX=DMPMGR,PARM='LIMIT=85 DELETE=BG MSG PRINT=F2'
```

when using the PARM-operand to specify the parameters.

```
// EXEC REXX=DMPMGR
MSG
DELETE=(F5,F6,F7)
OFFLOAD=(F2,F8)
LIMIT=89
TAPE_UNIT=181
DATE=03-04-30
CLASS=C
DISP=D
```

when using SYSIPT to specify parameters.

DMPMGR ends with one of these return codes:

0 Successful, limit not exceeded.

1 Successful, limit exceeded.

4 Syntax error.

8 Invocation of LIBR LD L=SYSDUMP failed.

12 Submission of INFOANA - PRINT DATA job failed.

14 Retrieval of INFOANA - PRINT DATA job output failed.

16 Submission of INFOANA - PRINT job failed.

18 Submission of INFOANA - OFFLOAD job failed.

20 Submission of INFOANA - DELETE job failed.

The following skeleton SKDMPMGR in ICCF library 59 demonstrates how you can schedule a job to handle the dump file repetitively. Figure 22 uses operands to schedule every 30 minutes for each day.

```
* $$ JOB JNM=DMPMGR,CLASS=Y,DISP=K,
* $$ DUETIME=0000,DUEDAY=DAILY,DUEFRQ=(0030,2400)
// JOB DMPMGR - REXX DUMP MANAGER
* ******************************************************************
*
*  ------ INVOKE REXX DUMP MANAGEMENT PROCEDURE --------------------
*
*  THIS SKELETON MAY BE USED TO CHECK CURRENT USAGE OF DUMP SPACE.
*  IF FILLED TO A CERTAIN DEGREE, CERTAIN ACTIONS ARE INITIATED.
*
* ******************************************************************
// EXEC REXX=DMPMGR,PARM='LIMIT=85 TAPE_UNIT=181 MSG'
OFFLOAD=(F7 F8)
PRINT=(BG,DYN)
DELETE=F2
DISP=D
/*
/&
* $$ EOJ
```

Figure 22. Example of Job for Periodic Dump Library Management

Extending the SYSDUMP Library

SYSDUMP is allocated in VSAM space if newly installed, and if the system is upgraded via FSU it still is in BAM space. In order to migrate to VSAM space or extending the dump library refer to skeleton SKDMPEXT in ICCF library 59. The SYSDUMP library holds dumps from all static and dynamic partitions. If SYSDUMP is full, dumps will be written directly to SYSLST instead unless option SYSDUMPC is active.

Taking a Stand-Alone Dump

A *stand-alone dump* is a dump of selected parts of virtual storage. The program records the shared space and the partitions in separate files on the stand-alone dump tape or on a disk device. You are recommended to take a *stand-alone dump* at various points in this manual.

To be able to take a stand-alone dump on tape or disk, you first have to prepare the output device. You need a running system to do that. For that reason:

- If you wish to take stand-alone dumps on tape, always have a number of prepared tapes available. If you have an error situation that requires a stand-alone dump, it is too late to prepare a tape.

- If you wish to take stand-alone dumps to disk, you must have prepared the disk accordingly (the standalone dump program must reside on the disk).

Note: You should take a stand-alone dump only if normal system operation is no longer possible. In cases where a loop or wait condition can be isolated to a partition, create a partition dump with the DUMP command, as described under "DUMP (Dump Storage Areas)" in the *z/VSE System Control Statements* manual. The partition in error may then be canceled and restarted and normal system operation in other partitions need not be interrupted.

Creating the Stand-Alone Dump Program on Tape or Disk

z/VSE provides a dialog to create the stand-alone dump program, either on tape or on disk. From the *Problem Handling* panel select

```
6  (Dump Program Utilities)
```

From the Dump Program Utilities select

```
1  (Create Standalone Dump Program on Tape)
```

or

```
2  (Create Standalone Dump Program on Disk)
```

Note: You cannot create a standalone dump program on an SCSI disk.

Option 1 requests you to enter the tape address (cuu) and, optionally, the density/mode of the tape. z/VSE creates and submits a job stream to write the stand-alone dump program to tape. The job stream includes a // PAUSE statement that allows you to mount the tape.

Option 2 requests you to enter:

- The address of the disk to store the stand-alone dump program.
- The starting address and the length of the dump file (in tracks or blocks).
- Whether the disk is a SYSRES disk (which can be used for an IPL) or not.
  - If the disk is a SYSRES disk, each IPL will first cause a stand-alone dump to be taken. After the dump program has completed, it will transfer control to the IPL. To avoid a stand-alone dump from always being taken, you are recommended to use a work disk for the stand-alone dump program.

Taking the Dump

To take a stand-alone dump, follow the steps as outlined under "Taking a Stand-Alone Dump" in the *z/VSE Diagnosis Tools* manual.

Removing the Stand-Alone Dump Program from Disk

You can **remove** the stand-alone dump program from a disk by entering, from the *Dump Program Utilities* panel:

```
3  (Remove Standalone Dump Program from Disk)
```

Scanning the Dump Files on Tape or Disk

The SCAN function provides a file directory of the dump files on tape or disk. You can scan the latest dump file on tape and any of the dump files stored on disk. To scan the dump file, enter from the *Dump Program Utilities* panel:

```
4  (Scan Dump Files on Tape)
```

or

```
5  (Scan Dump Files on Disk)
```

The dialog 4 (Scan Dump Files on Tape) requests you to enter the VSE device address of the tape to be scanned by the dump program.

The dialog 5 (Scan Dump Files on Disk) requests you to enter the VSE device address of the disk, and also the starting position and the length of the dump file to be scanned by the dump program.

The SCAN function can be used to scan stand-alone dump tapes, dump files on disk, or dump tapes created with the AR DUMP command.

Printing Stand-alone Dump

You can print stand-alone dumps that reside on disk or tape. To print a stand-alone dump, enter

```
8  (Print Standalone Dump)
```

from the *Dump Program Utilities* panel. The dialog 8 (Print Standalone Dump) requests you to indicate whether the dump resides on disk or on tape, VSE device address, file number if the dump is stored on tape or starting position and length for the dump stored on disk.

Using the Info/Analysis Program

The z/VSE dialogs for dump processing create job streams which invoke the Info/Analysis program for dump analysis. You can use the Info/Analysis program in native or batch mode.

For batch mode, you run Info/Analysis in a z/VSE partition and invoke its functions via control statements. These can be entered either from the system console or from SYSRDR. The output goes to SYSLST. For a detailed description of the functions available see “Info/Analysis:Introduction” in the manual *z/VSE Diagnosis Tools*.

Using the DOSVSDMP Program

With the DOSVSDMP utility program you can create unformatted dump printouts directly from tape or disk. This is required if a stand-alone dump file is too big to fit into any of the existing dump sublibraries.

The *Dump Program Utilities* dialog (Fastpath 46) also provides you with various functions for use with the DOSVSDMP program:

- You can use Fastpath dialogs 461, 462, and 463 to create or remove the stand-alone dump program on tape or disk.
- You can use Fastpath dialogs 464 and 465 to scan the contents of a dump tape, or a file stored on disk.

- You can use Fastpath dialog 466 to scan for ICCF control blocks, format ICCF control blocks, and print ICCF control blocks.
- You can use Fastpath dialog 467 to print the output of the SDAID program.
- You can use Fastpath dialog 468 to print a stand alone dump.

For further details about the DOSVSDMP utility, refer to "The DOSVSDMP Utility" in the manual *z/VSE Diagnosis Tools*.

Taking a CICS Snap Dump

CICS allows to take so called snap dumps using CEMT P SNAP command. For details refer to the *CICS Problem Determination Guide*.

Handling CICS Transaction Dumps

When a CICS transaction comes to an abnormal end, a transaction dump is taken. In z/VSE, the Online Problem Determination program is activated as well. It collects error data and determines the cause of the error. For most errors the program is able to determine the cause of it. But for certain error situations the Online Problem Determination program may not be able to do that. The transaction dump is then the only data available for error diagnosis.

A transaction dump shows all the CICS areas related to the transaction that abended. Among them:

- The task control area (TCA).
- The common system area (CSA).
- The terminal control table (TCT).
- Transaction storage.
- Program Storage.

Analyzing a transaction dump requires a good knowledge of the CICS Transaction Server. Before you can analyze a transaction dump you have to print it.

Printing a CICS Transaction Dump

A transaction dump resides in the CICS Transaction Server dump file. The dump file is split into the DFHDMPA and DFHDMPB file. One of the two files is active, the other inactive. The CICS Transaction Server always writes a transaction dump into the active file. However, you can only print a dump stored in the **inactive** file. If more than one dump is stored there, all the dumps are printed. This may cause a problem for identifying a particular dump. For that reason, consider printing a dump as soon as it has been created. To print a dump, enter CEMT followed by:

```
INQUIRE DUMP
```

Find out whether the dump you want to print is in the active file. If so, set it inactive by switching the two files. Enter:

```
SET DUMP SWI
```

z/VSE provides jobs in the VSE/POWER reader queue for printing:

Job **PRTDUMPA** for printing file DFHDMPA.
Job **PRTDUMPB** for printing file DFHDMPB.
Job **PRTDUC2A** for printing file DFHDMPA for a second CICS.
Job **PRTDUC2B** for printing file DFHDMPB for a second CICS.

If these jobs were accidentally removed, modify and use skeleton SKCOLD to reload them into the VSE/POWER reader queue. Refer to the manual *z/VSE Administration* for details about skeleton SKCOLD.

The jobs described above use the REXX procedure REXDFHDU, which allows you to select specific transaction dumps. For details, refer to the source member REXDFHDU stored in library IJSYSRS.SYSLIB.

Note:

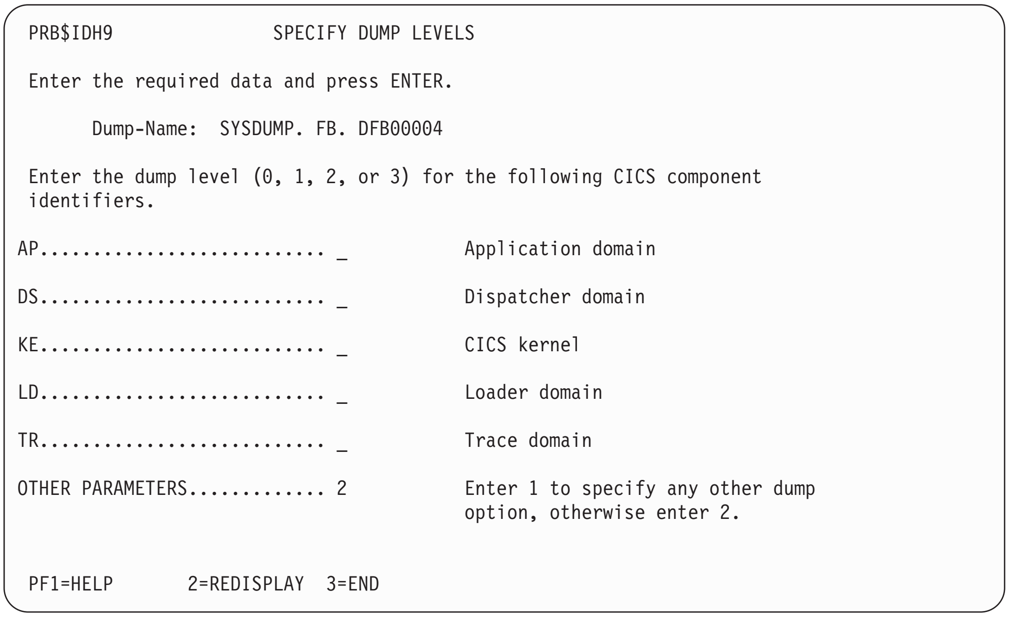
1. If you try to print the active file, the CICS Transaction Server may abend.
2. When switching the dump file, the contents of the new active dump file is erased.

Analyzing a CICS Transaction Dump

The *Storage Dump Management* dialog supports dumps produced by the CICS Transaction Server (CICS TS).

If you select Option 9 (ANALYZE CICS DUMP) on the *Storage Dump Management* panel, several panels (such as shown in Figure 23) are displayed in which you can select the dump levels of various CICS component identifiers.

When you finished entering the required information, press PF5. z/VSE will then create the required job.



```
 PRB$IDH9               SPECIFY DUMP LEVELS

 Enter the required data and press ENTER.

       Dump-Name:  SYSDUMP. FB. DFB00004

 Enter the dump level (0, 1, 2, or 3) for the following CICS component
 identifiers.

AP........................... _            Application domain

DS........................... _            Dispatcher domain

KE........................... _            CICS kernel

LD........................... _            Loader domain

TR........................... _            Trace domain

OTHER PARAMETERS............. 2            Enter 1 to specify any other dump
                                           option, otherwise enter 2.


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

Figure 23. Specify Dump Levels for CICS TS Dumps

“Dump levels” identify the level of detail for the dump report:

- At one extreme, **0** means "none" (no dump details).
- At the other extreme, **3** means “all dump details together with a summary”.
- If you enter a **1**, a series of panels will be displayed in which you can specify various dump parameters.

Dump levels are passed on to the CICS dump utility.

Note: The partition size must be at least 4MB.

Taking a VSE/POWER Disk Dump

If you report a VSE/POWER problem to IBM, you might be asked to supply a dump of the VSE/POWER files. Program IPW$$DD serves this purpose. IPW$$DD prints the data requested on the printer assigned to SYSLST. See also the sample skeleton SKPWRDMP in ICCF Library 59.

For further details refer to *VSE/POWER Administration and Operation* under "Requesting a Dump of the Spool File".

Taking a VSE/VSAM Snap Dump (IKQVEDA)

Several VSE/VSAM modules contain a SNAP dump facility. IBM may ask you to create a SNAP dump to provide error diagnosis information. See also the sample skeleton SKVSMSNP in ICCF Library 59.

Refer to the *VSE/VSAM User's Guide and Application Programming* under "VSE/VSAM Snap Dump (IKQVEDA)" for how to activate and run a SNAP dump.

Chapter 7. Handling Resource Constraint Conditions

This chapter describes how to handle **resource full conditions** and **partition size** constraints.

Full Conditions

System storage areas may now and then reach their storage capacity. If a system file, for example, has reached its limit, it is no longer available to the system. However, z/VSE system files are created with sufficient space to satisfy the needs of the majority of z/VSE users.

This chapter describes what to do when such an exceptional situation occurs. It points to warning signals so that you can take measures before a full condition becomes a reality.

VSE/POWER File Full Condition

A full condition may be caused by the queue file, the data file, or the account file. The queue file holds pointers to areas on the data file where the data elements are stored. The account file is used for saving account information.

VSE/POWER maintains and uses the following queues:

- The RDR (reader) queue, which holds jobs submitted for execution in a batch partition.
- The LST (list) queue, which holds temporarily printer output that has not yet been printed on an actual printer.
- The PUN (punch) queue, which holds temporarily punch output. This output can be read into a VSE/ICCF member. Also, members being exchanged between interconnected systems are stored in the punch queue of the receiving system.
- The XMT (transmit) queue, which holds temporarily job streams and files to be exchanged between interconnected systems.
- The Wait for Run Subqueue, which is a subqueue of the reader queue, and where the subqueue contains jobs that are ready to run.
- The In-Creation (CRE) Queue, which shows the jobs that are in the process of being created in the four VSE/POWER queues (RDR, LST, PUN, and XMT).
- The Deletion (DEL) Queue, which shows the jobs that are in the process of being deleted in the four VSE/POWER queues (RDR, LST, PUN, and XMT).

Any new element stored in a queue or in the account file may lead to a full condition. If this happens, the program trying to store something in a queue or in the account file goes into a wait. The system issues messages such as:

```
1Q38I  NO DASD SPACE AVAILABLE FOR task,cuu
1Q32A  NO MORE ACCOUNT FILE (IJAFILE) SPACE FOR task,cuu
1QF4I  NO FREE QUEUE RECORD AVAILABLE FOR task,cuu
```

When using one of the z/VSE *Manage Batch Queues* dialogs, a display station user may get a message that the queue file is not accessible. If this happens, go to the system console to either delete or print the job.

Preventive Measures

To avoid a full condition, keep track of the amount of VSE/POWER space available. Enter:

```
D Q
```

The system responds with the following type of messages:

```
1R49I  QUEUE FILE 045% FULL - 406 FREE QUEUE RECORDS
1R49I  USED QUEUE RECORDS: 328, CRE-Q: 2, DEL-Q: 0
1R49I  RDR-Q: 97, LST-Q: 223, PUN-Q: 5, XMT-Q: 1
1R49I  QUEUE FILE EXTENT ON CKD-230, SYS001, 1005, 8
1R49I  DATA FILE 042% FULL - 733 FREE DBLK GROUPS
1R49I  CURRENT DBLK SIZE=07548, DBLK GROUP SIZE=00008
1R49I  DATA FILE EXTENT 1 ON CKD-231, SYS002, 6690, 2025
1R49I  ACCOUNT FILE 17 % FULL
1R49I  ACCOUNT FILE EXTENT ON CKD-231, SYS000, 8715, 90
```

If you notice critical values, consider doing the following:

- Start printer if it has not been started yet.
- Delete queue entries that are no longer needed.
- If possible, set in hold those jobs that produce a lot of print output.
- Use the POFFLOAD command to save entries from a queue and reload them later.

Handling a VSE/POWER File Full Condition

If a **data** or **queue** file full condition occurs, space is or can be made available as follows:

1. Automatically.

   Space is made available whenever the system removes a queue entry. For example, when a print operation finishes, the space occupied by the segment which was just printed is freed for reuse. Similarly, when a job finishes execution, the RDR queue entry and its space are freed, or when network transmission is completed, the XMT queue entry and its space is released. The VSE/POWER task or the program which encountered the full condition resumes processing.

2. Through operator action.

   You can clear the condition yourself. Scan the VSE/POWER queues for any data no longer needed. If the LST queue has output which will not be printed, delete it from the queue. If possible, release print output that you may have kept in hold status up to now.

   You may also use the POFFLOAD command to save entries from VSE/POWER queues on tape and reload them later. Consult the manual *z/VSE Operation* for details on the POFFLOAD command.

   When a **looping program** produces excessive output, you cannot completely recover simply by deleting something from the queue. In this situation, you have to cancel the partition (pp in the commands below) where the looping program is running. First try the

   ```
   PFLUSH pp
   ```

   command. If this does not lead to a cancelation, try:

   ```
   CANCEL pp,NODUMP
   ```

   If this command does not work either, try:

   ```
   RC
   15 CANCEL pp,NODUMP
   ```

15 is the reply ID from the system response to the RC command. If you do not succeed in canceling the partition, take a stand-alone dump. Refer to “Taking a Stand-Alone Dump” on page 97 for details. Then re-IPL the system.

3. By extending the VSE/POWER files.

   For details, refer to “Extending the Space for the VSE/POWER Files” below.

If the VSE/POWER **account file** causes a full condition, use the PACCOUNT command for removing the condition. For details about the PACCOUNT command, refer to the manual *z/VSE Operation*.

Extending the Space for the VSE/POWER Files

Refer to the manual *z/VSE Administration* which shows a sample job stream and describes the related tasks in detail under “Extending VSE/POWER Files”.

You might also wish to refer to the skeleton SKPWREXT stored in ICCF library 59, which you can use to extend the VSE/POWER files.

You can extend the VSE/POWER queue and data files by performing a VSE/POWER warm start or reallocate:

- *To extend the VSE/POWER queue file*, at the next VSE/POWER warm start you should run the VSE/POWER queue file reallocation process. For details, refer to the manual *VSE/POWER Administration and Operation*.
- *To extend the VSE/POWER data file* by one or more additional extents, you should use the VSE/POWER data file extension utility. For details, refer to the manual *VSE/POWER Administration and Operation*.

Hardcopy File Full Condition

The system writes to the hardcopy file **any message** (each line of text) that appears on the system console screen. When 80 percent of the hardcopy file space is occupied, the system issues the following message:

```
0D20E  HARD COPY FILE SHOULD BE PRINTED
```

When this message appears on the system console, print the hardcopy file. A hardcopy file printout is essential for any error diagnosis. For details on printing the hardcopy file refer to “Printing the Hardcopy File” on page 113.

If you fail to notice the above message, the system will alert you once more. Message

```
0D25E  HARD COPY FILE IN OVERLAY MODE
```

is displayed if all available space has been used up. At that point, further recording of console messages continues at the beginning of the hardcopy file. As a result, the oldest part of the file is overwritten. You can still print what is left of the old file contents.

If you decide to increase the size of the hardcopy file, you must:

1. Modify the DLBL for the hardcopy file.
2. Initialize the hardcopy file by following the steps described below.

1. Get control in BG during system during IPL, before the first job card is executed. To do so, you must perform an IPL stop, and then specify a JCL procedure that does not exist.
2. Enter the following to recreate the hardcopy file:
   a. Enter `0 set hf=create`.

b. The system then responds with the message `BG-.... .D16D READY FOR COMMUNICATIONS`.

c. Enter `0 // JOB XXXX`.

d. The system then responds with `BG-.... 4433D EQUAL FILE ID IN VTOC IJSYSHC ....`

e. Enter `0 delete`. The file is now formatted, and you can re-IPL your system using the hardcopy file that is increased in size.

f. IPL your system as usual.

Note: z/VSE also provides you with the REXX procedure REXXPRTL, which you can use to manage your hardcopy file. For details, refer to the REXXPRTL source member stored in library IJSYSRS.SYSLIB.

Recorder File Full Condition

The system writes information about **hardware (device) errors** to the recorder file. As more and more information accumulates, space on the recorder file may be exhausted. When this occurs, recording of error information is suspended and the following message is issued:

```
0T05E  RECORDER FILE FULL. RUN EREP
```

You may continue running your system. But you should run the EREP program as soon as possible to save and clear the recorder file. Afterwards, delete message 0T05E from the system console screen. Move the cursor to the message line and press ENTER. This enables the system to continue recording device errors.

For further information refer to “Printing the Recorder File” on page 114.

Preventive Measures

You can keep track of the amount of space available on the recorder file. During IPL, after the first job starts running, the following message appears:

```
1I93I RECORDER FILE IS nnn% FULL [RUN EREP]
```

Watch out for this message or check your latest listing of the hardcopy file for this message. **If the recorder file becomes full, no further recording is done**. This may cause the loss of information essential for error diagnosis.

VSE/VSAM Space Full Condition

A VSE/VSAM space full condition may be caused by one of the following:

- A new file or library is being created but the space available is insufficient for primary allocation.
- During processing, VSE/VSAM tries to allocate additional space to a file or library. But the space available is insufficient for a secondary allocation.

Note: VSE/VSAM allocates additional space according to the value specified for secondary allocation. For the first allocation on a volume, VSE/VSAM always uses the value specified for primary allocation.

From time to time, monitor the system's use of VSE/VSAM space. Try to prevent VSE/VSAM from reaching a situation where no more space is available. A job affected by a shortage of VSE/VSAM space is canceled. Use the Interactive Interface of z/VSE to monitor space utilization. From the *File and Catalog Management* panel select the *Display or Process a Catalog, Space* dialog. Select option:

```
1  (Show Space)
```

The display shows, for the catalog specified, the values of the catalog space and of the data space(s) per volume. Figure 25 on page 116 shows a sample display. For further details on monitoring VSE/VSAM space refer to “Displaying VSE/VSAM Space Usage” on page 116.

You can increase the amount of available VSE/VSAM space by performing one of the following tasks:

- Deleting files.
- Defining additional VSE/VSAM space.

Deleting VSE/VSAM Files

You may delete VSE/VSAM files no longer needed. Access the *File and Catalog Management* panel and select:

```
1  (Display or Process a File)
```

You get a fulist of the files managed by the catalog specified. Select the DELETE option for the file you want to delete. The manual *VSE/ESA Programming and Workstation Guide* describes the dialog in detail under “Display or Process a File”.

You may also save files on tape and delete them afterwards. Save important files twice on different tapes. To save files on tape, access the *Backup/Restore VSAM Objects* panel and select the

```
3  (Backup VSAM File)
```

dialog. The manual *z/VSE Operation* describes the dialog in detail under “Backing Up VSE/VSAM Files”.

Defining Additional VSE/VSAM Space

From the *File and Catalog Management* panel select the *Display or Process a Catalog, Space* dialog. Select the DEFINE SPACE option. With this option you can assign free space on a volume to VSE/VSAM. For details about the dialog, refer to the *z/VSE Administration* manual under “Displaying or Processing a Catalog or Space”.

VSE/VSAM User File Full Condition

This condition occurs if a file reaches the maximum of 123 possible secondary allocations. You must redefine the file and increase the values for primary and secondary allocations. Proceed as follows:

1. Save the file on tape.

   Save an important file twice on different tapes. To save a file, select the COPY option of the “Display or Process a File” dialog. The manual *VSE/ESA Programming and Workstation Guide* describes the dialog in detail.
2. Delete the file.

   From the *File and Catalog Management* panel select the “Display or Process a File” dialog. Select the DELETE option for the file to be deleted. The manual *VSE/ESA Programming and Workstation Guide* describes the dialog in detail.
3. Define a new file.

   From the *File and Catalog Management* panel select the “Define a New File” dialog. Select larger values for the primary and secondary allocations of the file. The new primary allocation should be at least as large as the original primary allocation plus all original secondary allocations. For further details about the dialog, refer to the manual *z/VSE Administration*.
4. Restore the file from tape.

To restore the file, select the COPY option of the “Display or Process a File” dialog as you did when saving the file. The manual *VSE/ESA Programming and Workstation Guide* describes the dialog in detail.

VSE/VSAM User Catalog Full Condition

With the LISTCAT function of VSE/VSAM or with the dialog *File and Catalog Management* you must first determine the contents of the catalog. Proceed as follows:

1. Save all VSE libraries managed by the catalog by using the BACKUP function of the Librarian program.
2. Save the remaining VSE/VSAM files by using the BACKUP function of VSE/VSAM.
3. Now delete all files in the catalog and also the VSE/VSAM space. Be aware of the fact that system files cannot be erased by using the dialogs. For such files, the generated jobs have to be modified and submitted for processing. It is also not possible to erase files that are currently in use.
4. When all files and spaces associated with the catalog have been deleted, the corresponding catalog can be deleted and redefined with new values.
5. Next you have to redefine the libraries and restore their contents by using the Librarian program LIBR.
6. Finally, restore the remaining VSE/VSAM files by using the RESTORE function of VSE/VSAM. If system files are contained in the catalog, you should not use the Interactive Interface until all files have been restored. The same is true for other applications, which means that the applications should not be used until the catalog has been completely repaired.

Non-VSE/VSAM User File Full Condition

If one of your non-VSE/VSAM files becomes full, you can extend the space defined for it as follows:

1. Use the *Display VTOC* dialog to obtain disk volume information needed for extending space. For details on the dialog, refer to “Displaying the Table of Contents (VTOC) of a Disk Volume” on page 118.
2. Save the file on tape.

   Save an important file twice on different tapes. Use the DITTO/ESA for VSE program which is part of z/VSE.
3. Update the label area information by enlarging the EXTENT values as required.
4. Restore the file from tape.

   Use the DITTO/ESA for VSE program which is part of z/VSE.

   Note: z/VSE provides the application profile IESDITTO. Adding this profile to a selection panel for a user provides access to DITTO via the Interactive Interface.
   Consult also the DITTO/ESA for VSE documentation.
5. Do not forget to update label procedure STDLABUS.

VSE Library Space Full Condition

Monitor your libraries to detect a possible full condition before it occurs. Use the Librarian command LISTD to create a listing of the library directory. This listing indicates how many blocks or tracks are occupied by a particular library. For details on the LISTD command refer to "Using the LISTD Librarian Command" on page 128. When a library is full, the system issues one of the following messages:

```
L201I  LIBRARY xx...xx IS FULL

L268I  MAXIMUM NUMBER OF EXTENTS (32) ALLOCATED TO LIBRARY
       xx...xx - NO FURTHER EXTENSION POSSIBLE

L278I  VSAM DATA SPACE EXHAUSTED WHEN ATTEMPTING TO EXTEND
       LIBRARY xx...xx
```

Try to regain library space by:

1. Deleting library members, **if possible**. Use the DELETE command of the Librarian program LIBR. For details, refer to "Delete a Library, Sublibrary, or a Member" in the *z/VSE Guide to System Functions*. Before you issue the DELETE command, ensure that no library members are deleted which are still being shared and used by another partition or CPU.
2. Issuing the Librarian RELEASE command. This should be done if the library directory shows a substantial value for DELAYED SPACE. Create a library directory listing with the LISTD command to find out about delayed space.

