SC33-0070-5

Customer Information Control System (CICS/DOS/VS) Version 1 Release 6

Program Product

Installation and Operations Guide

Program Number 5746-XX3 (CICS/DOS/VS)

First Edition (June 1983)

This edition applies to Version 1 Release 6 (Version 1.6) of the IBM program product Customer Information Control System/Disk Operating System/Virtual Storage (CICS/DOS/VS), program number 5746-XX3.

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

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 program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the addresses given below. Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed either to:

International Business Machines Corporation,
Department 812HP,
1133 Westchester Avenue,
White Plains, New York 10604.

or to:

IBM United Kingdom Laboratories Limited,
Publications, Mail Point 095,
Hursley Park,
Winchester, Hampshire, England, SO21 2JN.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever.

About This Book

THE PURPOSE OF THIS BOOK

The purpose of this book is to guide the CICS system programmer through the task of installing a CICS system.

WHO THIS BOOK IS FOR

This book is intended for the system programmer who is responsible for setting up a CICS system. The system programmer may also have wider responsibilities, such as the installation of VSE, VTAM, and other programs that will run with CICS.

The system programmer will need to refer to other CICS books such as the CICS/VS Resource Definition Guide to find the required CICS resource definition macro instructions for use when creating a particular CICS system.

This book assumes that the system programmer has a working familiarity with VSE job control language.

THE SCOPE AND ORGANIZATION OF THIS BOOK

This book covers two major topics: **installation** and **operations**. The installation process will generally be carried out only once in creating your system, whereas the operations process may be repeated from time to time during the day-to-day running of your system.

1. Installation is described in Parts 1 and 2.
2. Operations are described in Parts 3, 4, and 5.

The contents of each part are as follows:

- Part 1. "Installing CICS" describes the installation process, from receipt of the distribution tape to the installation of an initial CICS system.

- Part 2. "Extending Your CICS System" describes how to extend a CICS system. This may be either a system that you have just created with the help of Part 1 of this book, or a system that existed earlier.

  A sample application package is supplied with CICS Version 1 Release 6, and Part 2 explains how to install this package and use it to check that your system is running satisfactorily.

  Part 2 also gives information on installing some additional CICS functions and associated programs:

  - Adding DL/I
  - Using the VSE shared virtual area
  - Adding CICS monitoring support
  - Installing resource definition online (RDO).

  For detailed information on intercommunication, see the CICS/VS Intercommunication Facilities Guide.

- Part 3. "Installing Control Tables and Application Programs" explains how to modify, assemble, link-edit, and install CICS tables, map sets, and application programs. Part 3 also describes which libraries and job control statements you will need.

- Part 4. "Setting Up CICS Data Sets" describes the various CICS data sets and the jobs necessary to define, initialize, use, and subsequently process them.

- Part 5. "Running the CICS System" describes how to start up, run, and shut down the CICS system. It provides a typical startup job stream, and shows the messages that you can expect at the system console during startup. It also shows how to enter startup override parameters, what you must do while CICS is running, and what to do if there is a system failure.

- Part 6. "Migrating and Applying Service to the CICS System" describes how to migrate from CICS Version 1 Release 5 to CICS Version 1 Release 6. Part 6 also describes what is meant by **service**, and how to apply it.

- Appendixes

  - Appendix A. Contents of the distribution libraries
  - Appendix B. CICS modules eligible for the SVA
  - Appendix C. Pregenerated modules
  - Appendix D. Sample tables and table entries
  - Appendix E. Aids and sample application programs
  - Appendix F. Sample job streams.

OTHER CICS PUBLICATIONS

This publication is one of a library of CICS publications providing information on all aspects of CICS. It replaces Part 1 and the appendixes of the CICS/VS System Programmer's Guide (DOS/VS). (Part 2 of the CICS/VS System Programmer's Guide (DOS/VS) is replaced by the CICS/VS Performance Guide, SC33-0134.)

The publications that deal with the design, installation, generation, and performance of a CICS system are:

1. CICS/VS System/Application Design Guide, SC33-0068: provides information to help the system design team decide which CICS facilities will best suit the needs of their organization.

2. CICS/VS Customization Guide, SC33-0131: gives details of the macro instructions that you need when generating the required CICS management programs. Read it in conjunction with the more general guidance given in this book.

3. CICS/VS Resource Definition Guide, SC33-0149: gives details of the macro instructions that you need when generating the required CICS control tables. Read it in conjunction with the more general guidance given in this book.

4. CICS/VS Application Programmer's Reference Manual (Command Level), SC33-0077: is for Assembler, COBOL and PL/I application programmers. It contains both introductory and reference material and assumes a knowledge of the General Information manual and of whichever programming language will be used.

5. CICS/VS Application Programmer's Reference Manual (RPG II), SC33-0085: is similar to the CICS/VS Application Programmer's Reference Manual (Command Level), but is issued as a separate publication because the command format for RPGII is very different from that of PL/I, COBOL, and assembler languages.

6. CICS/VS Application Programmer's Reference Manual (Macro Level), SC33-0079: is for application programmers who are maintaining programs written to the CICS macro-level interface in assembler, COBOL, or PL/I. It is also intended for system programmers who are writing user extensions to CICS in assembler.

7. CICS/VS Recovery and Restart Guide, SC33-0135: provides information on the recovery and restart aspects of a CICS system.

8. CICS/VS Intercommunication Facilities Guide, SC33-0133. You will need this book if the CICS system will be using intersystem communication (ISC) or multiregion operation (MRO).

9. CICS/VS Performance Guide, SC33-0134: describes how to adjust the optimization parameters to obtain the required performance from your CICS system.

The CICS/DOS/VS Program Directory, supplied with the distribution tape, contains the latest information on installing CICS. Use it in conjunction with this book. Where differences occur, regard the CICS/DOS/VS Program Directory as the latest level of information. It may contain information that is not in this book.

When running CICS, you will also need:

1. CICS/VS Operator's Guide, SC33-0160: tells terminal operators how to initiate CICS transactions, control various parts of the CICS system, switch messages, collect statistics, retrieve terminal error information, and use the processor console as a CICS terminal.

2. CICS/VS Messages and Codes, SC33-0156: contains all CICS messages, and is for use by the terminal operator, system programmer, and application programmer.

Publications on some other IBM products are listed in the Bibliography at the end of this book.

TERMINOLOGY

The formal title of the IBM program product described in this book is "Customer Information Control System/Disk Operating System/ Virtual Storage." For brevity, this program is called simply "CICS" throughout most of this book, although the formal abbreviation "CICS/DOS/VS" or "CICS/VS" is retained when quoting the titles of other CICS publications. Where it is necessary to refer to a particular release of CICS, CICS/DOS/VS Version 1 Release 6 is abbreviated to CICS 1.6, and CICS/DOS/VS Version 1 Release 5 to CICS 1.5.

Similarly, the name of the Virtual Storage Extended/Advanced Functions program product is abbreviated to VSE/AF or VSE.

In this book, the term VTAM refers to ACF/VTAM and ACF/VTAME. The term BTAM refers to BTAM-ES.

In the programming examples throughout this book, the dollar symbol ($) is used as a national currency symbol. In countries where the dollar is not the national currency, the local currency symbol should be used.

There is a glossary of terms and abbreviations at the back of this book.

Contents

Figures

Questionnaire CICS/DOS/VS Installation and Operations Guide

(CICS/VS Version 1 Release 6)

To help us produce books that meet your needs, please fill in this questionnaire. It would help us if you provide your name and address in case we need to clarify any of the points you raise. Please understand that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation whatever to you.

1. Please rate the book on the points shown below

The book is:						
accurate	1	2	3	4	5	inaccurate
readable	1	2	3	4	5	unreadable
well laid out	1	2	3	4	5	badly laid out
well organized	1	2	3	4	5	badly organized
easy to understand	1	2	3	4	5	incomprehensible
adequately illustrated	1	2	3	4	5	inadequately illustrated
has enough examples	1	2	3	4	5	has too few examples
And the book as a whole?						
excellent	1	2	3	4	5	poor

2. When using this book, did you find what you were looking for? ______
   What were you looking for? ______
   What led you to this book? ______
   Did you come straight to this book? ______

3. Which topics does the book handle well?

4. And which does it handle badly?

5. How could the book be improved?

6. How often do you use this book?
   Less than once a month? ☐ Monthly? ☐ Weekly ☐ Daily? ☐

7. What sort of work do you use CICS for? ______

8. How long have **you** been using CICS? ______ years/months

9. Have you any other comments to make?

Thank you for your time and effort. No postage stamp necessary if mailed in the USA. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to either address in the Edition Notice on the back of the title page.)

Note: *Staples can cause problems with automated mail sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.*

Questionnaire

Fold and Tape Fold and Tape Fold and Tape

No postage necessary if mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT 40 ARMONK, NEW YORK

Postage will be paid by addressee:

International Business Machines Corporation
Department 812H
1133 Westchester Avenue
White Plains, New York 10604

Fold and Tape Fold and Tape Fold and Tape

Name and address:

....

Job Title .... Company ....

Address ....

.... Zip ....

IBM®

CICS/DOS/VS Version 1 Release 6 Installation and Operations Guide Printed in U.S.A. SC33-0070-5

The IBM Customer Information Control System/Virtual Storage (CICS/VS) is a general purpose data-base/data-communication system. The term data-base/data-communication (DB/DC) is descriptive of the type of processing carried out by online systems as opposed to batch-processing systems. Generally, online systems involve the transmission of information from a remotely located terminal to a computer, the use of that information to access data maintained by the computer (referred to as a data base), and the transmission of processed information back to the terminal.

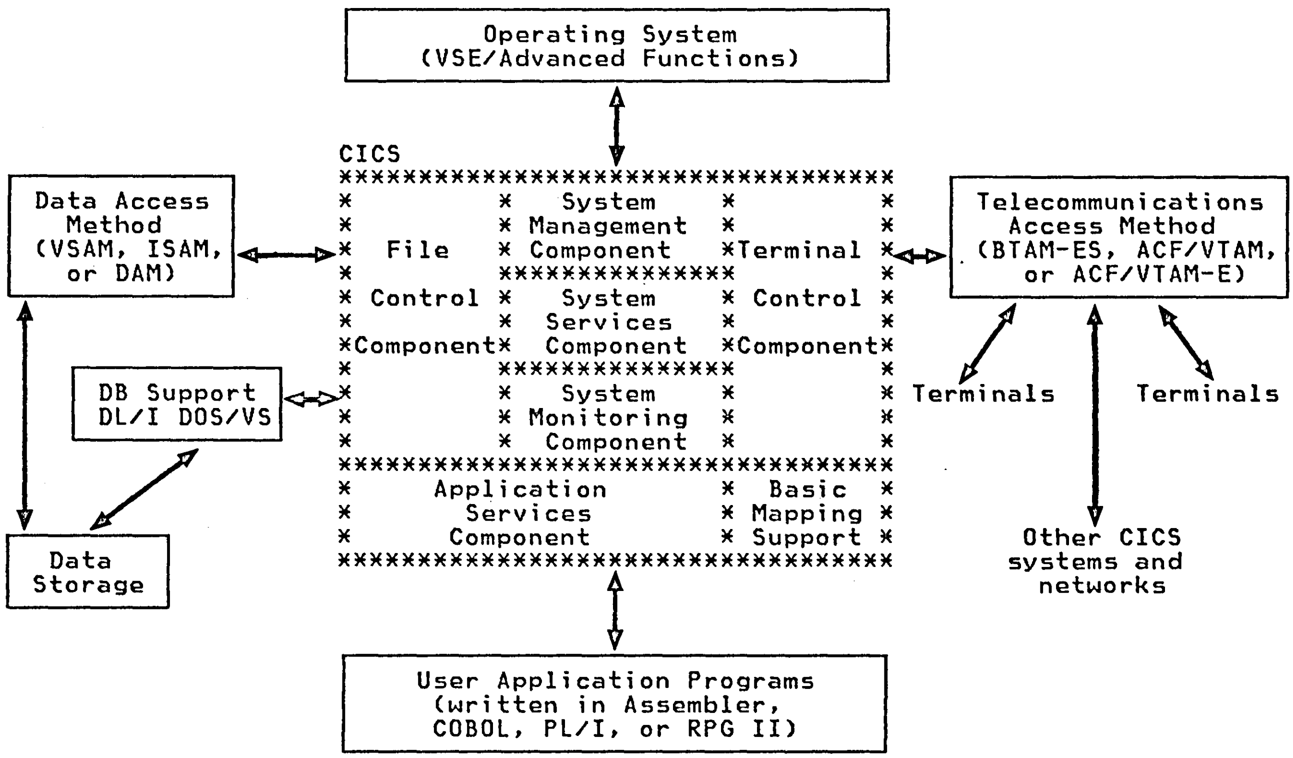


Figure 1. CICS and its Software Environment: the area outlined by asterisks represents CICS.

Part 1. Installing CICS

PART 1

Chapter 1.1. What is Installation?

This book is about the installation of a CICS system.

At its simplest, **installation** means mounting the distribution tape onto a tape drive and copying the information into libraries in your VSE system. This is a straightforward process that may take, for example, as little as 3 minutes 42 seconds of machine time on an IBM 4331 Processor. At the end of this process:

- You will have copied a core image library, a relocatable library, and a source statement library
- You will have merged the history file on the distribution tape into your system history file.

In short, you will have installed CICS, as it is supplied.

Of course, in order to use the CICS system to do productive work, you will need to do some other things. CICS provides a framework of management modules that is common to all CICS systems. But you must provide the application programs or application packages that will perform the processing that is unique to your requirements. You will need to:

- Install these application programs or packages into the VSE system.
- Define them to the CICS system with the aid of CICS control tables.
- Define other resources to CICS — for example, the precise configuration of terminals that is to be used when running the application programs.
- Define CICS data sets and user files that will enable the application programs to carry out their tasks.

All these activities must be completed in order to construct a total CICS system, suited to your own production environment.

The word "installation" is, therefore, often used for all the activities necessary to construct a working system. But these activities are not necessarily all carried out at the same time. You may wish to modify your system from time to time or to extend it by installing new application packages. You may wish to add new terminals to the system, or make it possible for CICS to communicate with other CICS systems. You will then have to redefine the control tables or re-create data sets or rebuild the job stream used to run the CICS system.

In this book, all these activities, which may be performed repeatedly during the life of a CICS system, are called **CICS operations**. In this way, the book distinguishes between installation, which is performed only once, and operations, which are the recurrent tasks necessary to ensure that a CICS system continues to satisfy the changing demands of its users.

The installation tasks — as described above — are described in Part 1 and in Chapters 2.5 through 2.10 of this book. The operations tasks are described in Parts 3, 4, and 5.

The distribution tape contains a set of sample application programs, together with appropriate tables that can be used to test them. For those users who are new to the installation of a total CICS system, the first few chapters of Part 2 have been written to provide a tutorial, step-by-step description of the procedures involved, using the sample application programs for illustration. The discussion, however, frequently goes further and aims to explain concepts not reflected in the samples. A user who has worked through Chapters 2.1, 2.2, 2.3, and 2.4 should be able not only to install a simple working system, but also to understand some of the underlying processes.

Some experienced users may choose to skip the tutorial chapters and turn to the reference material in the later chapters in Part 2 and in Parts 3, 4, and 5.

THE DISTRIBUTION TAPE (BASIC TAPE)

CICS is supplied on a single distribution tape known as the **basic tape**. Its contents are listed in Appendix A.

The basic tape contains all the material normally required to install and run a CICS system. The great majority of CICS modules are supplied only in pregenerated form, relieving users of the need to carry out system generation. A small number of CICS modules are also supplied in source form, so that users can, if necessary, generate them to suit their own requirements. For many users, this will be unnecessary, because the modules will meet the users' requirements in their standard, pregenerated form.

The basic tape thus contains:

- The **core image library**, containing the pregenerated modules and the sample application programs and maps
- The **source statement library**, containing the source code of those modules for which system generation is permitted in this release, as well as the macros used for building CICS control tables and for compiling CICS application programs
- The **relocatable library**, containing elements such as VSE logic modules, modules required when link-editing application programs, or items that are required if it is found necessary to generate alternative versions of modules in the core image library
- The MSHP **history file**, containing information about the CICS product.

THE OPTIONAL TAPE

In addition to the basic tape, it is possible to order separately the **optional tape**. The optional tape contains the source code for those CICS modules that are distributed only in the pregenerated form on the basic tape. The means to generate these modules is not supplied, and the tape is not required for product installation or operation. No service is applied to the contents of the optional tape, nor to any modules that may have been assembled from it. Most users will not need the optional tape.

THE INSTALLATION PROCESS

You copy the contents of the CICS distribution tape to your direct access storage device (DASD) using the Maintain System History Program (MSHP), which is an integral part of VSE. This procedure is commonly known as restoring the libraries on the tape.

During this MSHP installation job, the libraries on the distribution tape are copied to spaces you have defined on your DASD. The information in the history file on the distribution tape is merged into your system history file; it will subsequently be used during the servicing of CICS.

Chapter 1.2. What is Needed to Install and Run a CICS System

PART

PROGRAM DIRECTORY

Read the CICS/DOS/VS Program Directory first, before starting the installation process. The program directory is a source of additional or updated information about the installation process. The program directory should be used together with this manual but, where differences occur, the program directory will contain the latest level of information.

Consult the program directory for the release level of other program products against which CICS was assembled.

If you are an existing CICS user, you should also read at least "Chapter 6.1. Migrating to CICS 1.6" before you install CICS 1.6.

SERVICE REQUIRED

You should call your IBM program support representative to confirm that no other PTFs or APAR fixes are required — either to your system before CICS is installed — or to CICS before you can use it. Make sure that, if a fix is available, you install it only if it might affect your particular hardware or software.

DISTRIBUTION MATERIAL

The basic machine-readable material for CICS/DOS/VS is distributed on a single 9-track magnetic tape of either 1600 or 6250 bits per inch (bpi). The optional machine-readable material is also available at either 1600 bpi or 6250 bpi. The recording mode for all tapes is EBCDIC.

The basic and optional tapes are ordered separately.

The basic distribution tape supplied will contain CICS service up to the level currently available.

The optional tape, if ordered, will contain **no** service, and the source members on the tape will **not** be serviced.

Both the basic and optional tapes have been created using the VSE BACKUP utility program to dump complete libraries from disk to tape. All libraries are private libraries.

The contents of the basic tape must be restored (that is, copied from tape to disk) using the MSHP installation program (as described in the next chapter).

The libraries on the optional tape can be restored using the VSE RESTORE utility program. The LIST function of that program can be used to list all libraries on the optional tape, including file name, file identifier, and library sizes for a specific direct access storage device (DASD) type.

HARDWARE REQUIREMENTS

CICS/DOS/VS runs in a virtual storage environment on any IBM System/370, 303x series, 308x series, or 43xx series system configuration that supports the VSE operating system. Each system must have sufficient real storage to satisfy the combined requirements of CICS, the host operating system, appropriate access methods, batch requirements, and application programs that you may require.

The configuration must include sufficient I/O devices to support the requirements for system output, system residence, and system data sets. Sufficient direct-access storage must be available to satisfy your information storage requirements, and may consist of any direct-access facility supported by the system configuration and the programming system. See also the documentation for the host operating system for details of any features and engineering change levels that are required.