   For details about the LISTD command, refer to "Using the LISTD Librarian Command" on page 128.

   For details about the RELEASE command, refer to "Release Space for a Library or Sublibrary" in the *z/VSE Guide to System Functions*.

If you cannot regain space in this way, you have to extend the library space. You must distinguish between libraries in VSE/VSAM and non-VSE/VSAM managed space.

Extending a User Library in VSE/VSAM Managed Space

VSE/VSAM Space Exhausted: Message **L278I** indicates that no more VSE/VSAM space is available for secondary allocations. To extend the VSE/VSAM space, access the *File and Catalog Management* panel and select the "Displaying or Processing a Catalog or Space" dialog. Select the DEFINE SPACE option. With this option you can assign free space on a volume to VSE/VSAM. For details about this dialog, refer to the *z/VSE Administration* manual.

Secondary Allocations for Library Exhausted: Message **L268I** indicates that the maximum number of 16 secondary allocations has been reached. To regain library space, you must perform a backup/restore run and specify new values for the primary and secondary allocations of the library. Proceed as follows:

1. Save the library on tape. Save an important library twice on different tapes. Access the *Backup/Restore Library Objects* panel. Select the dialog:

   ```
   1  (Backup VSE Library on Tape)
   ```

   For further details about this dialog, refer to the manual *z/VSE Operation* under "Backing Up VSE Libraries".
2. Delete the library.

   Use the Librarian program LIBR and create a job stream to delete the library.

   Refer to "Delete a Library, Sublibrary, or a Member" in the *z/VSE Guide to System Functions*.

3. Delete the file definition.

   From the *File and Catalog Management* panel select the “Display or Process a File” dialog. Select the DELETE option to delete the VSE/VSAM file definition for the library. For details about this dialog, refer to the *VSE/ESA Programming and Workstation Guide* manual.
4. Define a new library.

   From the *File and Catalog Management* panel select the “Defining a Library” dialog. Select larger values for primary and secondary allocation. The new primary allocation should be at least as large as the original primary allocation plus 16 times the original secondary allocation. For further details about the dialog, refer to the manual *z/VSE Administration*.
5. Restore the library from tape.

   Access the *Backup/Restore Library Objects* panel. Select the dialog:

   ```
   2  (Restore VSE Library from Tape)
   ```

   For details about this dialog, refer to the manual *z/VSE Operation* under “Restoring VSE Libraries”.

Extending a User Library in Non-VSE/VSAM Managed Space

Message **L201I** is issued if a library full condition occurs. For libraries in VSE/VSAM managed space, additional messages appear. If message L201I appears alone, it indicates that a library in non-VSE/VSAM space is full.

z/VSE provides skeleton SKLIBEXT for extending such a library. Refer to the manual *z/VSE Administration* under “Extending a VSE User Library in Non-VSE/VSAM Space” for details about the skeleton.

Use the *Display VTOC* dialog to obtain disk volume information needed for extending the library space. Refer to “Displaying the Table of Contents (VTOC) of a Disk Volume” on page 118 for details about the dialog.

VSE/ICCF DTSFILE Full Condition

All VSE/ICCF libraries are part of the **DTSFILE**. During startup, a message informs you of the DTSFILE space status:

```
K088I  HI FILE RECORDS=nnn (mm%)
```

The nearer the value of mm is to zero, the less space is available.

If the space limit of the DTSFILE has been reached, the following VSE/ICCF message is displayed at a user display station:

```
*LIBRARY FILE IS FULL
```

As a short term solution, a display station user might try to purge library members owned (using the /PURGE command). Other users might do the same and delete members that are no longer needed.

As a long term solution, increase the amount of space that is allocated to the DTSFILE.

z/VSE provides skeleton SKDTSEXT for extending the DTSFILE. *z/VSE Administration* describes the skeleton and the related tasks in detail under “Using Skeleton SKDTSEXT”.

Dump Library Full Condition

SYSDUMP is now allocated in VSAM space. Thus the full condition only occurs if VSAM space is exhausted. Refer to “VSE/VSAM Space Full Condition” on page 106 for more details.

In order to manage the dump library successfully, you can do the following:
- Deleting dumps that are no longer needed.
- Deleting dumps that are older than a specified number of days.
- Moving dumps to the PRD2.DUMP dump archive.
- Saving dumps onto tape. For details of how to interactively save dumps onto tape, refer to “Using the Interactive Interface for Dump Processing” on page 75.

Note: If your dump library still resides in BAM space, see “Extending the SYSDUMP Library” on page 97 how to extend the dump library.

System Library (IJSYSRS.SYSLIB) Full Condition

In a **DASD-sharing environment**, z/VSE saves system control information in IJSYSRS.SYSLIB during startup. The members created by z/VSE are deleted but the space occupied is not released. This may eventually cause a full condition. To avoid it, perform a MINI startup from time to time and issue the librarian RELEASE command for system library IJSYSRS.SYSLIB.

Partition Size Constraints

Changing VSE Partition Allocations

Each VSE partition is split into two parts:
- A “SIZE” part which is used primarily to hold the programs that are being executed.
- A “GETVIS” part where programs reserve dynamically temporary buffer space during processing.

The partitions of your z/VSE system have been allocated according to the environment chosen during initial installation. It may happen that a program needs more partition space than allocated. The system informs you by issuing messages. For example:

```
0S00I  xx name CANCELED
0P77I  INVALID STORAGE ADDRESS
4879I  GETVIS FAILED RC=001
1Q26I  GETVIS AREA TOO SMALL
```

A frequent occurrence of message

```
1Q85I  task,cuu WAITING FOR VIRTUAL STORAGE, xxxxxxxx BYTES REQUESTED
```

also indicates a lack of GETVIS storage.

z/VSE provides skeletons (SKALLOCx) to let you tailor the static partition allocations. The manual *z/VSE Administration* describes the skeletons in detail under “Skeletons for Static Partition Allocations”. For dynamic partitions, you must update the dynamic class table described in the same manual.

Changing Interactive Partition Allocations

Changing the interactive partition sizes might be necessary if the current sizes of your interactive partitions are insufficient to run specific dialogs. For example, if you have defined a high number of devices in your system, you can get the ST01 abend in the *Configure Hardware* dialog.

If you change the size of your interactive partitions, you must tailor VSE/ICCF with new size values. To perform this task, you can either use:

- Skeleton SKICFGEN, which is described in the section skeleton SKICFGEN of the manual *VSE/ICCF Administration and Operation*.
- The optional VSE/ICCF generation phase DTSIGENM, which is shipped in library IJSYSRS.SYSLIB. You can select and activate this phase to increase the sizes of your VSE/ICCF partitions. For example, this might be necessary if the current sizes of your interactive partitions are insufficient to run specific dialogs.

Chapter 8. Identifying the Problem Area

This chapter describes how you produce the reports and listings that help to determine the area where the problem lies within your system. These reports and listings might also be required by IBM Support. This chapter contains these main sections:

- "Printing the Hardcopy File"
- "Printing the Recorder File" on page 114
- "Displaying VSE/VSAM File Information Online" on page 115
- "Printing the Contents of a VSE/VSAM File" on page 116
- "Displaying VSE/VSAM Space Usage" on page 116
- "Printing a VSE/VSAM Catalog" on page 117
- "Displaying the Table of Contents (VTOC) of a Disk Volume" on page 118
- "Listing I/O Assignments" on page 120
- "Printing Label Information" on page 120
- "Printing the System History File" on page 121
- "Displaying the z/VSE Level" on page 121

Note: This chapter contains several job streams. As documented, these job streams run in the BG partition. However, you can run these job streams in any batch partition of your system.

Printing the Hardcopy File

The system writes to the hardcopy file each line that appears at the system console.

During processing there may be a need to print the hardcopy file or parts of it. For example, the hardcopy file becomes full. You should then print its contents before it is overwritten (see "Hardcopy File Full Condition" on page 105). Also, there may be a need to check what happened during the day so far or which messages were issued for a certain partition. A printout of selected data from the hardcopy file provides that information.

To request a printout of the hardcopy file, proceed as follows:

1. Type // EXEC PRINTLOG and press ENTER. The following message appears on the screen:

   ```
   ENTER OPTIONS FOR PRINTLOG OR ? FOR A LIST OF OPTIONS
   ```

   By entering a ?, the following list of options is displayed:

   ```
   VALID OPTIONS:          (CHOOSE MAX ONE OF EACH LINE)

   ALL OR NEW              TOTAL FILE OR ONLY NEW RECORDS
   CMD, IPL, AR, BG, F1... COMMANDS, IPL-MSGS OR PARTITION ID
   A, S OR N               ACTION, SUPPRESSED OR 'NETVIEW' RECORDS
   JOBNAME=NAME            ONLY RECORDS OF THAT JOB
   MM/DD/YYYY              ONLY RECORDS OF THAT DATE
   ```

2. Select the desired options and enter them. The options must be separated by commas; intervening blanks are not permitted.

   You may enter a null line (simply press ENTER). The system then uses the option **ALL** by default.

Refer to the manual *z/VSE System Utilities* for further details about the PRINTLOG utility and the options available.

Printing the Recorder File

Whenever a hardware error occurs, the system writes error information into the recorder file. With the **Environmental Recording, Editing and Printing (EREP)** program you can produce reports based on this error information.

You are not expected to interpret these reports. They are intended for use by IBM's service personnel who will instruct you to provide a particular report.

Note that the EREP reports show the 4-digit physical device address (pcuu), not the 3-digit VSE device address (cuu).

The EREP program produces two types of reports:

1. Statistics of device failures, tailored to each device type. Here the input is the original recorder file, that is: still on disk. This type of report should be produced right after the device malfunction became apparent.

   How to request a particular EREP report, is described in the manual *EREP User's Guide*.
2. All records of the recorder file.

   Input to the EREP program is a magnetic tape. This tape is created by copying the recorder file (from disk) to tape.

You should save the information stored in the recorder file if you have a full condition. Proceed as follows:

1. If you want to perform POWER OFF or IML after running the EREP program, you should first issue the ROD command. Otherwise, go directly to step 2.

   The ROD command saves specific hardware information that might otherwise be lost. Enter at the system console:

   ```
   R RDR,PAUSEBG
   0 ROD
   ```

   0 is the partition ID.
2. To invoke EREP, enter the following job stream at the system console:

   ```
   R RDR,PAUSEBG
   0 // TLBL HISTOT
   0 // ASSGN SYS009,cuu
   0 // EXEC IFCOFFLD,SIZE=AUTO
   ```

   where 0 is the BG partition ID. The TLBL and ASSGN statement define the tape on which the recorder file is saved. *cuu* is the tape unit.

   The job stream does the following:

   - It creates a system summary report of the data stored in the recorder file. The system summary report is printed on SYSLST.
   - It writes the contents of the recorder file to the tape defined.
   - It reinitializes the recorder file.

Displaying VSE/VSAM File Information Online

As a z/VSE user, you can display file information directly at a user display station. Proceed as follows:

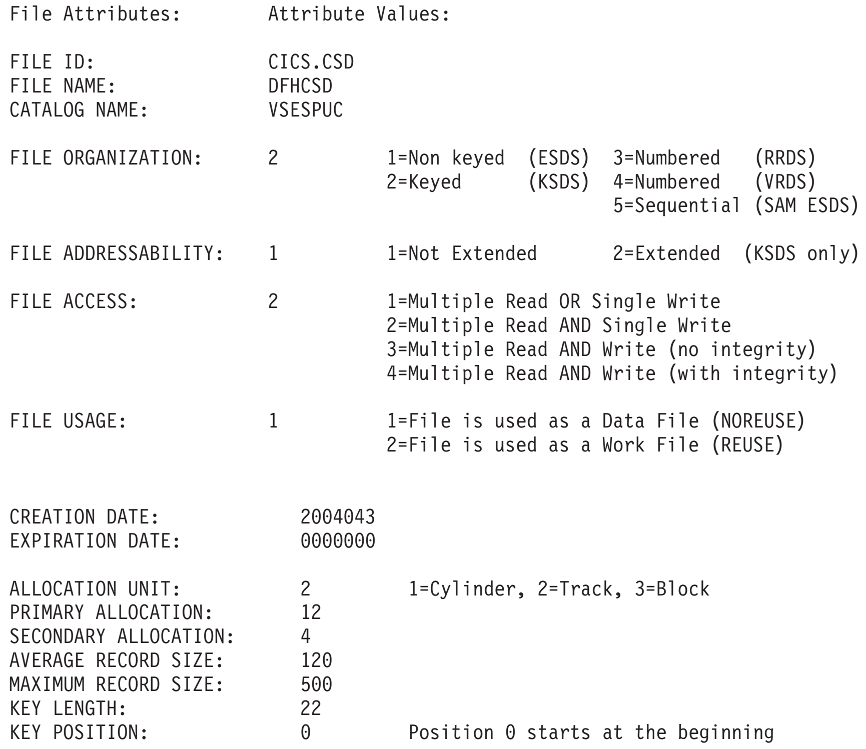
1. Select the *File and Catalog Management* panel.
2. From it, select the *Display or Process a File* dialog.

   You get a FULIST display of the files stored under the catalog name displayed. The catalog name displayed is user-profile defined. Usually, the catalog name displayed is the master catalog (IJSYSCT). You can change that name if your file is defined in another catalog.
3. The FULIST display you get identifies the files by **file ID** and **file name**. Enter

   ```
   1  (Show)
   ```

   in the option column for the file you are looking for. You get a display similar to that shown in Figure 24.

Use **PF10** to display space information for a particular file.



```
File Attributes:        Attribute Values:

FILE ID:                CICS.CSD
FILE NAME:              DFHCSD
CATALOG NAME:           VSESPUC

FILE ORGANIZATION:      2          1=Non keyed  (ESDS)  3=Numbered   (RRDS)
                                   2=Keyed      (KSDS)  4=Numbered   (VRDS)
                                                        5=Sequential (SAM ESDS)

FILE ADDRESSABILITY:    1          1=Not Extended       2=Extended  (KSDS only)

FILE ACCESS:            2          1=Multiple Read OR Single Write
                                   2=Multiple Read AND Single Write
                                   3=Multiple Read AND Write (no integrity)
                                   4=Multiple Read AND Write (with integrity)

FILE USAGE:             1          1=File is used as a Data File (NOREUSE)
                                   2=File is used as a Work File (REUSE)


CREATION DATE:             2004043
EXPIRATION DATE:           0000000

ALLOCATION UNIT:           2         1=Cylinder, 2=Track, 3=Block
PRIMARY ALLOCATION:        12
SECONDARY ALLOCATION:      4
AVERAGE RECORD SIZE:       120
MAXIMUM RECORD SIZE:       500
KEY LENGTH:                22
KEY POSITION:              0         Position 0 starts at the beginning
```

Figure 24. Example of a File Information Display

Printing the Contents of a VSE/VSAM File

You can print the contents (records) of a VSE/VSAM file on the system printer. Proceed as follows:

1. From the *File and Catalog Management* panel select the *Display or Process a File* dialog.

   You get a FULIST display of the files stored under the catalog name displayed. You may change the catalog name in the *File and Catalog Management* panel.
2. Enter

   ```
   3  (Print)
   ```

   in the option column for the file you are looking for. You can now specify the characteristics of your printout:
   - Print format (character, hexadecimal, or both)
   - Print entire or partial file.

     If you choose 'partial', you can start printing with a particular key (KSDS files only) or with a particular record number. You stop printing by specifying the last key (KSDS files only) or the number of records (last record number for RRDS files).

You can submit the job stream created by z/VSE immediately or later. The job stream is saved as member F$xxxx in your primary VSE/ICCF library, where xxxx is your user ID.

The output is stored in the VSE/POWER List Queue. As long as the output is in the List Queue you can display it on your screen before it is printed on the system printer.

Displaying VSE/VSAM Space Usage

You can monitor how z/VSE uses the space defined for VSE/VSAM files. This information is stored in the VSE/VSAM catalog(s). To obtain a display, proceed as follows:

- From the *File and Catalog Management* panel select the *Display or Process a Catalog, Space* dialog.

  You get a FULIST display of the VSE/VSAM catalogs.
- Enter

  ```
  1  (Show Space)
  ```

  in the option column for the catalog you are looking for. You get a display similar as that shown in Figure 25.

```
CATALOG:   VSESP.USER.CATALOG                             VSESPUC


                                CYLINDERS OR TRACKS / BLOCKS
                VOLID   TYPE         ALLOCATED     USED       FREE

CATALOG SPACE:  SYSWK1  3380               150      150          0

DATA SPACE(S):  DOSRES  3380              2010      684       1326
                SYSWK1  3380               975      842        133
```

Figure 25. Example of VSE/VSAM Space Usage Display

Your VSE/VSAM space is most likely distributed over several volumes. To determine the free space available for extension on a particular volume, you need the information provided by a VTOC display. Refer to “Displaying the Table of Contents (VTOC) of a Disk Volume” on page 118 for details.

Printing a VSE/VSAM Catalog

A printout of a VSE/VSAM catalog gives you detailed information about the volumes, VSE/VSAM space, and VSE/VSAM files stored on disk devices. For example:

- Creation date.
- Key length (if applicable).
- Record length (average and maximum).
- Statistics
  - Total number of records in the file.
  - Number of records deleted.
  - Number of records inserted.
  - Number of records updated.
- Number of blocks (cylinder/tracks) allocated.
- Disk areas (‘extents’) used for the allocation.

To obtain a printout, proceed as follows:

1. From the *File and Catalog Management* panel select the *Display or Process a Catalog, Space* dialog. You get a FULIST display of the VSE/VSAM catalogs.
2. Enter

   ```
   3  (Print Catalog Contents)
   ```

   in the option column for the catalog you are looking for.

   You can submit the job stream created by z/VSE immediately or later. The job stream is saved as member F$xxxx in your primary VSE/ICCF library, where xxxx is your user ID. You can change the member name assigned by the system.

   The output is stored in the VSE/POWER List Queue. As long as the output is in the List Queue you can display it on your screen before it is printed on the system printer (SYSLST).

The printout is a detailed account of **all VSE/VSAM files** and **all volumes** that hold VSE/VSAM files related to the catalog specified. Look for the serial numbers of those volumes that contain your VSE/VSAM files. The following figure shows a volume PORTION OF A catalog printout. For FBA disk devices you get a similar printout. The space unit is then a BLOCK.

```
 ...
CLUSTER ------- CICS.AUTO.STATS.A
 ...
CLUSTER ------- CICS.AUTO.STATS.B
 ...
VOLUME  ------- DOSRES
 ...
VOLUME  ------- SYSWK1
 ...
VOLUME  ------- SYSWK7
 ...
CLUSTER ------- CICSTS11.SAMPLE.FILEA
     HISTORY
       OWNER-IDENT-------(NULL)     CREATION--------2000.132
       RELEASE----------------2     EXPIRATION------2099.366
     PROTECTION----------(NULL)
     ASSOCIATIONS
       DATA-----CICSTS11.SAMPLE.FILEA.@D@
       INDEX----CICSTS11.SAMPLE.FILEA.@I@
   DATA ------- CICSTS11.SAMPLE.FILEA.@D@
     HISTORY
       OWNER-IDENT-------(NULL)     CREATION--------2000.132
     RELEASE----------------2     EXPIRATION------2099.366
   PROTECTION----------(NULL)
   ASSOCIATIONS
     CLUSTER--CICSTS11.SAMPLE.FILEA
   ATTRIBUTES
     KEYLEN-----------------6     AVGLRECL--------------80     BUFSPACE-------
     RKP--------------------1     MAXLRECL--------------80     EXCPEXIT-------
     SHROPTNS(2,3)   RECOVERY     SUBALLOC         NOERASE     NOCOMPRESS
     NOREPLICAT     UNORDERED     NOREUSE       NONSPANNED     NEVEREXPIR
   STATISTICS
     REC-TOTAL-------------41     SPLITS-CI--------------0     EXCPS----------
     REC-DELETED------------0     SPLITS-CA--------------0     EXTENTS--------
     REC-INSERTED-----------0     FREESPACE-%CI---------15     SYSTEM-TIMESTAM
     REC-UPDATED------------0     FREESPACE-%CA----------7          2001.122
     REC-RETRIEVED---------14     FREESPACE--------------0          X'B5C607CD
   ALLOCATION
     SPACE-TYPE------CYLINDER
     SPACE-PRI--------------1     USECLASS-PRI-----------0     HALRBA-OR-CI---
     SPACE-SEC--------------1     USECLASS-SEC-----------0     HUSRBA-OR-CI---
   VOLUME
     VOLSER------------SYSWK1     PHYREC-SIZE---------4096     HALRBA-OR-CI---
     DEVTYPE-------------3380     PHYRECS/TRK-----------10     HUSRBA-OR-CI---
     VOLFLAG------------PRIME     TRACKS/CA-------------15
     EXTENTS:
     LOW-CCHH-----X'03230000'     LOW-RBA-OR-CI----------0     TRACKS---------
     HIGH-CCHH----X'0323000E'     HI-RBA-OR-CI------614399
 ...
```

Figure 26. Portion of a VSE/VSAM Catalog Printout

Displaying the Table of Contents (VTOC) of a Disk Volume

A VTOC display helps you determine the following for a particular disk volume:

- Free space.
- Files stored.
- Volume layout.

To get a VTOC display, proceed as follows:

1. From the *Resource Definition* panel select the *Display VTOC* dialog. You get a FULIST display of the volumes attached.

2. In the option column enter one of the following options for the volume you are looking for:

   1 (Free Space)

   2 (Files List)

   3 (Volume Layout)

You get displays similar as those shown on the following pages.

Free Space Display

```
IESADMFREC            VOLUME FREE SPACE INFORMATION             Page  1 of  1
VOLUME: DOSRES              ADDRESS: 150              TYPE: 3380

TOTAL NUMBER OF TRACKS:  13259       FREE TRACKS:    893        94 % FULL

  BEGIN           END             EXTENT          RESERVED FOR
 CYL  TRK       CYL  TRK      RELTRK  NUMTRKS       SYSTEM USE

   67  8          67 14         1013       7            *
  210  0         210 10         3150      11            *
  392 10         408 14         5890     245            *
  842  0         883 14        12630     630
```

Figure 27. Example of a VTOC Free Space Display

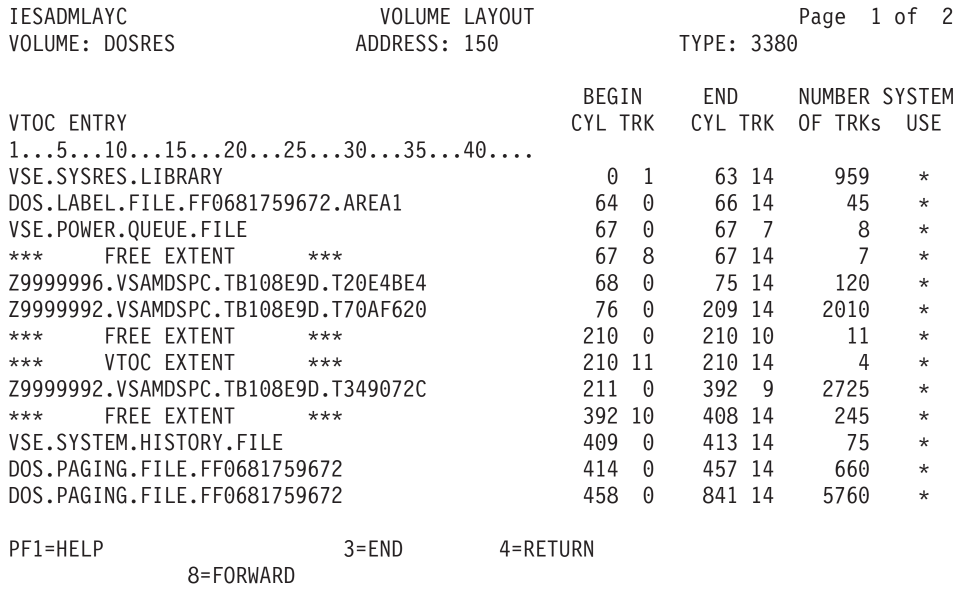
Note: To get correct free space values in a VTOC listing, all files must have been opened. The space of a file that has not been opened is shown as free space.

Files List Display

```
IESADMLSTC                  VOLUME FILES LIST                   Page  1 of  1
VOLUME: DOSRES             ADDRESS: 150             TYPE: 3380

                                                  BEGIN     END      EXTENT
FILE ID                                       SEQ  CYL TRK  CYL TRK RELTRK NUMTRK
1...5...10...15...20...25...30...35...40....
***     VTOC EXTENT      ***                    0   210 11   210 14   3161      4
DOS.LABEL.FILE.FF0681759672.AREA1               0    64  0    66 14    960     45
DOS.PAGING.FILE.FF0681759672                    0   414  0   457 14   6210    660
                                                1   458  0   841 14   6870   5760
VSE.POWER.QUEUE.FILE                            0    67  0    67  7   1005      8
VSE.SYSRES.LIBRARY                              0     0  1    63 14      1    959
VSE.SYSTEM.HISTORY.FILE                         0   409  0   413 14   6135     75
Z9999992.VSAMDSPC.TB108E9D.T349072C             0   211  0   392  9   3165   2725
Z9999992.VSAMDSPC.TB108E9D.T70AF620             0    76  0   209 14   1140   2010
Z9999996.VSAMDSPC.TB108E9D.T20E4BE4             0    68  0    75 14   1020    120




PF1=HELP                   3=END        4=RETURN
                           9=SRT.BEGIN 10=SRT.NAME
```

Figure 28. Example of a VTOC Files List Display

Volume Layout Display



```
IESADMLAYC                   VOLUME LAYOUT                      Page  1 of  2
VOLUME: DOSRES             ADDRESS: 150              TYPE: 3380

                                                BEGIN     END     NUMBER SYSTEM
VTOC ENTRY                                     CYL TRK   CYL TRK  OF TRKs  USE
1...5...10...15...20...25...30...35...40....
VSE.SYSRES.LIBRARY                               0  1     63 14     959    *
DOS.LABEL.FILE.FF0681759672.AREA1               64  0     66 14      45    *
VSE.POWER.QUEUE.FILE                            67  0     67  7       8    *
***     FREE EXTENT      ***                    67  8     67 14       7    *
Z9999996.VSAMDSPC.TB108E9D.T20E4BE4             68  0     75 14     120    *
Z9999992.VSAMDSPC.TB108E9D.T70AF620             76  0    209 14    2010    *
***     FREE EXTENT      ***                   210  0    210 10      11    *
***     VTOC EXTENT      ***                   210 11    210 14       4    *
Z9999992.VSAMDSPC.TB108E9D.T349072C            211  0    392  9    2725    *
***     FREE EXTENT      ***                   392 10    408 14     245    *
VSE.SYSTEM.HISTORY.FILE                        409  0    413 14      75    *
DOS.PAGING.FILE.FF0681759672                   414  0    457 14     660    *
DOS.PAGING.FILE.FF0681759672                   458  0    841 14    5760    *

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

Figure 29. Example of a VTOC Volume Layout Display

Listing I/O Assignments

The listing shows the device I/O assignments per partition. The output device can be the system console (SYSLOG) or SYSLST.

To request the listing on SYSLOG, enter the following commands at the system console:

```
R RDR,PAUSEBG
0 LISTIO ALL
```

where 0 is the BG partition ID. If using the command in the format // LISTIO, the output is directed to SYSLST. For further details refer to “LISTIO (Query I/O Assignments)” in the manual *z/VSE System Control Statements*.

Printing Label Information

Some error messages recommend to list label information. Use the LSERV program for that purpose. The output shows the relation between the DLBL and EXTENT statements. At the system console, enter the following :

```
R RDR,PAUSEBG
0 // EXEC LSERV
```

z/VSE stores the output in the VSE/POWER list queue. When displaying the output, you get a panel similar as that shown in Figure 30 on page 121.