For a list of the terminals that can be used with CICS, see the CICS/VS General Information manual, GC33-0155.

To use the sample application programs supplied as part of the distribution material, you will need an IBM 3277 Display Station or a compatible terminal device.

Distribution and maintenance of CICS requires one 9-track tape drive.

DISK STORAGE SPACE FOR INSTALLATION

The information contained here refers to the CICS distribution tape at initial shipment. The distribution tape is periodically refreshed with the currently available service and so the libraries on the tape you have may have changed in size. The CICS/DOS/VS Program Directory contains information about the space requirements required for the libraries on the distribution tape received with that directory.

The basic distribution tapes contain a core image library, a relocatable library, and a source library. You have to specify your own data set identifiers for these libraries when they are copied to disk. For convenience, these libraries are referred to as CICS160.CIL, CICS160.RELO, and CICS160.SOURCE throughout this book, but these identifiers can be replaced by identifiers of your own choice. Similarly, this book refers to your own private core image library as USER.PRIVATE.CIL.

The size of the CICS data in the distributed libraries is approximately 27 million bytes.

Library	No. of Library Blocks	No. of Directory Entries
CICS160.CIL	2903	357
CICS160.RELO	4313	281
CICS160.SOURCE	124245	907

Figure 2. CICS Library Sizes (At Initial Shipment)

The amount of direct-access storage you will need to install the libraries depends on the DASD type. Figure 3 shows the space needed for the libraries of CICS 1.6 as defined for all IBM direct-access storage devices supported as library residence volumes. The figures in brackets refer to the space to be reserved for directory entries, out of the total allocation.

PART 1

Library	3330 Cyls	3340 Cyls*	3350 Cyls*	3310/3370 Blocks
CICS160.CIL	16(3)	39(4)	8(3)	6387(33)
CICS160.RELO	10(2)	25(2)	5(2)	3142(15)
CICS160.SOURCE	164(4)	439(5)	84(3)	43187(41)
Total	190	503	97	52716

Figure 3. CICS Space Requirements

Note: Entries marked with an asterisk have been calculated using the formulas in "Chapter 4. Storage Requirements" of the VSE/Advanced Functions System Generation manual. All sizes are approximate and include a 10% margin for servicing requirements. You should check the figures given above with the CICS/DOS/VS Program Directory — if there are any differences, **use the CICS/DOS/VS Program Directory figures.**

You should keep CICS in its own libraries and catalog your own library members in other private user libraries.

SOFTWARE REQUIREMENTS

OPERATING SYSTEM

CICS/DOS/VS Version 1 Release 6 operates under IBM System/370 VSE/AF Version 1 Release 3. No specific supervisor options are required for CICS.

DATA ACCESS METHODS

CICS can use Virtual Storage Access Method (VSAM), Indexed Sequential Access Method (ISAM), or Direct Access Method (DAM) to support user files and some CICS data sets.

VSAM

This release of CICS assumes that you have VSAM on your system. In addition, VSAM Access Method Services (AMS) are required for generating various CICS data sets and files.

ISAM

For users who have ISAM files, CICS/DOS/VS requires a special version of the ISAM logic module to provide support for in-line WAITs and the browse functions. The logic module required by CICS/DOS/VS for access to an ISAM file can be provided in either of two ways:

- By a separate assembly of the DFHISMOD macro instruction, following which the module is cataloged in the relocatable library, and then link-edited automatically with the CICS file control table when required.

- By assembling DFHISMOD in-line within the file control table, using the DFHFCT TYPE=LOGICMOD macro instruction.

A file control table may require up to four logic modules providing support (or specifically without support) for rotational position sensing (RPS), and the "prime data in main storage" option of ISAM record addition (CORDATA=YES option) — a total of four combinations of functional support. In general, even if the DFHFCT TYPE=LOGICMOD macro instruction is used, it is usually necessary to link-edit at least one additional logic module from the relocatable library. MNOTEs in the assembly listing for the file control table provide guidance on the generation of these additional logic modules. See the description of the file control table in the CICS/VS Resource Definition Guide for more details.

Four preassembled ISAM logic modules, with all options included, are provided in the relocatable library supplied on the CICS distribution tape (see "Relocatable Library" on page 333).

DAM

CICS supports DAM as a file access method. CICS 1.6 does not provide a pregenerated DFHSDAM module, because the access method module is now provided by VSE and is loaded into the VSE shared virtual area (SVA) at OPEN time.

TELECOMMUNICATION ACCESS METHODS

Terminals can be attached to CICS by means of Virtual Telecommunications Access Method (ACF/VTAM or ACF/VTAM-E) or Basic Telecommunications Access Method (BTAM-ES).

VTAM

Special considerations apply to CICS terminal control for the following reasons:

- The use of ACF/VTAM requires complementary definitions in CICS and VTAM.

- Some users who do **not** have ACF/VTAM installed in their system may have assembly errors if attempts are made to apply service to CICS-supplied modules containing VTAM macros.

  The suffixable terminal control modules are listed in Appendix C. Because these modules include VTAM macros, servicing of the pregenerated system presumes that you have ACF/VTAM installed.

  Note: The pregenerated system also includes the nonsuffixable modules DFHZCA, DFHZCC, DFHZCW, and DFHZCY, which all provide VTAM terminal support only. Servicing of these modules also presumes that you have ACF/VTAM installed.

CICS can communicate with different releases of VTAM. It can determine which level you are using, and hence what level of function is available. This means that you can upgrade CICS and VTAM at different times. Use of the release level indicator facility in ACF/VTAM Version 2 Release 1 enables CICS to determine, at execution time, whether certain facilities are available in VTAM, and modify the execution to take the necessary actions.

The pregenerated modules in CICS have been assembled against ACF/VTAM Version 2 Release 1. These pregenerated modules can be used with any release of ACF/VTAM from Version 1 upwards.

PART

When service is applied (see "Chapter 6.2. Applying Service to CICS" on page 327), CICS modules may need to be assembled against the installation libraries. This will cause some modules to be assembled with the lower level of function suitable for the VTAM release you have installed. If and when you upgrade to a later VTAM release, the executing system will give information about the modules that can be reassembled to obtain enhanced function.

You can assemble CICS for one version of VTAM, assemble its terminal control table for another version, and execute the whole system with either. CICS recognizes, at execution time, any mismatches in level, and selects a valid subset of function. This means, for example, that you can now develop and assemble all terminal control tables at a central location, on a system that contains the latest levels of program products. You can then ship the tables to remote nodes, which can use them unchanged.

Intersystem communication facilities through ACF/VTAM, other than for communication within the same processor, require the installation of the multisystem networking facility of ACF/VTAM. To be able to use the extended LU6 (LU6.2) for intersystem communication (ISC), you must install ACF/VTAM Version 2 Release 1. The CICS/VS Intercommunication Facilities Guide describes the ACF/VTAM options that are required when communicating with other CICS systems or with IMS/VS.

BTAM

If BTAM is used, a BTAM logic module (BTMOD) is required by the CICS system. It is link-edited automatically to the terminal control table if a BTAM line group or control unit is specified. This module must be cataloged in a relocatable library.

Seven different BTMODs are supplied on the distribution tape, and the source of these modules is included in the distribution source library. See Appendix A for a description of these modules. If you apply service to BTMOD, you should reassemble the BTMOD that you are using and re-link-edit the terminal control table that you are using.

DATA LANGUAGE/I (DL/I)

To use the optional CICS/DOS/VS interface to Data Language/I (DL/I), you must previously install the DL/I DOS/VS (5746-XX1) program product.

The pregenerated modules in CICS 1.6 were assembled against DL/I DOS/VS Version 1 Release 6. If you require a different level of DL/I, see "Chapter 2.7. Adding DL/I Support" on page 125.

COMPILERS

Depending on the language to be used for application programs, you will need one or more of the following compilers:

- Full ANS COBOL Version 3 Compiler, 5736-CB2, and Full ANS COBOL Library, 5736-LM2 (see Note 1)
  - CICS does not support application programs using the optimizing option of 5736-CB2.
- DOS/VS COBOL Compiler and Library, 5746-CB1, or Library only, 5746-LM4 (see Note 1)
  - CICS supports application programs compiled using the optimizing option of 5746-CB1, provided that the application programs include a new compiler control statement at appropriate places within the source code.
- ANS COBOL Subset, 5736-CB1 (see Note 1)
- Full ANS COBOL Version 2, 360N-CB-482 (see Note 1)
- DOS PL/I Optimizing Compiler and Libraries, 5736-PL3 (see Note 2) or
  - DOS PL/I Optimizing Compiler, 5736-PL1 (see Note 2)
  - DOS PL/I Resident Library, 5736-LM4 (see Note 2)
  - DOS PL/I Transient Library, 5736-LM5 (see Note 2)
- DOS/VS RPG II Compiler, 5746-RG1

Notes:

1. ANS COBOL Version 3 or DOS/VS COBOL must be installed to use the command-level interface within COBOL programs.

2. DOS PL/I Version 1 Release 5.1 or later must be installed to use the command-level interface within PL/I programs.

   You must ensure that the DFHSAP module, provided by PL/I, is available when running CICS with PL/I. In addition, DFHPL1I must be available when link-editing command-level application programs written in PL/I. Versions of these modules are no longer provided by CICS.

OTHER SOFTWARE COMPONENTS

Appendix C in the CICS/VS General Information manual, GC33-0155, contains a list of other software products that you may wish to run with CICS.

RUNNING UNDER VM

CICS operates in a VM/370 system under control of VSE, subject to the following considerations:

1. CICS operating in a virtual machine has the same requirements as CICS operating in a real machine. Other software components (for example, access methods, compilers, and the release of VSE under which CICS runs) must be valid for that release of CICS.

2. The minimum hardware requirements of CICS operating in a virtual machine are the same as those for CICS running in a real machine and should be considered as additional to the minimum requirements for VM/370 itself and any other virtual machines within the VM/370 environment.

PUBLICATIONS

If you are new to CICS, you should read most of this book. Start with Chapter 1.1, which will give you an overview of the installation process. If you are an existing CICS user, you should at least read "Chapter 6.1. Migrating to CICS 1.6" on page 323 before you install CICS 1.6.

CICS PUBLICATIONS

The other CICS publications, and their relationship to this book, are described in "About this Book" on page iii.

VSE PUBLICATIONS

You may need some of these VSE publications while you install CICS:

- VSE/Advanced Functions System Control Statements, SC33-6095
- VSE/Advanced Functions Maintain System History Program User's Guide, SC33-6072
- VSE/Advanced Functions Messages, SC33-6098
- VSE/Advanced Functions Operating Procedures, SC33-6097.

OTHER PUBLICATIONS

Publications describing some other products are listed in the Bibliography at the back of this book.

Chapter 1.3. Installing the CICS Distribution Tape

FORMAT OF THE DISTRIBUTION TAPE

The distribution tape is available at either 1600 bpi or 6250 bpi. In both cases, the tape contains seven files:

1. Dummy file
2. Dummy file
3. Basic library file
   a. Basic core image library
   b. Basic relocatable library
   c. Basic source statement library
4. Dummy file
5. MSHP history file
6. Dummy file
7. Dummy file

THE MSHP HISTORY FILE

The CICS distribution libraries are installed under control of the Maintain System History Program (MSHP), which is an integral part of VSE. MSHP is the means used to install not only CICS but also the VSE system itself and other components and features such as VSAM, VTAM, and so on.

MSHP is also designed to assist in the application of IBM-supplied software corrections, and to maintain records of changes made to the system; MSHP automates certain checking requirements (see "Chapter 6.2. Applying Service to CICS" on page 327).

MSHP records installation and service activities in an MSHP **system history file** in order to reflect the current status of your operational system. MSHP can subsequently use the history file to compare an operational system's history with IBM-supplied information on requirements for installing additional programming support or applying corrections to the installed programming support.

You will already have a system history file in existence for your operational system. During the CICS installation program, the MSHP history file on the distribution tape is merged with this system history file to create an updated system history file. However, MSHP can keep information about only one release of a program product in your system history file. If you have CICS 1.5 installed and you install CICS 1.6, the service history of CICS 1.5 will be overwritten by the new CICS release and no maintenance can then be applied to CICS 1.5.

If you wish to maintain both CICS 1.5 and CICS 1.6 simultaneously, you will have to create a separate system history file for the CICS 1.6 system (see below under "Existing CICS Users Only" on page 16).

SPACE PLANNING

You must decide where to allocate space for the CICS libraries. The amount of space required has been given in Figure 3 in Chapter 1.2. You must first identify, on the DASD volume (or volumes) that is to hold the libraries, suitable areas of free space. In order to do so, you must list the volume table of contents (VTOC) of the DASD. As an example, the following job can be used to list the contents of the DASD volume 111111, and to send the listing to SYSLST:

```
// JOB LVTOC LIST VOLUME TABLE OF CONTENTS
// ASSGN SYS004,DISK,TEMP,VOL=111111,SHR
// ASSGN SYS005,SYSLST
// EXEC LVTOC
/&
```

Figure 4. Listing the Contents of a DASD Volume

The DASD space you identify is used in the MSHP installation job described later in this chapter.

INSTALLATION JOB STREAM

EXISTING CICS USERS ONLY

If you have an existing CICS 1.5 system installed, and you wish to keep this system operational while you install and test out the CICS 1.6 system, you will have to create a new system history file, to hold information relating to CICS 1.6 alone. At a later stage, when you are ready to move to exclusive use of the CICS 1.6 system, you will merge the information in the CICS 1.6 system history file with the system history file of your operational system.

The steps required are as follows:

1. Create a new, empty system history file.
2. Using this new system history file, install CICS 1.6 as described later in this chapter.
3. Use this new system history file when applying service to CICS 1.6 but, for all other maintenance, use your old system history file.
4. Test CICS 1.6 and establish a new production CICS system.
5. When you no longer require CICS 1.5, merge your new history file information into your old system history file.
6. Delete the history file with only CICS 1.6 information in it.

Creating a New (Empty) System History File

The suggested amount of space required for the system history file on various device types is as follows:

Device Type	Size
3330	38 tracks
3340	96 tracks
3350	30 tracks
3310/3370	900 blocks

You should now run the MSHP job that creates the new system history file. Be careful that you do not overwrite your existing history file. The following job is an example of the required job control statements:

```
// JOB CREATE NEW SYSTEM HISTORY FILE

// ASSGN SYS019,DISK,TEMP,VOL=111111,SHR
// EXEC MSHP
 CREATE HIST SYS
 DEFINE HIST SYS EXTENT=num1:num2 UNIT=SYS019 ID='cics160.sys.history'
/*
/&
```

In the above job, the new system history file has been created on volume 111111 with file identifier cics160.sys.history. The value of num1 is the starting address (relative to zero) of the file, num2 is the size of the file, both numbers being expressed in terms of tracks or blocks, depending on DASD type.

NEW CICS USERS

If you are a new user of CICS and you do not have any other CICS system installed on your VSE system, you will need to refer to the MSHP system history file that already exists as part of your VSE system. This system history file may be defined in your standard labels; if not, check your installation for its name and location. You must include the file identifier and the extent information of the system history file in the job stream to install CICS.

ALL USERS

The installation job below defines an auxiliary history file in addition to the system history file (you do not have to run a job to create the auxiliary history file explicitly). The auxiliary history file is needed during the installation process only. The contents of the history file on the distribution tape are copied to the auxiliary history file, and any prerequisite checking is done at this stage before the contents are merged into the system history file.

You need to specify the size of the auxiliary history file to be the same as that of the system history file.

The three CICS libraries (core image library, relocatable library, and source statement library) are installed onto DASD defined as logical units SYS007, SYS008, and SYS009 respectively.

Before you install any program product, you should create a backup copy of your system, including the system history file.

Figure 5 shows the installation job stream. Read the notes carefully and then modify the job stream to suit your system.

```
// JOB INSTALL COMPONENT FROM TAPE

// ASSGN SYS019,DISK,TEMP,VOL=111111,SHR   CICS system history file   1
// ASSGN SYS002,DISK,TEMP,VOL=111111,SHR   Auxiliary history file        2
// ASSGN SYS006,cuu                        Distribution tape                3
// ASSGN SYS007,DISK,TEMP,VOL=111111,SHR   Private core image library
// ASSGN SYS008,DISK,TEMP,VOL=111111,SHR   Private relocatable library
// ASSGN SYS009,DISK,TEMP,VOL=111111,SHR   Private source statement library
// MTC  REW,SYS006

// OPTION CATAL
// EXEC MSHP

INSTALL COMPONENT FROMTAPE ATTACH                      4

DEFINE CLIB PRIVATE EXTENT=num1:num2 DIRECTORY=num3 ID='cics160.cil'       5
DEFINE RLIB PRIVATE EXTENT=num1:num2 DIRECTORY=num3 -
               ID='cics160.relo'                              6
DEFINE SLIB PRIVATE EXTENT=num1:num2 DIRECTORY=num3 ID='cics160.source'
DEFINE HIST AUX EXTENT=num1:num2 UNIT=SYS002 ID='cics160.aux.history'
DEFINE HIST SYS EXTENT=num1:num2 UNIT=SYS019 ID='cics160.sys.history'     1, 7
/*
// MTC  RUN,SYS006
/&
```

Figure 5. Installing into New Libraries

Notes:

1 If you are an existing CICS user and you have created an empty history file, use that new history file in this job. Otherwise use your system history file.

2 In this sample installation job stream, all the CICS libraries are installed onto the same pack; this pack also contains enough space for the auxiliary history file. Each library and the auxiliary history file could be on different packs, however. Replace the volume identifiers in this example with your own names. If you do not have unique volume identifiers, you will have to change all assign statements to:

```
// ASSGN  SYS00x,cuu
```

where **cuu** is the address of the DASD.

3 **cuu** is the address of the tape drive where the distribution tape is to be mounted.

4 The ATTACH parameter will install the distribution libraries into **new** libraries on your DASD. If libraries already exist on your DASD in the space allocated in the DEFINE statements for the new CICS libraries, you will get message

```
4444D OVERLAP ON UNEXPIRED FILE
```

and you must reply DELETE to overwrite the existing libraries. If you did not expect to receive this message, check your space calculations.

5 **num1** is the starting address for the library or history file. For CKD devices, it specifies the track number and should be the start of a cylinder. For FBA devices, it must be the first FBA block of the library or history file.

num2 is the size of the library or history file as expressed in either tracks or FBA blocks (see Figure 3). For CKD devices, this number must correspond to an integral number of cylinders.

num3 is the number of tracks or FBA blocks to be allocated for the library directory (see Figure 3).

ID='datasetname' The data set names specified here are the ones you wish to specify for the libraries on your DASD. Change these to suit your naming conventions.

6 If any MSHP control statement extends beyond column 72, you should split it before a keyword, using a dash (-) preceded and followed by at least one blank as a continuation character, and continue the statement on the next line.

7 If your system history file is defined in your standard labels, you may leave out this statement.

After you have modified the job stream, mount the distribution tape and run the job.