```
IJQFILE
       FILE IDENTIFIER                              VSE.POWER.QUEUE.FILE
       FILE SERIAL NUMBER                           DOSRES
       VOLUME SEQUENCE NUMBER                       01
       CREATION DATE                                OMITTED
       EXPIRATION DATE                              1999/365
       FILE TYPE                                    DIRECT ACCESS
  EXTENT INFORMATION
       EXTENT SEQUENCE NUMBER                       000
       EXTENT TYPE                                  1 (PRIME DATA)
       RELATIVE START ADDRESS IN TRACKS/BLOCKS      055862
       NUMBER OF TRACKS/BLOCKS                      372
       SYMBOLIC UNIT                                SYS001  LOGICAL UNIT
       VOLUME SERIAL NUMBER                         DOSRES

PRD1
       FILE IDENTIFIER                              VSE.PRD1.LIBRARY
       FILE SERIAL NUMBER                           DOSRES
       VOLUME SEQUENCE NUMBER                       01
       CREATION DATE                                OMITTED
       EXPIRATION DATE                              1999/365
       FILE TYPE                                    SEQUENTIAL
  EXTENT INFORMATION
       EXTENT SEQUENCE NUMBER                       000
       EXTENT TYPE                                  1 (PRIME DATA)
       RELATIVE START ADDRESS IN TRACKS/BLOCKS      138136
       NUMBER OF TRACKS/BLOCKS                      063240
       SYMBOLIC UNIT                                OMITTED
       VOLUME SERIAL NUMBER                         DOSRES
          .
          .
```

Figure 30. Example of a LSERV Display

For further details refer to the manual *z/VSE System Utilities*.

Printing the System History File

The printout of the system history file gives you an up-to-date change level record of your system. Whenever you apply a change to your system (a PTF, for instance) you should create such a printout.

To request the listing, access the *Problem Handling* panel and select:

```
5  (Retrace History File)
```

The manual *z/VSE System Upgrade and Service* provides further details about this dialog under “Selection 6 – Retrace History File”.

Displaying the z/VSE Level

The procedure SPLEVEL displays the:

- z/VSE level
- Date when the system was refreshed or built.
- Copyright statement

```
// EXEC PROC=SPLEVEL
```

Chapter 9. Useful Commands and Utilities

This chapter describes how you can collect detailed debugging information using various commands and utilities.

This chapter contains these main sections:

Using the DEBUG Command

DEBUG consists of a set of tracing "hooks" placed at various points within the z/VSE system (but mainly within the supervisor), which are activated using a z/VSE operator command. Once it has been activated, DEBUG will:

1. Create trace entries.
2. Save these trace entries in 31-bit fixed SVA storage, in wrap-around mode.

The trace buffer is very useful for determining the cause of system errors. Since the trace buffer helps to decrease the time required to solve system problems, you should activate it whenever you suspect a system failure. However, when DEBUG is active it can negatively affect your z/VSE system performance.

The command DEBUG ON activates a standard set of traces. The trace types which are defined by default, monitor the:

- Dispatcher program
- First level interrupts
- I/O activity

If you do not explicitly specify the size of the buffer, the system allocates three buffers where each has a size of 16KB.

The command DEBUG TRACE allows you to modify the set of tracing functions. Figure 31 on page 124 shows how you activate and modify the DEBUG command.

The reports described in this section illustrate the use of functions available from VSE/ESA 2.4 onwards. If your VSE system is older than VSE/ESA 2.4, some of the functions described here may be missing and therefore the appropriate command might be invalid.

DEBUG ON[,nnnk]
Activate tracing.

DEBUG id
Activate tracing for the specified partition where **id** is the partition SYSLOG-id (BG, F1,F2...,Zn).

DEBUG OFF
Stop DEBUG temporarily.

DEBUG END
Stop tracing and free all allocated buffers.

DEBUG
Query tracing status. It will display the currently ACTIVE trace points. Any trace point can be selectively enabled or disabled as the following example will show.

DEBUG TRACE=REGS,TASK
Activate REGiSter and TASK-entry trace.

DEBUG TRACE=NOINT,NOSIO
Deactivate INTerrupt and Start-IO trace.

DEBUG TRACE=ALL,NOSVC
Activate all traces, except SVC trace.

DEBUG TRACE=NONE,DISP
Deactivate all traces, then activate DISPatching trace.

Figure 31. DEBUG Activation and Modification Commands

DEBUG STOP

The DEBUG address STOP command compares the contents of a specified storage location with a specified pattern. The z/VSE system performs the compare operation whenever one of the DEBUG event occurs, regardless of whether that entry is active or not. You can compare for:
- equal (EQ)
- not equal (NE)
- low (LO)
- high (HI)

When z/VSE detects a match, it enters a hard wait state with PSW = 000A0000 0000EEEE.

To recover from this hard wait state, you can use either:
- The RESTART feature of the processor's monitoring facilities.
- The SYSTEM RESTART command (when running under VM).

Figure 32 on page 125 shows how you set a DEBUG address STOP.

DEBUG STOP,4B504.4,EQ,FE12ABCD
STOP if the full word at 4B504 equals the pattern FE12ABCD.

DEBUG STOP,F4,5AC00C.1,NE,00,OR,180.4,HI,0004ABC0
STOP if the byte at 5AC00C within the F4 addressing scope differs from zero, OR STOP if the four bytes at address 180 are greater than 0004ABC0. You can also establish an AND condition if that should be required. The above example would then read as follows:

DEBUG STOP,F4,5AC00C.1,NE,00,AND,180.4,HI,0004ABC0
The system would now STOP only if **both** of the previously given conditions are true.

Figure 32. DEBUG STOP Commands

DEBUG [{P|N}]SHOW[,ALL]

The DEBUG command eventually fills three trace buffers in wrap-around mode. Whenever an application program terminates abnormally, the DEBUG command freezes the current buffer and switches to the next buffer (unless switching has been supressed (TRACE=NOSWCH). The buffer switching ensures that important trace data is not overlaid by entries generated by the dump routines.

The DEBUG SHOW command provides some options to show ALL or part of any of the three DEBUG-areas. The DEBUG SHOW command cause the traced entries to be formatted and displayed. It is possible to display the buffer contents on the screen or on a printer device. The SHOW command displays the most recent entries in the current buffer. The PSHOW command displays the most recent entries from the previous trace buffer and the DEBUG NSHOW command displays the most recent entries from the trace buffer which is going to be used next. This next buffer, if it contains trace data at all, contains the oldest trace entries which are available in the system.

You can also restrict the DEBUG SHOW output to certain event types only. In this case you would specify the event option, for example **DEBUG SHOW=SIO,INT** to get only the SIO entries PLUS the INT entries to be displayed/printed.

You are recommended to issue the command DEBUG OFF before you use the DEBUG SHOW command (not necessary for PSHOW or NSHOW). Instead of the DEBUG SHOW command you can also use the command DUMP DEBUG,cuu to print the current DEBUG buffer.

DEBUG SHOW[,CUU=cuu]
Display the current DEBUG buffer.

DEBUG PSHOW[,CUU=cuu]
Display the previous DEBUG buffer.

DEBUG NSHOW[,CUU=cuu]
Display the next DEBUG buffer.

Figure 33. DEBUG SHOW Commands

Using the GETVIS Command

The GETVIS command includes options which allow the retrieval of information about the occupancy of either system, or partition GETVIS storage.

The syntax of the GETVIS command is as follows:

```
GETVIS [{SVA|id}[,{ALL|DETAIL}]]
```

This command is especially useful in case you are running short of GETVIS storage and suspect a program error.

SVA Provides information about the SYSTEM GETVIS areas.

id The partition SYSLOG-id of the partition for which you want GETVIS storage allocation/usage information.

> **ALL** Indicates that you want a summary report about how many total storage has been consumed for the different sub-pools in the appropriate area.
>
> **DETAIL**
> Indicates that you want a detailed report with all the addresses of the pages that have been reserved for the different sub-pools in the appropriate area.

In addition to the information that is provided in the *z/VSE System Control Statements*, these fields are especially useful:

SUBPOOL
Contains the name of the subpool that has either explicitly been specified or which has been assumed by the system (Default). Appended to the 6-byte subpool name, you may find additional information to fully qualify the sub-pool-name within the system.

REQUEST
Contains information about the type of GETVIS request that had been requested.

> **SPACE**
> Indicates that this was initially a DYNAMIC SPACE GETVIS request that had been routed into the SVA, because the program was not running in a dynamic partition.

<--SVA-24-AREA---
Contains information about the areas that have been reserved in the 24-bit storage area. Since the 24-bit area is normally the most critical area, all the entries have been ordered according to their consumption within this area.

--SVA-ANY-AREA-->
Contains information about the areas that have been reserved in the 31-bit storage area.

Below are two examples of how the GETVIS SVA command is used. A partition-example would be similar to the first example, except that the sub-pool names might be different.

```
getvis sva,all
GETVIS USAGE   SVA-24    SVA-ANY                    SVA-24    SVA-ANY
 AREA SIZE:    1,404K     4,880K
 USED AREA:      280K       624K MAX. EVER USED:      428K     1,320K
 FREE AREA:    1,124K     4,256K LARGEST FREE:      1,008K     2,584K
SUMMARY REPORT
SUBPOOL       REQUEST  <--SVA-24-AREA---    --SVA-ANY-AREA-->
Default                              128K                   4K
IPWPWR                                32K                   0K
IJBFCB                                16K                   0K
INLSLD                                12K                   0K
IJBPRC0020    SPACE                    8K                   0K
IJBPRC0090    SPACE                    8K                   0K
IJBPRC00B0    SPACE                    8K                   0K
IJBPRC00A0    SPACE                    8K                   0K
IPTIB                                  8K                   8K
IJBFF100C0    SPACE                    8K                   0K
IJBPRC00C0    SPACE                    8K                   0K
IJBPRC0010    SPACE                    8K                   0K
ISTMAQ                                 4K                   0K
IJBSSP                                 4K                   0K
IPSAV                                  4K                   0K
ISXECB                                 4K                   0K
INLC002200C0  SPACE                    4K                   0K
$ARSMF                                 4K                  32K
IINIT                                  4K                   8K
$ARVIS                                 0K                  16K
$ARVIS                                 0K                  16K
$ARGFT                                 0K                  64K
IJBCSC                                 0K                   4K
VMCFSP                                 0K                   4K
IJBCSM                                 0K                 136K
ILSTCK                                 0K                   4K
SUBPOOL TOTALS                       280K                 296K
```

```
getvis sva,detail
 GETVIS USAGE   SVA-24    SVA-ANY                    SVA-24    SVA-ANY
  AREA SIZE:    1,404K     4,880K
  USED AREA:      280K       624K MAX. EVER USED:      428K     1,320K
  FREE AREA:    1,124K     4,256K LARGEST FREE:      1,008K     2,584K
 SUMMARY REPORT
 SUBPOOL       REQUEST  <--SVA-24-AREA---    --SVA-ANY-AREA-->
 Default                              128K                   4K
                         00236000-00236FFF    02DC0000-02DC0FFF
                         00238000-0023BFFF
                         0023D000-0023DFFF
                         0023F000-00240FFF
                         00242000-00243FFF
                         00245000-00257FFF
                         00260000-00262FFF

 IPWPWR                                32K                   0K
                         00266000-0026DFFF
 IJBFCB                                16K                   0K
                         0023C000-0023CFFF
                         0023E000-0023EFFF
                         00241000-00241FFF
                         00244000-00244FFF
 INLSLD                                12K                   0K
                         00258000-00258FFF
                         00264000-00265FFF
 IJBPRC0020    SPACE                    8K                   0K
                         00277000-00278FFF
 IJBPRC0090    SPACE                    8K                   0K
                         0027D000-0027EFFF
 IJBPRC00B0    SPACE                    8K                   0K
                         00275000-00276FFF
 IJBPRC00A0    SPACE                    8K                   0K
                         00273000-00274FFF
 IPTIB                                  8K                   8K
                         00271000-00271FFF    02DBE000-02DBEFFF
                         00298000-00298FFF    02D31000-02D31FFF
                         00298000-00298FFF    02D31000-02D31FFF
 IJBFF100C0    SPACE                    8K                   0K
                         00263000-00263FFF
                         0026E000-0026EFFF
 IJBPRC00C0    SPACE                    8K                   0K
                         0025B000-0025BFFF
                         0025D000-0025DFFF
 IJBPRC0010    SPACE                    8K                   0K
```

```
                        0025A000-0025AFFF
                        0025C000-0025CFFF
ISTMAQ                                4K                   0K
                        0027B000-0027BFFF
IJBSSP                                4K                   0K
                        00272000-00272FFF
IPSAV                                 4K                   0K
                        00270000-00270FFF
ISXECB                                4K                   0K
                        0026F000-0026FFFF
INLC002200C0  SPACE                   4K                   0K
                        0025E000-0025EFFF $ARSMF
                                      4K                  32K
                        00259000-00259FFF    02DD1000-02DD8FFF
IINIT                                 4K                   8K
                        00237000-00237FFF    02DD9000-02DDAFFF
ILCKSP                                0K                   4K
                                             02D35000-02D35FFF
IJBDSP                                0K                   4K
                                             02D73000-02D73FFF
IXXPDC                                0K                   4K
                                             02DBD000-02DBDFFF
IJBALE                                0K                   4K
                                             02DBF000-02DBFFFF
$ARVIS                                0K                  16K
                                             02DB9000-02DBCFFF
$ARGFT                                0K                  64K
                                             02DB9000-02DBCFFF
IJBCSC                                0K                   4K
                                             02DDB000-02DDBFFF
VMCFSP                                0K                   4K
                                             02DDC000-02DDCFFF
IJBCSM                                0K                 136K
                                             02DDD000-02DFEFFF
ILSTCK                                0K                   4K
                                             02DFF000-02DFFFFF
SUBPOOL TOTALS                      280K                 296K
1I40I  READY
```

Using the LISTD Librarian Command

The LISTD librarian command allows you to display library status and contents at the system console. Assume the system could not find a phase requested for execution, or you need to know the free space of a library. These are typical problems where the LISTD command can help you. Enter at the system console:

```
R RDR,PAUSEBG
0 // EXEC LIBR
```

You are requested for further information, such as:

```
0 LISTD LIB=IJSYSRS OUTPUT=STATUS
0 LISTD SUBLIB=IJSYSRS.SYSLIB
```

The first LISTD command gives you the STATUS DISPLAY of library IJSYSRS; the second the DIRECTORY DISPLAY of sublibrary IJSYSRS.SYSLIB. For examples of both displays, refer to Figure 34 on page 129. For further details on the LISTD command refer to “List Library, Sublibrary, or Member Information” in the *z/VSE Guide to System Functions*.

```
 STATUS DISPLAY      LIBRARY=IJSYSRS                DATE: 2004-02-18
                                                     TIME: 12:09
 --------------------------------------------------------------------
 FILE-ID        : (NOT DISPLAYED FOR IJSYSRS)
 CREATION DATE  : 2004-02-02   19:24
 SUBLIBRARIES   :        1

 LOCATION (BAM) : DEVICE=3380  VOLID=DOSRES CYL =     0.08 -    63.14

 LIBRARY BLOCK  : SIZE=  1024 BYTES   DATA SPACE=   988 BYTES

 TOTAL   SPACE  :   29512 LIBRARY BLOCKS   (100 %)
 USED    SPACE  :   21219 LIBRARY BLOCKS   ( 72 %)
 DELAYED SPACE  :      40 LIBRARY BLOCKS   (  0 %)
 FREE    SPACE  :    8253 LIBRARY BLOCKS   ( 28 %)

 --------------------------------------------------------------------
 SUBLIBRARY CREATION   SPACE     NO. OF      USED  DELAYED    % LIBR.
            DATE       REUSAGE   MEMBERS     LB'S  LB'S      SPACE
 --------------------------------------------------------------------
 SYSLIB     2004-02-02   AUTO       2655    21214       40      72 %


 ====================================================================
0 LD $I*.PHASE
 DIRECTORY DISPLAY    SUBLIBRARY=IJSYSRS.SYSLIB      DATE: 2004-02-18
                                                     TIME: 12:09
 --------------------------------------------------------------------
  M E M B E R      CREATION   LAST     BYTES    LIBR CONT SVA  A- R-
  NAME     TYPE     DATE      UPDATE   RECORDS   BLKS STOR ELIG MODE
 --------------------------------------------------------------------
 $IESEDEF PHASE    99-02-02   -  -     2640 B       3 YES  NO  31 ANY
 $IJBALE  PHASE    99-02-02   -  -     8104 B       9 YES YES  31 ANY
 $IJBALET PHASE    99-02-02   -  -     2040 B       3 YES YES  31 ANY
 $IJBAR   PHASE    99-02-02   -  -    94104 B      96 YES YES  24  24
 $IJBASGN PHASE    99-02-02   -  -     2880 B       3 YES YES  24  24
 $IJBATTN PHASE    99-02-02   -  -     2704 B       3 YES YES  24  24
 $IJBCJC  PHASE    99-02-02   -  -     2936 B       3 YES YES  24  24
 $IJBCRT  PHASE    99-02-02   -  -   136336 B     138 YES YES  31 ANY
 $IJBCSIO PHASE    99-02-02   -  -   176296 B     179 YES YES  31 ANY
 $IJBCSIW PHASE    99-02-02   -  -    82224 B      84 YES YES  31 ANY
```

Figure 34. Example of a LISTD Display

Using the LOCATE Command

The LOCATE command scans the virtual storage for the next occurrence of either a character-string or a hexadecimal-character string in which parts of the string may be unknown. The total string is limited to 16 characters or 32 hexadecimal digits.

The syntax of the LOCATE command is as follows:

```
LOCATE [id,][']string  [([FROM=start][,TO=end][,RUN]]
```

id specifies the SPACE or the PARTITION which is to be scanned for the specified **string**. If **id** is omitted, the space will be defaulted to private space number 0. Valid IDs are:

S SHARED space.

R REAL space.

1 Private space number one.

2 Private space number two.

n Private space number “n”.

BG BG (Background) partition.

F1 F1 (Foreground) partition one.

F2 F2 (Foreground) partition two.

...

FB FB (Foreground) partition eleven.

xy ID of dynamic partition.

' The single, special character that must be used to indicate that the following string is a **character string.** If the **'** is missing, the string is assumed to be hexadecimal digits.

string The string that the user wants to be located; this is limited to 32 characters (32 bytes pattern), or 32 hexadecimal digits (16 bytes pattern):

- Any character or hexadecimal digit that should be **excluded** from the scan, must be presented by a **.** (dot).
- An even number of hexadecimal digits, including **.** (dots), must be specified in case of hexadecimal digit scan.

OPTIONS Any additional processing option(s) that the user wants to specify must be separated from each other by a comma and the first OPTION specified must be preceded by a left parenthesis. The following processing options can be specified:

FROM=start
Specifies an address within the specified **id** where the scan should begin. The FROM option, if omitted will force the scan to be started at the first byte of the given **id**. You may want to specify an offset within the specified **id** rather than an absolute address. In this case, the **offset** must immediately be preceded by a **+**.

TO=end
Specifies the address within the specified **id** where the scan should end:

- If omitted, the TO option will force the scan to be ended at the last byte of the given **id**.
- You may want to specify an **offset** or a **length** rather than an absolute address to indicate where the scan should be ended within the specified **id**. In this case, an **offset** must be indicated by a **+** immediately preceding the offset relative to the begin of the specified **id**, whereas a length can be identified by a leading **.** (dot).

RUN Indicates that all addresses where a match is found (within the calculated or given boundaries) should be logged onto the console without prompting for an operator response:

- The RUN option if omitted will cause the system to display 64 bytes of information starting at the next lower 16-byte boundary which precedes the next subsequent **string** match. The system will then WAIT for an operator response. A NULL response (ENTER) will cause the system to check for the **next** occurrence of a matching **string.** This will be repeated until a NONULL response is received, or, until the scan boundaries are reached.
- “Operator prompting mode” is the default option.

You can terminate the LOCATE function by using the RC command. For a description of the RC (Request Communication) command, refer to *z/VSE System*

Control Statements.

Using the MAP Command

The MAP command provides information about virtual and real storage areas of the system, individual static or dynamic partitions, or the SVA. The output shows, for example:

- Address space identifier.
- Start address.
- Size.
- Execution mode (virtual or real).
- Name of job per partition.

In many cases a MAP display provides valuable information for error diagnosis. For example, the partition addresses provided may be needed for dump analysis. At the system console, enter:

```
MAP
```

Note: To display the actual partition and SVA values during processing, z/VSE provides the "Display Storage Layout" dialog. The manual *z/VSE Administration* describes the dialog in detail under "Using the Display Storage Layout Dialog". You can also use the MAP command to display the address limits of all currently defined **dynamic** partition classes in the system. At the system console, enter:

```
MAP CLASS=ALL
```

If you are interested in a specific dynamic class only, enter:

```
MAP CLASS=class
```

For details, see "MAP (Display Storage Layout)" in the *z/VSE System Control Statements* manual.

Using the /MAP Command

The /MAP command allows you to display the following information about the interactive partitions of VSE/ICCF:

- Partition number and ID.
- Scheduling class(es).
- Virtual start address.
- Size.
- Status.

You may need part of this information to analyze a dump, for example. At the system console, enter:

```
/MAP
```

You get a display similar as that shown in Figure 35 on page 132.

```
/map
AR 0015 1C39I COMMAND PASSED TO VSE/ICCF
F2 0091 K119I  *** INTERACTIVE PARTITION MAP ***
F2 0091   P=01(1) CLS=I    ADR=9C0000 SIZ=1024K USER=     WORK=4
F2 0091   P=02(2) CLS=A    ADR=AC0000 SIZ=0384K USER=     WORK=4
F2 0091   P=03(3) CLS=A    ADR=B20000 SIZ=0384K USER=     WORK=4
F2 0091   P=04(4) CLS=BA   ADR=B80000 SIZ=0512K USER=     WORK=4
F2 0091   P=05(5) CLS=BA   ADR=C00000 SIZ=0512K USER=     WORK=4
```

Figure 35. Example of a /MAP Display

Using the QUERY IO Command

From z/VSE 4.3 onwards, z/VSE supports device addresses (that is, *physical addresses*) of up to X'FFFF'. This support is implemented as follows:

- z/VSE applications, messages, commands, and so on, do not address a device by the physical address (**pcuu**), but instead by the *VSE address* (**cuu**).
- VSE addresses are in the range from X'000' to X'FFF'.
- To each physical address (pcuu) there is a corresponding VSE address (cuu).
- If the physical address is less than or equal to X'FFF', the VSE address (cuu) is equal to the physical address (pcuu).
- If the physical address is higher than X'FFF' (and therefore outside the range of VSE addresses), the physical address (pcuu) and VSE address (cuu) will be different.

CP commands (under z/VM) always use *physical addresses*.

z/VSE jobs, commands, dialogs, and messages use *VSE addresses*. However, in specified cases (for example, when using the `QUERY IO` command) *physical addresses* might be used.

Note: Throughout the z/VSE documentation, the term *address* of a device (used on its own) always refers to the *VSE address*.

QUERY IO command is used to visualize the relationship of the physical address of a device and the address used by z/VSE for the device.

AR , JCC, JCS Format

Syntax of the QUERY IO command:

```
►►──┬────┬──QUERY IO──┬─────────────────────────────────────────────┬──────────────►◄
    └─//─┘            └─,──CUU──=──┬─cuu─┬──┬──────────────────────┬┘
                                   └─ALL─┘  └─,──SORT──=──PHYS─────┘
```

CUU=cuu

For cuu you can specify one to four hexadecimal digits. If you enter one to three digits, the query looks for matching **VSE addresses**. However, if you enter a four digit address, the query automatically searches for a matching **physical device address**. The default display sequence is: VSE address, physical device address, and device class. One line is displayed for each device whose address or part of its address matches cuu.

CUU=ALL

Displays one line for each defined device.

SORT=PHYS

If you specify the operand SORT=PHYS, regardless of how many digits you have entered, the device with the corresponding **physical device address** is identified. The information is now displayed in the sequence: physical address, VSE address, and device class.

If you specify QUERY IO without any other operand, general information about devices is displayed.

```
QUERY IO
AR 0015 MAXIMUM OF I/O DEVICES (IODEV): nnnn, CURRENTLY DEFINED dddd
```

Figure 36. Output example of QUERY IO

Where nnnn displays the number specified in the IODEV operand of the IPL command and dddd the number of devices defined by ADD statements.

Following are examples of the QUERY IO command with additional operands specified. The device class is displayed according to the information in a z/VSE control block.