OUTPUT FROM THE SAMPLE INSTALLATION JOB STREAM

Figure 6 shows sample output from running the installation job stream. The job took 3 minutes 42 seconds to run on an IBM 4331 Processor with an IBM 3310 Direct Access Storage Device and IBM 3420 tapes.

```
// JOB INSTALL COMPONENT FROM TAPE          DATE 01/11/83,CLOCK 15/25/44
// ASSGN SYS019,DISK,VOL=111111,SHR          CICS system history file
1T20I  SYS019 HAS BEEN ASSIGNED TO X'241'
// ASSGN SYS002,DISK,VOL=111111,SHR          Auxiliary history file
1T20I  SYS002 HAS BEEN ASSIGNED TO X'241'
// ASSGN SYS006,181                          Distribution tape
// ASSGN SYS007,DISK,VOL=111111,SHR          Private core image library
1T20I  SYS007 HAS BEEN ASSIGNED TO X'241'
// ASSGN SYS008,DISK,VOL=111111,SHR          Private relocatable library
1T20I  SYS008 HAS BEEN ASSIGNED TO X'241'
// ASSGN SYS009,DISK,VOL=111111,SHR          Private source statement library
1T20I  SYS009 HAS BEEN ASSIGNED TO X'241'
// MTC  REW,SYS006
// OPTION CATAL
// EXEC MSHP


INSTALL COMPONENT FROMTAPE ATTACH
DEFINE CLIB PRIVATE EXTENT=1000:6400 DIRECTORY=33 ID='CICS160.CIL'
DEFINE RLIB PRIVATE EXTENT=7400:3200 DIRECTORY=15 -
               ID='CICS160.RELO'
DEFINE SLIB PRIVATE EXTENT=10600:43200 DIRECTORY=41 ID='CICS160.SOURCE'
DEFINE HIST AUX EXTENT=53800:900 UNIT=SYS002 ID='CICS160.AUX.HISTORY'
DEFINE HIST SYS EXTENT=100:900 UNIT=SYS019 ID='CICS160.SYS.HISTORY'
M038I INSTALL IN PROGRESS
M060I RESTORE  PROGRAM NOW INVOKED

 ALLOC PC=0000006400(0000000033),                                      X
       PR=0000003200(0000000015),                                      X
       PS=0000043200(0000000041)
8R35I  RESTORE OF PC LIBRARY IN PROGRESS
8R36I  RESTORE HAS BEEN SUCCESSFUL
8R35I  RESTORE OF PR LIBRARY IN PROGRESS
8R36I  RESTORE HAS BEEN SUCCESSFUL
8R35I  RESTORE OF PS LIBRARY IN PROGRESS
8R36I  RESTORE HAS BEEN SUCCESSFUL
8R38I  ***  RESTORE COMPLETE  ***

M300I MERGING HISTORY INFORMATION FOR COMPONENT 5746XX300 COMPLETED
M308I MERGING HISTORY INFORMATION FOR FEATURE H38         COMPLETED
M041I FUNCTION COMPLETED
// MTC  RUN,SYS006
EOJ INSTALL                    DATE 01/11/83,CLOCK 15/29/26,DURATION 00/03/42
```

Figure 6. Sample Installation Jobstream Output Using an IBM 3310 Direct Access Storage Device

WHAT TO DO IF THE INSTALLATION JOB FAILS

If the installation job fails, examine the output from the installation job. Possible errors are:

1. Library extents not big enough.

   Message:

   - 8R35I RESTORE OF xx LIBRARY IN PROGRESS

     followed by

   - 8R40I LIBRARY IS FULL

     and the job will cancel.

   Action: Recalculate sizes (see Figure 2), modify the job in Figure 5, and rerun the job.

2. Library extents overlapping.

   Message: 4444D OVERLAP ON UNEXPIRED FILE

   Action: Recalculate sizes (see Figure 2), modify the job in Figure 5, and rerun the job.

If you receive any other error messages, look them up in the VSE/Advanced Functions Messages manual, SC33-6098. Take the appropriate action, and then rerun the installation job. If the installation job fails again for the same reason, call your IBM support representative.

AFTER THE INSTALLATION JOB

After the MSHP installation job has been completed, the contents of the CICS basic libraries will have been copied from the distribution tape to your DASD. You will have a CICS pregenerated system containing management modules, sample control tables, and sample application programs.

Whenever you run CICS or any jobs that require access to CICS facilities, you will need to make reference to the location of the CICS libraries by means of the DLBL and EXTENT job control statements. The DLBL statement contains label information for a file on DASD; the EXTENT statement defines the area occupied by that file.

These labels can be included in the startup job itself, but more commonly are written into the system standard subarea where they will be accessible to all partitions, or into the partition standard subarea only.

Standard labels should, in fact, be used for all CICS libraries and files. The labels should be added to existing system or partition standard label information. Standard labels are used in all example jobs shown in the rest of this chapter.

It is possible to add standard label information to a running system. The job to do this must run in the background partition when system labels are being added, or in the relevant partition when partition labels are being added.

The following job can be used to add standard label information for the CICS libraries created by means of the MSHP installation job described in the last chapter. You must ensure that the file identifiers and the volume identifiers match those in the MSHP job.

```
// JOB LABELS
// OPTION STDLABEL=DEL       (or // OPTION PARSTD=DEL)
C160PCL
C160PRL
C160PSL
/*
// OPTION STDLABEL=ADD       (or // OPTION PARSTD=ADD)
// DLBL C160PCL,'CICS160.CIL'
// EXTENT ,111111
// DLBL C160PRL,'CICS160.RELO'
// EXTENT ,111111
// DLBL C160PSL,'CICS160.SOURCE'
// EXTENT ,111111
/*
/&
```

Figure 7. Adding Standard Label Information

The label information is stored permanently into the system or partition standard subarea without overwriting the information that already exists in that subarea.

After you have added the standard label information, it is available to the following job, which you can run to list the members of the CICS libraries you have installed:

```
// JOB DSERV DISPLAY ALL CICS FILES
// LIBDEF CL,FROM=C160PCL
// LIBDEF RL,FROM=C160PRL
// LIBDEF SL,FROM=C160PSL
//   EXEC DSERV
 DSPLYS CD,RD,SD
/*
/&
```

See the Appendixes at the back of this book for more details on the pregenerated CICS system:

- Appendix A summarizes the contents of the distribution libraries.
- Appendix B lists the CICS modules eligible for the SVA.
- Appendix C describes the pregenerated modules that can be selected to specify a particular CICS system.
- Appendix D describes the sample tables.
- Appendix E describes the sample applications.

If you are an existing CICS user, you will already have your own tables and startup job streams, as well as applications. If you wish to use these with CICS 1.6, read "Chapter 6.1. Migrating to CICS 1.6," which describes the changes you must make to your tables and job streams before starting up CICS 1.6.

If you are a new CICS user or an inexperienced user, read the next chapter of this book, which explains how to start up a very simple CICS system. Following on from that, Part 2 of this book describes how to extend this initial CICS system and run the CICS sample applications.

Chapter 1.4. Initial CICS Startup

THE INITIAL CICS SYSTEM

CICS consists of a set of control and service programs. The distributed libraries contain at least one version of every CICS program needed to run a CICS system. If you use these pregenerated modules, you avoid most of the system generation processes that would otherwise have to be carried out by yourself.

CICS gives you a wide choice of options of the facilities provided. For example, if you do not use ISAM, CICS gives you the opportunity not to include the CICS ISAM support, thus reducing the size of the CICS system during execution.

CICS gives this ability to select the functions you require by means of suffixable modules. Each CICS module name consists of the following parts:

- The first three letters: DFH
- Three characters describing the function of the module, for example: FCP (file control program)
- A 2-character suffix that distinguishes between different versions of certain CICS modules, thereby allowing more than one version of a module to exist. For example, DFHFCP5$ contains VSAM and ISAM support. DFHFCP3$ contains VSAM but no ISAM support.

CICS CONTROL TABLES

CICS is a table-driven system. Before CICS can be run, the resources it controls for a particular run must be defined in CICS control tables, which are prepared separately.

For example, user data sets are defined in the file control table, and terminals are defined in the terminal control table. These are suffixed like the CICS modules, allowing you to have more than one table defined in the system, from which you can choose particular tables for a particular startup. You might, for example, have one FCT, DFHFCTDA, for the day shift and use a different one, DFHFCTNT, for the night shift.

The list below gives the names of all CICS tables.

Table	Name
Application load table	DFHALTxx
Destination control table	DFHDCTxx
File control table	DFHFCTxx
Journal control table	DFHJCTxx
Monitor control table	DFHMCTxx
Nucleus load table	DFHNLTxx
Program control table	DFHPCTxx
Program list table	DFHPLTxx
Processing program table	DFHPPTxx
System initialization table	DFHSITxx
Sign-on table	DFHSNT
System recovery table	DFHSRTxx
Terminal control table	DFHTCTxx
Terminal list table	DFHTLTxx
Temporary storage table	DFHTSTxx
Transaction list table	DFHXLTxx

Versions of certain of these control tables are provided as samples in the pregenerated system (see Appendix D).

All these tables can be created in more than one version. Apart from the sign-on table, the different versions are distinguished by specifying a 2-character suffix when the tables are defined. (The sign-on table cannot be generated with a suffix; if you have more than one version, you must ensure that the one you are using for the CICS run is named DFHSNT.)

The 2-character table suffix can be used in the system initialization table (SIT) or as an override parameter during system initialization to identify the version to be used for a particular CICS run.

Certain CICS tables must be available before CICS can be run:

- The system initialization table
- The program control table
- The processing program table
- The terminal control table.

THE SYSTEM INITIALIZATION TABLE

The system initialization table defines which CICS tables, modules and other system parameters are to be used for a particular CICS system.

The distribution tape contains — in both source form and pregenerated module form — a sample system initialization table with the suffix 1$. A listing of the source of this table is also included in Appendix D. (See the CICS/VS Resource Definition Guide for detailed information on each parameter in the sample system initialization table.)

In this chapter, DFHSIT1$ will be used in the job to start up an initial CICS system.

PROGRAM CONTROL TABLE AND PROCESSING PROGRAM TABLE

CICS is a transaction-processing system. To communicate with CICS, you enter transaction names. These transaction names must have been defined in the program control table (PCT). The PCT contains the information to be used by CICS for identifying and initializing a transaction. Information includes the transaction priority and the security key, as well as the name of the program to which CICS must pass control.

The program name referenced by a transaction must also be defined in the processing program table (PPT).

A sample PCT and PPT (DFHPCT1$ and DFHPPT1$) are supplied with the distribution libraries (they are listed in Appendix D).

TERMINAL CONTROL TABLES

Pregenerated terminal control tables (TCTs) are supplied. One (DFHTCT1$) has support for a system console and a card reader/line printer terminal only. DFHTCT1$ will be used in this chapter. Appendix D lists the other TCTs provided on the distribution tape. "Chapter 2.3. Sample Application Package and CICS Tables," will explain how to extend this initial TCT to include BTAM and VTAM terminals.

STARTING UP CICS

Using the tables provided, you can now start up a CICS system. In order to make this initial startup as easy as possible, we have defined a CICS system without system data sets. This system will have very little function, but it can be used to enter the CICS CEMT transaction.

Figure 8 shows the job stream needed to start up the initial CICS system.

```
// JOB CICS TO START UP AN INITIAL CICS SYSTEM
// LIBDEF CL,SEARCH=C160PCL    (see note 1)
// EXEC DFHSIP,SIZE=1200K,PARM='CONSOLE'  (see notes 2 and 3)
/*
/&
```

Figure 8. Starting Up an Initial CICS System

Notes:

1. You must define, by means of LIBDEF statements, the sequence in which the libraries are to be searched for phases, modules, and so on. Two library concatenation chains can be built for each library type, one permanently for a partition and one temporarily for the duration of the CICS job only. If you have both a temporary and a permanent search chain, the temporary chain is searched before the permanent chain.

   Thus, to include the CICS core image library in the search chain for the duration of the CICS job, you should include the LIBDEF statement shown in the above startup job.

2. Sufficient virtual storage should be specified for the CICS partition to allow all user applications, CICS programs, and CICS tables to be resident in virtual storage. The CICS/VS Performance Guide should be used to calculate a figure for your requirements. For this initial CICS system, a partition size of 1500K should be specified using the ALLOC macro of VSE, with a minimum ALLOCR value of 100K. These sizes are more than sufficient to run a simple CICS system.

   The SIZE parameter specified on the EXEC statement is the size of that part of the virtual partition which will be directly available to the program. The remainder of the partition may be used as additional storage (GETVIS area) for other modules or data required by the program in that partition. The system always allocates a minimum partition GETVIS area of 48K bytes but certain programs, for example VSAM, require more than the minimum value.

3. A PARM option of 'CONSOLE' is included here to enable system initialization overrides to be entered at the system console.

   The pregenerated system initialization table (DFHSIT1$) will be used to start up an initial CICS system. In response to the prompt to specify alternative override parameters, enter:

   ```
   sit=1$,$end
   ```

   If, however, you are using asynchronous operator communication support, you must also specify the reply identifier, for example:

   ```
   2 sit=1$,$end
   ```

   where '2' is the reply identifier.

RUNNING THE SAMPLE CICS JOB STREAM

Figure 9 shows the messages displayed at the system console while the job to start up CICS is being run. The job is running in the F2 partition. When the CICS system initialization has been completed (in the F2 partition in the present example), the final initialization message is displayed and, at this point, CICS is ready to be used for productive work.

```
 F2 002 // JOB CICS
                                                            (see note 1)
 DATE 03/11/83,CLOCK 22/49/24
 F2 002 // EXEC DFHSIP,SIZE=1200K,PARM='CONSOLE'            (see note 2)
 F2 002 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS.
 F2 002 DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
*F2 002           AND ENTER $END WHEN COMPLETED
 2 sit=1$,$end                                              (see note 3)
 F2 002 SIT=1$,$END
 F2 002 DFH1501 - DFHSIT1$ IS BEING LOADED
 F2 002 DFH1500 - LOADING CICS NUCLEUS
 F2 002 DFH1500 - CICS START-UP IS COLD
 F2 002 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
 F2 002 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
 F2 002 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS   820K
 F2 002 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE            (see note 4)
 F2 002 DFH1500 - STXIT MACROS ARE BEING ISSUED
 F2 002 DFH1500 - PROCESSING RESIDENT PROGRAMS
 F2 002 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS   744K
 F2 002 DFH1500 - CONTROL IS BEING GIVEN TO CICS            (see note 5)
```

Figure 9. Sample Output on Starting Up an Initial CICS System

Notes:

1. The libraries required for this job are defined in the partition standard subarea, and their search order is specified in a LIBDEF statement.

2. The PARM='CONSOLE' operand produces the prompt for system initialization overrides.

3. Overrides, terminated by $end.

4. The supplied TCT provides support for the system console.

5. Final CICS message — you can now communicate with CICS.

STARTUP ERRORS

If any problems occur during CICS initialization, look up the message in the CICS/VS Messages and Codes manual and take the appropriate action before rerunning the startup job.

AN INITIAL CICS SYSTEM

In the F2 partition, you now have an initial CICS system, but with very little function.

COMMUNICATING WITH CICS FROM THE SYSTEM CONSOLE

At this stage you will wish to communicate with CICS. You can use only the system console ("Chapter 2.3. Sample Application Package and CICS Tables" will explain how to use a full-screen CICS terminal).

As soon as CICS initialization has been completed and the message

```
DFH1500 - CONTROL IS BEING GIVEN TO CICS
```

has been received, you can communicate with CICS from the system console. On the console, enter:

```
msg partition-id
```

In the example dialog below, CICS is running in the F2 partition, so you enter:

```
msg f2
```

Figure 10 shows a typical dialog between the operator and the CICS system.

CICS transactions can be entered from the console prefixed by the appropriate reply identifier. In the example, the command:

```
2 cemt inquire terminal
```

was entered.

The CICS system is shut down by the CICS command CEMT PERFORM SHUTDOWN at the system console.

```
 F2 002 DFH1500 - CONTROL IS BEING GIVEN TO CICS
 msg f2
*F2-002
 2 cemt inquire terminal
 F2-002
 F2 002   Ter(CNSL) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti
 F2 002   Ter(SAMA)           Pri( 032 ) Aut Ins Ati Tti
*F2 002  RESPONSE: NORMAL TIME:  22.50.34   DATE:  83.070
 2 cemt perform shutdown
 F2-002
 F2 002 DFH1701 - C.I.C.S. IS BEING TERMINATED
 F2 002 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
 F2 002 EOJ CICS

 DATE 03/11/83,CLOCK 22/50/51,DURATION 00/01/27
```

Figure 10. Communicating with CICS Using the System Console

END OF INSTALLATION?

You have now run an initial CICS system. Part 2 of this book will help you extend this initial system by adding system and user data sets, CICS control tables, and the sample CICS application package.

Part 2. Extending Your CICS System

PART 2

Chapter 2.1. Introduction

You have now completed the installation of the CICS system as supplied on the distribution tape. You are able, as shown in the last chapter, to start the system up, perform some very basic operations, and shut it down again. You will now wish to extend it (and possibly modify it) by installing your own application programs. These application programs will have been written by you, or supplied, to enable you to use the system to satisfy the specific requirements of your installation.

The process of performing the various tasks required to complete the installation of a working system is best illustrated by means of an example. The CICS system, as supplied on the distribution tape, contains some sample application programs; these will now be used in this and the next two chapters to describe the procedures involved. The style adopted will be tutorial; if you are familiar with the topics discussed, you should refer direct to Parts 3, 4, and 5 of this book for the equivalent information in reference form.

If you are an existing CICS 1.5 user, you will have your own versions of CICS control tables and startup job streams. You may also have generated your own versions of CICS management modules. See "Chapter 6.1. Migrating to CICS 1.6" for a description of the changes you must be aware of before you run your own CICS 1.6 system.

CREATING A WORKING CICS SYSTEM

The installation of application programs into the VSE core image libraries is not, by itself, sufficient to enable them to be run as CICS application programs. You must also make them known to the CICS system by defining them in a CICS control table (the processing program table). By doing so, you define the programs as CICS resources; the process of assembling and link-editing the control table is known as **resource definition**.

An instance of the execution of an application program is known as a **transaction**. Transactions are initiated by sending a transaction identifier to a running CICS system. The transaction identifier can be between 1 and 4 characters in length. CICS associates this identifier with the corresponding application program, initiates the transaction, and passes control to the application program.

Transaction identifiers are a further instance of CICS resources. You must define them to CICS in the program control table, each entry in the table including both the transaction identifier and the name of the corresponding application program.

A number of further CICS resources, such as user files and terminals, must be defined to CICS in control tables. These are described in more detail in "Chapter 2.3. Sample Application Package and CICS Tables."

Apart from the control tables, you will need to define CICS data sets. Some of these, like the auxiliary temporary storage data set and the transient data intrapartition data set, are usually required to satisfy requests made by application programs. Others, such as the dump data set or the auxiliary trace data set, are more commonly required to enable the CICS system or the operator to perform certain operations. The CICS data sets are described in "Chapter 2.2. System Data Sets."

Finally, you will need to construct a job stream to run the CICS system. In doing so, you will need to be aware of the function provided by the various pregenerated modules on the distribution tape (listed in Appendix C). You will need to choose your own versions (or the pregenerated versions) of the CICS control tables for the CICS run. The various modules and tables are identified by 2-character suffixes and these are defined to CICS during CICS initialization as initialization parameters. In this way, CICS is able to build up a running system with the precise components you have specified at CICS startup time. The CICS startup job stream is described in "Chapter 2.3. Sample Application Package and CICS Tables."

All the jobs mentioned in the following three chapters are supplied, in general form, in the single Z.DFHSPJCL book in the source statement library. Appendix F lists this source book. It is not intended that this source book should be run as a single job stream. You should punch out this source book and edit individual jobs with your specific installation values, for example, data set names and space allocations.

If you are using the VSE interactive computing and control facility (ICCF), the following job can be used to punch the sample job stream to the POWER punch queue.

```
* $$ JOB JNM=PUNCHSP
* $$ PUN CLASS=Q
// JOB PUNCHSP
// LIBDEF SL,FROM=C160PSL
// EXEC SSERV
 PUNCH Z.DFHSPJCL
/*
/&
* $$ EOJ
```

Figure 11. Punching the Sample Job Stream to a POWER Punch Queue

The job stream can then be retrieved using the ICCF GETP procedure:

```
GETP PUNCHSP Q DELETE MEM=DFHSPJCL
```

The DFHSPJCL sample job stream will need to be modified as follows:

Notes:

1. In the DFHSPJCL sample job stream, the fields included in **lowercase** should be replaced with your own values before each job is run.

2. In the DFHSPJCL sample job stream, the *$$ should be deleted.

```
*$$ // should be replaced with //
*$$ /* should be replaced with /*
*$$ /& should be replaced with /&
```

The source books for the sample tables are also supplied in the distribution source library. These are included as A. books, to be used as copy files in your assembly of CICS control tables. Appendix D names and describes each source book.

Chapter 2.2. System Data Sets

This chapter (together with the next two chapters) is intended for a new user, to provide an introductory and more tutorial approach to the installation process. This chapter contains a general description of the data sets, concentrating particularly on those which need to be defined to run the sample application programs. In order to place the description in context, this chapter also includes a short overview of how the data sets are used in a CICS system.

PART 2

The CICS data sets can be divided into classes according to the type of activity for which they are used. The following list shows such a classification:

Purpose	Type of Data Set	Access Method
Application programming	Temporary storage	VSAM
	Transient data (intrapartition)	VSAM or DAM
	Transient data (extrapartition)	SAM
Problem determination	Dump	SAM
Problem determination / Performance monitoring	Auxiliary trace	SAM
Performance monitoring	Automatic statistics	SAM
	Monitoring journals	SAM
Restart, recovery, and journaling	System log and journals	SAM
	Restart	VSAM
Resource definition online	CICS system definition file	VSAM

Not all of these data sets need necessarily be defined when running a CICS system. If your application programs are written in such a way that none of them ever makes any request to read or write to intrapartition transient data, the transient data intrapartition data set need not be defined. If you will never be requesting automatic (as opposed to termination or requested) statistics, the automatic statistics data sets need not be defined. Again, if your procedures are such that CICS is always started up without reference to the preceding run, if no data needs to be preserved from run to run, and if information does not need to be recovered or backed out (see "Chapter 2.4. Restart and Recovery of CICS") after the unlikely event of a system failure, you can run CICS without defining a restart data set and CICS system log. In most cases, however, these data sets will be found to be required at some stage in the lifetime of a production system. As part of your installation procedure, therefore, you should define these data sets after the initial system described in Part 1 of this book has been set up.

The amount of space to be allocated to the data sets depends on the requirements of your installation and on the DASD available to you. Part 4 of this book contains information to aid you in making this choice. In the examples given in the next few chapters, suggested space allocations have been chosen to be sufficient to run the sample application programs.

Part 4 of this book describes the CICS data sets in some detail. The chapters in that part explain which CICS functions use the data sets, how the data sets are defined and initialized, how they are manipulated during CICS execution, and how the information stored in them is postprocessed by CICS-supplied utility programs.

Part 4 is intended as reference material for system programmers who are familiar with the basic CICS functions and the data set requirements for their use.

VSAM REQUIREMENTS

As indicated in the above table, certain of the CICS data sets are defined as VSAM files. Before this can be done, you must define the VSAM data space(s) that is to contain the data sets and indicate which VSAM catalog(s) is to own the data space(s). In general, for performance reasons, you may wish to define the various VSAM files to reside on different volumes: you should then define a separate user catalog for each volume.

You may also wish to define one or more of your VSAM files as UNIQUE. A unique file is such that no other file can occupy its data space. In this case, the data space is not defined separately; it is defined as part of the definition of the unique file itself.

In this book, however, the sample jobs assume that the data sets are all defined on a single VSAM data space, owned by a user catalog defined to have the name CICS160.UCAT.

The following jobs can be used to define a user catalog occupying 19 tracks on a 3350 volume with volume identifier M3350A, and to define VSAM space of a further 171 tracks belonging to that catalog. It is assumed that the VSAM master catalog for the system has already been defined as part of your VSE installation.

```
// JOB DEFUCAT CREATE VSAM USER CATALOG
// EXEC IDCAMS,SIZE=AUTO
 DEFINE USERCATALOG                -
          (NAME(CICS160.UCAT)      -
           ORIGIN(19)              -
           VOLUME(M3350A)          -
           TRACKS(19))             - (see note)
        CATALOG(VSAM.MASTER.CATALOG)
/*
/&
// JOB DEFSPACE DEFINE SPACE
// EXEC IDCAMS,SIZE=AUTO
  DEFINE SPACE                     -
          (VOLUMES(M3350A)         -
           ORIGIN(38)              -
           TRACKS(171))            - (see note)
        CATALOG(CICS160.UCAT)
/*
/&

              (These jobs are included in Z.DFHSPJCL)
```

Figure 12. Defining a VSAM User Catalog and Space

Note: The job to define the user catalog, included in Z.DFHSPJCL, has the BLOCKS parameter coded. If you are using CKD DASD, you must replace the BLOCKS parameter in the job with a TRACKS parameter, as shown in the example above.

The equivalent sizes for a user catalog on devices other than a 3350 are:

Device Type	User Catalog Size	VSAM Space Size
3330	19 tracks	171 tracks
3340	24 tracks	216 tracks
3310/3370	438 blocks	3942 blocks

After you have defined the data space(s) and associated them with a VSAM catalog, you are in a position to define the VSAM files that are to be used as CICS data sets. All definitions are performed using VSAM Access Method Services; sample jobs for this purpose are given later in this and in following chapters.

PART 2

Finally, every time you run the job to start up CICS, each VSAM file used by CICS must be identified by means of a DLBL control statement. No EXTENT statement should be included, because the file is owned by VSAM. The VSAM catalog that owns the file must, however, be specified explicitly or implicitly.

TEMPORARY STORAGE AND TRANSIENT DATA

Temporary storage and that type of transient data known as **intrapartition** are storage mechanisms provided by CICS to enable an application program to pass data created by one transaction, to be used later by the same transaction or by a different transaction. The type of transient data known as **extrapartition** enables an application program to read data originating outside the CICS system or to store data intended for a destination outside the CICS system. The difference between temporary storage and intrapartition transient data depends largely on the way the saving of data and later retrieval is performed.

Temporary storage is essentially used for storing and retrieving data randomly, that is, in any order. The data is, in fact, stored in the form of queues of records, but these records may be read or updated in any order, by specifying the position in the queue.

Transient data, on the other hand, is designed to be a sequential storage technique. The elements of data are written, or read, in sequence as a queue, and no mechanism exists for updating an element already written. When elements in an intrapartition transient data queue are read, the elements are destroyed, that is, they may not be read again during the CICS run.

TEMPORARY STORAGE

Data may be written to temporary storage either in main storage or in auxiliary storage, the choice of storage type being made by the application program at the time the data is written.

Data in main temporary storage does not survive from one CICS run to the next: main storage is, therefore, used if the data is needed for only short periods of time. It is also used if only relatively small amounts of data are to be stored.

Auxiliary temporary storage is used if large amounts of data are to be stored or if the data is to be kept for extended periods of time and, in particular, maintained from one CICS run to the following one. The ability of CICS to restore temporary storage data in this way relies, however, on the existence of a restart data set and the possibility of performing a "warm restart" (see "Chapter 2.4. Restart and Recovery of CICS").

Data is written to temporary storage in the form of a queue of elements or records. The queue name (referred to as DATAID) is assigned dynamically by an application program at the time the queue is created by writing the first record in it. Further records may then be added to the queue by the same transaction or a different transaction, and the queue may be explicitly deleted when no longer required. When the queue is deleted, the storage allocated to it is released and may be reused by subsequent temporary storage requests.

Temporary Storage Data Set

Use of auxiliary temporary storage requires the definition of a single VSAM auxiliary temporary storage data set for the CICS run. The data set must be large enough to satisfy all demands for storage made during the run, the storage within the data set being allocated and released dynamically in response to requests made by application programs. Note also that CICS itself makes requests to temporary storage, on behalf of application programs, when performing certain CICS functions (see "Chapter 4.2. Temporary Storage Data Set" for details).

How to Define

The following is a sample job for defining the temporary storage data set. The job defines a nonindexed VSAM file (cluster) with name CICS160.TEMP.STORAGE on volume M3350A in a VSAM data space owned by user catalog CICS160.UCAT.

```
// JOB CRATS CREATE AUXILIARY TEMPORARY STORAGE DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                   -
          (NAME(CICS160.TEMP.STORAGE)             -
           RECORDSIZE(4089,4089)                  -
           RECORDS(144)                           -
           NONINDEXED                             -
           CONTROLINTERVALSIZE(4096)              -
           SHAREOPTIONS(2)                        -
           VOLUMES(M3350A))                       -
        DATA                                      -
          (NAME(CICS160.TEMP.STORAGE.DATA))       -
        CATALOG(CICS160.UCAT)
/*
/&

                (This job is included in Z.DFHSPJCL)
```

Figure 13. Defining the Auxiliary Temporary Storage Data Set

JCL Required for CICS Execution

When the temporary storage data set (and any VSAM file) is defined in the CICS startup job, only a single DLBL statement is required, indicating that the file is managed by VSAM.

The catalog that owns the file can be identified **explicitly** by including the CAT= operand in the DLBL statement. This operand specifies the filename of a DLBL statement for the catalog. This enables you to associate different catalogs with the various VSAM files in the system. A catalog might, for example, be identified by the statement:

```
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
```

and the DLBL statement for the VSAM file would then refer to CICSUCT.

If the temporary storage data set is defined as above, and the catalog that owns the data set is identified to CICS by a DLBL statement with filename CICSUCT, the following DLBL statement should also be included in the job to start up CICS:

```
// DLBL DFHTEMP,'CICS160.TEMP.STORAGE',,VSAM,CAT=CICSUCT
```

where the filename DFHTEMP is a required name.

PART 2

The catalog is specified **implicitly** by omitting the CAT= operand from the DLBL statement for the VSAM file. The DLBL statement for the catalog is then included with reserved filename IJSYSUC:

```
// DLBL  IJSYSUC,'CICS160.UCAT',,VSAM
```

and the one catalog is assumed to own all the VSAM files in the job.

SIT Overrides

The version of the temporary storage program used in the initial system described in Part 1 of this book is DFHTSP1$. This allows temporary storage to be written and read in main storage only. In order to start up CICS and enable auxiliary temporary storage to be used, you may specify that the pregenerated module DFHTSP2$ is to be loaded when the system is being initialized. This can be achieved by including the initialization override parameter

```
TSP=2$
```

with the CICS startup job.

TRANSIENT DATA

As in the case of temporary storage, data written to transient data is stored in the form of named queues of elements. The method of defining and processing these queues is, however, somewhat different.

The queues, referred to as destinations, must be predefined in a table, known as the destination control table (DCT). Any number of DCTs may be defined; the particular DCT selected for a CICS run determines which destinations may be used during that run.

Destinations are of two types:

1. Intrapartition, which are accessible only to application programs within the CICS system
2. Extrapartition, which are known not only within the CICS system, but can also be processed by programs independent of the CICS system.

Intrapartition Destinations

Intrapartition destinations are queues in which items of data are stored for subsequent retrieval by a transaction running in the same CICS partition; the transaction refers to the queue by its name (DESTID) and retrieves at each stage the oldest item of data that has not yet been read. Thus, an item of data cannot be read more than once, and the extent of an intrapartition queue is described at any moment by the two pointers to the oldest and newest items of data in the queue.

The most common use of intrapartition data is one in which a number of transactions accumulate a collection of records by writing to the same queue. Subsequently, a separate transaction is initiated to process the entire queue. This transaction may be initiated:

1. Explicitly, by an operator entering the transaction identifier at a terminal, or

2. Automatically, by CICS, when a specified period of time has elapsed or a specified time of day has been reached, or

3. Automatically, by CICS, when a specified number of records has been collected on the queue.

The last method of initiating a transaction is achieved by including in the destination control table entry definition for the queue, the number of queue elements that will trigger off the transaction (trigger level — TRIGLEV) and the identifier of the transaction to be initiated (TRANSID).

An example of an intrapartition destination with a trigger level is given in the definition of the L860 queue included in the DFHXDCTS sample copy book described in "Destination Control Table (DCT)" on page 71.

Extrapartition Destinations

Extrapartition destinations are sequential data sets (queues) external to the CICS partition, residing on any sequential device (DASD, tape, printer, and so on).

Extrapartition destinations may be defined in the destination control table either as sequential input files or as sequential output files. They cannot be processed as both.

Input transient data files may be defined as residing on disk or magnetic tape, or they may be defined as a card reader or simulated card reader (SYSIPT) held on disk or tape.

Output transient data files may again be defined as disk or tape, or they may be sent direct to a printer (for example, SYSLST). They may alternatively be defined to simulate SYSLST or SYSPCH on disk or tape.

Data written to extrapartition destinations is, in general, intended for subsequent input to a batch program or, alternatively, is routed direct to an output device such as a line printer.

An example of an extrapartition destination routed to SYSLST is contained in the pregenerated destination control table (DFHDCT1$) as the termination statistics destination, CSSL.

An example of an extrapartition destination routed to a disk or tape file is one of the automatic statistics data sets (CSSM or CSSN). The use of automatic statistics is described in "Chapter 4.7. Automatic Statistics Data Sets."

Transient Data Intrapartition Data Set

If intrapartition transient data is to be used, it is necessary to define an intrapartition data set for the CICS run. There is, in this case, no option to define transient data queues in main storage: the data frequently needs to be preserved for long periods of time, and from one CICS run to another. The intrapartition data set is a VSAM (or DAM) data set in which, unlike temporary storage, a particular control interval (or track) may not hold elements from more than one queue (destination) at any given time. The data set should, therefore, be defined large enough to allow at least a control interval (or track) for every destination that is active at any given time.

The destination may be defined with the reusable queue space feature (REUSE=YES in the DCT). If this is done, a control interval (or track) allocated to the queue is released for reuse when all the items of data stored in the control interval (or track) have been read. Otherwise, the application program must purge the space allocated to the queue explicitly at appropriate points to prevent the space growing unboundedly or remaining allocated for an indefinite time.

How to Define

The following sample job defines a transient data intrapartition data set as a nonindexed VSAM file with name CICS160.INTRA.TRANSD on volume M3350A in a data space owned by user catalog CICS160.UCAT.

```
// JOB CRITD CREATE INTRAPARTITION TRANSIENT DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                   -
          (NAME(CICS160.INTRA.TRANSD)             -
           RECORDSIZE(1529,1529)                  -
           RECORDS(100)                           -
           NONINDEXED                             -
           CONTROLINTERVALSIZE(1536)              -
           VOLUMES(M3350A))                       -
        DATA                                      -
          (NAME(CICS160.INTRA.TRANSD.DATA))       -
        CATALOG(CICS160.UCAT)
/*
/&

                  (This job is included in Z.DFHSPJCL)
```

Figure 14. Defining the Intrapartition Data Set

JCL Required for CICS Execution

If the transient data intrapartition data set has been defined as above, and the user catalog has been identified in the CICS startup job with filename CICSUCT, you should also include the following statement:

```
// DLBL DFHNTRA,'CICS160.INTRA.TRANSD',,VSAM,CAT=CICSUCT
```

The filename DFHNTRA is a required name.

SIT Overrides

The pregenerated system initialization table used to bring up the initial system described in Part 1 of this book is DFHSIT1$. This contains (see Appendix D) the entries TDP=1$ and DCT=1$. The pregenerated transient data program DFHTDP1$ supports extrapartition transient data only (see Appendix C). Correspondingly, the pregenerated sample table DFHDCT1$ defines only extrapartition destinations (including CSSL, which is used by CICS for termination statistics). If you wish to include intrapartition transient data in your CICS system, you will need to:

1. Include a system initialization override for an appropriate version of the transient data program, for example, TDP=6$, and

2. Code, assemble, and link-edit in your system a destination control table that includes entries for your own

intrapartition queue (destination) names. When assembling the table, you can choose the 2-character suffix by which you wish it to be known, and then identify the table in the CICS startup job by means of the DCT= initialization override parameters.

More guidance on how to assemble and link-edit your own destination control table is given in the next chapter.

DUMPS

Dumps (representations of the contents of selected areas of main storage) provide the fundamental means of:

1. Determining that a program is functioning as intended (especially while the program is being developed), and

2. Analyzing problems that may occur from time to time.

Dumps are not necessarily produced as the result of a CICS system failure. They may be written in a variety of circumstances, either when the CICS system detects an error or as a result of the user making an explicit request. Any number of dumps may be recorded during the progress of a CICS run. A description of the types of dumps and occasions when they are produced is given in "Chapter 4.9. Dump Data Sets."

If the CICS system is still in control after the problem has occurred, it may — depending on circumstances — produce a transaction dump or a formatted dump. In contrast to basic, unformatted dumps of storage, formatted and transaction dumps are intended to make it easier to diagnose errors by performing automatically the mechanical process of finding the control blocks in storage. Each control block is printed separately in the dump, and is preceded by a heading; for some of the blocks, the important fields are printed by name. In addition, the dump may contain diagnostic error messages if the nature of the failure is simple enough for the CICS system to identify.

More detailed information on formatted and transaction dumps and advice on how to interpret them is given in the CICS/VS Problem Determination Guide.

Dump Data Sets

Dumps produced by CICS are placed in a CICS data set. Because, in general, it is impossible to foresee when an error producing a dump will occur, it is important that every CICS run should include a dump data set. In order to print a dump in the data set, a CICS utility program — DFHDUP (Dump Utility Program) — is provided. This runs in batch mode and requires the dump data set to have been closed by the CICS system before the utility program can be run.

If a large number of dumps is produced in the course of a CICS run, a single dump data set may be insufficiently large to hold them all. To overcome this problem, CICS allows two dump data sets to be defined. When the first data set becomes full, the operator (or the CICS system) may switch to the other data set. The switching is performed by means of the CEMT master terminal transaction either explicitly or automatically (see "Chapter 4.9. Dump Data Sets" for details). The first data set is closed and may be printed down using DFHDUP while CICS is still executing.

When a dump data set is opened for output, the pointers in the data set are reset and any recorded dumps are effectively lost. This takes place at CICS startup time, when the first data set that is to be used is opened, and again when the alternate data set is opened after a switch. You should remember, therefore, that if a data set has not been opened during a CICS run, it will either be empty or will contain data unrelated to that run. In order to ensure that both data sets are empty at the start of a CICS run, you might use the technique of switching the data sets twice as soon as CICS has been initialized.