```
QUERY IO,CUU=1
 AR 0015  VSE ADDR  PHYSICAL ADDR  DEVICE CLASS
 AR 0015      120            0120         DASD
 AR 0015      121            0121         DASD
 AR 0015      150            0150         DASD
 AR 0015      151            0151         DASD
 AR 0015      152            0152         DASD
 AR 0015      190            0190         DASD
 AR 0015      192            0192         DASD
 AR 0015      194            0194         DASD
 AR 0015      19C            019C         DASD
 AR 0015      19D            019D         DASD
 AR 0015      19E            019E         DASD
 AR 0015 1I40I READY
```

Figure 37. Output example of QUERY IO,CUU=1

```
QUERY IO,CUU=ALL,SORT=PHYS
 AR 0015  PHYSICAL ADDRESS  VSE ADDRESS  DEVICE CLASS
 AR 0015         0009           009            SYSLOG
 AR 0015         000C           00C            UNIT-RECORD DEVICE
 AR 0015         000D           00D            UNIT-RECORD DEVICE
    .
    .
    .
 AR 0015         0FEE           FEE            UNIT-RECORD DEVICE
 AR 0015         0FFF           FFF            TERMINAL
 AR 0015         1200           200            DASD
 AR 0015         1201           201            DASD
 AR 0015         2480           480            TAPE
 AR 0015         2481           481            TAPE
    .
    .
    .
 AR 0015 1I40I READY
```

Figure 38. Output example of QUERY IO,CUU=ALL,SORT=PHYS

Using the STACK Command

Note:

1. The STACK command is provided on an unofficial basis and can be changed due to internal purposes. This command is not covered by the official IBM error-reporting processes (for example, the raising of APARs).
2. For a complete description of the STACK command, including many examples, refer to the manual *Hints and Tips For z/VSE* on the z/VSE Homepage (see "Where to Find More Information" on page xiii).

The STACK command is a command that can be used for different purposes. It enables the system administrator or operator to:

- Prepare a sequence of commands and/or replies, give it a name and have this sequence executed whenever that name is being entered or submitted as a command.
- Suppress or change any z/VSE command.
- Abbreviate long z/VSE commands to just a few characters.

In addition:

- If your IPL-device is a CKD-type device, the STACK-data will be preserved throughout subsequent IPL attempts.
- The STACK command, without any further operands, will cause all stacked commands (STACK-buffer) to be displayed on the console.

The STACK command was first intended to provide a means to shut down a z/VSE system by using a single command. For example:

STACK SHUT|MSG F2|23 CEMT P,SHUT|Z NET QUICK|PEND|1 ROD

SHUT Would be the name of the command that the operator would have issued to close down CICS, VTAM and finally POWER.

Note: You are strongly recommended to first start using the STACK command on a test system, and only when you have a good understanding of the command's capabilities.

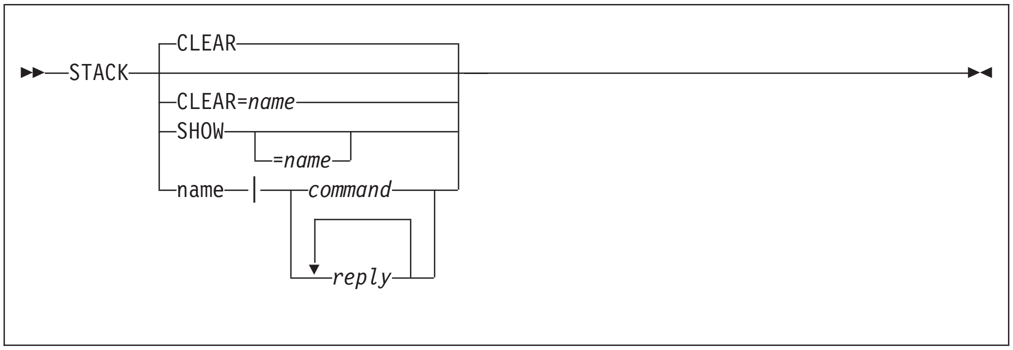


Figure 39. Syntax: STACK Command

STACK
Identifies this command as a STACK command.

name A STACK name that you can freely choose and that you would have to specify to get the sequence of the specified commands and/or replies executed in the specified order. The whole entry is referred to as a STACK-entry and it will be saved in a special STACK-buffer for subsequent use and the whole STACK-buffer will be preserved throughout IPL. The preserved STACK data can only be reset or purged via the STACK CLEAR command. Please note that a STACK-name does supersede a z/VSE command with the same name until this STACK command has been removed (see the option CLEAR).

CLEAR
This is a special keyword indicating that STACK entries currently defined in the STACK-buffer are to be cleared. If you do not supply a STACK-name, then all STACK-entries (the whole STACK-buffer), will be purged. Otherwise, only the named STACK entry will be purged. The option CLEAR is required to force the system to also purge any eventually preserved STACK-data (a single STACK-entry versus the total STACK-buffer).

SHOW
A special keyword indicating that STACK entries currently defined in the STACK-buffer are to be displayed on the operator console. If you do not supply a STACK-name, all STACK entries (the whole STACK-buffer) will be displayed. Otherwise, only the named STACK entry will be displayed. SHOW is the default option that will be executed in case the STACK command has been issued without any further operands.

command
A complete z/VSE-command which may have incorporated special variables (&0 through &9) which will be explained later. The command must be followed by a vertical bar (|) which serves as a separator.

reply is a complete z/VSE-reply which may contain incorporated special variables (&0 through &9) which will be explained later. The reply must be followed by a vertical bar (|) which serves as a separator. A system control statement is considered a reply in this sense.

You can concatenate as many commands or replies assuming the whole STACK command does not exceed 126 bytes, which is the maximum z/VSE command buffer length. None of the operands in the STACK command will be validated at the time it is being entered, it will be saved unchanged into the STACK-buffer. An eventually existing STACK-entry with an identical STACK-name will be overwritten unconditionally. You may specify up to a maximum of 32 STACK-entries. Any of the STACK-entries may contain variables, identified by &n where n is an integer ranging from 0 through 9. These variables will eventually be substituted by operands at execution time, which is when the STACK-name is being entered as a command. &0 will get the first positional parameter assigned, &1 the second positional parameter and so on. If a parameter is being omitted at execution time, then the appropriate variable will be omitted also.

Notes:

- If for some reason you should ever have problems with the STACK processing, then you simply need to issue a **STACK CLEAR** command to get the STACK processing deactivated.

- While STACK processing is ongoing, you can not normally enter other z/VSE commands. To do so, you have to issue the **RC** command which will cause the currently ongoing STACK command processing to be suspended.
- If you are running a z/VSE-system with fix-level DY45926 or above, then the STACK-buffer will be preserved throughout subsequent IPL's.
- The whole STACK-buffer information will be lost in case you have NOT applied the fix for APAR DY45629 and your system has been newly IPLed. In this case, to ensure that you have always your STACK-entries set properly, you could run a simple job at start-up-time which could set up the STACK-entries by using the SVC-30 interface.

  An example of this type of program is shown later in this section. Once it has been assembled and catalogued, you can execute it during your system startup-procedure and it will load the STACK entries similar to what we discussed on the previous pages. The only purpose of this example is to give you an idea of what can be done and how it should be done.
- You need of course to make modifications to this program (mainly the STACK1 and subsequent constants) to make it suit your installations needs. You can of course also use this skeleton program to submit any other z/VSE command during system-start-up time.

Using the STATUS Command

Note: For details of the:
- `STATUS BG|Fn` command
- `STATUS SYS` command

refer to the manual *z/VSE System Control Statements*.

The syntax of the STATUS command is as follows:

```
STATUS cuu[,ALL]
```

The **STATUS cuu** command allows you to retrieve device status information and, if appended with the ALL option, will also provide addressing information about the related I/O control blocks. This command is especially useful in case of device hang conditions.

The command will produce similar output as shown in the example below.

```
status 480,all
  (1.0) SCHIB DEV  INT-PARM ISC FLG LP PNO LPU PI MBI  PO PA CHPID0-3
  (2.0) 0010  0480 00003710   3  83 80  00  80 80 0000 80 80 0AFFFFFF
  (3.0)            KEY SLCC FPIAUZEN FCTL ACTL SCTL CCW-ADDR DS CS CNT
  (4.0)             0   0      00     40  0000  07  00027358 0C 00 000C
  (5.0) REQUEST IS STARTED            DEVICE IS BUSY
  (6.0)   PUB=00003710   PUBX=0007A228   PUB2=00072288   POWN=00003BBC
  (6.0)  VCTE=000712FA  POWNX=00237BC4
AR 0015 1I40I  READY
```

Figure 40. Example of STATUS Command

Explanation:

(1.0) This line is the heading line describing the information provided in the next line.

SCHIB Contains the Subchannel Number of the device that has been inspected.

DEV Contains the device number (cuu) which is being used within z/VSE to identify and address the device.

INT-PARM
Contains the INTerrupt-PARaMeter that had been passed in the ORB at SSCH time.

ISC Contains the I/O Interruption Subclass Code that has been assigned to this subchannel. z/VSE normally only uses the subclass code three, except for SDAID, which has a different subclass code assigned.

FLG The FLaG byte which identifies the status plus some features that have been enabled for this subchannel.

LP this field identifies the Logical-Path(es) that z/VSE did allow the channel subsystem to use in accessing the device.

PNO Identifies the Path(es) that the channel subsystem found Not Operational when attempting to address the device via the identified path(es).

LPU Identifies the Last-Path-Used by the channel subsystem to communicate with the device.

PI Identifies the Path(es)-Installed for this device (as defined in the IOCDS).

MBI The Measurement-Block-Index used by the channel subsystem to calculate the address of the measurement block for this subchannel, assuming that measuring is active.

PO Identifies the Path(es) that where found Operational by the channel subsystem last time it did use that path.

PA Identifies the Path(es) that where found Available but not necessarily operational by the channel subsystem.

CHPID0-3 and CHPID4-7
Contains all the channel path Id's that have been defined (IOCDS) to access a certain device.

(2.0) This line describes how device X'480' with the subchannel number x'0010' and the interrupt parameter X'00003710' has been ENABLED (FLG=8x) for I/O interrupts for subclass 3 and can only be accessed via a single path (PI=X'80') with the CHannel-Path-ID X'0A'.

(3.0) The heading line of dynamic subchannel information provided in the next subsequent line. This line as well as the next line will only be presented if an I/O interrupt is outstanding, or if they had explicitly been requested (option ALL or DEBUG active).

KEY Defines the storage protection KEY that had been used at SSCH time.

SLCC Contains information about the progress of the I/O operation. The important information is the cc-bits, which is the deferred condition code of the I/O operation in progress if it is unequal to zero.

FPIAUZEN
Contains control bits which are of minor interest.

FCTL Contains Function ConTroL information which is:

40 start function

20 halt function

10 clear function

ACTL Contains Activity ConTroL information where the first byte contains PENDING I/O instruction information

04 start pending

02 halt pending

01 clear pending

and the second byte contains the subchannel/device activity information

80 subchannel active

40 device active

SCTL Contains Status ConTroL information where the bits have the following meaning assigned.

10 ALERT status which normally indicates an error condition or the presentation of an unsolicited interrupt.

04 primary I/O interrupt status which normally indicates the completion of data transfer operations.

02 secondary I/O interrupt status which indicates the completion of an I/O operation at the device level.

01 status pending which indicates that I/O interrupt status as defined by the other SCTL bits is PENDING in the subchannel and waiting to be presented to the z/VSE system as soon as interrupts are enabled.

CCW-ADDR
Contains the address+8 of the last CCW that had been executed.

DS Contains the device status information

CS Contains the channel status information

CNT Contains the residual count, that is the number of bytes that had not been transferred.

(4.0) this line is now telling us, that a SSCH operation (FCTL=40) has just completed its operation since primary plus secondary I/O interrupt status is pending (SCTL=07), which means it has not yet been presented to the system. The last executed CCW was at address X'27350 and has completed normally (CE+DE interrupt). 12 bytes of data (CNT=000C) had not been transferred.

(5.0) this line is simply repeating in plain text form the status of the device at the time it was interrogated. It should be self explanatory. REQUEST IS STARTED DEVICE IS BUSY; this is in sync with what we found out before (ref:4.0), because the system has not yet received the I/O interrupt so from its point of view, the device is still busy.

(6.0) this line contains the addresses of some I/O control blocks which might be interesting to know.

Using VM Commands

If your z/VSE system runs as a virtual machine under an IBM VM operating system, you will not have the hardware functions like Alter/Display and Stop-on-Address available. Instead, VM provides CP commands to perform such functions. Here is a selection of CP commands:

TRACE

Use the TRACE command to trace virtual machine activity and to display the results at the terminal, or on a virtual spooled printer, or on both.

ADSTOP

Use the ADSTOP command to halt the execution of a virtual machine at a virtual instruction address. Execution halts when the instruction at the address specified in the command is the next instruction to be executed.

DISPLAY

Use the DISPLAY command to display the virtual machine components at your terminal.

STORE

Use the STORE command to alter the contents of specified registers and locations of the virtual machine.

DUMP

Use the DUMP command to print the contents of various components of the virtual machine on the virtual spooled printer.

You should see the appropriate VM documentation or the VM HELP function (if available) for more information on the use of CP commands.

Using the VSE/VSAM Catalog Check Service Aid Utility

The Catalog Check Service Aid utility helps determine whether a catalog has been damaged. It also determines the type and extent of the damage. In certain situations VSE/VSAM calls the Catalog Check Service Aid automatically. In addition, you should run the aid in the following situations to assess catalog integrity:

- After a system failure.
- When a file or catalog is not behaving as expected.
- As part of regular system maintenance.

For further details refer to the *VSE/VSAM User's Guide and Application Programming* under “VSE/VSAM Catalog Check Service Aid”.

Using the DITTO Utility

DITTO/ESA for VSE (where “DITTO” is an abbreviation for “Data Interfile Transfer, Testing and Operations Utility”) is a general-purpose utility program. It can scan, display, list, alter, and create files or parts of files on card, tape, disk, and diskette input/output devices.

For a detailed description of DITTO functions, refer to the *DITTO/ESA for VSE* documentation.

Note:

1. z/VSE provides the application profile IESDITTO. Adding this profile to a selection panel for a user provides access to DITTO via the Interactive Interface.
2. Alternatively, you can use PF6 and the CICS transaction DITT.

Using DSF Utilities

The DSF (Device Support Facilities) program contains a set of device utilities. For example, you might need them to initialize a new disk device or to assign an alternate track or block. DSF is part of z/VSE. For details on how to use a particular utility, consult the manual *Device Support Facilities User’s Guide and Reference*.

Commands and Utilities Documented in Other Manuals

In the manual *z/VSE Diagnosis Tools* you find details about the following commands and utilities:

- LIBLIST (display library chains).
- LOG (print job control statements).
- LIST (print language translator source code).
- ACTION (print linkage editor map).
- STOP/PAUSE (suspend program execution).
- LISTLOG (list job related information).
- LSERV utility (display the label information).
- LVTOC utility (display VTOC information).
- System console functions:
  - DSPLY/ALTER command.
  - Display/Alter feature.
  - Instruction stepping feature.
  - Stop-on-address compare feature.

In the manual *z/VSE System Utilities* you find details about the following utilities:

- Clear Disk (CLRDK)
- VSE/Fast Copy
- Initialize Tape (INTTP).

Chapter 10. Using Traces

This chapter describes various traces that you can use while attempting to solve system problems.

This chapter contains these main sections:

Using the Interactive Trace Program

The Interactive Trace Program traces the execution of user programs running in static or dynamic partitions. It operates at the level of machine instructions and virtual storage addresses, similar to the CP debugging facilities in z/VM®. z/VSE users may activate the trace program independently in different partitions.

For a detailed description of the Interactive Trace Program, refer to the manual *z/VSE Diagnosis Tools*.

Using SDAID Event Traces

You can use the SDAID program to trace internal events such as:

- Branch instructions.
- Buffer contents (when overflow).
- Cancel or EOJ condition.
- External Interrupt.
- All or selected instructions.
- GETVIS or FREEVIS virtual storage requests (described in "Using an SDAID Getvis/Freevis Trace" on page 143).
- I/O interrupts.
- LOCK or UNLOCK a resource (described in "Using an SDAID LOCK / UNLOCK Trace" on page 143).
- Monitor calls.
- Program checks.
- Program fetch/load.
- SSCH (Start Subchannel) instructions.
- Storage alterations.
- Supervisor calls.

- VTAM buffers.
- VTAM I/O operations.
- XPCC cross-partition communication (described in "Using an SDAID XPCC Trace" on page 143).

The output, called event records, can be stored in a wraparound buffer, written to magnetic tape, or printed on a line printer.

The SDAID event traces are described in "SDAID Overview" of the *z/VSE Diagnosis Tools* manual.

These are the sub-sections of this section:

- "Printing and Analyzing SDAID Information"
- "Using an SDAID Getvis/Freevis Trace" on page 143
- "Using an SDAID LOCK / UNLOCK Trace" on page 143
- "Using an SDAID XPCC Trace" on page 143
- "Sample Jobs Showing the Use of SDAID Traces" on page 144

Printing and Analyzing SDAID Information

z/VSE provides a dialog to analyze and print the contents of an SDAID trace tape. Two types of trace files may be processed:

- The output written to tape by the **SDAID** trace program.
- The output of the attention command **DUMP BUFFER,cuu**.

One or more SDAID files of either type may be stored on the same tape.

You can analyze SDAID information on your screen by performing the following steps:

1. Access the *Problem Handling* panel and select the *Print SDAID Tape* dialog (Fastpath 468).
2. You are requested for the tape address **cuu** and the file number (**1** through **9999**). Enter the file number of the SDAID file you want to analyze.

   z/VSE places the output of the job created and submitted in the VSE/POWER List Queue.
3. To view and analyze the output use the *Problem Handling* panel and select the *Inspect Dump Management Output* dialog (Fastpath 44).

   The *List Queue* panel is displayed on your screen.
4. Enter

   ```
   1  (Display)
   ```

   in the option column for the related jobname and press **ENTER**.

   The requested SDAID information is displayed on your screen.

Note: To interpret SDAID trace information, a good system knowledge is required.

Using an SDAID Getvis/Freevis Trace

A Getvis / Freevis trace provides information about requests made to obtain or release virtual storage. These requests can be made using:

- SVC 3D
- SVC 3E
- An internal Getvis call via BRANCH and LINK

The simple trace of the SVC's 3D and 3E only show the existence of SVCs at the point of invocation. However, the Getvis / Freevis trace records the results of a virtual-storage request *after* it has been evaluated by the z/VSE Getvis / Freevis routines.

You can limit the tracing of your Getvis / Freevis requests to:

- A specific partition.
- The supervisor.
- A specific subpool name.
- A Getvis location (24-bit or 31-bit area).

For details of:

- Sample jobs (for example, SKSDGTVJ and SKSDGTVA) which you can use to create SDAID traces, see “Sample Jobs Showing the Use of SDAID Traces” on page 144.
- How to use a Getvis / Freevis trace, refer to the manual *z/VSE Diagnosis Tools*.

Using an SDAID LOCK / UNLOCK Trace

An SDAID LOCK/UNLOCK trace provides information about requests made to lock or unlock a resource.

You can limit the tracing of your LOCK / UNLOCK requests to:

- A specific partition.
- The supervisor.
- A specific resource name.
- A lock type.
- The scope of the lock request.
- A volume ID.
- A dedicated return code.

For details of:

- Sample jobs which you can use to create SDAID traces, see “Sample Jobs Showing the Use of SDAID Traces” on page 144.
- How to use an LOCK / UNLOCK trace, refer to the manual *z/VSE Diagnosis Tools*.

Using an SDAID XPCC Trace

An SDAID XPCC trace provides information about connections between different applications (cross-partition communication). The information is gathered after the requested function has been processed and completed by the VSE cross-partition communication routine.

You can limit the tracing of your XPCC requests by using one or more of the XPCC trace definitions.

For details of:

- Sample jobs which you can use to create SDAID traces, see "Sample Jobs Showing the Use of SDAID Traces."
- How to use an XPCC trace, refer to the manual *z/VSE Diagnosis Tools*.

Sample Jobs Showing the Use of SDAID Traces

The VSE/ICCF library 59 contains these sample jobs which you can use to create your own SDAID traces.

Skeleton
Description

SKSDBRA
Defines the SDAID branch trace area.

SKSDBRJ
Defines the SDAID branch trace jobname.

SKSDGTVA
Defines the SDAID GETVIS/FREEVIS trace areas.

SKSDGTVJ
Defines the SDAID GETVIS/FREEVIS trace jobnames.

SKSDINSA
Defines the SDAID instruction trace area.

SKSDINSJ
Defines the SDAID instruction trace jobname.

SKSDIOA
Defines the SDAID I/O interrupt trace area.

SKSDIOJ
Defines the SDAID I/O interrupt trace jobname.

SKSDPGCA
Defines the SDAID program check trace area.

SKSDPGCJ
Defines the SDAID program check trace jobname.

SKSDPGMA
Defines the SDAID program load trace (FETCH/LOAD) area.

SKSDPGMJ
Defines the SDAID program load trace (FETCH/LOAD) jobname.

SKSDSTA
Defines the SDAID storage alteration trace area.

SKSDSTJ
Defines the SDAID storage alteration trace jobname.

SKSDSVCA
Defines the SDAID SVC trace area.

SKSDSVCJ
Defines the SDAID SVC trace jobname.

Using the VSE/POWER RJE,BSC and PNET Trace Facility

VSE/POWER provides a combined input/output (I/O) and buffer content trace. Together with the system console log and dump information, the output from the trace area helps you identify PNET and RJE,BSC problems. For example, it allows you to reconstruct I/O sequences.

For a detailed description of the commands mentioned in this section, refer to the manual *VSE/POWER Administration and Operation*.

The operator can activate the trace by specifying the TRACE operand in the PSTART command. Tracing takes place as long as the RJE,BSC line or the connection to the other node is active.

A trace record is written if one of the following VSE/POWER events occurs:

PNET,BSC/CTC I/O completed.

PNET,SNA SEND request completed.

PNET,SNA RECEIVE request completed.

RJE,BSC I/O completed.

Each trace record is 256 bytes long. The following types of trace records are created:

- PNET,BSC/CTC trace record.

  Table 3 on page 146 describes the contents of this record.
- PNET,SNA SEND/RECEIVE trace record.

  Table 4 on page 147 describes the contents of this record.
- RJE,BSC trace record.

  Table 5 on page 148 describes the contents of this record.

The trace records created by VSE/POWER are recorded in wraparound mode. The amount of storage allocated for the trace area is specified with the TRACESZ=xx(x) parameter of the VSE/POWER generation macro.

As shipped with z/VSE, VSE/POWER does not include a trace area and does not support the RJE, BSC and PNET function. If you re-generate VSE/POWER and specify RJEBSC=YES or PNET=phasename, VSE/POWER defines a trace area with a default size of 12K. You can change this value with the TRACESZ parameter. The maximum value allowed is 252K.

The trace area is divided into two parts, the primary and the alternate trace area. Both trace areas are of the same size. If one trace area is full, VSE/POWER switches automatically to the other one. To avoid that the contents of a trace area is overwritten, you can save the contents of the active trace area in the dump sublibrary of the VSE/POWER partition. You save a trace area by issuing the PSTART DUMPTR command. A message notifies the operator if the contents of a trace area has been saved.

Analyzing Trace Information

There are three ways to analyze trace information:

1. You can print the contents of the active trace area. Issue first the PDISPLAY TRINFO command to get the following addresses:
   - Trace area start address.
   - Trace area end address.
   - Address of the next free trace area entry.

   Use these addresses together with the attention routine DUMP command to print the trace area content on the system printer.
2. You can display and analyze the contents of a trace area saved in the dump sublibrary with the z/VSE *Dump Management* dialogs. The dialogs are described in Chapter 6, "Using Dumps for Error Diagnosis," on page 71.

Trace Record Formats

PNET BSC/CTC Trace Record Format

Table 3. PNET BSC/CTC Trace Record Format

Bytes (Hex)	Data Length	Format	Data Description
00-0F	16	binary	CCB
10-37	40	binary	Last 5 executed CCWs
38-3F	8	-	Reserved
40-5F	32	char	Sent buffer content (first 16 and ast 16 bytes)
60-7F	32	char	Received buffer content (first 16 and ast 16 bytes)
80-81	2	binary	Sense bytes 1 and 2
82-83	2	-	Reserved
84-87	4	binary	Last executed CCW + 8
88-8B	4	packed	Time of day (0HHMMSSF)
8C-8F	4	-	Reserved
Status of Communication			
90-93	4	binary	Address of input buffer in use
94-97	4	binary	Address of output buffer being sent
98-9B	4	binary	Address of buffer used by read CCW
9C-9F	4	binary	Address of buffer used by write CCW
A0	1	binary	Expected block sequence count (BCB)
A1	1	binary	Transmitted block sequence count (BCB)
A2-A3	2	binary	Received FCS bytes
A4-A5	2	binary	Transmitted FCS bytes
A6	1	binary	Request code for I/O manager
A7	1	binary	Last sent request code
A8	1	binary	Last non-NAK request code
A9	1	binary	Request from line-driver
AA	1	binary	Interface flag I/O manager
AB-AF	5	-	Reserved

Table 3. PNET BSC/CTC Trace Record Format (continued)

Bytes (Hex)	Data Length	Format	Data Description
Error Counts			
B0-B3	4	binary	Total transmission count
B4-B5	2	binary	Total time out count
B6-B7	2	binary	Error count
B8	1	binary	Retry count
B9	1	binary	Time out retry count
BA	1	binary	Error count
BB-BF	5	-	Reserved
Status of Node			
C0-C1	2	binary	Action bytes 1, 2
C2	1	binary	Process byte
C3-C5	3	binary	Status bytes 1, 2, 3
C6	1	binary	Termination code
C7	1	binary	Termination subcode
C8	1	binary	Stop code qualifier
C9-F7	47	-	Reserved
F8-FB	4	char	Task ID of task writing trace record
FC	1	-	Reserved
FD-FF	3	char	Line address (cuu)

PNET, SNA SEND/RECEIVE Trace Record Format

Table 4. PNET, SNA SEND/RECEIVE Trace Record Format

Bytes (Hex)	Data Length	Format	Data Description
00-07	8	char	Remote node name
08-1F	24	binary	Buffer header
20-83	100	binary	VTAM RPL
84-87	4	packed	Time of day (0HHMMSSF)
88-97	16	char	First 16 bytes of data
98-A7	16	char	Last 16 bytes of data
Status of Node			
A8-A9	2	binary	Action bytes 1, 2
AA	1	binary	Process byte
AB-AC	2	binary	Status bytes 1, 2
AD	1	-	Reserved
AE	1	binary	Node termination code
AF	1	binary	Termination subcode
B0	1	binary	Termination code qualifier
B1-B3	3	-	Reserved

Table 4. PNET, SNA SEND/RECEIVE Trace Record Format (continued)

Bytes (Hex)	Data Length	Format	Data Description
B4-B5	2	binary	Number of receivers active
B6-B7	2	binary	Number of transmitters active
B8-BB	4	binary	Address of free input buffer queue
BC-BF	4	binary	Address of to-be-sent queue
C0-C3	4	binary	Tail pointer to-be-sent output queue
C4-C7	4	binary	Address of receive buffer
C8-CB	4	binary	Address of send buffer
CC-CD	2	binary	Buffer size
CE	1	binary	Maximum number of input buffers
CF	1	binary	Maximum number of output buffers
D0-D1	2	binary	Number of acquired input buffers
D2-D3	2	-	Reserved
D4-D7	4	binary	Address of input buffer in use
D8-DB	4	binary	Address of output buffer sent
DC-E7	12	-	Reserved
E8-EB	4	binary	Address of suspended buffer queue
EC-EF	4	binary	Address free buffer ahead queue
F0-F3	4	binary	Address send ahead queue
F4	1	binary	Send gate
F5	1	binary	Receive gate
F6-FA	5	binary	Session status byte 1 to 5
FB	1	-	Reserved
FC-FF	4	binary	Task ID of task writing trace record

RJE, BSC Trace Record Format

Table 5. RJE, BSC Trace Record Format

Bytes (Hex)	Data Length	Format	Data Description
00-0F	16	binary	CCB
10-37	40	binary	Last 5 executed CCWs
38-77	64	binary	Data sent and received
78-7B	4	binary	Address of last executed CCW in trace
7C-7D	2	binary	Sense bytes 1 and 2
7E	1	binary	Last request code
7F	1	binary	Current request code
80	1	binary	Stop code
81-83	3	binary	LCB flag bytes 1, 2, and 3
84-85	2	binary	PUB pointer
86-87	2	-	Reserved

Table 5. RJE, BSC Trace Record Format (continued)

Bytes (Hex)	Data Length	Format	Data Description
88-89	2	binary	Transmission count
8A-8B	2	binary	Session time out count
8C-8D	2	binary	Error count
8E-8F	2	binary	Time out count
90-91	2	binary	Invalid response count
92	1	binary	Time out count (pre-signon)
93	1	binary	Terminal error count
94	1	binary	Retry count
95	1	binary	Retry count
96-97	2	-	Reserved
98	1	binary	Message index (remote message queue)
99	1	binary	Line mode control byte
9A	1	binary	Output switches
9B	1	binary	Activity control byte
9C-A1	6	binary	LCB PLINE entry
A2-B9	24	binary	LCB PRMT entry
BA-BB	2	-	Reserved
BC-C3	8	packed	Time when channel end processed
C4-D3	16	-	Reserved
D4-D7	4	binary	LST DCT flags
D8-DB	4	binary	PUN DCT flags
DC-DF	4	binary	MSG DCT flags
E0-E3	4	binary	RDR DCT flags
E4-EF	12	binary	Reader buffer status
F0-FB	12	binary	Writer buffer status
FC-FE	3	char	Line address (cuu)
FF	1	binary	Remote ID

PNET BSC/CTC Console Trace

This trace can be used to record the sequence of input/output (I/O) events that occur between two nodes linked by a BSC communication line or a channel-to-channel adapter. The trace can be used to check whether a node or both nodes follow the protocol. Figure 41 on page 150 shows the information recorded at I/O completion time at the central operator's console.

The operator can start the trace with a PSTART CNSLTR command and stop it with a PSTOP CNSLTR command.