When you are defining the dump data sets, the space allocated to them should be dictated by the frequency with which dumps are expected to occur and, of course, by the amount of DASD space you have available. A frequent technique, which avoids excessive switching of dump data sets, is to define a primary data set large enough to hold all the dumps that are likely to occur during a CICS run. A secondary, smaller data set is then also defined, for use on the rare occasions when the primary data set becomes filled up.

If this secondary data set is required, it will be used only for the length of time that it takes to print down the primary data set (using DFHDUP). The data sets are then switched again and the primary data set comes into operation in the normal way.

The sample job control statements given below show a suggested allocation of 40 tracks for the primary data set DFHDMPA, and 10 tracks for DFHDMPB.

JCL Required for CICS Execution

Because the dump data sets are SAM files, they are defined to CICS by the usual DLBL and EXTENT control statements: there is, in this case, no need to predefine them by means of a separate job.

You might wish to include the DLBL and EXTENT statements for these and the other CICS data sets as part of the definitions in the partition standard labels, to take effect during initialization of the VSE system. You can then omit mentioning these statements explicitly in the CICS startup job. In the examples in this book, however, the DLBL and EXTENT statements will be given in full in the jobs to which they belong. This means that the extent information is repeatedly established; there is no disadvantage in this, so long as it is not modified from run to run.

The control statements below define two dump data sets named CICS160.DUMP.DATASET.A and CICS160.DUMP.DATASET.B, occupying 40 tracks and 10 tracks beginning at tracks numbered 210 and 250 respectively on an IBM 3350 Direct Access Storage Device. The equivalent sizes for devices other than a 3350 are:

Device Type	Sizes
3330	50 and 10 tracks
3340	128 and 32 tracks
3310/3370	1200 and 300 blocks

For an explanation of the other operands in these statements, see VSE/Advanced Functions System Control Statements.

```
// DLBL DFHDMPA,'CICS160.DUMP.DATASET.A',0,SD
// EXTENT SYS001,,1,0,210,40
// DLBL DFHDMPB,'CICS160.DUMP.DATASET.B',0,SD
// EXTENT SYS001,,1,0,250,10
```

Figure 15. Defining the Dump Data Sets

Filenames DFHDMPA and DFHDMPB are required names.

SIT Overrides

The system initialization table used to define the initial system described in Part 1 of this book contains the entry DCP=NO. This means that, when the initial system is initialized, a dummy version of the dump control program is loaded. If the conditions that would result in a dump occur, the dummy version is invoked but control is returned immediately without the dump being actually recorded. The version of the dump control program that supports recording of dumps on disk is DFHDCPD$ (see Appendix C). After you have defined the dump data sets on disk, therefore, you should include the override parameter

```
DCP=D$
```

with the CICS startup job.

TRACE

CICS trace is used to record the progress of CICS transactions and is a further aid in debugging problems that may occur. Whereas a dump gives a snapshot of conditions at a particular instant of time, CICS trace provides a a history of events leading up to a certain situation. CICS trace can also be useful in analyzing performance problems such as excessive waiting on events in the system, or bottlenecks resulting from inefficient system setup or application program design.

In many CICS installations, recording of CICS requests in the trace table is arranged to take place continuously while CICS is running. If an error occurs and this causes a dump to be taken, the trace entries are then immediately available as part of the dump, and can be printed out in interpreted form as part of DFHDUP processing. For information on the format of trace entries and for hints on how to use trace to analyze problems, consult the CICS/VS Problem Determination Guide.

Trace entries are stored for all, or selected, requests made by application programs to CICS management modules. The entries are of standard length (32 bytes each) and are stored sequentially in a trace table, whose size is determined at CICS startup time by means of the TRT= initialization parameter. When the table becomes full, subsequent entries overwrite the oldest entries in a wraparound manner.

Auxiliary Trace Data Sets

Trace entries can be preserved from being overwritten by writing them to an auxiliary trace data set as well as the main storage trace table (see "Chapter 4.8. Auxiliary Trace Data Sets" for details). When auxiliary tracing has been completed and after the trace data set has been closed, the information can be printed down, as in the case of the dump data set, with the help of the CICS-provided DFHTUP utility program (trace utility program). The utility also allows trace entries to be selected for printing according to defined criteria.

Tracing of CICS requests in auxiliary storage is normally controlled by means of the CEMT master terminal transaction. (For other methods of initiating and terminating both main storage and auxiliary storage trace, see "Chapter 4.8. Auxiliary Trace Data Sets.") Because of the volume of output produced, tracing is usually activated only for limited periods of time, typically when a situation that has caused a problem requires to be rerun because the in-core trace has not provided sufficient information.

Two auxiliary trace data sets may be defined: in this way, tracing to auxiliary storage may continue after the other data set has been closed and while it is being printed. The switch from the one data set to the other is carried out by the master terminal operator using the CEMT master terminal transaction. If the data set becomes full, the master terminal operator is prompted to perform the switch by the CICS system which detects the end of data set condition, closes the data set, and displays a message at the system console.

JCL Required for CICS Execution

The following statements define two auxiliary trace data sets named CICS160.AUX.TRACE.A and CICS160.AUX.TRACE.B. The data sets begin at tracks 260 and 300, and occupy 40 tracks each.

The comments in the previous section of this chapter about DLBL and EXTENT statements for the dump data sets apply equally to the auxiliary trace data sets.

```
// DLBL     DFHAUXT,'CICS160.AUX.TRACE.A',0,SD
// EXTENT   SYS001,,1,0,260,40
// DLBL     DFHBUXT,'CICS160.AUX.TRACE.B',0,SD
// EXTENT   SYS001,,1,0,300,40
```

Figure 16. Defining the Trace Data Sets

Filenames DFHAUXT and DFHBUXT are required names.

SIT Overrides

The pregenerated version of the trace control program selected in the initial system initialization table is DFHTRP5$ (main storage only). To allow auxiliary tracing to disk to be available, the override parameter

TRP=D$

must be included with the startup job.

The pregenerated system initialization table DFHSIT1$ contains the entry:

TRT=512

indicating that 512 trace entries can be accommodated in the in-core trace table. This value can be changed, if required. If TRT is set to be 0, no tracing will take place.

CICS STATISTICS

CICS statistics are the simplest means of gathering information that can be used to analyze the behavior and performance of the system. They provide information relating to most of the activities in the CICS system, including for example:

- Number of tasks in the system
- Number of transactions accepted and the identity of each
- Number of programs used and the identity of each
- Identity of each terminal used.

A complete description of the information provided by CICS statistics is included in the CICS/VS Performance Guide.

Three methods of obtaining the statistics are available:

1. The accumulated statistics for the CICS run are always recorded when CICS is shut down. As already mentioned, these statistics (referred to as **termination statistics**) are written to a CICS-defined extrapartition transient data destination CSSL. There is no way of suppressing these statistics except by not defining the destination CSSL in the destination control table (DCT), or by running the CICS system with no DCT at all defined. If the CSSL destination is defined, it must be associated with a filename in the DCT which, in turn, must be defined to VSE as a sequential output file (in the initial system described in "Chapter 1.4. Initial CICS Startup," this file is defined to be SYSLST).

2. Making an explicit request at any time while CICS is running. The accumulated statistics are then recorded in the same format as the termination statistics described above, and are written to the extrapartition destination CSSL (or to a different extrapartition destination if specified with the request). **Requested statistics**, as this method is called, are initiated by means of the master terminal CSTT transaction. Both requested statistics and termination statistics are formatted by CICS so that they can be sent direct to an output printer.

3. Automatic periodic recording of the statistics, the frequency of recording being specified with the CSTT transaction request which initiates the recording. This method is referred to as **automatic statistics**. After each recording is made, the statistics counts are reset to zero, so the values recorded refer only to the period of time between recordings.

Statistics Data Sets

As mentioned above, for both termination and requested statistics, the output is formatted by CICS in such a way that it can be directed immediately to an output printer. This fact is reflected in the pregenerated destination control table DFHDCT1$ where the extrapartition destination CSSL is defined to be associated with the VSE symbolic unit SYSLST. No explicit job control statements are then required to define a statistics data set when running CICS.

In the case of automatic statistics, the output is not formatted for immediate printing, but instead must be sent to a sequential output data set. The mechanism for doing this also uses extrapartition transient data destinations; in this case, the destination names are CSSM and CSSN, and these must correspond to sequential access method (SAM) files defined on disk or tape with filenames DFHSTM and DFHSTN as specified on the DLBL system control statements.

"Chapter 4.7. Automatic Statistics Data Sets" describes how to define automatic statistics data sets and how to control the recording of them.

SIT Overrides

The initial system described in Part 1 of this book included the transient data program DFHTDP1$ (extrapartition only). This version can be used to record automatic and requested statistics as long as the destinations are defined in the destination control table. Because the pregenerated table DFHDCT1$ does not include CSSM and CSSN, only requested statistics (sent to destination CSSL) will be demonstrated in the sample run at the end of this chapter.

OTHER CICS DATA SETS

The remaining CICS data sets are described later in this book.

The restart data set has been mentioned already in connection with maintaining temporary storage data from one CICS run to the next. The restart data set, together with the system log, is concerned, more generally, with recreating the same CICS environment at startup time as was in effect at the time of the previous shutdown. These concepts are discussed further in "Chapter 2.4. Restart and Recovery of CICS."

The CICS system definition file is used to hold information necessary to enable the definition of resources (programs and transactions) to be carried out while CICS is running. Online definition of resources is described in the CICS/VS Resource Definition Guide.

JOB TO START UP CICS WITH THE CICS DATA SETS DEFINED

At this stage, it may be instructive to start up CICS again, including all the essential CICS data sets described so far. The following facilities have been defined:

- Temporary storage
- Transient data
- Dump
- Auxiliary trace
- Statistics.

The system initialization table used for this startup is the same as the one used to start up the initial CICS system described in "Chapter 1.4. Initial CICS Startup" (DFHSIT1$). To the startup parameters defined in this table are added override parameters, included here in the SYSIPT file. Because the initial system initialization table includes only console support, the demonstration in this chapter will again be limited to the console. "Chapter 2.3. Sample Application Package and CICS Tables" explains how to install a terminal control table that allows transactions to be run from terminals.

Although no user application programs have yet been defined in the system, CICS-provided transactions (such as those shown in Figure 19) can be used to demonstrate some of the functions that can be performed with a subset of these data sets.

The functions that can be demonstrated in this way are dump, auxiliary trace, and statistics — both requested and termination.

The temporary storage and intrapartition data sets are included in the startup job, but are not used at this stage. To demonstrate these functions, it is necessary to install application programming in the system together with a destination control table (not provided in the pregenerated system). How this is done is explained in the next chapter.

The system initialization parameters necessary to start up a system with these facilities enabled are included here in the form of system initialization override parameters read from SYSIPT. The system is notified that SYSIPT parameters are to be included, either by means of the SYSIN or SI parameter in the PARM option (as shown in the following example), or alternatively by means of an UPSI job control statement with a setting of 1. (See Chapter 5.3 for a full description of UPSI settings for a CICS startup job stream.)

The CICS startup job is followed by jobs to print down the dump and auxiliary trace data sets to show the type of output that may be expected. Only a single data set is processed in each case; in a production system where data-set switching is needed, the utilities might be run against data sets that had become full while CICS was still running.

```
// JOB CICS BRING UP CICS SYSTEM WITH DATA SETS
* User VSAM Catalog
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
* Auxiliary Temporary Storage
// DLBL DFHTEMP,'CICS160.TEMP.STORAGE',,VSAM,CAT=CICSUCT
* Intrapartition Transient Data
// DLBL DFHNTRA,'CICS160.INTRA.TRANSD',,VSAM,CAT=CICSUCT
* Dump Data Sets
// DLBL DFHDMPA,'CICS160.DUMP.DATASET.A',0,SD
// EXTENT SYS001,,1,0,210,40
// DLBL DFHDMPB,'CICS160.DUMP.DATASET.B',0,SD
// EXTENT SYS001,,1,0,250,10
* Auxiliary Trace
// DLBL DFHAUXT,'CICS160.AUX.TRACE.A',0,SD
// EXTENT SYS001,,1,0,260,40
// DLBL DFHBUXT,'CICS160.AUX.TRACE.B',0,SD
// EXTENT SYS001,,1,0,300,40
* ASSGN of DASD Containing CICS and User Files
// ASSGN SYS001,DISK,TEMP,VOL=M3350A,SHR
* Search Order for Core Image Libraries
// LIBDEF CL,SEARCH=C160PCL
* Start Up CICS With Override Parameters From SYSIPT
// EXEC DFHSIP,SIZE=1200K,PARM='SIT=1$,SI'
 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 $END
/*
/&

                  (This job is included in Z.DFHSPJCL)
```

Figure 17. Starting Up CICS With System Data Sets Defined

The messages displayed at the system console while the above job is being run, are shown in Figure 18. After the final message shown has been displayed, CICS is ready to accept transaction identifiers and initiate application programs to perform useful work.

The CEMT and CSTT transactions are provided by CICS and can be used to demonstrate a few simple operations on the CICS data sets. Examples of these commands are shown in Figure 19, where they are given in their full, unabbreviated forms. CICS does, however, accept abbreviated forms of the commands, as long as the meaning is not ambiguous, thus

```
cemt set auxtrace on open
```

can be written

```
cemt s a on op
```

and

```
cemt perform shutdown
```

can be written

```
cemt p shut
```

The cemt perform shutdown command is used to terminate the CICS run. As part of shutdown processing, CICS records termination statistics which, in the present example, are sent to SYSLST.

CONSOLE OUTPUT

```
F2 002 // JOB CICS BRING UP CICS WITH SYSTEM DATA SETS

DATE 03/11/83,CLOCK 23/22/54
F2 002 * User VSAM Catalog
F2 002 * Auxiliary Temporary Storage
F2 002 * Intrapartition Transient Data
F2 002 * Dump Data Sets
F2 002 * Auxiliary Trace
F2 002 * ASSGN of DASD containing CICS Data Sets
F2 002 1T20I  SYS001 HAS BEEN ASSIGNED TO X'329'
F2 002 * Search Order for Core Image Libraries
F2 002 * Start Up CICS With Override Parameters From SYSIPT
F2 002 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS.
F2 002 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
F2 002  TSP=2$,TDP=6$,DCP=D$,TRP=D$
F2 002  $END
F2 002 DFH1501 - DFHSIT1$ IS BEING LOADED
F2 002 DFH1500 - LOADING CICS NUCLEUS
F2 002 DFH1500 - CICS START-UP IS COLD
F2 002 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
F2 002 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
F2 002 DFH1500 - DUMP DATA SET IS BEING OPENED
F2 002 DFH1500 - INITIALIZING TEMPORARY STORAGE
F2 002 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS   786K
F2 002 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
F2 002 DFH1500 - STXIT MACROS ARE BEING ISSUED
F2 002 DFH1500 - PROCESSING RESIDENT PROGRAMS
F2 002 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS   710K
F2 002 DFH1500 - CONTROL IS BEING GIVEN TO CICS
```

Figure 18. Sample Output on Starting Up CICS With Essential Data Sets

```
 F2 002 DFH1500 - CONTROL IS BEING GIVEN TO CICS
 msg f2
*F2-002
 2 cemt inquire dump
 F2 002
 F2 002   Dum(DFHDMPA ) Ope
*F2 002  RESPONSE: NORMAL TIME:  23.30.02   DATE:  83.070
 2 cemt set dump close
 F2 002
 F2 002   Dum(DFHDMPA ) Clo      NORMAL
*F2 002  RESPONSE: NORMAL TIME:  23.30.14   DATE:  83.070
 2 cemt set dump open
 F2 002
 F2 002   Dum(DFHDMPA ) Ope      NORMAL
*F2 002  RESPONSE: NORMAL TIME:  23.30.27   DATE:  83.070
 2 cemt perform snap title('User''s Sample Dump')
 F2 002
 F2 002   Sna
*F2 002  RESPONSE: NORMAL TIME:  23.31.25   DATE:  83.070
 2 cemt set dump switch
 F2 002
 F2 002   Dum(DFHDMPB ) Ope      NORMAL
*F2 002  RESPONSE: NORMAL TIME:  23.31.48   DATE:  83.070
```

```
 2 cstt aor
*F2 002  DFH1800 STATISTICS REQUEST IS BEING PROCESSED
 2 cemt inquire auxtrace
 F2 002
 F2 002   Aux(DFHAUXT ) Off Clo
*F2 002  RESPONSE: NORMAL TIME:  23.32.16   DATE:  83.070
 2 cemt set auxtrace on open
 F2 002
 F2 002   Aux(DFHAUXT ) On  Ope  NORMAL
*F2 002  RESPONSE: NORMAL TIME:  23.32.34   DATE:  83.070
 2 cemt set auxtrace off close
 F2 002
 F2 002   Aux(DFHAUXT ) Off Clo  NORMAL
*F2 002  RESPONSE: NORMAL TIME:  23.32.47   DATE:  83.070
 2 cemt set auxtrace switch
 F2 002
 F2 002   Aux(DFHBUXT ) Off Ope  NORMAL
*F2 002  RESPONSE: NORMAL TIME:  23.33.03   DATE:  83.070
 2 cemt perform shutdown
 F2 002
 F2 002 DFH1701 - C.I.C.S. IS BEING TERMINATED
 F2 002 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
 F2 002 EOJ CICS

 DATE 03/11/83,CLOCK 23/33/28,DURATION 00/10/34
```

Figure 19. Sample Transactions To Demonstrate Data Set Operation

OTHER OUTPUT

The output from the CICS run consists of the following:

1. A file of requested statistics and a file of termination statistics, both sent direct to the printer (SYSLST).
2. A snap dump of the CICS partition, which has been put to the primary data set DFHDMPA.
3. A CICS trace of activity during the short period of time between setting auxiliary trace on and off. This has been put to the primary auxiliary trace data set, DFHAUXT.

If they are still open, the data sets themselves are automatically closed when CICS is shut down. The data sets can now be processed by using the CICS-provided utility programs.

On the following pages are reproduced statistics, dump, and trace printouts produced by running the utility jobs in a batch partition. No job is required for statistics, because the output is directed to SYSLST.