```
LINE=cuu  CCB=cccc  SENSE=ssss  OUT:  BCB=bb  FCS=ffff  BSC=aaaa  RCB=rrrr
                                 IN:  BCB=bb  FCS=ffff  BSC=aaaa  RCB=rrrr
```

Figure 41. Record Format of PNET BSC/CTC Console Trace

The meanings are as follows:

LINE=cuu
is the 2-byte device address in binary format.

CCB=cccc
are the CSW status bits.

SENSE symbol=text=ssss
is the 2-byte sense information as returned from the device.

OUT or IN
is the direction of the buffer.

BCB=bb
is the block control byte either sent or received.

FCS=ffff
are the 2 bytes of the function control sequence field.

BSC=aaaa
are the first two BSC control characters used in the buffer being sent or received. Valid characters are:

- X'1070' - ACK0
- X'3D00' - NAK
- X'1002' - start of text (data)
- X'012D' - SOH ENQ

RCB=rrrr
is the record control byte (RCB) and subrecord control byte (SRCB) of the first record in the buffer sent or received.

Using the VSE/POWER Task Dispatching Trace

To aid in problem determination, a task dispatching trace parameter can be used with the PSTART and PSTOP commands. Once the trace is enabled, each VSE/POWER task dispatching event is recorded in wraparound mode in a trace area in processor storage. With the PVARY command, tracing may be enabled or disabled.

Because of the number of VSE/POWER tasks concurrently active it is extremely difficult (without a trace) to identify a task causing errors.

Using the Trace

- To activate the trace, enter:
  ```
  PSTART TASKTR,,6
  ```
  This will start the trace and fill up an area of 6K with trace records in wraparound mode.
- To stop tracing temporarily, enter:
  ```
  PVARY TASKTR,DISABLE
  ```
- To resume tracing again, enter:
  ```
  PVARY TASKTR,ENABLE
  ```

- To stop tracing definitely, enter:

```
PSTOP TASKTR
```

To print and analyze the trace information, you need a dump of the VSE/POWER partition and have this dump processed by the Info/Analysis program. You can use the dialogs described in Chapter 6, “Using Dumps for Error Diagnosis,” on page 71 for that purpose. The trace area, if present, is included in the Info/Analysis output (section 6 of the symptom record) and pointed to by a locator entry.

Format of the Trace Output

The trace area has a minimum and default size of 2K (the maximum is 64K) and resides in real storage. Each trace entry is 96 (X'60') bytes long and the default trace area holds 21 trace entries.

The trace area begins at an address stored in field CATTRA located at address X'4E4' in the VSE/POWER partition. The first 32 bytes of the trace area contain a header with the following information:

```
0 - F      "DISPATCH TRACE"
10 - 13    Address of first trace entry
14 - 17    Address of last trace entry
18 - 1B    Address of last used entry during tracing
1C - 1F    Total number of bytes of the trace area including the header
```

This is followed by trace entries showing the history of the last n dispatched tasks and their status at dispatch time. This helps to identify errors which may otherwise no longer be visible at the time an error is noticed.

The following table shows the layout of a task dispatching trace entry.

Table 6. Layout of a Task Dispatching Trace Entry

Displacement	Type	Contents
0-7	Character	Task ID and cuu
8-B	Binary	TCB address
C	Character	Function trace byte
D	Character	Termination byte
E	Character	1st byte of task selection field
F		Reserved
10-16	Binary	Flag bytes 2-8
17-18		Reserved
19-1F	Binary	Log. interface and function request bytes
20-57	Binary	Registers 12-9
58-5F	Binary	STCK value

Using CICS Traces

Execution Diagnostic Facility (EDF)

This trace is of interest to you if you are an application programmer. Using this CEDF transaction, you can trace the execution of a CICS Transaction Server online application program (transaction). This might be necessary during program development or after the program has been installed.

This trace is of interest to you if you are an application programmer. Using thie CEDF transaction, you can trace the execution of a CICS Transaction Server online application program (transaction).

For details on EDF refer to the *CICS Application Programming Reference*.

The CICS Trace Facility

The CICS Transaction Server maintains a wraparound trace table in main storage. The entries reflect CICS macros or commands issued by an application program, or by the CICS Transaction Server itself.

The trace entries can also be stored on auxiliary storage on disk. Since no wraparound occurs, all entries are preserved and a complete history is provided. For analysis, you can print selected entries or the complete contents of the auxiliary storage.

In ICCF Library 59, z/VSE provides the skeleton DFHAUXPR for the CICS Transaction Server, including a job stream for printing.

The CICS Trace Facility is described in the *CICS Application Programming Reference*.

Using VTAM Traces

VTAM provides traces like the following:

- VTAM Network Traces.

  Several traces are available to record the internal flow of network events. Each trace becomes active at a different point in the network. This allows reconstruction of the flow within the network. The trace records generated are stored in the VTAM trace file. The contents of the file can be printed for analysis.
- VTAM Buffer Use (SMS) Trace (skeleton SKVTAMBU). The VTAM buffer use trace records information on the use and availability of VTAM buffer pools.
- VTAM I/O Trace (skeleton SKVTAMIO).

Refer to the VTAM documentation for details.

Using the TCP/IP for VSE/ESA BSD Trace

Overview

TCP/IP for VSE/ESA provides two C language application-programming interfaces:

- The BSD C-language interface.
- The TCP/IP C-language interface of LE/VSE.

The above interfaces are related, because the TCP/IP C-language interface of LE/VSE is based upon the BSD C-language interface.

You can use the BSD trace capability of TCP/IP for VSE/ESA to determine the BSD C-language functions that are called by an application using the TCP/IP C-language interface of LE/VSE.

Activating the BSD Trace Capability

To get the BSD trace active the phase XSOCKDBG.PHASE which is contained in the TCP/IP installation library PRD2.TCPIPC must be renamed or copied to the member $SOCKDBG.PHASE. After this change, the TCP/IP partition and any application which needs to be debugged must be recycled. For tracing it is important that the application which should be traced has the $SOCKDBG.PHASE in access. If multiple TCP/IP applications exist but only one should be traced, the $SOCKDBG.PHASE can be copied to a private library which is only accessible to this specific application.

Output From the Trace

Immediately after the TCP/IP-C-LE application is started again, which means as soon as the first socket() call is issued by the program, the following message is issued

```
BSD001I $SOCKLST 01.04.00 05/30/00 14.16  ...addresses...
```

and additionally the following (or similar) information is issued:

```
BSD001I IPNRSOCK 01.04.00 06/07/00 11.42  ...addresses...
BSD002I IPNRSOCK R15=00000000 RETCD=GOOD ERRNO=NONE
```

In this example, the IPNRSOCK indicates that the BSD-C call socket() was issued at that point in time.

The BSD001I message indicates that the module was entered. Here the date and time when this module was compiled can also be found.

The BSD002I message indicates that the module was left. R15=0 indicates no error during the processing which is also shown through RETCD=GOOD and ERRNO=NONE.

These are the module names representing the most important BSD C-language function calls:

```
IPNRACCP - accept()
IPNRAIOR - aio_read()
IPNRAIOW - aio_write()
IPNRBIND - bind()
IPNRCLOS - close()
IPNRCONN - connect()
IPNRGETC - getclientid()
IPNRGETS - getsockopt()
IPNRGHBA - gethostbyaddr()
IPNRGIVE - givesocket()
IPNRLIST - listen()
```

```
IPNRRECV - receive()
IPNRSELE - select()
IPNRSEND - send()
IPNRSETS - setsockopt()
IPNRSOCK - socket()
IPNRTAKE - takesocket()
```

Special Considerations for Some Function Calls

The following has to be considered when using some of the function calls.

When an accept() or takesocket() is successful, the R15 does not contain and error indication but contains the returned socket number.

If a call is returning unsucessful with R15=FFFFFFFF (representing -1) the ERRNO contains the TCP/IP internal error indication. This error indication is translated by the TCP/IP-C-LE interface to an LE-compliant ERRNO value.

Recommendation

In case a problem situation requires a more insight view as to which TCP/IP function calls are issued by a TCP/IP-C-LE application, the BSD trace should be analyzed together with other important trace data. Also, if an MQSeries® for VSE/ESA V2.1 problem has to be investigated, the BSD trace should be analyzed together with a CICS AUX trace and relevant MQSeries trace information.

Documentation

Further documentation of the BSD trace capability of TCP/IP for VSE/ESA can be found in Info APAR II11836. The BSD-C function calls are described in the manual *TCP/IP for VSE 1.4 Programmer's Reference*, and the TCP/IP-C-LE interface in the manual *z/VSE TCP/IP Support*.

Using the IPv6/VSE BSD Trace

Overview

Similar as the above trace IPv6/VSE also supports a BSD trace, for details refer to the BSI documentation.

Using the Linux Fast Path Trace

Overview

For details refer to the manual *z/VSE TCP/IP Support*.

Using the VSE Connector Server Trace

To get a full trace of the VSE Connector Server, issue the following command at the console:

```
MSG nn,DATA=SETTRACE DD:SYSLST 0xFFFFFFFF
```

In the above example, the trace is sent to SYSLST. However, you can specify SYSLOG to have the trace messages displayed on the console.

To end the trace of the, issue the following command at the console:

```
MSG nn,DATA=SETTRACE DD:SYSLST 0x00000000
```

Using the Virtual Tape Data Handler Trace

To start a trace of the Virtual Tape Data Handler, you must:

1. Change the TAPESRVR job (skeleton SKVTASTJ in ICCF library 59) as shown below:

   ```
   // EXEC $VTMAIN,SIZE=$VTMAIN,PARM='TRACE'
   ```
2. Issue a DEBUG ON at your system console.

The trace output is always sent to SYSLST.

Note: When the trace is activated, the TAPESRVR will complete with a return code of **0008**. As a result, VSE/POWER will not remove the SYSLST output.

Using the Virtual Tape Server Trace

To activate a trace of the Virtual Tape Server that is running on a Java™ platform, you must:

1. Set `messages=on` in the file **VirtualTapeServer.properties** . The trace output will then be sent to **stdout**. You should then check (in particular) the entries in the "Exceptions" part of this trace.
2. Change the Java call in the start script **run.bat** or **run.cmd** as follows:

   ```
   java -Dcom.ibm.vse.vtape.trace=trace.txt com.ibm.vse.vtape.VirtualTapeServer
   ```

   This will activate the internal trace. The trace output is sent to file **trace.txt**, which is contained in the current directory.
3. For TCP/IP from CSI: Set the TCP/IP BSD trace to ON by renaming the file XSOCKDBG.PHASE (contained in library PRD2.TCPIPC) to $SOCJDBG.PHASE. For each socket call, the trace output will then be sent to SYSLOG or SYSLST.
4. For IPv6/VSE refer to the manual how to activate the trace.
5. For using the Linux Fast Path refer to the *z/VSE TCP/IP Support* manual.

Using the Workstation File Transfer Trace

This trace is available with the Workstation File Transfer support when exchanging data between IBM Personal Computers and a z/VSE host system.

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 PC 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 PC 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, that is, whether EXTDS has been defined for a PC in DFT mode.

However, a user will not be able to solve problems like incorrect data transmission or protocol errors. Such errors require support from IBM for trace output analysis.

Starting the Trace

You start the workstation file transfer trace at the PC by issuing the SEND or RECEIVE command with the TRACE or TRACECUT option.

Format

```
SEND pc-filename host-filename (TRACE|TRACECUT .. other options ..
RECEIVE pc-filename host-filename (TRACE|TRACECUT .. other options ..
```

For a complete description of the SEND and RECEIVE commands, refer to the manual *VSE/ESA Programming and Workstation Guide*.

Note: 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.
The TRACE and TRACECUT options are not supported by the workstation file transfer PC dialog. You must use the command format for SEND and RECEIVE.

Setup Checking

To perform file transfer with a PC running in DFT mode, the EXTDS (Extended Data Stream) feature must be set in the CICS System Definition file for this device. You can define a PC with EXTDS with the *Configure Hardware* dialog by selecting

```
19  (PS/2 PC DFT)
```

in the device selection panel.

During normal file transfer (without the trace being active), checking for CUT or DFT mode is controlled by the presence of the EXTDS attribute:

- If EXTDS is specified, a QUERY REPLY command is sent to the PC 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** option, a QUERY REPLY command is **always** sent, regardless of whether the EXTDS attribute has been specified or not.

Note: If the PC 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 163.

Output Files Created by the Trace

Each trace record is written to two files:

- 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. If the user ID starts with a number, the trace will not be generated because the CICS Report Controller does not allow such user IDs. In order to trace the file transfer, please use a user ID starting with an alphabetic character.
- 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 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 for this device.
  - Whether the PC 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 159 and “Trace Output Example CUT Mode” on page 163.

Using the File Transfer Trace

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 PC, but no completion message was issued in the PC session or in the host session. Your PC 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
   ```
   where nnnn is your user ID.

   In the PC 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 PC 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 PC. If the data is still intact at this point, the error occurred at the host. Otherwise, the data was destroyed during transmission from the PC to the host.
      - If you issued a RECEIVE command, the trace output shows the data before it is sent to the PC. If the data is still intact at this point, the error occurred during transmission to the PC or was caused by the PC control program. Otherwise, the data was destroyed at the host.

   b. If your PC 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 PC 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 are displayed. An incorrect setup may be the missing specification of the EXTDS attribute if your PC is a DFT device. In some cases, depending on the type of error and the PC 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 PC (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 PC 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 PC 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 3270 PC in DFT mode with control program version 3.0. Traces obtained with other PC equipment may show slightly different output in the QUERY REPLIES returned by the PC.

Discussion of Output

The trace output shows each transmission from the PC to the host as INBOUND and from the host to the PC 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 z/VSE 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 the IBM 5550 PC, where ASCII-EBCDIC translation is done at the PC, 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 PC and before further processing (conversion) is done at the host. For OUTBOUND transmission, the data is displayed before it is sent to the PC 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

"1. Example of Trace For a PC in DFT Mode (SEND Command)" is an example of a trace output for a SEND command issued at a PC in DFT mode.

"2. Example of Trace For a PC in DFT Mode (RECEIVE Command)" is an example of a trace output for a RECEIVE command issued at a PC 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.

1. Example of Trace For a PC in DFT Mode (SEND Command)

SEND Command Used: SEND PCFILE HOSTFILE (ASCII CRLF TRACE

```
FLE TRANSFER TRACE:  USERID=SYSA TERMID=D1C2 DATE=89252 TIME=090120...01

INBOUND: INPUT COMMAND IND$FILE ................ MODULE=INWPROOT.......09
1140C1C9D5C45BC6C9D3C540D7E4E340C8D6E2E3C6C9D3C5 . AIND$FILE PUT HOSTFILE
404DC1E2C3C9C940C3D9D3C640E3D9C1C3C5              (ASCII CRLF TRACE

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
000F81808081858687889597999DA6                   ..a..aefghnpr.w
001781810100005000180100010003006400C4090E0780   ..aa...&;.........D....
0010818500000910000000000003000000               ..ae...........
00168186000800F4F1F1F2F2F3F3F4F4F5F5F6F6F7F7     ..af...411223344556677
000D81870400F0F1F1F2F2F4F4                       ..ag..0112244
00078188000102                                   ..ah...
000C819500000E000E000101                         ..an........
000681990000                                     ..ar..
001181A600000B01000050001800500018               ..aw......&;..&;.

INFO: TERMINAL IS IN DFT MODE .................. MODULE=INWPQUER.......17

OUTBOUND: OPEN REQUEST FOR DATA TRANSFER ....... MODULE=INWPOPN1.......30
0029D0001201060101040030A0A000100000000000100500552 .)...................P.R
03F0080627040DEF030946543A44415441               ....'.....FT:DATA

INBOUND: OPEN REPLY (POSITIVE) ................. MODULE=INWPOPN1.......31
0005D00009                                       .....

OUTBOUND: SET CURSOR / GETDATA REQUEST ......... MODULE=INWPPUT1.......41
000FD0451101050006000905010300000 9D0461101040080 ...E..............F.....

INBOUND: GETDATA ACKNOWLEDGEMENT + DATA ........ MODULE=INWPPUT1.......42
CONTROL INFORMATION
00A8D04605630600000001C08061009D                 ...F.c.......a..
```

SEND Command Used: SEND PCFILE HOSTFILE (ASCII CRLF TRACE

```
USER DATA
5448495320495320524543 4F524420303030310D0A544849 THIS IS RECORD 0001..THI
532049532052454 34F524420303030320D0A544849532049 S IS RECORD 0002..THIS I
53205245434F52442030303030330D0A5448495320495320 52 S RECORD 0003..THIS IS R
```

```
45434F524420303030340D0A5448495320495320524543 4F ECORD 0004..THIS IS RECO
```

```
45434F524420303030340D0A54484953204953205245434F ECORD 0004..THIS IS RECO
524420303030350D0A54484953204953205245434F524420 RD 0005..THIS IS RECORD
303030360D0A5448495320495320544845204C4153542052 0006..THIS IS THE LAST R
45434F52440D0A1A                                 ECORD...

OUTBOUND: SET CURSOR / GETDATA REQUEST ......... MODULE=INWPPUT1.......41
000FD04511010500060009050103000009D0461101040080 ...E..............F.....

INBOUND: END OF FILE ........................... MODULE=INWPPUT1.......43
0009D0460869042200                               ...F.i.".

OUTBOUND: CLOSE FILE REQUEST ................... MODULE=INWPCLS1.......34
0005D04112                                       ...A.

INBOUND: CLOSE FILE REPLY (POSITIVE) ........... MODULE=INWPCLS1.......35
0005D04109                                       ...A.

OUTBOUND: OPEN REQUEST FOR MESSAGE ............. MODULE=INWPOPN1.......33
0023D0001201060101040030A0A000000001101010050055 .#...................P.R
03F0030946543A4D534720                           ....FT:MSG

INBOUND: OPEN REPLY (POSITIVE) ................. MODULE=INWPOPN1.......31
0005D00009                                       .....

OUTBOUND: INSERT REQUEST FOR MESSAGE ........... MODULE=INWPGET1.......40
000AD04711010500800000 5AD04704C080610055494E5730 ...G.......Z.G...a.UINW0
303031492020466696C65207472616E7366657220636F6D70 001I  File transfer comp
6C6574652E20240000000000000000000000000000000000 lete. $.................
000000000000000000202020202020202020202020202020 .........
20202020

INBOUND: INSERT REPLY (POSITIVE) ............... MODULE=INWPGET1.......38
000BD04705630600000001                           ...G.c.....

OUTBOUND: SEND MESSAGE TO HOST SCREEN .......... MODULE=INWPMSG .......45
C9D5E6F0F0F5F0C94040E399818385408396949793 85A385 INW0050I  Trace complete
406040838885839240E3E24098A485A48540C3C6E3D9E2E8  - check TS queue CFTRSY
E2C1409699409389A2A34098A485A48540C9E6E2E3D9C1C3 SA or list queue IWSTRAC
C54040404040404040404040404040404040404040404040 E
404040404040404040

E N D   O F   T R A C E                          MODULE=INWPMSG ..... .05
```

2. Example of Trace For a PC in DFT Mode (RECEIVE Command)

RECEIVE Command Used: RECEIVE PCFILE HOSTFILE (ASCII CRLF TRACE

```
FILE TRANSFER TRACE:  USERID=SYSA TERMID=D1C2 DATE=89252 TIME=090347...01

INBOUND: INPUT COMMAND IND$FILE ................ MODULE=INWPROOT.......09
1140C1C9D5C45BC6C9D3C540C7C5E340C8D6E2E3C6C9D3C5 . AIND$FILE GET HOSTFILE
404DC1E2C3C9C940C3D9D3C640E3D9C1C3C5              (ASCII CRLF TRACE

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
000F81808081858687889597999DA6                   ..a..aefghnpr.w
0017818101000050001801000100030006400C4090E0780   ..aa...&;.........D....
00108185000009100000000003000000                 ..ae...........
00168186000800F4F1F1F2F2F3F3F4F4F5F5F6F6F7F7     ..af...411223344556677
000D81870400F0F1F1F2F2F4F4                       ..ag..0112244
00078188000102                                   ..ah...
```

```
000C819500000E000E000101                          ..an........
000681990000                                      ..ar..
001181A600000B01000050001800500018                ..aw......&;..&;.

INFO: TERMINAL IS IN DFT MODE .................. MODULE=INWPQUER.......17

OUTBOUND: OPEN REQUEST FOR DATA TRANSFER ....... MODULE=INWPOPN1.......30
0023D000120106010104030A0A0000000011010100500552 .#...................P.R
03F0030946543A44415441                           ....FT:DATA

INBOUND: OPEN REPLY (POSITIVE) ................. MODULE=INWPOPN1.......31
0005D00009                                       .....

OUTBOUND: INSERT REQUEST FOR DATA .............. MODULE=INWPGET1.......37
CONTROL INFORMATION
000AD04711010500800000A2D04704C08061009D         ...G.........G...a..
```

RECEIVE Command Used: RECEIVE PCFILE HOSTFILE (ASCII CRLF TRACE

```
USER DATA
544849532049532052454434F524420303030310D0A544849 THIS IS RECORD 0001..THI
532049532052454434F524420303030320D0A544849532049 S IS RECORD 0002..THIS I
532052454434F524420303030330D0A544849532049532052 S RECORD 0003..THIS IS R
45434F524420303030340D0A54484953204953205245434F ECORD 0004..THIS IS RECO
524420303030350D0A54484953204953205245434F524420 RD 0005..THIS IS RECORD
303030360D0A544849532049532054484520 4C4153542052 0006..THIS IS THE LAST R
45434F52440D0A1A                                 ECORD...

INBOUND: INSERT REPLY (POSITIVE) ............... MODULE=INWPGET1.......38
000BD04705630600000001                           ...G.c.....

OUTBOUND: CLOSE FILE REQUEST ................... MODULE=INWPCLS1.......34
0005D04112                                       ...A.

INBOUND: CLOSE FILE REPLY (POSITIVE) ........... MODULE=INWPCLS1.......35
0005D04109                                       ...A.

OUTBOUND: OPEN REQUEST FOR MESSAGE ............. MODULE=INWPOPN1.......33
0023D000120106010104030A0A0000000011010100500552 .#...................P.R
03F0030946543A4D534720                           ....FT:MSG

INBOUND: OPEN REPLY (POSITIVE) ................. MODULE=INWPOPN1.......31
0005D00009                                       .....

OUTBOUND: INSERT REQUEST FOR MESSAGE ........... MODULE=INWPGET1.......40
000AD047110105008000005AD04704C080610055494E5730 ...G.......Z.G...a.UINW0
30303149202046696C65207472616E7366657220636F6D70 001I  File transfer comp
6C6574652E2024000000000000000000000000000000000000 lete. $.................
000000000000000000202020202020202020202020202020 .........
20202020

INBOUND: INSERT REPLY (POSITIVE) ............... MODULE=INWPGET1.......38
000BD04705630600000001                           ...G.c.....

OUTBOUND: SEND MESSAGE TO HOST SCREEN .......... MODULE=INWPMSG .......45
C9D5E6F0F0F5F0C94040E3998183854083969497938 5A385 INW0050I  Trace complete
40604083888583924 0E3E24098A485A48540C3C6E3D9E2E8  - check TS queue CFTRSY
E2C1409699409389A2A34098A485A48540C9E6E2E3D9C1C3 SA or list queue IWSTRAC
C540404040404040404040404040404040404040404040404040 E
404040404040404040

E N D   O F   T R A C E                          MODULE=INWPMSG ..... .05
```

Inbound Transmission Errors (DFT Mode)

Whenever an unexpected reply is received by the host from the PC, 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 PC 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 DFHTCT entry 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 CUT mode is mostly no longer used. However, to start a trace under CUT mode, you should use the trace option as described for the DFT mode. Alternatively, you can use the option TRACECUT to produce a trace.

For details of the output that is produced using the CUT mode, refer to earlier editions of this manual. These editions provide trace examples that were produced on an IBM PC/AT with the IBM 3278/79 Emulation Adapter and the IBM PC 3270 Emulation Program, Entry Level Version 1.0.

Setup Errors (CUT Mode)

This section describes the most common setup errors that might occur when using 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 PC that has the EXTDS feature defined, 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 ................. .. MODULE=INWPQUER.......04
ERROR: EXTDS FEATURE INVALID FOR TRACECUT OPTION MODULE=INWPQUER.......08
```

If the TRACE option is used for a PC 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 ................. .. 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 ............. .. 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.

Tracing the SSL Connection

Using the OPEN SSL Trace

The following JCL variable controls the OpenSSL trace:

```
// SETPARM SSL$DBG = [ 'YES' | 'NO' ]
```

When variable SSL$DBG is set to YES, SSL trace output is written to SYSLST.

For more information refer to the *z/VSE TCP/IP Support* manual.

Chapter 11. Diagnosis Procedures for More Experienced Users

This chapter describes diagnosis procedures that experienced users might use to solve system problems. It contains these main sections:

- "Incorrect Output Symptom"
- "Abend Symptom" on page 167
- "Wait Symptom" on page 170
- "Loop Symptom" on page 174

Also, if you contact IBM for help you may be asked to:

- Locate various control blocks in a formatted dump – for details on a formatted dump refer to Chapter 6, "Using Dumps for Error Diagnosis," on page 71.
- Produce a trace with the SDAID program – for details on SDAID traces refer to "SDAID Overview" of the manual *z/VSE Diagnosis Tools*.

Incorrect Output Symptom

Errors in program logic or in the system setup for program execution may cause errors in the output. For example, the use of incorrect data for input, mistakes in device assignments, or incorrect job control statements and commands often result in unexpected output.

Data You Should Collect

- System console log (printout of hardcopy file).
- All output associated with the program in error.
- System/partition dump. If this is not available, rerun the job and take a dump when the incorrect output occurs again.
- Job stream used.
- A list of all I/O files and volumes used by the particular program.
- EREP output, if possible.

Diagnosing Steps

1. Identify the SYSLOG output for the job in error and, if possible, the input stream for the job submitted.
2. Was there any incorrect operator's response to a message during execution of the failing program? If so, rerun the program and ensure that the operator's responses are correct (see *z/VSE Messages and Codes*).
3. Find out whether input data read by the program from magnetic tape or disk is still available. If so, make sure that the data will be saved for later problem analysis.
4. Find out about the status of the hardware at the time the program was running (for example, whether an I/O unit used showed an error). Analyze possible effects of any unusual hardware status on the program.

   If you suspect that a hardware error may have caused the incorrect output, check the hardware and rerun the program.
5. Determine the execution environment.

If the program was executed under control of VSE/POWER, find out whether other programs running under control of VSE/POWER were also generating incorrect output. If so, consult the documentation of VSE/POWER.

6. Classify the incorrect output by the type of the output error:
   - **Duplicate lines** written to the I/O device. This suggests unintended loops in the program's I/O routines.
   - **Missing data**. This suggests that specific routines of the failing program either did not receive control, or they lost control before they were completely executed.
   - **Too much or unexpected output data**. I/O areas may have been overwritten or, if the output consists of variable-length records, these areas may not have been cleared properly.

   If the incorrect output falls into one of the above categories, rerun the program with the SDAID trace on. Trace the (1) execution of instructions involved in building the output records and (2) alteration of storage areas in which the faulty output is built or written from. The trace information provided by SDAID should give you an indication of the cause of the incorrect output. If the incorrect output does not fall into one of those categories, continue with the next step.
7. Find out about the system condition at the time of program execution.

   Was there any abnormal interruption in system operation? If so, analyze the possible effects of that interruption on program execution and rerun the program.
8. Check the job stream used for the program.

   This check is also indicated if a cataloged procedure is used. Use LIBLIST to display library chains involved. When you have isolated the error, make the necessary corrections in the program and rerun it.
9. Determine whether the input is at fault.
   a. Verify the label information.

      Use program LSERV to list label information, program LVTOC to list VTOC information, and program IDCAMS to list catalog information. Compare the output of these programs with the label information provided in the job stream.
   b. Isolate suspected areas in the input files by comparing the input for the program with the associated output.

      For example, a missing record in the output could indicate an erroneous area in the appropriate input file. If the program's input is the output of another program, you may have to analyze the program whose output is used as input for the failing program.

      Obtain a dump of the suspected area(s) of the appropriate input file. It might be worthwhile to check file definitions (for example, DTF macros when logical IOCS is used or the CCB or CCW if physical IOCS is used) against the characteristics of data in the input file.
10. Examine the output of the partition dump.

    Inspect the contents of the I/O area(s) and, if the program's I/O routines use physical IOCS, also of the involved CCBs or IORBs. In the latter case, check also the channel program (CCW chain). If this does not help you in isolating the cause of the error, continue with the next step.
11. Rerun the program with the SDAID trace program.

    Use the program listings and the information obtained so far to determine the approximate area of the failure.

Select the tracing instructions related to the incorrect output. Consider also a storage-alter trace. Requesting a program-load trace in addition may be helpful since it provides a record of the phases called by the program.

Request SDAID to provide a dump of the I/O areas used by the program. If the program uses physical IOCS, let SDAID print the contents of the applicable CCBs or CCWs on certain events.

Evaluate the SDAID output:

- Compare the I/O areas as defined in the program listings with the contents of these areas in the SDAID dump.
- Inspect the contents of the CCBs or CCWs, if applicable.
- Check the sequence, names, and load- or entry-point addresses of the phases used by the failing program and compare them with the addresses given in the linkage editor map.
- Check the storage-alter trace output for possible field overlay in output (or input) records.
- Use the program listings and the SDAID dump output for inspecting the values in registers and storage locations used for intermediate results.

If any of the above steps help you finding the cause of the error, correct your program and rerun it.

12. Check for unusual conditions generated by the program and its input.

    Ensure that the program logic and control counters, if any, can handle unusual input. For example, changes in the date (year) or extraordinary changes in amounts and quantities.

13. Gather additional error information.

    Rerun the failing program with input that is known to be correct. Recreate the conditions identical to those which existed when the failure occurred. For this run, dump the contents of the I/O areas before and after an I/O operation using the PDUMP macro.

    If the program uses VSE/VSAM files, use the macros TESTCB or SHOWCB before and after each OPEN, GET, and PUT.

14. If necessary, contact IBM for support and have the data collected available for problem analysis.

Abend Symptom

Data You Should Collect

- System console log (printout of the hardcopy file).
- A dump taken when the job was canceled.
- SYSLST (or other printer) output of the failing program.
- Job stream of the failing program.
- Link map produced by the ACTION MAP linkage editor statement.
- Source listing (or source deck) of the program that caused the abend.

Diagnosing Steps

Determine the Type of Program Cancelation

Check the cancel message issued. The message indicates one of the following:

- Program check: Continue with the next step.
- Illegal SVC: Go to “Procedure: Program Canceled Because of Illegal SVC” on page 169.
- Other reason: Go to “Procedure: Program Canceled for Undetermined Reason” on page 170.

Determine Where the Program Cancelation Occurs

1. Determine whether the *failing instruction* was executed from within the *SVA* by examining the output of the system or partition dump if the program check occurred in the SVA. The SVA phase containing the failing instruction is included in the dump output under the heading:

   SVA PHASE IN ERROR = phasename ADR=xxxxxx

   Locate the failing instruction in the system dump output. Try to determine the reason for the cancelation by investigating the interrupt code. If that instruction is part of an IBM-supplied phase, call IBM for support. If the instruction is part of another phase, consider the recommendations in steps 4c on page 169 and 4d on page 169 below for your code that is being executed from the SVA.
2. Determine whether the program check occurred in the *LTA*.

   This is indicated by message 0S08I, which displays the name of the phase that was last loaded into the LTA. This is most likely an IBM-supplied phase. If so, contact IBM for support.

   Analyze the dump:

   a. Examine the contents of the general registers for unreasonable or unexpected values. The symptom REGS/xxyyy shows you the registers which may be related to the failing instruction. Examine the code that loads those registers.

      For the symptom REGS/xxyyy, refer to “Example 1: Displaying a Symptom Record” on page 81.
   b. Locate the areas that were referred to by the code which loads those registers. Check the contents of those areas for validity, duplication, or lack of information. Examine (in the source listings) the code that manipulates those areas.

   If the above steps do not give an indication of the cause of the error, consider a rerun of the canceled program with the SDAID trace. Trace the execution of the instructions related to the problem and, as applicable, the program check and alterations of storage locations. This should give you sufficient error information for reporting the problem.
3. Check the symptoms RIDS and PIDS to determine whether the program check occurred within an *IBM Program* running in the partition where the canceled program was being executed. If so, contact IBM for support.

   For the symptoms RIDS and PIDS, refer to “Example 1: Displaying a Symptom Record” on page 81.
4. If the program check was caused by *your own code,* isolate the cause of the failure by using the techniques recommended below.

   a. Examine any *error message* that may have been issued during execution of the failing job and consult *z/VSE Messages and Codes* for an error indication. Examine the job stream, the program's source listings, or both, and evaluate your findings.

b. If your failing program used an abnormal termination exit routine (defined by STXIT AB), examine the *cancel code* provided in register 0. For a list of these codes, see the *z/VSE Messages and Codes* manual.

c. If, at this point, the gathered information is insufficient to isolate the cause of the problem, consider reassembling and relinking your program (phase) with:

   - The PRINT GEN assembler instruction (which prints the instructions generated by the assembler as the result of a macro), and
   - The PDUMP macro (to obtain a dump of selected areas such as I/O) and the DUMP or JDUMP macro (to obtain a dump of the partition *and* the supervisor) inserted in the critical routine(s) at a convenient point before the program check occurs.

   Re-analyze the problem using the output obtained from the rerun of the malfunctioning program.

d. Locate the failing instruction in the dump and compare its hex representation with that of the instruction in the source listings. If they differ, find out why they differ. Either by further offline analysis of the source code or by a rerun of the job with the SDAID trace on. Trace the storage alter operation that changes the code of the particular instruction.

e. Locate (in the dump) the partition's save area and inspect the contents of the general purpose registers. If you find suspicious values in any register that is related to the failing instruction, locate, in the program listings, the code that loads those registers and inspect that code.

   Consider a rerun of the job with the SDAID trace on. Trace selected instructions like branch operations; specify appropriate address limits for this tracing activity. Request SDAID to provide a dump of critical areas on the occurrence of certain storage-alter or branch events.

f. Use the program listings and the linkage editor map (produced by ACTION MAP) to locate data areas used by the failing instruction in the dump output. Determine the relocation factor based on the partition's start address at the time when the failing phase was executed.

   Inspect the contents of these areas for validity, duplication, missing information, or sequence errors, whichever applies depending on the generated VSE message(s). Locate the code that is responsible for moving data to those areas and inspect that code.

   Consider a rerun of the job with the SDAID trace on. Trace events as recommended above.

Correct the faulty code and rerun the job.

Procedure: Program Canceled Because of Illegal SVC

An illegal SVC cancelation of a program is indicated by *message 0S04I,* which displays the faulty SVC code.

1. See *z/VSE Messages and Codes* and follow the instructions given under “Programmer Action” for that message. If this does not correct the problem, continue with the next step.
2. Check whether the supervisor includes support for the SVC code displayed in the message.

   If your supervisor does not support the displayed SVC, either reassemble the supervisor to include the required support or change your program to avoid this SVC. Rerun the program.
3. Check whether a correctly coded SVC instruction has been altered during program execution.

To do this, use the linkage editor map and a source statement listing that shows the expansions of the macros used in the program. (This type of a listing is obtained by a program assembly run with the PRINT GEN assembler instruction included in the source code.) Locate the SVC instruction in the source statement listing.

4. Rerun the program with the SDAID trace on.

   If the code in the source instruction differs from the code displayed in the message, that code has probably been overwritten during program execution. Consider a rerun of the failing program with SDAID trace on. Trace the alteration of the storage locations that contain that SVC instruction.

   The trace information provided by SDAID indicates which instruction of the failing program altered the code. If that instruction is part of an IBM-supplied routine, call IBM for support. Keep your problem analysis records (and notes), including the assembly listings of the supervisor used at the time of program failure, for further problem analysis. If the instruction is part of your own code, correct your program and rerun it.

Procedure: Program Canceled for Undetermined Reason

1. Inspect the messages printed on SYSLOG and SYSLST prior to program cancelation.

   Ensure that the responses to VSE messages were correct and follow the recommendations given in the *z/VSE Messages and Codes* manual.

   The program may have been canceled because of an unusual condition. This condition may have been caused by an incorrect response to a message issued by the system during program execution.
2. Rerun the program with the job control statement // OPTION PARTDUMP inserted in the job stream.

If necessary, contact IBM for support and have the data collected available for problem analysis.

Wait Symptom

A CPU is in a wait state if it does not execute any instruction. Whenever bit 14 (W-bit) in the current PSW is on, a CPU is said to be in a wait state. A visible indication is the wait indicator at the system console panel which is turned on.

You can obtain detailed status information using the STATUS command: see "Using the STATUS Command" on page 136.

There are two types of wait states: a **Soft Wait** or a **Hard Wait**.

- In a hard wait state, the I/O and external interrupts are disabled; bits 6 and 7 of the current PSW are both off. This indicates that a hardware or programming error cannot be associated with a single program only. A hard wait requires a re-IPL of the system.
- In a soft wait state, I/O and external interrupts are enabled. Any interrupt causes the system at least temporarily to get out of the wait state. Soft waits are normally caused by non-posted event control blocks (CCBs, IORBs, or ECBs). Usually, no re-IPL of the system is required.

  Soft waits are grouped into:
  - Temporary soft waits.

The system can escape from the wait without any operator intervention. A temporary soft wait indicates *normal processing* and not a failure, except the wait occurs again and again. This may indicate an I/O instruction loop.

- Permanent soft waits.

  The system cannot escape from the wait without operator intervention. A permanent soft wait is most likely an indication of an operating or programming error. To exclude an operating error, the operator must ensure that all outstanding replies to messages have been answered and that required attention routine commands have been entered.

When a wait occurs, collect the information about the system status *during the time* the wait exists.

Data You Should Collect

- For a **Soft Wait**:
  - Saved status information.
  - Stand-alone dump or partition dump if problem can be isolated to a partition (DUMP pp,cuu).
  - System console log (printout of the hardcopy file).
  - EREP output covering a reasonable time before the error occurred.
- For a **Hard Wait**:
  - Saved status information.
  - Stand-alone dump.
  - System console log (printout of the hardcopy file).
  - EREP output.

Diagnosing Steps

Procedure: System or Program Entered Wait State

If you contact IBM for support, ensure that all the data about the failure is available: The attention command DEBUG ON should have been used prior to recreating the problem whenever possible.

1. Determine the type of wait: Soft Wait or Hard Wait. For details, refer to “Recovering from a Wait State” on page 173.

   In case of a soft wait, turn to “Procedure: Soft Wait” on page 172.

   In case of a hard wait, proceed with the next step.

2. Compare the contents of bytes 0 through 3 of processor storage with the hard wait codes as listed in *z/VSE Messages and Codes*.

   If the code in those bytes is not listed, continue with “Procedure: Hard Wait with No Wait Code” on page 172 below.

   If the wait code suggests that the hard wait condition has been caused by an IBM program, call IBM for support. Examples of such wait codes are:

   ```
   X'00 00 0F FF' - a program check in the supervisor.

   X'00 00 0F FB' - a page fault in a supervisor routine.
   ```

3. Attempt to isolate the possible cause of the error by using:
   - The dump (or notes) of low address storage for locating the old PSW applicable to the wait code.
   - The stand-alone dump output.

- The DEBUG trace information pointed to by storage address X'270' may be especially useful in locating the error.
- If the hard wait code seems to be device related, try to get the appropriate device corrected or use a different device if possible.
- If the hard wait code points to an IBM program or to a system error, contact IBM for support.

Procedure: Hard Wait with No Wait Code

1. Check whether VSE/POWER was used when the system entered the wait state (establish this from the SYSLOG output).

   If VSE/POWER was used, re-IPL your system and rerun the problem program in a partition whose unit record input and output is not being spooled. Consider also running appropriate SDAID traces with dumps specified. If the error still appears, continue with the next step.

   If the problem program runs successfully, consult the VSE/POWER documentation.
2. Check the available dump for unexpected or unreasonable values in the following areas:
   - Old PSWs in low address storage (machine check, program, I/O, SVC, external). Note the interrupt codes.
   - General purpose registers.
   - Control registers.
   - CSW and CAW.
3. In the dump, examine the contents of relevant control blocks (for example: SYSCOM, LUB, PUB, CHANQ, PIB, ERBLOC).
4. You may want to examine the LTA. The first eight bytes of that area contain the name of the transient phase that was being executed in the LTA at the time the wait state was entered; provided that name was not modified by program action. (You can find the address of the LTA in the SYSCOM).

   If a user-written transient was executed last, obtain the listings of that phase (in source code) and check the use of SVC22. The first occurrence of that SVC seizes the system, the second one releases it.
5. Recreate the hard wait situation while the SDAID trace is running.

   If the above two steps do not provide an indication for the cause of the malfunction, trace the suspected instructions and the alteration of specific storage areas. Request, in the TRACE command, dumps of critical storage areas on occurrence of storage-alter events. This should give you enough information for final error diagnosis.

   If the output of the SDAID trace does not give an indication of the cause of the error, contact IBM for support.

Procedure: Soft Wait

1. Determine which of the active tasks were waiting and why.

   In the dump output, locate the PIB extension (PIB2TAB) table. The four-byte pointer at displacement 8 of the table points to the Partition Control Block (PCB). Locate the task ID string at displacement X'20' of the PCB. All non-zero bytes represent the task IDs of all the tasks that have been started in this partition. The reason of a soft wait can be found by examining the task status flag (TIB+X'20'=TIBRQID) of all tasks that have been started. Any task status other than X'80' and X'83' is indicating a task wait state. Byte 4 through 7 of the TIB, in this case, contain bound state information. This can be the address of a control block (CCB/IORB, ECB, or TECB) if the task is I/O bound (X'82') or it could be the address of a resource for which the task is waiting for.

2. Examine (and take notes of) the return PSW of each affected task.

   Use the instruction address contained in that PSW to locate the instruction that will be executed when the partition (or task) regains control.

3. Analyze the program(s) running when your system entered the wait state.

   Use the program listings and the linkage editor map for each of the programs that were running. Together with the instruction address in the associated return PSW, locate the re-entry instruction in the program listings.

   Check the coding of the applicable routine for correct usage of macros such as ENQ, DEQ, and WAITM (possibly, a task waited for a particular resource). For details about these macros and their formats, refer to the manual *z/VSE System Macros Reference*.

4. If the above actions did not help you resolve the problem, consider executing the problem program with the SDAID trace on. Trace the applicable instructions and, possibly, the alteration of involved registers or storage areas. This should provide sufficient information to isolate the cause of the soft wait condition.

   If necessary, contact IBM for support and have the data collected available for problem analysis.

Recovering from a Wait State

1. Did you run the SDAID trace?

   If YES, check for the SDAID wait codes in the address part of the wait PSW.

   - If the PSW contains X'00EEEE', an event occurred for which OPTION=HALT was specified.
   - If the PSW contains X'EEEEEE', the SDAID output device requires attention. Display the low storage bytes 0 - 3 for further device information.

   In both cases, refer to “Starting/Terminating Tracing in a System Wait Condition” in the *z/VSE Diagnosis Tools* for the restart possibilities (and the meaning of the codes stored in bytes 0 and 1).

   If NO, continue with the next step.

2. If it is a **Hard Wait** (bits 6 and 7 of the current PSW are both off), proceed as follows:

   a. If the address part of the current PSW (bits 40 through 63) is unequal to X'00EEEE' there is no way to recover. Take a stand-alone dump and re-IPL your system as described under step 6.

   b. If the address part of the current PSW (bits 40 through 63) is equal to X'00EEEE', *restart* the system. If this is not possible, take a stand-alone dump and re-IPL your system as described under step 6.

3. If it is a **Soft Wait** (bit 6 or 7 or both of the current PSW are on), proceed as follows:

   a. Display low address storage bytes 0 - 3 (with ALTER/DISPLAY). Bytes 2 through 3 may contain the cuu address of a failing device and the system is unable to report this problem on the system console. In this case, byte 0 contains the message number and byte 1 contains the message type. The operator must perform the manual procedures as described in the *z/VSE Messages and Codes* manual for the message number in byte 0.

   b. Enter an appropriate attention or subsystem command to retrieve additional information that helps you in finding the problem area.

   c. Try to find a PUB entry which has an I/O operation enqueued, but which has not yet been started (DEVBSY bit in PUBCSFLG is off). Issue the *ONLINE cuu* command to get that device selected for restart.

4. If the problem affects the whole system, do a STORE STATUS/MACHINE SAVE and take a stand-alone dump for problem analysis. Refer to “Taking a Stand-Alone Dump” on page 97 for details. Re-IPL the system.

   If the problem can be isolated to a single partition, use the command **DUMP pp,cuu** to create a partition dump for problem analysis.

   If necessary, contact IBM for support and have the data collected available for problem analysis.

Loop Symptom

A loop may be caused by one of the following:

- A coding or logic error in the program.
- An error in setting up the job.
- Malfunction of an I/O device.

Data You Should Collect

- System console log (printout of the hardcopy file).
- Stand-alone dump.
- Output of the MAP command.
- Output on SYSLST; if the loop includes user-written code, also output on the line printer owned by the particular partition.
- Trace information (notes if the loop was traced by manual operation, otherwise the output produced by SDAID). Trace information should include a recording of PSW and general-register displays.
- Link map of virtual storage of the running program (ACTION MAP).
- Source listing (or source deck) of the program that was running when the loop occurred.

Diagnosing Steps

1. Can you communicate with the **Attention Routine?**

   ==> If YES, go to the next step.

   ==> If NO,

   - Are there any messages when you attempt to activate the attention routine or to reply to a pending reply ID?

     ==> If NO, the loop is probably a supervisor disabled loop. Sample the current PSW to determine if the loop is tight or long, and at the same time display low address bytes to see if the loop contains any interrupts.

     ==> If YES, the loop is long and is enabled for interrupts. Sample the loop by displaying the current PSW. At the same time display low address bytes and note any interrupt data changes.

   Note: The method for *tracing a loop* depends on whether the loop is tight or long, and whether it is interruptible or not. You determine this by sampling the current PSW to see if the loop is tight, and at the same time display low address bytes to see if the loop contains interrupts. If the loop is tight, do instruction stepping (as indicated in the Operation manual for your processor).

If the loop is long example the loop. In either case, if the loop is interruptible, note the interrupt data in low storage at the same time.

2. Is more than one partition active?

   ==> If NO, trace the loop as described in the Note above and record the contents of low address bytes. Do a STORE STATUS and take a stand-alone dump.

   ==> If YES, (optionally) invoke an SDAID trace.

   For an indication of the range of the loop, trace applicable branch and I/O instructions and possibly program fetch and load. Limit the trace to the partition with the fault.
3. If necessary, do a STORE STATUS and take a stand-alone dump.

Additional Considerations

1. Determine where (in IBM or user code) the loop occurred.

   Use for this purpose (1) the trace of the loop (SDAID output of a branch trace or the notes on a manual trace) and (2), if necessary, the linkage editor map and the stand-alone dump output.

   If the loop is entirely within IBM code, make sure that all your records on the failure are available and contact IBM for support.

   If the loop is not entirely within IBM code, continue with the next step.

   Isolate the loop to a single routine or section of code.
2. Use the listings and the documentation provided for the code to analyze the reason for the loop. This may require a code inspection, instruction by instruction. You might also consider a rerun of the program with the SDAID trace on. Trace specific instructions and the alteration of specific storage areas. If your loop extends beyond phase boundaries, include a trace of phase fetch and load activities.

   You should pay particular attention to the following items when checking the code:
   - Counters that can never reach their limits, that are not updated, or that are not initialized.
   - Counters that are not tested properly each time an intended loop is executed.
   - I/O operations that cannot be completed because of the program's logic flow.
   - Correctness of values passed to a program.

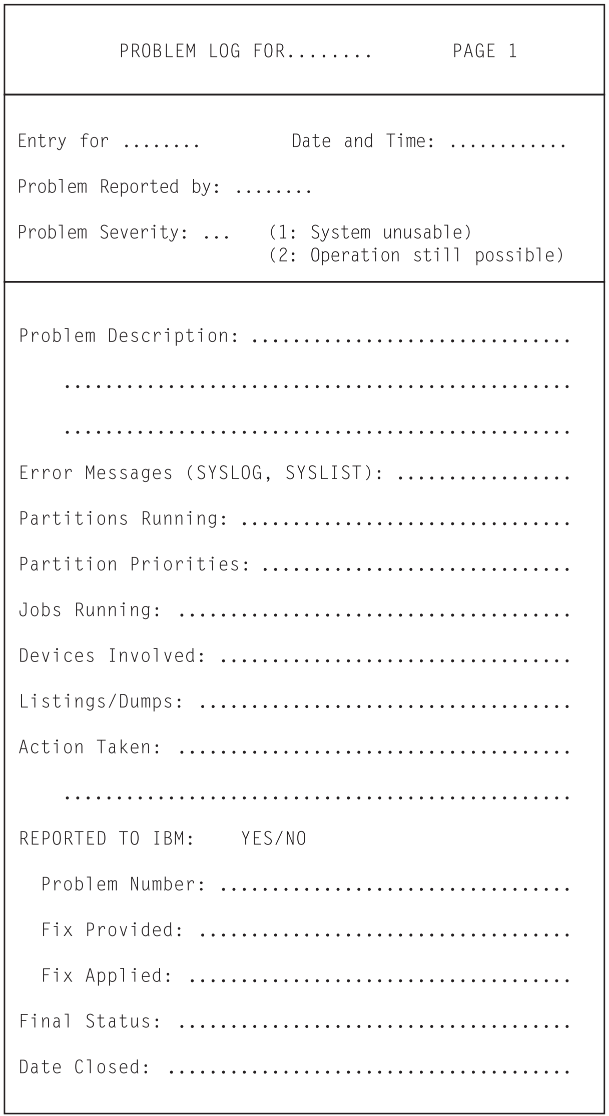
   If necessary, contact IBM for support and have the data collected available for problem analysis.

Appendix A. Problem Log and Change Log

Problem Log

To keep a detailed record of system problems, you should create a Problem Log. Figure 42 shows a problem log example. In the log, record the problem in detail and what was done to correct it.

Note: The documentation for the applications run at your installation should include error recovery procedures.



```
        PROBLEM LOG FOR........       PAGE 1

Entry for ........        Date and Time: ...........

Problem Reported by: ........

Problem Severity: ...   (1: System unusable)
                        (2: Operation still possible)

Problem Description: ..............................

    ...............................................

    ...............................................

Error Messages (SYSLOG, SYSLIST): ................

Partitions Running: ...............................

Partition Priorities: .............................

Jobs Running: .....................................

Devices Involved: .................................

Listings/Dumps: ...................................

Action Taken: .....................................

    ...............................................

REPORTED TO IBM:    YES/NO

  Problem Number: .................................

  Fix Provided: ...................................

  Fix Applied: ....................................

Final Status: .....................................

Date Closed: ......................................
```

Figure 42. Problem Log Example

Change Log

Maintaining system resources is an important task of the system administrator. For example:

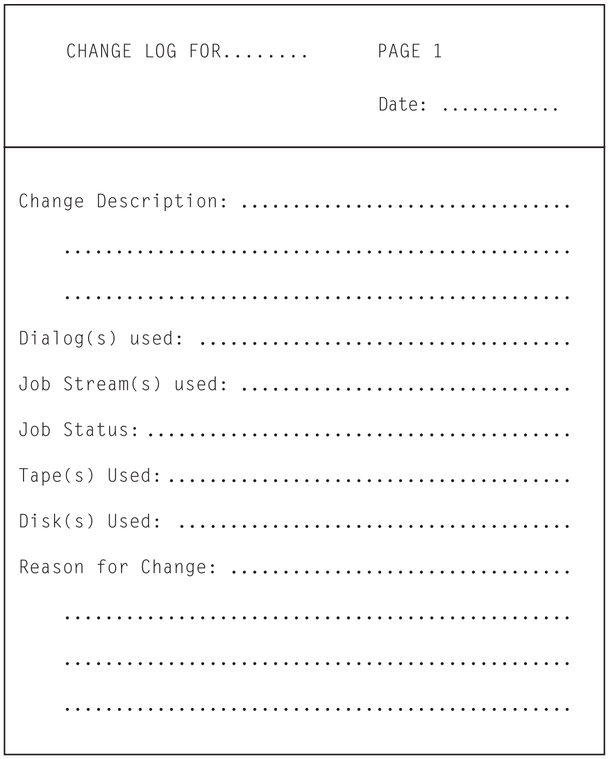
- Define or delete a file.
- Create a user library.
- Extend file space.
- Install an IBM licensed program.

If an erroneous change was implemented, messages like the following may be the result:

```
1A80I  SYSTEM FILE OPEN FAILURE
4181I  NO LABEL INFORMATION ...
4228I  FILE filename OPEN ERROR X'nn'(nnn) ...
4n01I  NO FORMAT 1 LABEL FOUND ...
4n42D  NO MATCHING EXTENT ...
4n98I  OVLAP UNEXPRD SECRD FILE ...
```

Note: To avoid such problems, check the output when performing such tasks carefully. Especially, look for any error messages.

You should **carefully plan** for any system changes required and create a **permanent record** of them. Figure 43 shows an example of a change log form. A Change Log reduces the chances that you may accidentally damage your system. For example, by restoring a back-level version of a library. It also helps you identify problems caused by system changes.



```
    CHANGE LOG FOR........      PAGE 1

                                Date: ...........

Change Description: ..............................
    ..............................................
    ..............................................
Dialog(s) used: ..................................
Job Stream(s) used: ..............................
Job Status:.......................................
Tape(s) Used:.....................................
Disk(s) Used: ....................................
Reason for Change: ...............................
    ..............................................
    ..............................................
    ..............................................
```

Figure 43. Change Log Example

Appendix B. Problem Symptom String and APAR Submission

The Problem Symptom String

Any problem that seems to be caused by IBM code should be described to IBM in the form of a symptom string. An example of the problem symptom string that IBM expects is shown in Figure 44. It consists of:

1. Product ID, which identifies the system and its release level.
2. The system change (refresh) level.
3. The affected component.
4. The change level of the affected component.
5. The applicable failure-type keyword.
6. Supplementary information.

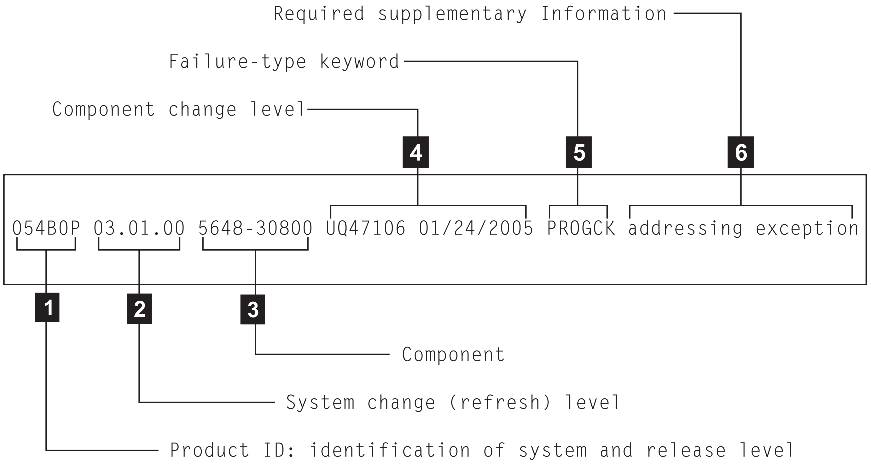

Required supplementary Information
Failure-type keyword
Component change level
4
5
6
054B0P 03.01.00 5648-30800 UQ47106 01/24/2005 PROGCK addressing exception
1
2
3
Component
System change (refresh) level
Product ID: identification of system and release level

Figure 44. Elements of a Problem Symptom String

You can retrieve items 1, 2, and 4 from your system's history file. Therefore, make it a habit to obtain a printout of your system's history file each time you install an IBM program, a system refresh, a PTF, an APAR, or a local fix.

Keep a copy of this printout on file, in case you need to present a software problem to IBM. Item 3 is the identification of the IBM program (component) causing your problem.

Obtaining a System History Printout

To obtain an up-to-date change level record of your system, access the *Problem Handling* panel and select:

```
5  (Retrace History File)
```

To get a history overview retrace as shown in Figure 45 on page 180, use the *Retrace* dialog. To get a component retrace as shown in Figure 46 on page 180, use the *Retrace Components* dialog.

Following is a description of each of the information items shown in Figure 44, telling you where to find the required information.

Displaying the System or Refresh Level of Your System

At the system console, you can display the system or refresh level of your installed system by executing the procedure SPLEVEL. This displays the z/VSE level, the installation date, and the Copyright statement.

For details of the SPLEVEL procedure, refer to “Displaying the z/VSE Level” on page 121. This information is also shown in the header of the retrace overview. See Figure 45.