The printed output from the previous run consists of a file of requested statistics followed by a file of termination statistics. In each of these, you should, for example, be able to find the following information:

1 Number of tasks

2 Which transactions were used

3 Which programs were used

4 Which terminals were used.

The <u>CICS/VS Performance Guide</u> describes the contents of each type of statistics.

```
     REQUESTED STATISTICS  03/11/83

*********************************** TASK CONTROL STATISTICS *************************************
            PEAK NUMBER OF TASKS                  1
            NUMBER OF TIMES AT MAX TASK           0
            TOTAL NUMBER OF TASKS                 7             1
            MAX. NO. ACTIVE TASKS REACHED         1

************************************** STORAGE STATISTICS ***************************************
            NUMBER OF STORAGE ACQUISITIONS          101
            NUMBER OF STORAGE RELEASES               69
            * NO STORAGE VIOLATIONS *

*********************************** TRANSACTION STATISTICS **************************************
         TRANSACTION  PROGRAM     TIMES      TIMES   TIMES ADD'L STORAGE      TIMES
         ID (L-LONG)    NAME    CALLED BY    STALL-  REQ'D FOR ANTICIPAT  TRANSACTION
           (S-SHORT)            TRANSACTION  PURGED    PAGING TASK         RESTARTED
          CEMT(L)     DFHEMTP          5       0                             0
          CSTE(L)     DFHTACP          1       0                             0           2
          CSTT(L)     DFHSTKC          1       0                             0
          *****TOTALS*****             7       0               0             0

************************************* PROGRAM STATISTICS ****************************************
                      PROGRAM      TIMES          TIMES
                        NAME    PROGRAM USED   PROGRAM FETCHED
                      DFHEITMT         5            1
                      DFHEMB           5            1
                      DFHEMTD          5            1
                      DFHEMTP          5            1                    3
                      DFHFELG          1            1
                      DFHOCP           3            1
                      DFHSTKC          1            1
                      DFHSTPD          1            1
                      DFHTACP          1            1
                      DFHTEP           1            1
                      DFHTEPT          1            1
          *****TOTALS*****            29           11

*************************************** DUMP STATISTICS *****************************************
               NUMBER OF STORAGE DUMPS          0

*********************************** TERMINAL STATISTICS *************************************
        LINE    NO OF   TERMINAL     INPUT      OUTPUT      TRANS- TRANS-      TRANS-       PIPELINE MESSAGES
         ID     POLLS      ID        MSGS        MSGS      MISSION ACTION      ACTION      ---  NOT PROCESSED  ---
                                                           ERRORS              ERRORS      TOTAL  GROUPS  MAX CONSEC
        0001      6
                          CNSL         6           6          0        6          0          4
        0002      1
                          SAMA         0           0          0        0          0
**TOTALS**        7                    6           6          0        6          0          0       0

********************************** TRANSIENT DATA STATISTICS **********************************
             DESTINATION EXTRAPARTITION INTRAPARTITION  INDIRECT  AUTOMATIC          REMOTE
                IDENT         REQUESTS       OUTPUTS     REQUESTS  TRANSACTION        REQUESTS
                                                                INITIATION OUTPUTS
              CPLI              66
              CSSL                                             59
              CSMT                                              8
              **TOTALS**        66             0               67          0              0
          NUMBER OF TRACKS/CIS USED BY INTRAPARTITION TRANSIENT DATA       0

**************************** TEMPORARY STORAGE STATISTICS ***************************************
              NUMBER OF RECORDS PUT/PUTQ MAIN                         0

              NUMBER OF RECORDS PUT TO UNIQUE ID S                    0

              MAX VIRT STRG USED FOR RECORDS                          0

              NUMBER OF CIS AVAILABLE FOR USE                       240

*************************** DYNAMIC TRANSACTION BACKOUT STATISTICS *****************************
              NUMBER OF RECORDS LOGGED BY DTB             0
              NUMBER OF RECORDS SPILLED BY DTB            0
     CICS ELAPSED TIME IS  0000 HOURS 10 MIN 06 SEC.  RELATIVE DAY IS    0
```

```
     REQUESTED STATISTICS  03/11/83

************************************ TASK CONTROL STATISTICS *************************************
            PEAK NUMBER OF TASKS                  1
            NUMBER OF TIMES AT MAX TASK           0
            TOTAL NUMBER OF TASKS                12             1
            MAX. NO. ACTIVE TASKS REACHED         1

*************************************** STORAGE STATISTICS ***************************************
            NUMBER OF STORAGE ACQUISITIONS           164
            NUMBER OF STORAGE RELEASES               117
            * NO STORAGE VIOLATIONS *

************************************ TRANSACTION STATISTICS *************************************
         TRANSACTION  PROGRAM      TIMES      TIMES   TIMES ADD'L STORAGE     TIMES
         ID (L-LONG)    NAME     CALLED BY    STALL-  REQ'D FOR ANTICIPAT  TRANSACTION
          (S-SHORT)             TRANSACTION   PURGED    PAGING TASK        RESTARTED
          CEMT(L)     DFHEMTP        10          0                             0
          CSTE(L)     DFHTACP         1          0                             0          2
          CSTT(L)     DFHSTKC         1          0                             0
          *****TOTALS*****           12          0                 0           0

************************************** PROGRAM STATISTICS **************************************
                      PROGRAM      TIMES         TIMES
                        NAME    PROGRAM USED   PROGRAM FETCHED
                      DFHEITMT       10           1
                      DFHEMB         10           1
                      DFHEMTD        10           1
                      DFHEMTP        10           1
                      DFHFELG         1           1
                      DFHJCSDJ        1           1                 3
                      DFHOCP          3           1
                      DFHSTKC         2           1
                      DFHSTLK         1           1
                      DFHSTP          1           1
                      DFHSTPD         2           1
                      DFHSTTD         1           1
                      DFHSTTR         1           1
                      DFHTACP         1           1
                      DFHTEP          1           1
                      DFHTEPT         1           1
          *****TOTALS*****           56          16

***************************************** DUMP STATISTICS ****************************************
               NUMBER OF STORAGE DUMPS        0

************************************ TERMINAL STATISTICS **************************************
       LINE    NO OF   TERMINAL     INPUT     OUTPUT      TRANS- TRANS-      TRANS-       PIPELINE MESSAGES
        ID     POLLS      ID         MSGS      MSGS       MISSION ACTION      ACTION      --- NOT PROCESSED ---
                                                          ERRORS              ERRORS      TOTAL  GROUPS  MAX CONSEC
       0001      11
                       CNSL           11        11         0        11        0              4
       0002       1
                       SAMA            0         0         0         0        0
**TOTALS**       12                   11        11         0        11        0           0       0

********************************* TRANSIENT DATA STATISTICS ***********************************
             DESTINATION EXTRAPARTITION INTRAPARTITION  INDIRECT  AUTOMATIC            REMOTE
                IDENT        REQUESTS       OUTPUTS     REQUESTS  TRANSACTION          REQUESTS
                                                                  INITIATION OUTPUTS
               CPLI            147
               CSSL                                         140
               CSMT                                           8
               **TOTALS**      147             0            148              0              0
       NUMBER OF TRACKS/CIS USED BY INTRAPARTITION TRANSIENT DATA           0

**************************** TEMPORARY STORAGE STATISTICS **************************************
               NUMBER OF RECORDS PUT/PUTQ MAIN                               0

               NUMBER OF RECORDS PUT TO UNIQUE ID S                          0

               MAX VIRT STRG USED FOR RECORDS                                0

               NUMBER OF CIS AVAILABLE FOR USE                             240

*************************** DYNAMIC TRANSACTION BACKOUT STATISTICS ****************************
               NUMBER OF RECORDS LOGGED BY DTB                0
               NUMBER OF RECORDS SPILLED BY DTB               0
```

Printing the Dump Data Set

After the CICS system has been shut down (or after a dump data set has become full and a switch to the alternate data set has taken place), the data set can be printed down using the CICS-provided DFHDUP utility program. This program formats the output on the printed page and, where appropriate, interprets the hexadecimal values in storage with their EBCDIC equivalents.

A sample job that can be used to print down a data set is shown below:

```
// JOB PRTDUMP
// DLBL DTFDISK,'CICS160.DUMP.DATASET.A',,SD
// EXTENT SYS001,,1,0,210,40
// ASSGN SYS001,DISK,TEMP,VOL=M3350A,SHR
// EXEC DFHDUP,SIZE=80K
 DEVICE=DISK
/*
/&
                (This job is included in Z.DFHSPJCL)
```

Figure 20. Processing the Dump Data Set

In the above sample job, the file name must be DTFDISK if disk dump data sets are being used. You may also need to add:

```
// LIBDEF CL,SEARCH=C160PCL
```

before the EXEC statement.

A description of the job control statements to be used when running DFHDUP and of the parameters that can be specified are given in "Chapter 4.9. Dump Data Sets."

Sample CICS Formatted Dump Output

The dump, only a portion of which has been reproduced here, can be divided into several parts. These include:

1 Short symptom string

2 Trace table containing a record of the flow within CICS

3 List of control blocks used in the CICS run

4 List of programs used in the CICS run.

The CICS/VS Problem Determination Guide, SC33-0163 describes a formatted dump in some detail.

```
CUSTOMER INFORMATION CONTROL SYSTEM FORMATTED DUMP                                    DATE=03/11/83   TIME=23:31:24   PAGE    1
FORMATTED DUMP INVOKED BY MASTER TERMINAL, TITLE = USER'S SAMPLE DUMP
SYMPTOMS= AB/UCSMT PIDS/5746XX300 FLDS/F000KC RIDS/DFHEMTD
CICS/VS LEVEL = 0160                                             1
005345E8   COMMON SYSTEM AREA
         004C CSACDTA  00570140     0074 CSAPLBA  00533800      0078 CSAPUBA  006C37FF      0080 CSATDTCA 005355C8
         008C CSASITBA 00658E08     011C CSATRTBA 00633DC0      0128 CSATCTBA 00533898      012C CSAFCTBA 00000000
         0130 CSADCTBA 005661A0
00000000   00000000 00000000 00000000 5055A850  00000003 C0000800 0056E4B0 00534120   *..............Y...........U.....*
00000020   5061B022 0053441E 50554442 A05544D2  005708B0 005349E8 00000008 E2C1D4C1   *./.............K.......Y....SAMA*
00000040   00533898 00570140 0011999C 00570140  2331241F 00000000 07D00100 00000000   *...Q... ..R.....................*
00000060   008137BB 0005DC00 00000000 000148E8  00003000 00533800 006C37FF 0083070F   *.A.............Y.............C..*
00000080   005355C8 F0FFFFFE 0000004B 00658E08  00000000 00000000 0056AB50 C513C616   *...H0.......................E.F.*
000000A0   00000000 00000000 00534684 00534684  0056F800 00534CA0 00534E70 C500FF00   *...........D...D..8.......+.E...*
000000C0   000A0001 000C0400 005349E8 00000000  00000000 00000000 00000000 00000000   *...........Y....................*
000000E0   405362B0 0055A540 0055E800 036574F0  0064DCDC 005339D0 00563E44 0056655C   * .....V ..Y....0...............**
00000100   005643BC 0064F78A 0063B774 00630888  00000000 0053CDBC 01541920 00633DC0   *......7.........H...............*
00000120   00000000 00000000 00533898 00000000  005661A0 0056A820 0056F000 00000000   *...........Q....../...Y...0.....*
00000140   00000000 406308B0 00534410 00000000  005660A4 006302E0 00000000 00000000   *.... ...............-U..........*
00000160   006422C2 00000000 00000000 3A96511A  0181709C 02000000 00000000 006303F0   *...B.........O...A.............0*
00000180   00000000 00000000 00000000 0065A2D8  00000000 00000000 00000000 FF534598   *...............SQ..............Q*
000001A0   07FE58F0 D19C07FF 0183A733 0183A77C  00096000 000C0000 E6D6D9D2 C1D9C5C1   *...0J....CX..CX...-.....WORKAREA*
000001C0   0000000C 000C0000 001C001C 00005C00  00082C00 00046C00 0C000C00 0C000C00   *...............*................*
000001E0   0C000C00 000C0000 0C000C00 0C00000C  00000C00 000C0000 0C00000C 00000C00   *................................*
00000200   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
00000220   LINES TO 000003E0 SAME AS ABOVE
005349E8   OPTIONAL FEATURES LIST
         0000 CSAATP   00000000     0008 CSADLI     00000000     0034 CSAJCTBA 00000000
00000000   00000000 00000000 00000000 00655150  005684F4 00000000 00656E58 00656E88   *....................D4..........H*
00000020   006421C4 00641EC0 206417E4 00000000  00000000 00000000 00000000 00000000   *...D.......U....................*
00000040   00000000 00000000 00000000 00000000  0063BDF6 00000000 00000000 00000000   *...................6............*
00000060   80000000 01F4D5D6 00000000 0000FE1C  0064D440 0083070F 00000000 005345E8   *.....4NO...........M .C.........Y*
00000080   00536068 0064CFC0 0064A9A0 00643030  00000000 00000000 00000000 00000000   *..-........Z....................*
000000A0   00000000 00000000 0053B33C 0053A808  005401D8 00563350 005662C0 00564020   *...............Y....Q.......... .*
000000C0   0055A420 00560230 00658440 00536208  00656EC0 0053CC18 00567C60 0063B5E0   *..U.......D ...............-....*
000000E0   0064EF80 0064D490 006567D0 005673C0  00000000 00000000 00000000 0053F828   *......M......................8.*
00000100   005532B8 0064CE00 0064CB00 00000000  00000000 00000000 0054150C 00000000   *................................*
00000120   00554378 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
00000140   00000000 00000000 00000000 00000000  00000000 006401E0 0055DD24 00000000   *................................*
00000160   00000000 00558024 00637E10 0053FEB8  005393B8 00000000 00000000 00000000   *..........=.......L.............*
00000180   00633CE0 0055F960 00000000 00000000  00000000 00000000 00536008 00534C28   *......9-....................-....*
000001A0   00633C90 00534228 00000000 00000000  00000000 00630448 00000000 00000000   *................................*
000001C0   00000000 00000000 00000000 00001000  00000000 00000000 00000000 00000000   *................................*
000001E0   00000000 00000000 00000000 00000000  00000000 00000000 953A9642 1B0E2000   *...........................N.O.....*
00000200   953A9642 1B0E2000 00000000 00000000  00659288 00000000 00000000 00000000   *N.O...............KH............*
00000220   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
00534C28   CLASS MAXIMUM TASK COUNTERS
00000000   00000000 00000000 0000010C 000C000C  000C000C 010C000C 000C000C 000C010C   *................................*
00000020   000C000C 000C000C 010C000C 000C000C  000C010C 000C000C 000C000C 010C000C   *................................*
00000040   000C000C 000C010C 000C000C 000C000C  010C000C 000C000C 000C010C 000C000C   *................................*
00000060   000C000C 010C000C 000C000C 000C                                            *..............                  *
00630448   PROGRAM CHECK / ABEND TRACE TABLE
00000000   D7D9D6C7 D9C1D440 C3C8C5C3 D261C1C2  C5D5C440 E3D9C1C3 C540E3C1 C2D3C57A   *PROGRAM CHECK/ABEND TRACE TABLE:*
00000020   00630488 00630488 00630788 00000000  00000000 00000340 00000000 00000000   *...H...H...H........... ........*
00000040   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
00000060   LINES TO 00000320 SAME AS ABOVE
00658E08   SYSTEM INITIALISATION TABLE
00000000   C513C616 00000000 04000200 00000000  00000000 00659220 006599A0 00659240   *E.F...................K...R...K *
00000020   00659208 0064256C 00000000 00000000  0005DC00 00096000 07D0004B 0000000C   *..K.....................-.........*
00000040   00000000 00003000 00000800 00000200  00040000 01F4FE00 000003E8 03E80000   *......................4.....Y.Y..*
00000060   00000000 00000000 C4C2C4C3 C3C9C3E2  40404040 40D5F020 C1406CF1 E840D501   *........DBDCCICS      N0.A .1Y N.*
00000080   40D54040 10D50000 00000000 00000000  00C3C9C3 E2D7E2C2 C3C9C3E2 E3D9D5D5   * N  .N............CICSPSBCICSTRNN*
000000A0   40404040 40404040 00404040 40404040  00404040 40404040 00404040 40404040   *        .       .       .       *
000000C0   00404040 40404040 40400000 0C004040  D5D6F15B D5D6D5D6 D5D6F15B D5D6D5D6   *.         ....  NO1$NONONO1$NONO*
000000E0   F15BF15B F15BD5D6 D5D6D5D6 40404040  40404040 4040F35B C45B4040 E25BE25B   *1$1$1$NONONO          3$D$  S$S$*
00000100   4040F65B F25BC45B F15B4040 D5D6D5D6  D5D6D5D6 D5D6D5D6 D5D6D5D6 F15BD5D6   *  6$2$D$1$  NONONONONONONONO1$NO*
00000120   40404040 D5D6D5D6 D5D6D5D6 D5D6D5D6  D5D64040 40404040 4040D5D6 4040D5D6   *    NONONONONONONO          NO  NO*
00000140   4040D5D6 40404040 40404040 40404040  40404040 40404040 40404040 40404040   *  NO                            *
00000160   40C34040 00400000 40404040 00400000  000A999C 010C010C 010C010C 010C010C   * C  . ..    . ....R.............*
00000180   010C010C 010C010C 000C000C 000C000C  40404040 40404040 40404040 40404040   *................                *
000001A0   40404040 40404040 40404040 40404040  40404040 40404040 40404040 40404040   *                                *
000001C0   LINES TO 000003E0 SAME AS ABOVE
00659208   INITIALISATION MODULES
```

```
00000000   00180018 C1F1C2F1 C3F1C4F1 C5F1C6F1  C7F1C8F1 C9F1D1F1                     *....A1B1C1D1E1F1G1H1I1J1       *
00659240   OVERRIDE PARAMETERS
00000000   0023E2C9 E37EF15B 6BE3E2D7 7EF25B6B  E3C4D77E F65B6BC4 C3D77EC4 5B6BE3D9   *..SIT=1$,TSP=2$,TDP=6$,DCP=D$,TR*
00000020   D77EC45B 6B                                                                *P=D$,                          *
00659220   ATP CSA EXTENSION
00000000   00000000 03E803E8 001C998C 000C0000  0C                                    *.....Y.Y..R......              *
006599A0   SIP COMMUNICATIONS AREA
00000000   00005000 00659C08 00659DE8 00659F98  0065A2D8 0065A522 0065A654 005345E8   *...........Y...Q..SQ..V...W....Y*
00000020   00658E08 00000000 00659240 00659C08  00000000 00561940 0065E800 00000000   *..........K ........... ..Y.....*
00000040   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
00000060   406308B0 00570130 0055E800 8055E99E  00633A70 0055D024 00570000 00573200   * .........Y...Z.................*
00000080   00561940 00000144 005732E0 8055F112  00570140 005345E8 5055E278 00658E08   *... ...........1.... ...Y..S.....*
000000A0   005350C0 0065BC7C 00535110 0065AC40  006599A0 00658E08 00534D30 00000000   *............... ..R.......(.....*
000000C0   005349E8 0065BC40 8065BED4 0056AB50  00534E70 005345E8 5065B67A 005362B0   *...Y... ...M......+....Y...:....*
000000E0   00F14400 00534020 00000000 00000F13  00000018 0065AC28 00000000 00000000   *.1.... .........................*
00000100   00000000 00000080 00569800 00568800  00000FF9 00000000 002C8000 40404040   *...........Q...H....9........   *
00000120   40404040 40404040 40404040 40404040  40404040 40404040 40404040 40404040   *                                *
00000140   LINES TO 000001E0 SAME AS ABOVE
00000200   40404040 40404040 40404040 C4C6C8E2  C9D1F140 00930B0D 000701E2 4A000320   *            DFHSIJ1 .L.....S....*
00000220   78032090 00120003 20780000 0000009A  2ED44000 00000000 0065AC28 000019E2   *.................M .............S*
00000240   00000018 00000000 00000000 006598A0  00659920 0065A09C 00000000 00534650   *...............Q...R.............*
00000260   00000000 000000F0                                                          *.......0                       *
00659920   SUBTASK SAVE AREA
00000000   C4C6C8D1 C3E2E4C2 00000000 00000000  00000000 00000000 00000000 00000000   *DFHJCSUB........................*
00000020   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
00000040   LINES TO 00000060 SAME AS ABOVE
006598A0   SUBTASK SAVE AREA
00000000   C4C6C8D3 D6C1C4D9 076D0000 0065A0BC  0065AC28 00659265 00000000 00000445   *DFHLOADR...............K.........*
00000020   0065934C 6065B610 00000000 00000000  00659BE4 00000001 00659C08 006599A0   *..L.-...............U..........R.*
00000040   00658E08 00659BE4 00000000 0065B7BD  0000953A 96384370 00000000 00000000   *.......U..........N.O..........*
00000060   00000000 00000000 00000000 00000000  00000000 00000000 30C9E37E F15B6BE3   *..........................IT=1$,T*
00633DC0   TRACE TABLE
00000000   00635C40 00633DE0 00637DC0 00000000  008118FF AE56990F 00000000 00000000   *..* ......'......A....R..........*
   TIME OF DAY    ID  REG 14  REQD TASK     FIELD A  FIELD B  CHARS    RESOURCE     TRACE TYPE
23:23:14.620864   F0 4055E6E4 4004 00001 88000000 00562BAE ........              KCP WAIT DCI=CICS
23:23:14.721280   F1 4061C0DC 8C04 00001 00610064 20533F14 ./......              SCP GETMAIN
23:23:14.721408   C8 5055A850 0004 00001 005708B0 8C610078 ...../..              SCP ACQUIRED USER STORAGE
23:23:14.786176   F6 6061BF84 4003 00001 005708B8 00000000 ........ CSMT    2     TDP PUT
23:23:14.786240   EA 40566678 0003 00001 01000600 005701C4 .......D CSMT          TMP DCT LOCATE
23:23:14.786400   EA 405590E8 0055 00001 01000600 00566204 ........              TMP RETN NORMAL
23:23:14.788640   F0 40566846 4004 00001 20000000 00562BAE ........              KCP WAIT DCI=DISP
   TIME OF DAY    ID  REG 14  REQD TASK     FIELD A  FIELD B  CHARS    RESOURCE     TRACE TYPE
23:23:14.788800   F6 40566864 0055 00001 01000600 00566204 ........              TDP RETN NORMAL
23:23:14.788896   E6 7061C3F2 1001 00001 00000085 E2C1D4C1 ....SAMA              TACP
23:23:14.788928   F2 5061C410 8104 00001 00000000 00000000 ........ DFHFELG       PCP LINK-CONDITIONAL
23:23:14.788992   EA 4055ED36 0003 00001 01000300 005701C4 .......D DFHFELG       TMP PPT LOCATE
23:23:14.789120   EA 405590E8 0005 00001 01000300 00560F68 ........              TMP RETN NORMAL
23:23:14.789152   F1 4055E96E 8904 00001 00570050 0056E4B0 ...&..U.              SCP GETMAIN
23:23:14.789280   C8 5055A850 0004 00001 00570930 89570058 ........              SCP ACQUIRED RSA STORAGE
23:23:14.837056   F1 4055E642 8804 00001 00570072 0056E4B0 ......U.              SCP GETMAIN
23:23:14.840224   C8 5055A850 0004 00001 0061B000 88000800 ./......              SCP ACQUIRED PGM STORAGE
23:23:14.841408   F0 4055E6E4 4004 00001 88000000 00560F8E ........              KCP WAIT DCI=CICS
23:23:14.894912   FC 4061B074 0204 00001 01010000 E2C1D4C1 ....SAMA              ZCP ZLOC LOC REQ ID LOCAL
23:23:14.895008   F2 4061B348 1004 00001 00000000 00000000 ........ DFHFELG       PCP RETURN
23:23:14.895072   F1 6055ED9E 4004 00001 00570930 0056E4B0 ......U.              SCP FREEMAIN
23:23:14.895104   C9 5055A886 0004 00001 00570930 89570058 ........              SCP RELEASED RSA STORAGE
23:23:14.895200   F2 5061C424 8104 00001 00000000 00000000 ........ DFHTEP        PCP LINK-CONDITIONAL
23:23:14.895264   EA 4055ED36 0003 00001 01000300 005701C4 .......D DFHTEP        TMP PPT LOCATE
23:23:14.895392   EA 405590E8 0005 00001 01000300 00562BD0 ........              TMP RETN NORMAL
23:23:14.895424   F1 4055E96E 8904 00001 00570050 0056E4B0 ...&..U.              SCP GETMAIN
23:23:14.895456   C8 5055A850 0004 00001 00570930 89570058 ........              SCP ACQUIRED RSA STORAGE
23:23:14.914112   F1 4055E642 8804 00001 00570164 0056E4B0 ......U.              SCP GETMAIN
23:23:14.915936   C8 5055A850 0004 00001 0061A000 88001000 ./......              SCP ACQUIRED PGM STORAGE
23:23:14.920864   F0 4055E6E4 4004 00001 88000000 00562BF6 .......6              KCP WAIT DCI=CICS
23:23:14.972544   F0 40541660 4004 TCP   44000000 00534400 ........              KCP WAIT DCI=TCP
23:23:14.991296   F3 5061A044 1003 00001 1061A008 00000000 ./......              ICP GETIME
23:23:14.991360   F3 40657626 0065 00001 0061A000 88001000 ./......              ICP RETN NORMAL
23:23:14.991392   F2 5061A058 A404 00001 00000000 00000000 ........ DFHTEPT       PCP LOAD-CONDITIONAL
23:23:14.991456   EA 4055ED36 0003 00001 01000300 005701C4 .......D DFHTEPT       TMP PPT LOCATE
23:23:14.991584   EA 405590E8 0005 00001 01000300 00562C60 .......-              TMP RETN NORMAL
23:23:14.998112   F0 4055E6E4 4004 00001 88000000 00562C86 ........              KCP WAIT DCI=CICS
23:23:15.065184   F6 5061A850 4003 00001 005708B8 00000000 ........ CSMT          TDP PUT
23:23:15.065248   EA 40566678 0003 00001 01000600 005701C4 .......D CSMT          TMP DCT LOCATE
23:23:15.065376   EA 405590E8 0055 00001 01000600 00566204 ........              TMP RETN NORMAL
23:23:15.067616   F0 40566846 4004 00001 20000000 00562C86 ........              KCP WAIT DCI=DISP
23:23:15.067712   F6 40566864 0055 00001 01000600 00566204 ........              TDP RETN NORMAL
23:23:15.067776   F6 5061A850 4003 00001 005708B8 00000000 ........ CSMT          TDP PUT
23:23:15.067808   EA 40566678 0003 00001 01000600 005701C4 .......D CSMT          TMP DCT LOCATE
23:23:15.067904   EA 405590E8 0055 00001 01000600 00566204 ........              TMP RETN NORMAL
23:23:15.071936   F0 40566846 4004 00001 20000000 00562C86 ........              KCP WAIT DCI=DISP
23:23:15.072032   F6 40566864 0055 00001 01000600 00566204 ........              TDP RETN NORMAL
23:23:15.072224   F6 5061A850 4003 00001 005708B8 00000000 ........ CSMT          TDP PUT
23:23:15.072288   EA 40566678 0003 00001 01000600 005701C4 .......D CSMT          TMP DCT LOCATE
23:23:15.072352   EA 405590E8 0055 00001 01000600 00566204 ........              TMP RETN NORMAL
23:23:15.074272   F0 40566846 4004 00001 20000000 00562C86 ........              KCP WAIT DCI=DISP
23:23:15.074368   F6 40566864 0055 00001 01000600 00566204 ........              TDP RETN NORMAL
23:23:15.074560   F6 5061A850 4003 00001 005708B8 00000000 ........ CSMT          TDP PUT
23:23:15.074624   EA 40566678 0003 00001 01000600 005701C4 .......D CSMT          TMP DCT LOCATE
23:23:15.074688   EA 405590E8 0055 00001 01000600 00566204 ........              TMP RETN NORMAL
23:23:15.076608   F0 40566846 4004 00001 20000000 00562C86 ........              KCP WAIT DCI=DISP
23:23:15.076704   F6 40566864 0055 00001 01000600 00566204 ........              TDP RETN NORMAL
23:23:15.076864   F6 5061A850 4003 00001 005708B8 00000000 ........ CSMT          TDP PUT
23:23:15.076928   EA 40566678 0003 00001 01000600 005701C4 .......D CSMT          TMP DCT LOCATE
23:23:15.076992   EA 405590E8 0055 00001 01000600 00566204 ........              TMP RETN NORMAL
23:23:15.078944   F0 40566846 4004 00001 20000000 00562C86 ........              KCP WAIT DCI=DISP
23:23:15.079040   F6 40566864 0055 00001 01000600 00566204 ........              TDP RETN NORMAL
23:23:15.079264   F6 5061A850 4003 00001 005708B8 00000000 ........ CSMT          TDP PUT
23:23:15.079296   EA 40566678 0003 00001 01000600 005701C4 .......D CSMT          TMP DCT LOCATE
23:23:15.079392   EA 405590E8 0055 00001 01000600 00566204 ........              TMP RETN NORMAL
23:23:15.081312   F0 40566846 4004 00001 20000000 00562C86 ........              KCP WAIT DCI=DISP
23:23:15.081408   F6 40566864 0055 00001 01000600 00566204 ........              TDP RETN NORMAL
23:23:15.081472   F6 5061A850 4003 00001 005708B8 00000000 ........ CSMT          TDP PUT
```

```
23:23:15.081536  EA 40566678 0003 00001 01000600 005701C4 .......D CSMT      TMP DCT LOCATE
23:23:15.081600  EA 405590E8 0055 00001 01000600 00566204 ........           TMP RETN NORMAL
23:23:15.083552  F0 40566846 4004 00001 20000000 00562C86 ........           KCP WAIT DCI=DISP
23:23:15.083648  F6 40566864 0055 00001 01000600 00566204 ........           TDP RETN NORMAL
23:23:15.083680  F2 4061A188 1004 00001 00000000 00000000 ........ DFHTEP    PCP RETURN
23:23:15.083744  F1 6055ED9E 4004 00001 00570930 0056E4B0 ......U.           SCP FREEMAIN
23:23:15.083808  C9 5055A886 0004 00001 00570930 89570058 ........           SCP RELEASED RSA STORAGE
23:23:15.083904  E6 7061C452 2001 00001 00FF8805 00000000 ........           TACP
23:23:15.083968  F1 4061B908 4004 00001 0056E4B0 00533F14 ..U.....           SCP FREEMAIN
23:23:15.084000  C9 5055A886 0004 00001 0056E4B0 96000040 ..U....            SCP RELEASED TACLE STORAGE
23:23:15.084032  F2 5061B91C 1004 00001 00000000 00000000 ........ DFHTACP   PCP RETURN
23:23:15.084096  F0 4055EB0C 8004 00001 00000000 00000000 ........           KCP DETACH
23:23:15.084160  D8 60536EE0 0203 00001 02000888 00570880 ........           SPP SYSTEM
23:23:15.084224  F1 40656E7C DB04 00001 00000085 00533F14 ........           SCP GETMAIN INITIMG
23:23:15.084256  C8 5055A850 0004 00001 00570930 9B000098 ........           SCP ACQUIRED JCA STORAGE
23:23:15.084352  EA 6053D4D0 0003 00001 0D000000 47F09106 .....0..           TMP DWE
23:23:15.084448  EA 405590E8 0005 00001 0D000000 50E020F4 ....&..4           TMP RETN NORMAL
23:23:15.084480  D8 4053D71A 0005 00001 0D000000 50E020F4 ....&..4           SPP RETN
23:23:15.084512  F0 50536EEA 0304 00001 00562C86 00000000 ........           KCP DEQALL
23:23:15.084576  D0 505377B4 0704 00001 04000400 C3E2E3C5 ....CSTE           KCP TERMINATE
23:23:15.084608  F1 4053708A 4A04 KCP   00570000 00000000 ........           SCP FREEMAIN
23:23:15.084640  C9 5055A886 0004 KCP   00570000 8A040878 ........           SCP RELEASED TCA STORAGE
23:23:15.245536  F0 40541660 4004 TCP   44000000 00534400 ........           KCP WAIT DCI=TCP
23:23:20.251712  FD 0000013C 0104 TCP   00807E02 0080957A ..=.....           ... REPEAT 00013 TIMES
23:24:21.827424  F1 40545B66 E504 TCP   00000052 80534020 ...... .           SCP GETMAIN CONDITIONAL INITIMG
23:24:21.828064  C8 5055A850 0004 TCP   0056E5E0 85000068 ..Y.....           SCP ACQUIRED TERMINAL STORAGE
23:24:21.828096  F1 40545BB8 E404 TCP   0000005D 80534020 ...)... .          SCP GETMAIN CONDITIONAL INITIMG
23:24:21.828160  C8 5055A850 0004 TCP   0056E950 84000068 ..Z&....           SCP ACQUIRED LINE STORAGE
23:24:21.831488  F0 40541660 4004 TCP   44000000 00534400 ........           KCP WAIT DCI=TCP
23:24:22.154432  FD 0000013C 0104 TCP   00809630 0081172D ........           ... REPEAT 00013 TIMES
23:29:52.798784  F0 50551E42 1304 TCP   C3D5E2D3 C3C5D4E3 CNSLCEMT           KCP ATTACH-CONDITIONAL
23:29:52.798912  EA 705368A2 0003 TCP   01040100 00534F08 ........ CEMT      TMP PCT LOCATE
23:29:52.799040  EA 405590E8 0005 TCP   01040100 0055CFC4 .......D           TMP RETN NORMAL
23:29:52.799136  F1 40536A00 EA04 TCP   000006C0 80534020 ...... .           SCP GETMAIN CONDITIONAL INITIMG
23:29:52.802272  C8 5055A850 0004 TCP   00570000 8A0408C8 .......H           SCP ACQUIRED TCA STORAGE
23:29:52.802464  D0 505377B4 0604 00002 C3D5E2D3 C3C5D4E3 CNSLCEMT           KCP CREATE
23:29:52.802560  EA 40536D94 0003 TCP   0C000100 0055CFC4 .......D CEMT      TMP PCT TRANSFER
23:29:52.802688  EA 405590E8 0005 TCP   0C000100 00000000 ........           TMP RETN NORMAL
23:29:52.802912  F0 40541660 4004 TCP   44000000 00534400 ........           KCP WAIT DCI=TCP
23:29:52.803392  FC 055434F8 0503 00002 00000000 00534020 ...... .           ZCP ZSUP START UP TASK
23:29:52.814592  E5 405435AA 0C03 00002 00000000 00000000 ........ CEMT.*..  XSP SECURITY