```
DATE 01/27/2005 (MDY)     JOB RETRACE     MSHP-FUNCTION = RETRACE OVERVIEW                                     PAGE    1
                              CUSTOMER                IUI DEV
                              ADDRESS                 WUNDERBLICK STR.234
                              PHONE NUMBER            09090-77-0
                              SYSTEM PROGRAMMER       AMAD BUBR ELKE HSCZ WACK
                              ENVIRONMENT             Z/VSE, RELEASE  3.1
                              HISTORY FILE             24 % IN USE
                              REFRESH LEVEL           03.01.00 2
DATE 01/27/2005 (MDY)     JOB RETRACE     MSHP-FUNCTION = RETRACE OVERVIEW                                     PAGE    2
                              PRODUCT LIST

⋮
          PRODUCT  054B0P 1  INSTALLATION DATE = 01/24/2005  'Z/VSE 3.1.0 - CICS TS FOR VSE 1.1.1'
                             COMPRISES    5648-05400 3
                             RESIDES IN   PRD1.BASE         (PRODUCTION PART)
   PTF  |     AFFECTS     |  APAR  |   DATE   |
UQ46776  5648-05400-B0P    PQ40971  01/24/2005
         DFHUSXM .OBJ
⋮
  APAR   |   PTF   |  APAR   |   PTF   |  APAR   |   PTF   |  APAR   |   PTF   |  APAR   |   PTF   |  APAR   |   PTF   |
PQ40971   UQ46776   PQ41268   UQ58042   PQ41434   UQ47249   PQ41437   UQ47389   PQ41522   UQ75840   PQ41588
⋮
```

Figure 45. Example of a RETRACE Printout (History Overview)

```
DATE 01/27/2005 (MDY)     JOB RETRCMP     MSHP-FUNCTION = RETRACE COMPONENT                                    PAGE    1
                              CUSTOMER                IUI DEV
                              ADDRESS                 WUNDERBLICK STR.234
                              PHONE NUMBER            09090-77-0
                              SYSTEM PROGRAMMER       AMAD BUBR ELKE HSCZ WACK
                              ENVIRONMENT             Z/VSE, RELEASE  3.1
                              HISTORY FILE             24 % IN USE
                              REFRESH LEVEL           03.01.00
⋮
                              COMPONENT LIST
                              COMPONENT = 5648-05400   LEVEL = B0P   CORRECTLY   INSTALLED = 01/24/2005
                              PRODUCTION PART IN: PRD1.BASE
                                                                                  COMPONENT: 5648-05400-B0P  PTF: UQ46776
   PTF   |STATUS|   COMPONENT  | APARS  |     AFFECTS     |   DATE   |          REQUIRES         |SUPERSEDES|SUPERSEDED|
⋮
UQ46776    INTEG 5648-05400-B0P PQ40971  DFHUSXM .OBJ      01/24/2005 PRE = 054B0P

UQ47106 4  INTEG 5648-05400-B0P PQ40944  DFHPASYL.PHASE    01/24/2005 4  PRE = 054B0P
                                         DFHPADM .OBJ
                                         DFHPAGP .OBJ
                                         DFHPASYT.OBJ
⋮
 APAR  |      AFFECTS      |  PTF  |   DATE   |    FIX DESCRIPTION / REQUIRES INFORMATION
⋮
PQ40944  5648-05400-B0P    UQ47106  01/24/2005
PQ40971  5648-05400-B0P    UQ46776  01/24/2005
⋮
```

Figure 46. Example of a RETRACE COMPONENTS Printout

Identification of the Failing System

Product ID

In Figure 45 on page 180, **1** points to the *product ID* (054B0P in this example).

System Change (Refresh) Level

The *system change level* of your system is identical with its refresh level as shown in Figure 45 on page 180. In the example used, **2** points to the refresh level 03.01.00.

Identification of the Failing Component

The Component

If you have come to the conclusion that your problem was caused by an IBM program (product), the name of that program gives you an indication of the component that failed. In the example used, **3** points to component 5648-05400 of product 054B0P. Ask IBM for details, if necessary.

Component Change Level

Under "Component List", Figure 46 on page 180 lists all the components that are part of the installed system. That component's *change level* is given either:

- By the "Correctly Installed" date if no PTF had been applied to that component up to the time the problem occurred, or
- By the number and application date of the PTF most recently (last one) applied. In the example used, **4** points to PTF UQ47106 which is the PTF that was applied last.

Description of the Type of Failure

Information **5** and **6** of Figure 44 on page 179 must be supplied by the user and depends on the error that occurred. For failure-type keyword, see description below.

Failure-Type Keyword

For a search of IBM's data base RETAIN®, a program failure should be categorized by one or a combination of the failure-type keywords listed below. Report your problem to the IBM Support Center with the applicable failure-type keyword(s). The keywords are:

MSGx...x
An incorrect message occurred, or a message was incomplete or missing.

where

x...x = the message identifier printed by the system, which you use to find the message explanation in the *z/VSE Messages and Codes* manual.

PROGCK
A program check occurred.

Any problem that causes the system to terminate (cancel) a program before completion. This includes program abend situations.

INCORROUT
Program output is incorrect.

LOOP An unintended loop occurred.

WAIT A wait state occurred. This includes also error situations described as suspended online activities in this manual.

PERFM
A performance problem occurred.

DOC A publication contains incorrect information or required information is missing.

Each of these failure keywords is discussed in the following section.

An Incorrect Message Occurred (MSGx...x)

Message, in this context, may also be an assembler-originated MNOTE, compiler-originated diagnostic information, or any diagnostic code such as a return or error code in a register (normally register 15) or in a field of a control block.

A problem would fall into this category if, for example, you performed the action recommended by the particular message, but

The problem still persists, or

Your action created another problem.

If your system displayed a message which is not documented in *z/VSE Messages and Codes*, (or in another applicable “Messages” manual), report this as a documentation (DOC) problem.

An incorrect message problem would be submitted to IBM with a failure-type keyword such as the following example:

```
MSG4122I   (indicates a problem with message 4122I)
```

A Program Check Occurred (PROGCK)

When a program check condition occurs, the system indicates the type of program exception in message 0S03I. For a successful search of RETAIN in case of a program-check problem, IBM needs to know what type of a program exception (such as addressing exception or operation exception) was displayed by the system. A program abend or cancel situation falls into this category too.

Program Output is Incorrect (INCORROUT)

If you have come to the conclusion that IBM code is at fault, submit the problem with a failure-type keyword as follows:

```
 INCORROUT D/Tnnnn function
```

where

```
 D/Tnnnn  = IBM device type; for example: D/T3211.
```

function = the failing function; for example:

```
      SKIP    for skipping a line
      PRINT   for printing a line
      SELECT  for selecting a stacker
      WRITE   for an erroneous WRITE operation
```

or

```
         the name of the IBM program that produced
         the incorrect output.
```

For a successful search of RETAIN with INCORROUT as failure-type, IBM requires, as a component name, either the name of the IBM language translator that was used to compile or assemble the failing program or the name of the access method used.

A Loop Occurred (LOOP)

If your problem is an unintended program loop and you have good reason to believe that IBM code is at fault, use LOOP as the failure-type keyword. If possible, provide, as supplementary information, the name of the program in which the loop occurs.

A Wait Occurred (WAIT)

If you have isolated the cause for the wait condition to IBM-supplied code, submit the problem to IBM with WAIT as the failure-type keyword. If the system displayed a wait code, provide that code as supplementary information. For a list of wait codes that may be displayed by the system, see the *z/VSE Messages and Codes* manual.

A Performance Problem Occurred (PERFM)

A problem of this kind is indicated if the performance of your system has decreased notably.

Provide copies of run-time records available at your installation and ensure that sufficient configuration and job-mix information is available to reproduce the environments for both the satisfactory and the unsatisfactory performance.

Information in a Manual is Incorrect or Missing (DOC)

A problem should be reported with this keyword if the problem is caused by incorrect, missing, or ambiguous information in a manual.

Submitting an APAR

When the problem found is not yet known to IBM, and if probably IBM code is at fault, you may have to submit an APAR. APAR stands for Authorized Program Analysis Report. An APAR includes a detailed problem description, along with any related data collected.

If an APAR is required, follow the instructions of your IBM Support Representative who will process the APAR through RETAIN.

The data required for an APAR submission for any given component may be more than listed in this manual. If so, your IBM Support Representative will tell you which additional data is required.

Appendix C. List of IPL, JCL, and Label Procedures

This appendix lists the names of the IPL, JCL, and label procedures shipped with z/VSE.

```
   IPL           JCL           Label
Procedures:   Procedures:   Procedures:

  $IPLE80      $0JCL780      STDLAB80
  $IPLE90      $0JCL790      STDLAB90
  $IPLEGF      $0JCL7GF      STDLABGF

               $1JCL780
               $1JCL790
               $1JCL7GF

               $2JCL780
               $2JCL790
               $2JCL7GF

               $3JCL780
                 ...
               $4JCL780
                 ...
               $5JCL780
                 ...
```

Figure 47. IPL, JCL, and Label Procedures for Initial Installation

Note: z/VSE uses these procedures only once for initial installation.

The name of a procedure is determined by the disk device type used for DOSRES and SYSWK1. The following naming conventions apply for the procedures:

- The last two characters identify the type of disk device on which DOSRES and SYSWK1 reside. The manual *z/VSE Installation* shows the layout of DOSRES and SYSWK1 for each disk device type supported for initial installation under “z/VSE Disk Layouts”. Procedures are provided for initial installation for the following disk device types:

  ```
  80 = IBM 3380
  90 = IBM 3390
  GF = IBM FBA/SCSI
  ```

  The characters GF indicate that this is the generalized layout for FBA disk devices such as an FBA or SCSI disk under z/VM.
- In a running system, after installation, the procedures have been renamed as follows:
  - IPL procedure: $IPLESA
  - JCL procedures: $0JCL, $1JCL, $2JCL, and so on.
  - Label procedure: STDLABEL

    Procedure STDLABEL calls procedures STDLABUP and STDLABUS.

It might be useful to have printouts available of those procedures which are relevant for your system. Such a printout can help you identify the original contents of a procedure in case of a system startup problem because of user modifications. This should be done right after initial installation.

You can use the VSE/ICCF **LIBRP** command to copy a procedure from the system library IJSYSRS.SYSLIB to your VSE/ICCF library. From it, print the procedure using the *Program Development Library* dialog.

Example of an IPL Procedure for Initial Installation

Figure 48 shows the statements of an IPL procedure for initial installation (stored in system library IJSYSRS.SYSLIB) **as shipped** by IBM. As an example, the figure shows procedure $IPLE90, required for initial installation of a z/VSE system residing on an IBM 3390 disk device.

```
MEMBER=$IPLE90.PROC      SUBLIBRARY=IJSYSRS.SYSLIB   DATE:2013-12-10
                                                     TIME: 12:42
-------------------------------------------------------------------------
01F,$$A$SUPI,VSIZE=256M,VPOOL=64K,IODEV=1024,VIO=512K
ADD FDF,FBAV VIRTUAL DISK FOR LABEL AREA
ADD FEC,3505
ADD FFC,3505 ICCF DUMMY DEVICE DON'T DELETE
ADD FFA,3505 ICCF DUMMY DEVICE DON'T DELETE
ADD FED,2520B2
ADD FFD,2520B2 ICCF DUMMY DEVICE DON'T DELETE
ADD FEE,PRT1
ADD FEF,PRT1
ADD FFE,PRT1 ICCF DUMMY DEVICE DON'T DELETE
ADD FFF,CONS DEDICATED CONSOLE DON'T DELETE
DEF SYSCAT=DOSRES,SYSREC=SYSWK1
SYS DASDFP=YES
SYS JA=YES
SYS SPSIZE=0K
SYS NPARTS=60
SYS PASIZE=50M
SYS SDSIZE=96K
DPD VOLID=DOSRES,CYL=398,NCYL=36,DSF=N
DPD VOLID=DOSRES,CYL=434,DSF=NO
SVA PSIZE=(652K,6M),SDL=700,GETVIS=(768K,6M)
```

Figure 48. IPL Procedure for Initial Installation

Appendix D. Error Messages When Using Virtual Tapes

Related Sections:
- “Using the Virtual Tape Data Handler Trace” on page 155
- “Using the Virtual Tape Server Trace” on page 155

This appendix describes the error messages that can be generated when you use virtual tapes. It contains these main sections:
- “Understanding the Sense Information in an I/O Error Message”
- “Virtual Tape Support: All Error Messages”
- “Additional Actions for Data Handler Message 1YM7t” on page 189

Understanding the Sense Information in an I/O Error Message

The error messages that you might receive when using virtual tapes are provided by:
- Input/output (I/O) supervisor.
- The trace used with the Virtual Tape Data Handler.

For every *I/O error*, a message similar to the one below is generated. Each message contains “sense” (SNS) information, as shown below.

```
BG 0014 0P32I C  NON COMPAT SYS099=480
      CCSW=020064111002000050 CCB=6411A8
      SNS= 0849202E 00000020 00000000 00000133 00640088 04200220
           042000F0 000000FF
```

In this example:
- Reason Code X'0064' converts to a return code (decimal) of 100. In the list of error messages shown in “Virtual Tape Support: All Error Messages,” return code 100 means that a CONNECTION_ERROR has occurred.
- Reason Code X'0133' converts to a return code (decimal) of 307. In the list of error messages shown in “Virtual Tape Support: All Error Messages,” return code 307 means that a TCP_UNRECOVERABLE_READ_ERROR has occurred.

Note: Related error messages might also be displayed on the Virtual Tape Server that is running on your workstation.

Virtual Tape Support: All Error Messages

Here is a list of all *internal* error messages that can be generated by z/VSE's Virtual Tape Support. It includes additional explanations (where necessary).

```
/* TDH internal Error Messages  */
 Message                     RC    Explanation
 =======                     ==    ===========
 OK                          0     no error
 CONNECTION_ERROR            100   problem with TCP/IP connection
 VIRTUAL_TAPE_ERROR          200   problem with virtual tape
 INTERNAL_ERROR              300   Internal error
 CANCEL                      400   TDH was cancelled


/* Error Messages from VTM requests - VSAM */
 Message                     RC    Explanation
 =======                        ==    ===========
 VS_INVALID_REQ_TYPE            100   invalid req. type in VTM req
 NO_OPEN_TAPE                   101   no open tape for UndefCuu, GetBuffer or
                                      PutBuffer
 NULL_POINTER                   102   pointer is NULL
 VS_ALLOCATION_ERROR            103   no space allocated
 VTM_REQ_ERROR                  104   invalid request from Virtual Tape Manager
 VS_WRITE_NOT_ALLOWED           105   file not opened for write
 ANOTHER_REQUEST_PENDING        106   incomming request but previous request not
                                      completed
 VS_UNKNOWN_RETURN_CODE         107   unknown rc from AWS routine
 INVALID_VIRTAPE_STRUCT         108   invalid field content found in  virtape
                                      structure
 NO_PENDING_READ_WRITE          109   status of vtm_req is in_progress but no read
                                      or write is pending
 INVALID_MEDIUM                 110   status is in_progress but medium is not TCPIP
 UNKNOWN_HOST_NAME              111   host name could not be resolved

/* Error Messages from VTM requests - TCP/IP */
 Message                        RC    Explanation
 =======                        ==    ===========
 TCP_INVALID_REQ_TYPE           300   invalid req. type in VTM req
 TCP_INVALID_BYTE_COUNT         302   counter for bytes to sent
 TCP_GETSOCKET_FAILURE          303   no socket received
 TCP_CONNECT_FAILURE            304   CONNECT not successful
 TCP_UNRECOVERABLE_SEND_ERROR   306
 TCP_UNRECOVERABLE_READ_ERROR   307
 TCP_CLOSE_ERROR                308
 TCP_ALLOCATION_ERROR           309
 TCP_IOCTL_ERROR                311   TCP/IP macro ioctl ended with error
 TCP_TIMEOUT                    312
 SELECT_ERROR                   313
 EXCEPTION_WRITE_CHECK          314   exception when checking socket for write ready
 EXCEPTION_READ_CHECK           315   exception when checking socket for read ready
 INET_NTOP_ERROR                316   inet_ntop error

/* Error Messages from Tape Data Handler Server - TCP/IP  */
 Message                        RC    Explanation
 =======                        ==    ===========
 COMMON_INVALID_REQUEST         201   invalid request from Tape Data Handler Client
 CUU_ALREADY_DEFINED            202   virtual tape already defined
 CUU_NOT_DEFINED                203   cuu has never been defined
 CUU_INVALID                    204   cuu in request does not match with cuu defined
 MULTIPLE_READ_WRITE            205   virtual tape is used concurrently with read and
                                      write access
 USER_REQ_CANCEL                206   user requested cancel
 TDH_CANCEL                     207   Tape Data Handler canceled
 NO_VIRTUAL_TAPES_DEFINED       208  No active virtual tapes
 ERR_UNKNOWN_HOST               209 /improper ip/port spec.

/* Error Messages from VSAM access services   */
 Message                        RC    Explanation
 =======                        ==    ===========
 AWSERR_INVALID_PARAM           401   invalid parameter
 AWSERR_NULL_POINTER            402   null pointer

 AWSERR_OPEN_FAILED             403   open has failed
 AWSERR_INVALID_FILE            404   file is not ESDS
 AWSERR_FORMAT_ERROR            405   format error
 AWSERR_NOT_EOF                 406   write rejected, not at EOF
 AWSERR_READ_ERROR              407   read error, possible eof
 AWSERR_WRITE_ERROR             408   write error
 AWSERR_GETPOS_ERROR            409   get pos error
 AWSERR_SETPOS_ERROR            410   set pos error

 AWSERR_EXISTS_ALREADY          411   Index Entry exists already
 AWSERR_NOT_FOUND               412   Index Entry not found
 AWSERR_BUFFER_ERROR            413   Buffer Handling error
 AWSERR_BUFFER_TOO_SMALL        414   Buffer is too small
 AWSERR_EXCEEDS_4GB             415   VSAM SIZE > 4GB
 AWSERR_LABEL_ERROR            416   Error during LABEL handling
 AWSERR_ZLIB_ERROR             417   Error during ZLIB handling

 BUFFERR_NO_ERROR              500
 BUFFERR_INVALID_PARAM         501
 BUFFERR_INVALID_LENGTH        502
 BUFFERR_NO_SPACE              503
 BUFFER_END_OF_VOLUME          504
 BUFFER_END_OF_BUFFER          505
 BUFFER_INVALID_BUFFER         506
 BUFFER_TOO_SMALL              507

/* Error messages for stacking tape support (since z/VSE 5.2) */
 Message                       RC    Explanation
 =======                       ==    ===========
 TAPE_LIMIT                    800   number of images on tape
 TAPE_INVALID_REQ_TYPE         600   invalid req. type in VTM req
 TAPE_IN_USE                   601   stape is in use
 TAPE_ALLOCATION_ERROR         602   no space left to allocate
 TAPE_ASSIGN_ERROR             603   stape cannot be assigned
 TAPE_UNKNOWN_RC               604   unknown rc from AWS routine
 TAPE_LABEL_ERROR              605   error with label processing
 TAPE_NO_VOL1                  606   tape has no standard label
 TAPE_NO_STACKING              607   tape is not a stacking tape
 TAPE_CORRUPTED                608   last file is not a directory
 TAPE_INITIALIZED              609   tape is already initialized
 TAPE_LIMIT_REACHED            610   tape is full
 TAPE_OPEN_ERROR               611   error when opening file
 TAPE_READ_ERROR               612   error when reading file
 TAPE_WRITE_ERROR              613   error when writing file
 TAPE_OTHER_ERROR              614   other C LE error
 TAPE_DUP_IMAGE                615   file name already used
 TAPE_NO_FILE                  616   file name not in directory
 TAPE_WRITE_NA                 617   file not opened for WRITE
```

Additional Actions for Data Handler Message 1YM7t

When starting a virtual tape, you might receive the *external* message “1YM7t Tape Data Handler Encountered Connection Error”.

In the Online Message Explanation (OME), the message explanation is: “The virtual Tape Data Handler tried to establish a TCP/IP connection to a foreign host and failed for one of the following reasons”. These are the listed possible causes of the error message:

- TCP/IP partition not active on the VSE system.
- Virtual Tape Server not active on the foreign host.
- No foreign host found with the specified IP address.
- Foreign host with specified IP address did not respond.

However, there are *two further* possible causes:

TCP/IP Started With a Different System ID

If TCP/IP for VSE/ESA has been started using a system ID, the appropriate `EXEC IPNET` statement will look like this:

```
// EXEC IPNET,SIZE=IPNET,PARM='ID=nn,INIT=... '
```

(where the default ID is `00`).

If you wish to use TCP/IP services from another partition, this partition has to recognize the system ID. To do so, you must specify:

```
// OPTION SYSPARM='nn'
```

(where *nn* is the system ID).

You can use skeleton SKVTASTJ (in ICCF library 59) to add the above statement. For further details, refer to the chapter “BSD/C Sockets - Connecting to TCP/IP” in the manual *TCP/IP for VSE, Programmer' Reference*. You can find this manual in the *TCP/IP for VSE VnRn PDFs* catalog of the VSE Collection kit, SK2T-0060.

Incorrect $EDCTCPV.PHASE Has Been Used

There are **two** copies of $EDCTCPV.PHASE that are supplied with each release of z/VSE:

- One copy in library PRD2.SCEEBASE.
- One copy in library PRD2.TCPIPC in case the CSI product is used, or in the product library, which is supplied as part of TCP/IP for VSE/ESA.

If z/VSE uses the 'dummy' phase contained in PRD2.SCEEBASE, TCP/IP socket calls **will not work**. An EDC-prefixed message will then be sent to SYSLST:

```
Function nnn not implemented
```

To correct this error, you should ensure that the active $EDCTCPV.PHASE is the one supplied as part of TCP/IP for VSE/ESA (you should correct the library search order in the LIBDEF statements).

Note: Related error messages might also be displayed on the Virtual Tape Server that is running on your workstation.

Appendix E. Problem Solving When Using SCSI Disk Devices

This appendix provides a few sentences only about problem solving when using SCSI disk devices. However, the main source of information for using SCSI disk devices is chapter "Configuring Your System to Use SCSI Disks" in the *z/VSE Administration*.

Internal SCSI Errors (0S4n)

Internal error messages in the range 0S40I to 0S46I can occur during the configuration of SCSI disk devices. For a details of these errors and how to remedy them, refer to the chapter “Configuring Your System to Use SCSI Disks” in the *z/VSE Administration*.

Checking Which SCSI Devices Are Available

To obtain the configuration of all SCSI devices in the system, you can use the JCL QUERY SCSI command:

```
QUERY SCSI
```

To obtain the configuration of a single SCSI device in the system, you can also use the JCL QUERY SCSI command:

```
QUERY SCSI,cuu
```

Glossary

This glossary defines technical terms and abbreviations used in the *z/VSE Guide for Solving Problems*. If you do not find the term you are looking for, view the *IBM Dictionary of Computing* located at:

http://www.ibm.com/ibm/terminology/

The glossary includes definitions with symbol * where there is a one-to-one copy from the IBM Dictionary of Computing.

*** 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 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.

address space
A range of up to two gigabytes of contiguous virtual storage addresses that the system creates for a user. Unlike a data space, an address space contains user data **and** programs, as well as system data and programs, some of which are common to all address spaces. Instructions execute in an address space (not a data space). Contrast with *data space*.

Advanced Function Printing (AFP)
A group of IBM licensed programs that support APA printers.

AFP Advanced Function Printing.

alternate block
On an FBA disk, a block designated to contain data in place of a defective block.

alternate index
In systems with VSE/VSAM, the index entries of a given base cluster organized by an alternate key, that is, a key other than the prime key of the base cluster. For example, a personnel file primarily ordered by names can be indexed also by department number.

*** alternate tape**
A tape drive to which the operating system switches automatically for tape read or write operations if the end of the volume has been reached on the originally used tape drive.

alternate track
On a CKD disk, a track designated to contain data in place of a defective track.

*** American National Standard Code for Information Interchange (ASCII)**
The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), 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)

APA All points addressable.

APAR Authorized Program Analysis Report.

appendage routine
A piece of code physically located in a program or subsystem, but logically an extension of a supervisor routine.

*** application profile**
A control block in which the system stores the characteristics of one or more application programs.

application program
A program written for or by a user that applies directly to the user's work. See also *batch program* and *online application program*.

*** ASCII**
American National Standard Code for Information Interchange.

ASI (automated system initialization) procedure
A set of control statements which specifies values for an automatic system initialization.

assemble
To translate 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.

attention routine
A routine of the system that receives control when the operator presses the Attention key. The routine sets up the console for the input of a command, reads the command, and initiates the system service requested by the command.

authorized program analysis report (APAR)
A report of a problem caused by a suspected defect in a current release of a program.

*** automated system initialization (ASI)**
A function that allows control information for system startup to be cataloged for automatic retrieval during system startup.

auxiliary storage
Addressable storage that is not part of the processor, for example storage on a disk unit. Synonymous with *external storage*.

*** 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.

binary synchronous communication (BSC)
Method of telecommunication using binary synchronous line discipline. Contrast with *SDLC*.

block Usually, a block consists of several records of a file that are transmitted as a unit. But if records are very large, a block can also be part of a record only. See also *control block*.

blocking
The process of combining (or cutting) records into blocks.

*** bps** Bits per second.

*** bringup**
The process of starting a computer system or a subsystem that is to operate under control of the system.

BSC Binary synchronous communication.

*** BTAM-ES (Basic Telecommunication Access Method Extended Storage)**
An IBM supplied telecommunication access method. It permits read and write communication with remote devices.

B-transient
A phase with a name beginning with $$B and running in the Logical Transient Area (LTA). Such a phase is activated by special supervisor calls.

*** cache**
A high-speed buffer storage that contains frequently accessed instructions and data; it is used to reduce access time.

cache storage
A random access electronic storage in selected storage controls used to retain frequently used data for faster access by the channel. For example, the IBM 3990 Model 3 contains cache.

catalog
1. A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors. (I) (A)

2. To store a library member such as a phase, module, or book in a sublibrary.

See also *VSAM master catalog, VSAM user catalog*.

*** cataloged procedure**
A set of control statements placed in a library and retrievable by name.

CCB Command control block.

CCW Channel command word.

central processing unit (CPU)
The hardware component that interprets and executes instructions. Synonym for *processor*.

chained sublibraries
A facility that allows sublibraries to be chained by specifying the sequence in which they must be searched for a certain library member.

chaining
A logical connection of sublibraries to be searched by the system for members of the same type (phases or object modules, for example).

channel-attached
Pertaining to the attachment of devices directly by data channels (I/O channels) to a computer. Contrast with *link-attached*. Synonymous with *locally attached*.

channel command word (CCW)
One or more CCWs make up the channel program that directs data channel operations.

channel program
One or more channel command words that control a sequence of data channel operations. Execution of this sequence is initiated by a single start I/O (SIO) instruction.

*** channel scheduler**
The part of the supervisor that controls all input/output operations.

channel-to-channel attachment (CTCA)
A function that allows data to be exchanged (1) under the control of VSE/POWER between two virtual VSE machines running under VM or (2) under the control of VTAM between two processors.

checkpoint
1. A point at which information about the status of a job and the system can be recorded so that the job step can be restarted later. 2. To record such information.

CICS (Customer Information Control System)
An IBM program that controls on-line communication between terminal users and a data base. Transactions entered at remote terminals are processed concurrently by user-written application programs. The program includes facilities for building, using, and servicing data bases.

CICS system definition (CSD) file
See CSD.

CICS/VSE
Customer Information Control System/VSE.

CICS TS
CICS Transaction Server

CKD device
Count-key-data device.

class In VSE/POWER, a group of jobs that either come from the same input device or go to the same output device.

CMS Conversational monitor system.

COBOL
Common business-oriented language.

command control block (CCB)
The name of a system control block to hold information about a specific instance of a command.

common business-oriented language (COBOL)
A high-level programming language based on English used primarily for business application programs.

common library
A library that can be interactively accessed by any user of the (sub)system that owns the library.

*** communication adapter**
A circuit card with associated software that enables a processor, controller, or other device to be connected to a network.

*** communication controller**
1. A device that directs transmission of data over the links of a network; its operation is controlled by a program executed in a processor to which the

controller is connected or it may be controlled by a program executed within the device. (T)2. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit. It manages the details of line control and the routing of data through a network.

communication line
See *telecommunication line*.

*** communication region**
An area of the supervisor that is set aside for transfer of information within and between programs.

*** 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.

control block
An area within a program or a routine defined for the purpose of storing and maintaining control information.

control program
A program to schedule and supervise the running of programs in a system.

control unit
See *communication controller*. Synonymous with *controller*.

*** conversational monitor system (CMS)**
A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities and operates under the control of VM/SP.

*** corrective service**
The installation of a PTF or an APAR fix that corrects a specific problem.

count-key-data (CKD) device
A disk device that stores data in the record format: count field, key field, data field. The count field contains, among others, the address of the record in the format: cylinder, head (track), record number and the length of the data field. The key field, if present, contains the record's key or search argument. CKD disk space is allocated by tracks and cylinders. Contrast with *FBA disk device*. See also *extended count-key-data device*.

CPU Central processing unit.

CTCA Channel-to-channel attachment.

DASD
Direct access storage device.

DASD sharing
An option that lets independent computer systems use common data on shared disk devices.

database
A set of data available online that is organized by a common system and used for a common purpose.

DATABASE 2 (DB2)
An IBM relational database management system.

*** data entry panel**
A panel in which the user communicates with the system by filling in one or more fields. See also *panel* and *selection panel*.

data file
See *file*.

*** Data Interfile Transfer, Testing and Operations (DITTO/ESA for VSE) utility**
An IBM program that provides file-to-file services for card I/O, tape, and disk devices.

The current version is called DITTO/ESA for VSE.

Data Language/I (DL/I)
A database access language used with CICS/VSE and CICS TS.

data link
In SNA, the combination of the link connection and the link stations joining

network nodes, for example, a System/370 channel and its associated protocols. A link is both logical and physical.

In SNA, synonym for *link*.

*** data management**
A major function of the operating system. It involves organizing, storing, locating, and retrieving data.

data security
See *access control*.

data set
See *file*.

data space
A range of up to two gigabytes of contiguous virtual storage addresses that a program can directly manipulate through ESA/370 instructions. Unlike an address space, a data space can hold only user data; it does not contain shared areas, system data or programs. Instructions do not execute in a data space, although a program can reside in a data space as non-executable code. Contrast with *address space*.

deblocking
The process of making each record of a block available for processing. Contrast with *blocking*.

default value
A value assumed by the program when no value has been specified by the user.

*** device address**
1. The identification of an input/output device by its channel and unit number.

2. In data communication, the identification of any device to which data can be sent or from which data can be received.

*** device class**
The generic name for a group of device types, for example, all display stations belong to the same device class. Contrast with *device type*.

*** Device Support Facilities**
An IBM program for performing operations on disk volumes so that they can be accessed by IBM and user programs. Examples of these operations are initializing a disk volume and assigning an alternate track.

*** device type code**
The four- or five-digit code to be used for defining an I/O device to a computer system.

*** dialog**
1. In an interactive system, a series of related inquiries and responses similar to a conversation between two people. 2. For VSE/SP, a set of panels that can be used to complete a specific task, for example, defining a file.

direct access
Accessing data on a storage device using their address and not their sequence. This is the typical access on disk devices as opposed to magnetic tapes. Contrast with *sequential access*.

Direct access storage device
A device in which access time is effectively independent of the location of the data.

directory
1. A table of identifiers and references to the corresponding items of data. (I) (A) 2. In VSE, specifically, the index for the program libraries. See also *library directory* and *sublibrary directory*.

disk sharing
An option that lets independent computer systems use common data on shared disk devices.

display station
A display screen with attached keyboard for communication with the system or a network. See also *terminal*.

*** distribution tape**
A magnetic tape that contains, for example, a preconfigured operating system like z/VSE. This tape is shipped to the customer for program installation.

DITTO utility
Data Interfile Transfer, Testing and Operations utility.

The current version is called DITTO/ESA for VSE.

DOSRES
Disk operating system residence volume.

dummy device
A device address with no real I/O device behind it. Input and output for that device address are spooled on disk.

dump 1. Data that has been dumped. (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)

*** duplex**
Pertaining to communication in which data can be sent and received at the same time.

dynamic partition
A partition created and activated on an 'as needed' basis that does not use fixed static allocations. After processing, the occupied space is released. Dynamic partitions are grouped by class, and jobs are scheduled by class. Contrast with *static partition*.

*** dynamic partition balancing**
A VSE facility that allows the user to specify that two or more or all partitions of the system should receive about the same amount of time on the processor.

EBCDIC
Extended binary-coded decimal interchange code.

ECKD™ device
Extended count-key-data device.

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.

Enterprise Systems Architecture/370
See *ESA/370* and *ESA/390*.

environmental record editing and printing (EREP) program
The IBM program that makes the data contained in the system recorder file available for further analysis.

EREP program
Environmental record editing and printing program.

error recovery procedures (ERP)
Procedures to help isolate and, where possible, to recover from errors in equipment.

ESA mode
An operation mode of the supervisor (generated with MODE=ESA) of the VSE system. Such a supervisor will run on Enterprise Systems Architecture processors (ESA/370 and ESA/390) and provides support for multiple virtual address spaces, the channel subsystem, and more than 16MB of real storage.

ESA/370
IBM Enterprise Systems Architecture/370. The extension to the IBM System/370 architecture which includes the advanced addressability feature that provides access registers.

ESA/390
IBM Enterprise Systems Architecture/390. The latest extension to the IBM System/370 architecture which includes the advanced addressability feature and advanced channel architecture.

*** escape**
To return to the original level of a user interface.

exit A routine, normally user-supplied, that receives control from the system when a certain event occurs (abnormal-end exit, for example).

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.

extent Continuous space on a disk or diskette occupied by or reserved for a particular file or VSAM data space.

extended binary-coded decimal interchange code (EBCDIC)
A coded character set consisting of 8-bit coded characters.

external storage
Storage that is not part of the processor.

fast service upgrade (FSU)
A service function of z/VSE for the

installation of a refresh release without regenerating control information such as library control tables.

FBA disk device
Fixed-block architecture disk device.

*** FCB** Forms control buffer.

fetch 1. To locate and load a quantity of data from storage. (A) 2. To bring a program phase into virtual storage from a sublibrary and pass control to this phase. 3. The name of the macro instruction (FETCH) used to accomplish 2. See also *loader*.

file A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

fixed-block architecture (FBA) disk device
A disk device that stores data in blocks of fixed size. These blocks are addressed by block number relative to the beginning of the file. Contrast with *CKD device*.

*** foreground partition**
A space 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.

*** forms control buffer (FCB)**
In the 3800 Printing Subsystem, a buffer for controlling the vertical format of printed output.

*** fragmentation (of storage)**
Inability to allocate unused sections (fragments) of storage in the real or virtual address range of virtual storage.

FSU Fast service upgrade.

GB Gigabyte.

generation
See *macro generation*.

*** GETVIS space**
Storage space within a partition or the shared virtual area, available for dynamic allocation to programs.

gigabyte (GB)
1024MB of storage (see MB). One gigabyte equals 1 073 741 824 bytes, which is 2 to the thirtieth power.

guest system
A data processing system that runs under control of another (host) system.

*** half-duplex**
In data communication, pertaining to transmission of data in only one direction at a time. Contrast with *duplex*.

hardcopy file
A system file on disk, used to log all lines of communication between the system and the operator at the system console, to be printed on request.

hard wait
The condition of a processor when all operations are suspended. System recovery from a hard wait is impossible without performing a new system startup.

hardware
Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation. (I) (A) Contrast with *software*.

help panel
A display of information provided by the system in response to a user's help request.

*** host system**
The controlling or highest level system in a data communication configuration.

ICA Integrated communication adapter.

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.

input/output control system (IOCS)
A group of IBM supplied routines that handle the transfer of data between main storage and auxiliary storage devices.

integrated communication adapter (ICA)
The part of a processor where multiple lines can be connected.

integrated console
In z/VSE, the service processor console available on ES/9000 processors that operates as the z/VSE system console. The integrated console is typically used

during IPL and for recovery purposes when no other console is available.

interactive
A characteristic of a program or system that alternately accepts input and then responds. An interactive system is conversational, that is, a continuous dialog exists between user and system. Contrast with *batch*.

Interactive Computing and Control Facility (VSE/ICCF)
An IBM program that serves as interface, on a time-slice basis, to authorized users of terminals linked to the system's processor.

interactive interface
A system facility which controls how different users see and work with the system by means of user profiles. When signing on, the interactive interface makes available those parts of the system authorized by the profile. The interactive interface has sets of selection- and data-entry panels through which users communicate with the system.

interactive partition
An area of virtual storage for the purpose of processing a job that was submitted interactively via VSE/ICCF.

interface
A shared boundary between two hardware or software units defined by common functional or physical characteristics. It might be a hardware component or a portion of storage or registers accessed by several computer programs.

*** intermediate storage**
Any storage device used to hold data temporarily before it is processed. See also *buffer storage*.

I/O (input/output)
See *input* and *output*.

IOCS Input/output control system.

IPL Initial program load.

*** irrecoverable error**
An error for which recovery is impossible without the use of recovery techniques external to the computer program or run. (T)

JCL Job control language.

JECL Job entry control language.

job accounting
A system function that lists how much every job step uses of the different system resources.

job control language (JCL)
A language that serves to prepare a job or each job step of a job to be run. Some of its functions are: to identify the job, to determine the I/O devices to be used, set switches for program use, log (or print) its own statements, and fetch the first phase of each job step.

job control statement
A particular statement of JCL.

job entry control language (JECL)
A control language that allows the programmer to specify how VSE/POWER should handle a job.

job step
One of a group of related programs complete with the JCL statements necessary for a particular run. Every job step is identified in the job stream by an EXEC statement under one JOB statement for the whole job.

job stream
The sequence of jobs as submitted to an operating system.

KB Kilobyte (KB equals 1024 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.

*** kilobyte (KB)**
1024 bytes of storage. One kilobyte equals 1024 bytes, which is 2 to the tenth power.

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 block**
A block of data stored in a sublibrary.

*** library directory**
The index that enables the system to locate a certain sublibrary of the accessed library.

*** library member**
The smallest unit of data to be stored in and retrieved from a sublibrary.

*** licensed program**
A separately priced program and its associated materials that bear an IBM copyright and are offered to customers under the terms and conditions of either the Agreement for IBM Licensed Programs (ALP) or the IBM Program License Agreement (PLA).

line Short for telecommunication line. Any physical medium such as a wire or microwave beam, that is used to transmit data. Synonymous with *transmission line*.

line printer
A device that prints a line of characters as a unit. (I) (A) Contrast with *character printer* or *page printer*.

link To connect items of data or portions of programs, for example linking of object programs by the linkage editor or linking of data items by pointers.

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-attached**
Pertaining to devices connected to a control unit by a data link. Synonymous with *remote*. Contrast with *channel-attached*.

link-edit
To create a loadable computer program by having the linkage editor process compiled (assembled) source programs.

loader A routine, commonly a computer program, that reads data or a program into processor storage. See also *relocating loader*.

*** lock file**
In a shared disk environment under VSE, a system file on disk used by the sharing systems to control their access to shared data.

*** logging**
The recording of data about specific events.

logical record
A user record, normally pertaining to a single subject and processed by data management as a unit. Contrast with *physical record* which may be larger or smaller.

logical unit name
In programming, a name used to represent the address of an input/ output unit.

logo A trademark or other art work that is associated with a firm or product. A logo often appears as the first screen of an interactive program.

LSR Local shared resources.

LU Logical unit.

macro (instruction)
1. In assembler programming, a user-invented assembler statement that causes the assembler to process a set of statements defined previously in the macro definition. 2. A sequence of VSE/ICCF commands defined to cause a

sequence of certain actions to be performed in response to one request.

macro definition
A set of statements and instructions that defines the name of, format of, and conditions for generating a sequence of assembler statements and machine instructions from a single source statement.

macro expansion
See *macro generation*.

macro generation
An assembler operation by which a macro instruction gets replaced in the program by the statements of its definition. It takes place before assembly. Synonymous with *macro expansion*.

* **main task**
The main program within a partition in a multiprogramming environment.

* **Maintain system history program (MSHP)**
A program used for automating and controlling various installation, tailoring, and service activities for a VSE system.

master console
In z/VSE, one or more consoles that receive all system messages, except for those that are directed to one particular console. Contrast with *user console* which receives only those messages that are specifically directed to it, for example messages issued from a job that was submitted with the request to echo its messages to that console. The operator of a master console can reply to all outstanding messages and enter all system commands.

* **MB** Megabyte (MB equals 1 048 576 bytes).

* **megabyte (MB)**
1024KB of storage (see KB). One megabyte equals 1 048 576 bytes, which is 2 to the twentieth power.

* **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.

* **microcode**
1. A code written using the instructions of a specific instruction set and implemented in a part of storage that is not program-addressable. 2. To design write, and test one or more micro instructions.

* **migrate**
To move to a changed operating environment, usually to a new release or version of a system.

* **module**
A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from, an assembler, a compiler, linkage editor, or executive routine. (A)

* **MSHP**
Maintain system history program.

* **multiprogramming**
1. A mode of operation that provides for interleaved execution of several programs by a single processor. (I) (A) 2. Pertaining to concurrent execution of several programs by a computer. (A)

multitasking
Concurrent running of one main task and one or several subtasks in the same partition.

MVS™ Multiple Virtual Storage. Implies MVS/390, MVS/XA, MVS/ESA, and the MVS element of the OS/390® operating system.

* **nest** To incorporate a structure or structures of some kind into a structure of the same kind. For example, to nest one loop (the nested loop) within another loop or to nest one subroutine (the nested subroutine) within another subroutine. (T)

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 program.

*** 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.

OCCF See VSE/OCCF.

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.

optional program
An IBM optional program that a user can install on VSE by way of available installation-assist support.

*** OS/390**
An IBM licensed program that not only includes and integrates functions previously provided by many IBM software products (including the MVS operating system) but also (a) is an open, secure operating system for the IBM S/390 family of enterprise servers, (b) complies with industry standards, (c) is Year 2000 ready and enabled for network computing and e-business, and (d) supports technology advances in networking server capability, parallel processing, and object-oriented programming.

page 1. In a virtual storage system, the unit of code or data or both which is transferred between processor storage and the PDS as needed for processing. 2. To transfer pages between processor storage and the page data set.

page data set (PDS)
One or more extents of disk storage in which pages are stored when they are not needed in processor storage.

page fault
A program interruption that occurs when a program page marked "not in processor storage" is referred to by an active page.

*** page fixing**
Marking a page so that it is held in processor storage until explicitly released. Until then, it cannot be paged out.

page frame
An area of processor storage that can contain a page.

page-in
The process of transferring a page from the PDS to processor storage.

page I/O
Page-in and page-out operations.

page-out
The process of transferring a page from processor storage to the PDS.

*** page pool**
The set of page frames available for paging virtual-mode programs.

panel The complete set of information shown in a single display on a terminal screen. Scrolling back and forth through panels is like turning manual pages. See also *selection panel* and *data entry panel*.

partition
A division of the virtual address area available for running programs. See also *dynamic partition, static partition*.

*** partition balancing, dynamic**
A VSE facility that allows the user to specify that two or more or all partitions of the system should receive about the same amount of time on the processor.

PDS Page data set.

*** phase**
The smallest unit of executable code that can be loaded into virtual storage.

*** physical record**
The amount of data transferred to or from auxiliary storage. Synonymous with *block*.

*** physical unit (PU)**
In SNA, the component that manages and monitors the resources of a node, such as attached links and adjacent link stations, as requested by an SSCP via an SSCP-SSCP session.

PL/I A programming language designed for use in a wide range of commercial and scientific computer applications.

PNET Programming support available with VSE/POWER; it provides for the transmission of selected jobs, operator commands, messages, and program output between the nodes of a network.

POWER
See *VSE/POWER*.

*** preventive service**
The installation of one or more PTFs on a VSE system to avoid the occurrence of anticipated problems.

*** primary library**
A VSE library owned and directly accessible by a certain terminal user.

Print Services Facility/VSE
An access method that provides support for the advanced function printers.

priority
A rank assigned to a partition or a task that determines its precedence in receiving system resources.

*** private library**
A user-owned library that is separate and distinct from the system library.

*** private partition**
Any of the system's partitions that are not defined as shared. See also *shared partition*.

procedure
See *cataloged procedure*.

*** processing**
The performance of logical operations and calculations on data, including the temporary retention of data in processor storage while this data is being operated upon.

processor
The hardware component that interprets and executes instructions. (I) (A)

processor storage
The storage contained in one or more processors and available for running machine instructions. Synonymous with *real storage*.

*** production library**
1. In a pre-generated operating system (or product), the program library that contains the object code for this system (or product). 2. A library that contains data needed for normal processing. Contrast with *test library*.

profile
A description of the characteristics of a user or a computer resource.

*** programmer logical unit**
A logical unit available primarily for user-written programs. See also *logical unit name*.

program product
See *licensed program*.

program service
The customer- or program-related IBM service of correcting design or implementation errors via APARs and PTFs.

program temporary fix (PTF)
A solution or by-pass of one or more problems documented in APARs. PTFs are distributed to IBM customers for preventive service to a current release of a program.

prompt
To issue messages to a terminal or console user, requesting information necessary to continue processing.

PSF/VSE
Print Services Facility/VSE.

PTF Program temporary fix.

PU Physical unit.

punch 1. To make holes in some data medium according to a signal code and thus save data on that medium. 2. A machine (output device) to punch 80-column punch cards.

*** punch card**
A card into which hole patterns can be punched; normally, it is characterized by 80 columns and 12 rows of punch positions.

*** 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.

queue record
A record in the queue file containing descriptive information about a job or job output.

*** random processing**
The treatment of data without respect to its location on disk storage, and in an arbitrary sequence governed by the input against which it is to be processed.

real address
The address of a location in processor storage.

*** real address area**
In VSE, the area of virtual storage where virtual addresses are equal to real addresses.

*** real address space**
The address space whose addresses map one to one to the addresses in processor storage.

real mode
In VSE, a processing mode in which a program may not be paged. Contrast with *virtual mode*.

real storage
See *processor storage*.

*** record**
A collection of related data or words, treated as a unit. See *logical record*, *physical record*.

recovery management support (RMS)
System routines that gather information about hardware failures and that initiate a retry of an operation that failed because of processor, I/O device, or channel errors.

*** reentrant**
The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by several tasks.

refresh release
An upgraded VSE system with the latest level of maintenance for a release.

relocatable module
In VSE, a library member of type object. It consists of one or or more control sections cataloged as one member.

relocating loader
A function that modifies addresses of a phase, if necessary, and loads the phase for running into the partition selected by the user.

*** remote job entry (RJE)**
Submission of jobs through an input unit that has access to a computer through a data link.

*** restore**
To write back on disk data that was previously written from disk to an intermediate storage medium such as tape.

RJE Remote job entry.

RJE workstation
Any workstation that is used for remote job submission and for the remote retrieval of output.

RMS Recovery management support.

*** routine**
Part of a program, or a sequence of instructions called by a program, that may have some general or frequent use. (I) (A)

*** routing**
The assignment of the path by which a message will reach its destination.

RPG II
A commercially oriented programming language suitable for writing application programs that meet common business data processing requirements.

* **run** 1. A performance of one or more jobs. (I) (A)

2. A performance of one or more programs. (I) (A)

3. To cause a program or job to be performed.

SAM Sequential access method.

SAM ESDS file
A SAM file managed in VSE/VSAM space, so it can be accessed by both SAM and VSE/VSAM macros.

schedule
To select a program or task for getting control over the processor.

SCSI (Small Computer System Interface)
A standard hardware interface that enables a variety of peripheral devices to communicate with one another.

SDL System directory list.

* **search chain**
The order in which chained sublibraries are searched for the retrieval of a certain library member of a specified type.

second-level directory
A table in the SVA containing the highest phase names found on the directory tracks of the system sublibrary.

security
See *access control*.

* **selection panel**
A displayed list of items from which a user can make a selection. Synonymous with *menu*.

sense Determine, on request or automatically, the status or the characteristics of a certain I/O or communication device.

sequential access
The serial retrieval of records in their entry sequence or serial storage of records with or without a premeditated order. Contrast with *direct access*.

sequential access method (SAM)
A data access method that writes to and reads from an I/O device record after record (or block after block). On request, the support performs device control operations such as line spacing or page ejects on a printer or skip a certain number of tape marks on a tape drive.

sequential file
A file in which records are processed in the order in which they are entered and stored.

* **shared spooling**
A function that permits the VSE/POWER account file, data file, and queue file to be shared among several computer systems with VSE/POWER.

* **shared virtual area (SVA)**
In VSE, a high address area that contains a list system directory list (SDL) of frequently used phases, resident programs shared between partitions, and an area for system support.

SIT (System Initialization Table)
A table in CICS that contains data used by the system initialization process. In particular, the SIT can identify (by suffix characters) the version of CICS system control programs and CICS tables that you have specified and that are to be loaded.

skeleton
A set of control statements and/or instructions that requires user-specific information to be inserted before it can be submitted for processing.

SNA System Networks Architecture.

SNA network
The part of a user-application network that conforms to the formats and protocols of SNA.

* **software**
Programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system.

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.

spanned record
A record that extends over several blocks.

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.

startup
The process of performing IPL of the operating system and of getting all subsystems and application programs ready for operation.

static partition
A partition, defined at IPL time and occupying a defined amount of virtual storage that remains constant. Contrast with *dynamic partition*.

storage dump
See *dump*.

storage fragmentation
Inability to allocate unused sections (fragments) of storage in the real or virtual address range of virtual storage.

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.

submit
A VSE/POWER function that passes a job to the system for processing.

*** subsystem**
A secondary or subordinate system or program, usually capable of operating independently of, or asynchronously with, the operating system.

subtask
A task that is initiated by the main task or by another subtask.

*** supervisor**
The part of a control program that coordinates the use of resources and maintains the flow of processor operations.

SVA Shared virtual area.

switched line
A telecommunication line in which the connection is established by dialing.

SYSRES
System residence volume.

*** system console**
A console, usually equipped with a keyboard and display screen for control and communication with the system.

system directory list (SDL)
A list containing directory entries of frequently-used phases and of all phases resident in the SVA. The list resides in the SVA.

*** system file**
In VSE, a file used by the operating system, for example, the hardcopy file, the recorder file, the page data set.

System Initialization Table (SIT)
A table in CICS that contains data used by the system initialization process. In particular, the SIT can identify (by suffix characters) the version of CICS system control programs and CICS tables that you have specified and that are to be loaded.

system logical unit
A logical unit available primarily for operating system use. See also *logical unit name*.

Systems Network Architecture (SNA)
The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

system recorder file
The file that is used to record hardware reliability data. Synonymous with *recorder file*.

system refresh release
See *refresh release*.

system residence volume (SYSRES)
The disk volume on which the system sublibrary is stored and from which the hardware retrieves the initial program load routine for system startup.

system sublibrary
The sublibrary that contains the operating

system. It is stored on the system residence volume (SYSRES).

*** tailor**
A process that defines or modifies the characteristics of the system.

*** task** The basic unit of synchronous program execution. A task competes with other tasks for system resources such as processing time and I/O channels.

task management
The functions of a control program that control the use, by tasks, of the processor and other resources (except for input/output devices).

TCP/IP
See Transmission Control Protocol/Internet Protocol.

TCT Terminal Control Table.

telecommunication
Transmission of data between computer systems and between such a system and remote devices.

telecommunication line
Any physical medium such as a wire or microwave beam, that is used to transmit data. Synonymous with *transmission line*.

terminal
A point in a system or network at which data can either enter or leave. Usually a display screen with a keyboard.

terminal access facility (TAF)
In the NetView® program, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of such subsystems simultaneously.

terminal control table (TCT)
A control block in which the system stores information about the characteristics and modes of operation of the terminals defined to the system.

time event scheduling support
In VSE/POWER, the time event scheduling support offers the possibility to schedule jobs for processing in a partition at a predefined time once or repetitively. The time event scheduling operands of the * $$ JOB statement are used to specify the desired scheduling time.

token A piece of information used as a unit.

*** telecommunication**
Transmission of data between computer systems and between such a system and remote devices.

telecommunication line
Any physical medium such as a wire or microwave beam, that is used to transmit data. Contrast with *data link*.

terminal
A point in a system or network at which data can either enter or leave. (A) Usually a display screen with a keyboard.

*** throughput**
1. A measure of the amount of work performed by a computer system over a given period of time, for example, jobs per day. (I) (A) 2. In data communication, the total traffic between stations per unit of time.

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.

*** track**
A circular path on the surface of a disk or diskette. Smallest unit of physical disk space.

track hold
A function that protects a track while it is being updated by one program from being accessed by another program.

*** transient area**
An area within the control program used to provide high-priority system services on demand.

transaction
In a batch or remote batch entry, a job or job step.

In CICS, 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.

*** Transmission Control Protocol (TCP)**
A communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol.

*** Transmission Control Protocol/Internet Protocol (TCP/IP)**
A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transmission line
Synonym for *telecommunication line*.

*** transmit**
To send data from one place for reception elsewhere. (A)

UCB Universal character set buffer.

*** UCS** Universal character set.

universal character set buffer (UCB)
A buffer to hold UCS information.

user console
In z/VSE, a console that receives only those system messages that are specifically directed to it. These are, for example, messages that are issued from a job that was submitted with the request to echo its messages to that console. Contrast with *master console*

*** 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. (A)

VAE Virtual addressability extension.

virtual address
An address that refers to a location in virtual storage. It is translated by the system to a processor storage address when the information stored at the virtual address is to be used.

virtual addressability extension (VAE)
A storage management support that gives the user of VSE multiple address spaces of virtual storage.

virtual address area
The virtual range of available program addresses.

virtual address space
In VSE, a subdivision of the virtual address area available to the user for the allocation of private (non-shared) partitions.

*** virtual I/O area (VIO)**
An extension of the page data set; used by the system as intermediate storage, primarily for control data.

*** virtual machine**
A functional simulation of a computer system and its associated devices.

*** virtual mode**
The operating mode of a program which may be paged.

virtual storage
Addressable space image for the user from which instructions and data are mapped into processor storage locations.

volume
A data carrier that is mounted and demounted as a unit, for example, a reel of tape or a disk pack. (I) Some disk units have no demountable packs. In that case, a volume is the portion available to one read/write mechanism.

volume ID
The volume serial number, which is a number in a volume label assigned when a volume is prepared for use by the system.

volume table of contents (VTOC)
A table on a disk volume that describes every file on it.

VSAM
See *VSE/VSAM*.

VSE (Virtual Storage Extended)
A system that consists of a basic operating system and any IBM-supplied and user-written programs required to meet the data processing needs of a user.

VSE and the hardware it controls form a complete computing system. Its current version is called z/VSE.

VSE/Advanced Functions
Part of VSE Central Functions, a base program of VSE.

VSE/DITTO for VSE (VSE/Data Interfile Transfer, Testing, and Operations Utility)
An IBM program that provides file-to-file services for disk, tape, and card devices.

*** VSE/Fast Copy**
A utility program for fast copy data operations from disk to disk and dump/restore operations via an intermediate dump file on magnetic tape or disk.

*** VSE/ICCF (VSE/Interactive Computing and Control Facility)**
An IBM program that serves as interface, on a time-slice basis 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/OCCF (Operator Communication Control Facility)
A z/VSE optional program that helps reduce operator interaction in the operation of a VSE-controlled installation and helps centralize data processing skills.

*** VSE/OLTEP (VSE/Online Test Executive Program)**
An IBM program for managing the online tests that are available for preventive service for I/O devices. Normally, only IBM service representatives use this program.

*** 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 Central Functions, a base program of z/VSE.

VSE/VSAM (VSE/Virtual Storage Access Method)
An IBM access method for direct or sequential processing of fixed and variable length records on disk devices.

*** VSE/VSAM managed space**
A user-defined space on disk placed under the control of VSE/VSAM.

VTAM (Virtual Telecommunications Access Method)
An IBM program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability; it supports application programs and subsystems (VSE/POWER, for example).

VTOC Volume table of contents.

wait state
The condition of a processor when all operations are suspended. System recovery from a hard wait is impossible without performing a new system startup. Synonym for *hard wait*.

Workstation File Transfer Support
Enables the exchange of data between IBM Personal Computers linked to a VSE host system where the data is kept in intermediate storage. PC users can retrieve that data and work with it independently of VSE.

z/VSE (z/Virtual Storage Extended)
The most advanced VSE system currently available.

Index

Readers' Comments — We'd Like to Hear from You

IBM z/VSE
Guide for Solving Problems
Version 5

Publication No. SC34-2615-00

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:
- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: +49-7031-163456
- Send your comments via email to: s390id@de.ibm.com
- Send a note from the web page: http://www.ibm.com/systems/z/os/zvse/

If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	Email address

Readers' Comments — We'd Like to Hear from You
SC34-2615-00

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Research & Development GmbH
Department 3282
Schoenaicher Strasse 220
71032 Boeblingen
Germany

Fold and Tape | **Please do not staple** | Fold and Tape

SC34-2615-00

Cut or Fold
Along Line

Product Number: 5609-ZV5

Printed in USA

SC34-2615-00