23:29:52.834336  F2 50543CE0 8804 00002 00000000 00000000 ........ DFHEMTP   PCP XCTL-CONDITIONAL
23:29:52.834432  EA 4055ED36 0003 00002 01000300 005701C4 .......D DFHEMTP   TMP PPT LOCATE
23:29:52.834592  EA 405590E8 0005 00002 01000300 00561988 ........           TMP RETN NORMAL
23:29:52.839136  F0 4055E6E4 4004 00002 88000000 005619AE ........           KCP WAIT DCI=CICS
23:29:52.906464  F1 4053B388 CC04 00002 00000230 01534020 ...... .           SCP GETMAIN INITIMG
23:29:52.906592  C8 5055A850 0004 00002 005708D0 8C000238 ........           SCP ACQUIRED USER STORAGE
23:29:52.942880  F1 6053B7C4 CC04 00002 000001DB 01534020 ...... .           SCP GETMAIN INITIMG
23:29:52.943040  C8 5055A850 0004 00002 00570B10 8C0001E8 .......Y           SCP ACQUIRED USER STORAGE
23:29:52.943072  E1 5062F32E 0004 00002 00570B20 00000208 ........           EIP ASSIGN ENTRY
23:29:52.943168  E1 5062F32E 00F4 00002 00000000 00000208 ........           EIP ASSIGN RESPONSE
23:29:52.943200  E1 5062F428 0004 00002 00570B20 00000C02 ........           EIP GETMAIN ENTRY
23:29:52.943232  F1 4055A498 0004 00002 C00C0000 9D570934 ........           SCP GETMAIN INITIMG
23:29:53.005952  C8 5055A850 0004 00002 00570D00 8C0021F8 .......8           SCP ACQUIRED USER STORAGE
23:29:53.006048  E1 5062F428 00F4 00002 00000000 00000C02 ........           EIP GETMAIN RESPONSE
23:29:53.006176  E1 4062F4E8 0004 00002 00570B20 0000020A ........           EIP IGNORE-CONDITION ENTRY
23:29:53.006272  F1 6053BEC0 CC04 00002 0000009E 01534020 ...... .           SCP GETMAIN INITIMG
23:29:53.006336  C8 5055A850 0004 00002 00572F00 8C0000A8 ........           SCP ACQUIRED USER STORAGE
23:29:53.006368  E1 4062F4E8 00F4 00002 00000000 0000020A ........           EIP IGNORE-CONDITION RESPONSE
23:29:53.006400  E1 5062F516 0004 00002 00570B20 00000402 ........           EIP RECEIVE-TC ENTRY
23:29:53.110976  E1 5062F516 00F4 00002 00000000 00000402 ........           EIP RECEIVE-TC RESPONSE
23:29:53.111200  E1 5062F6D0 0004 00002 00570B20 00000E06 ........           EIP LOAD ENTRY
23:29:53.111264  F2 60560258 8204 00002 00000000 00000000 ........ DFHEITMT  PCP LOCATE
23:29:53.111360  EA 4055ED36 0003 00002 01000300 005701C4 .......D DFHEITMT  TMP PPT LOCATE
23:29:53.111488  EA 405590E8 0005 00002 01000300 005619D0 ........           TMP RETN NORMAL
23:29:53.111520  F2 4053C144 0404 00002 00000000 00000000 ........ DFHEITMT  PCP LOAD
23:29:53.111584  EA 4055ED36 0003 00002 01000300 005701C4 .......D DFHEITMT  TMP PPT LOCATE
23:29:53.111616  EA 405590E8 0005 00002 01000300 005619D0 ........           TMP RETN NORMAL
23:29:53.113408  F0 4055E6E4 4004 00002 88000000 005619F6 .......6           KCP WAIT DCI=CICS
23:29:53.113760  F0 40541660 4004 TCP   44000000 00534400 ........           KCP WAIT DCI=TCP
23:29:53.265472  E1 5062F6D0 00F4 00002 00000000 00000E06 ........           EIP LOAD RESPONSE
23:29:53.265568  E1 5062F704 0004 00002 00570B20 00000E02 ........           EIP LINK ENTRY
23:29:53.265632  F2 60560258 8204 00002 00000000 00000000 ........ DFHEMTD   PCP LOCATE
23:29:53.265696  EA 4055ED36 0003 00002 01000300 005701C4 .......D DFHEMTD   TMP PPT LOCATE
23:29:53.265856  EA 405590E8 0005 00002 01000300 00561B80 ........           TMP RETN NORMAL
23:29:53.265920  F1 6053BEC0 CC04 00002 000000F0 01534020 ...0.. .           SCP GETMAIN INITIMG
23:29:53.267872  C8 5055A850 0004 00002 00572FB0 8C0000F8 .......8           SCP ACQUIRED USER STORAGE
23:29:53.268032  F2 5053C144 8104 00002 00000000 00000000 ........ DFHEMTD   PCP LINK-CONDITIONAL
23:29:53.268128  EA 4055ED36 0003 00002 01000300 005701C4 .......D DFHEMTD   TMP PPT LOCATE
23:29:53.268256  EA 405590E8 0005 00002 01000300 00561B80 ........           TMP RETN NORMAL
23:29:53.268320  F1 4055E96E 8904 00002 00570050 01534020 ...&.. .           SCP GETMAIN
23:29:53.268352  C8 5055A650 0004 00002 005730B0 69570058 ........           SCP ACQUIRED RSA STORAGE
23:29:53.270464  F0 4055E6E4 4004 00002 88000000 00561BA6 ........           KCP WAIT DCI=CICS
23:29:53.512352  F0 40541660 4004 TCP   44000000 00534400 ........           KCP WAIT DCI=TCP
23:29:53.934624  F1 6053B7C4 CC04 00002 00000232 01534020 ...... .           SCP GETMAIN INITIMG
23:29:53.934752  C8 5055A650 0004 00002 00573110 8C000248 ........           SCP ACQUIRED USER STORAGE
23:29:54.036896  E1 40621760 0004 00002 00573120 00008404 ........           EIP ENTRY
23:29:54.036960  F2 5053BA9E 0404 00002 00000000 00000000 ........ DFHEMB    PCP LOAD
23:29:54.037056  EA 4055ED36 0003 00002 01000300 005701C4 .......D DFHEMB    TMP PPT LOCATE
23:29:54.037216  EA 405590E8 0005 00002 01000300 00561C10 ........           TMP RETN NORMAL
23:29:54.074720  F1 4055E642 8804 00002 00570130 01534020 ...... .           SCP GETMAIN
23:29:54.076640  C8 5055A850 0004 00002 00619000 88001000 ./......           SCP ACQUIRED PGM STORAGE
23:29:54.079680  F0 4055E6E4 4004 00002 88000000 00561C36 ........           KCP WAIT DCI=CICS
23:29:54.141664  F1 6053BEC0 CC04 00002 000000C4 01534020 ...D.. .           SCP GETMAIN INITIMG
23:29:54.141792  C8 5055A850 0004 00002 00573360 8C0000D8 ...-...Q           SCP ACQUIRED USER STORAGE
23:29:54.141888  F2 5053BCDA 0804 00002 00000000 00000000 ........ DFHEMB    PCP DELETE
23:29:54.141952  EA 4055ED36 0003 00002 01000300 005701C4 .......D DFHEMB    TMP PPT LOCATE
23:29:54.142112  EA 405590E8 0005 00002 01000300 00561C10 ........           TMP RETN NORMAL
23:29:54.142144  E1 40621760 00F4 00002 00000000 00008404 ........           EIP RESPONSE
23:29:54.142816  E1 50621C92 0004 00002 00573120 00001002 ........           EIP ASKTIME ENTRY
```

```
23:29:54.142880   F3 6065848E 1303 00002 13570A30 00000000 ........          ICP GETIME
23:29:54.142944   F3 40657626 0055 00002 00573120 00001002 ........          ICP RETN NORMAL
23:29:54.142976   E1 50621C92 00F4 00002 00000000 00001002 ........          EIP ASKTIME RESPONSE
23:29:54.143040   E1 50620EEC 0004 00002 00573120 00000404 ........          EIP SEND-TC ENTRY
23:29:54.143168   F1 405406DE C304 00002 0000005F 00534020 ...... .          SCP GETMAIN INITIMG
23:29:54.143232   C8 5055A850 0004 00002 0056E9C0 85000078 ..Z.....          SCP ACQUIRED TERMINAL STORAGE
23:29:54.187360   FC 50540B70 0103 00002 00810000 00534020 ...... .          ZCP ZARQ APPL REQ ERASE WRITE
23:29:54.187424   FC 40541C3A 0105 00002 00810000 0056E9C0 ......Z.          ZCP RETN ZARQ APPL REQ ERASE WRITE
23:29:54.187584   E1 50620EEC 00F4 00002 00000000 00000404 ........          EIP SEND-TC RESPONSE
23:29:54.187712   F2 6055F3AC 1004 00002 00000000 00000000 ........ DFHEMTD  PCP RETURN
23:29:54.187776   F1 6055EFA2 4004 00002 00573110 01534020 ...... .          SCP FREEMAIN
23:29:54.187872   C9 5055A886 0004 00002 00573110 8C000248 ...... .          SCP RELEASED USER STORAGE
23:29:54.187904   F1 6055ED9E 4004 00002 005730B0 01534020 ...... .          SCP FREEMAIN
23:29:54.187936   C9 5055A886 0004 00002 005730B0 89570058 ........          SCP RELEASED RSA STORAGE
23:29:54.188064   F1 5053BEC0 4004 00002 00572FB0 01534020 ...... .          SCP FREEMAIN
23:29:54.188096   C9 5055A886 0004 00002 00572FB0 8C0000F8 .......8          SCP RELEASED USER STORAGE
23:29:54.188160   E1 5062F704 00F4 00002 00000000 00000E02 ........          EIP LINK RESPONSE
23:29:54.188192   E1 5062F736 0004 00002 00570B20 00000C04 ........          EIP FREEMAIN ENTRY
23:29:54.188256   F1 4055A4FA 0004 00002 40000000 9D570934  .......          SCP FREEMAIN
23:29:54.188288   C9 5055A886 0004 00002 00570D00 8C0021F8 .......8          SCP RELEASED USER STORAGE
23:29:54.190720   E1 5062F736 00F4 00002 00000000 00000C04 ........          EIP FREEMAIN RESPONSE
23:29:54.190912   F2 4055F3AC 1004 00002 00000000 00000000 ........ DFHEMTP  PCP RETURN
23:29:54.190944   F1 6055EFA2 4004 00002 00570B10 01534020 ...... .          SCP FREEMAIN
23:29:54.191008   C9 5055A886 0004 00002 00570B10 8C0001E8 .......Y          SCP RELEASED USER STORAGE
23:29:54.191072   F0 4055EB0C 8004 00002 00000000 00000000 ........          KCP DETACH
23:29:54.191200   F0 40536EEA 0304 00002 00561C36 00000000 ........          KCP DEQALL
23:29:54.191328   D0 505377B4 0704 00002 04000400 C3C5D4E3 ....CEMT          KCP TERMINATE
23:29:54.191360   FC 50536F3E 0B03 00002 04000000 01534020 ...... .          ZCP ZISP ISC FREE DETACH
23:29:54.237024   FC 40545510 0B05 00002 04000000 00534020 ...... .          ZCP RETN ZISP ISC FREE DETACH
23:29:54.237056   F1 4053708A 4A04 KCP   00570000 00000000 ........          SCP FREEMAIN
23:29:54.237120   C9 5055A886 0004 KCP   00573360 8C0000D8 ...-...Q          SCP RELEASED USER STORAGE
23:29:54.238208   C9 5055A886 0004 KCP   00572F00 8C0000A8 ........          SCP RELEASED USER STORAGE
23:29:54.239328   C9 5055A886 0004 KCP   005708D0 8C000238 ........          SCP RELEASED USER STORAGE
23:29:54.239392   C9 5055A686 0004 KCP   00570000 8A0408C8 .......H          SCP RELEASED TCA STORAGE
23:29:54.2               4004 TCP   44000000 00534400 ........          KCP WAIT DCI=TCP
23:2                             008117EA 008117EA ........          ... REPEAT 0000
                                 0056E9C0 80534020 ..Z... .          SCP
                                          85000078 ..Z.....
```

```
             00571800 80570BA8 0062                                              ..V..74*
00000CC0     80570CD5 00000000 00000003 000                                *...N............................*
00000CE0     00000000 005717E0 00571920 70624958                    00001  *................................*
00000D00     60624476 00000009 00000000 00000000   00000000          00570C10 005718F0   *-..............................0*
00000D20     00571880 00571800 00000000 00571A20   00000000 00000000 00570C0C 00571850   *................................*
00000D40     005718F0 A062298C 00622982 00000000   0057183C 0062E99C 60622814 00000009   *...0.......B.........Z.-........*
00000D60     00000000 00000000 00000000 00571940   00570C0C 00571880 00623538 00623510   *................................*
00000D80     00000000 005D1880 00000000 00000000   00000000 00000000 00000000 00000000   *.....).........................*
00000DA0     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00000DC0     00000000 00000000 00000000 00571AC0   00000000 00000000 00570C10 005718F0   *...............................0*
00000DE0     00000000 A062298C 00622982 00000000   005718DC 00000001 70622814 00000009   *...........B....................*
00000E00     00000000 00000000 00000000 00623813   00570C10 00571920 80570BA8 00000000   *.........................Y....*
00000E20     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00000E40     LINES TO 000013E0 SAME AS ABOVE
00001400     00000000 00000000 00000000 00000000   00000000 00000000 00000080 00000014   *................................*
00001420     00000080 00000000 08602001 00572138   00000020 00050000 00800000 00000000   *.........-......................*
00001440     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00001460     LINES TO 00001480 SAME AS ABOVE
000014A0     00000000 00000000 00000080 0000006E   00000080 00000000 D7C5D9C6 D6D9D440   *.........................PERFORM *
000014C0     40404040 40000000 00000001 0080E2D5   C1D74040 40404040 40404000 00000000   *     ..........SNAP        .....*
000014E0     00020080 D7C1D9E3 C9E3C9D6 D5404040   00038000 00010001 0000C6D6 D9D4C1E3   *....PARTITION   ..........FORMAT*
00001500     40404040 40400003 40000001 00010000   E3C9E3D3 C5404040 40404040 01012000   *      .. .......TITLE       ....*
00001520     01010001 00A00000 00000000 00000000   00000000 00000000 00000040 0000002C   *............................ ....*
00001540     00000040 00000000 00000000 005720F4   80572138 84108000 28A00000 00000000   *... ............4....D..........*
00001560     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00001580     00000000 00000000 00000080 00000020   00000080 00000000 E4E2C5D9 7DE240E2   *......................USER'S S*
000015A0     C1D4D7D3 C540C4E4 D4D74040 40404040   40404040 40404040 00000000 00000000   *AMPLE DUMP               ........*
000015C0     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000015E0     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00001600     00000000 00000000 00000000 00000000   00000000 00000000 00000040 0000000A   *........................ ......*
00001620     00000040 00000000 01000200 03010401   05010000 00000000 00000000 00000000   *... ............................*
00001640     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00001660     00000000 00000000 00000040 00000000   00000040 00000000 00000000 00000000   *.......... ....... ............*
00001680     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000016A0     00000000 00000000 00000000 00000000   00000000 00000000 00000040 00000000   *........................ .......*
000016C0     00000040 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *... ............................*
000016E0     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00001700     00000000 00000000 00000040 00000000   00000040 00000000 00000000 00000000   *.......... ....... ............*
00001720     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00001740     00000000 00000000 00000000 00000000   00000000 00000000 00000100 00000000   *................................*
00001760     00000100 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00001780     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000017A0     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000017C0     LINES TO 00001840 SAME AS ABOVE
00001860     00000000 00000000 00000080 0000002B   00000080 00000000 40D7C5D9 C6D6D9D4   *......................... PERFORM*
00001880     40E2D5C1 D740E3C9 E3D3C54D 7DE4E2C5   D97D7DE2 40E2C1D4 D7D3C540 C4E4D4D7   * SNAP TITLE('USER''S SAMPLE DUMP*
000018A0     7D5D4000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *') .............................*
000018C0     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000018E0     00000000 00000000 00000000 00000000   00000000 00000000 000008E8 00000000   *...........................Y....*
00001900     000008E8 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *...Y............................*
00001920     00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
00001940     LINES TO 000021C0 SAME AS ABOVE
000021E0     00000000 00000000 00000000 00000000   00000000 00000000 8C0021F8 005709B0   *...........................8....*
005709B0     USER STORAGE
00000000     8C0001E8 00570000 00000000 08000000   00000000 0057052C 00000000 5062F704   *...Y..........................7.*
00000020     00000000 00633A78 00570A78 0062F871   6062F266 00570ABC 0057051C 00000720   *..............8.-.2.............*
00000040     0062F260 00630265 00570A9C 00570C3C   005709C0 00570BA8 00000000 00571520   *..2-.....................Y.......*
00000060     00000000 005709C0 005345E8 00000000   00000000 00000000 00000000 00000000   *...........Y....................*
```

```
00000080   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
000000A0   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
000000C0   00000000 00000000 0062F871 00570ABC  00570BA8 80570A9C 80570A98 00000000   *..........8........Y........Q...*
000000E0   00570BA8 00000000 000021E8 21E80000  00000000 00000000 00000000 00000000   *...Y.......Y.Y..................*
00000100   00000000 00000000 00000000 C4C6C8C5  D4E3C440 00000000 00000000 00000000   *............DFHEMTD ............*
00000120   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
00000140   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000   *................................*
00000160   00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000108   *................................*
00000180   F0000018 004C0000 C4C6C8C5 D4E3C440  C4C6C8C5 C9E3D4E3 0233124C 0083070F   *0.......DFHEMTD DFHEITMT.....C..*
000001A0   C3C5D4E3 0000005C C3D5E2D3 00000000  00000000 00000000 00000000 00000000   *CEMT...*CNSL....................*
000001C0   00000000 00000000 00000040 40404040  40404000 00000000 00000000 00000000   *...........        .............*
000001E0   00000000 00000000 8C0001E8 00570000                                        *...........Y....                *
 ADDRESS   PAGE  C.B.    ERROR     ADDRESS    PAGE  C.B.   ERROR      ADDRESS    PAGE  C.B.   ERROR
            NO   TYPE    COUNT                 NO   TYPE   COUNT                  NO   TYPE   COUNT
00533898    33   TCT              0056BAD4     26   SKT               00659240     2   OVRPM
005339D0    15   ZEPD             0056BB24     26   SKT               006598A0     3   SUBSAV
00533E20    34   TCTDTF           0056BBD4     26   SKT               00659920     3   SUBSAV
00533E50    34   TCTDTF           0056BC44     24   DSG               006599A0     3   SIPCOM
00533E88    34   TCTDTF           0056C414     23   DSG
00533EB8    34   TCTLE            0056CBE4     22   DSG
00533F14    34   TCTLE            0056D3B4     20   DSG
00534020    51   TCTTE            0056DAC4     19   DSG
00534120    52   TCTTE            0056E1D4     26   DSG
00534228    52   TCTSE            0056E2E4     26   DSG                    3
005343D0    52   TCTSK            0056E4F4     21   SKT
00534400    34   TCTWL            0056E4F4     34   PPT
005345E8     1   CSA              0056E800     52   TIOA
005349E8     1   CSA              0056E8E0     52   TIOA
00534C28     2   CMXT             0056F000     15   TSUT
00534CA0    52   DCA              0056F600     54   DCA
00534D30    53   TCA              00570000     55   TCA
00534E70    53   TCA              00570140     54   TCA
005350C0    54   TMLL             00570440     57   PCL
00535110    54   LFS              00570470     56   EIS
00535488    16   TCA              00570628     56   TMLL
005355C8    15   TCA              00570680     56   LFS
005357B0    16   TMLL             005709B0     60   USER
00535800    16   LFS              00570BA0     58   USER
00536008    32   SECBL            00572DA0     58   USER
005660D0    32   DTF              00572E50     57   USER
005661A0    32   DCT              00572F50     57   RSA
005661B4    32   DCT              00572FB0     57   USER
005661C4    32   DCT              00573200     57   USER
005661D4    32   DCT              006302E0     32   SSA
005661E4    33   DCT              00630340     18   SSA
005661F4    33   DCT              00630448      2   STR
00566204    33   DCT              006308B0     27   SSA
00566214    33   DCT              006326F0     17   SSA
0056A820    14   TSMAP            006339A0     17   SSA
0056AA40    15   RPL              00633A20     18   SSA
0056AAB0    15   RPL              00633A70     27   SSA
0056AB50    14   PAM              00633B50     18   TMSS
0056AD70    14   MAP1             00633C90     17   SSAL
0056AED3    14   MAP2             00633DC0      3   TRT
0056B934    18   SKT              00658E08      2   SIT
0056B934    46   PCT              00659208      2   SIMODS
0056BA64    21   SKT              00659220      3   ATP
 ENTRY     LOAD      NAME    VERS'N TIME  DATE   OPTIONS                      ENTRY     LOAD      NAME    VERS'N TIME   DATE   OPTIONS
  POINT     POINT                                                              POINT     POINT
00533898 00533880 DFHTCT1$ 0160I                 00                          0063BDF6 0063BCC8 DFHFDP   0160I                   40
00536208 005361F0 DFHEKC   0160I                 40                          006401E0 006401C8 DFHSCR   0160I                   40
005362B0 00536290 DFHKCP   0160I                 40                          006417E4 006417C8 DFHKPPDY 0160I                   40
005393B8 005393A0 DFHALP   0160I                 40                          00641EC0 00641EA8 DFHSRT1$ 0160I                   00
0053A808 0053A7F0 DFHEEI   0160I                 40              4           006421C4 006421A8 DFHSRP1$ 0160I                   40
0053B33C 0053B320 DFHEIP   0160I                 40                          00643030 00643018 DFHMGT   0160I                   00
0053CC18 0053CC00 DFHESP   0160I                 40                          0064A9A0 0064A988 DFHMGP   0160I                   40
0053CDBC 0053CDA0 DFHSPP   0160I                 40                          0064CB00 0064CAE8 DFHEEX   0160I                   40
0053F828 0053F810 DFHERM   0160I                 40                          0064CE00 0064CDE8 DFHEBU   0160I                   40
0053FEB8 0053FEA0 DFHXSP   0160I                 40                          0064CFC0 0064CFA8 DFHLFO   0160I                   40
005401D8 005401C0 DFHETC   0160I                 40                          0064D440 0064D428 DFHDIPDY 0160I                   40
0054150C 005414F0 DFHZCPS$ 0160I                 4000FFEFFC                  0064D490 0064D478 DFHEDI   0160I                   40
005455B8 00545590 DFHTCPS$ 0160I                 4000EFFDFFBDFFFF7FFF        0064DCDC 0064DBB8 DFHDCPD$ 0160I                   40
005532B8 005532A0 DFHETL   0160I                 40                          0064EF80 0064EF68 DFHEDC   0160I                   40
00554378 00554360 DFHZCXS$ 0160I                 40                          0064F78A 0064F778 DFHSAP   20     00:00 58/ 2  F0
00558024 00558008 DFHTMP   0160U                 00                          00655150 00655138 DFHBFP1$ 0160I                   40
0055A420 0055A408 DFHESC   0160I                 40                          006567D0 006567B8 DFHEBF   0160I                   40
0055A540 0055A518 DFHSCP   0160I                 40                          00656E58 00656E38 DFHJCPDY 0160I                   40
0055E800 0055DD08 DFHPCP3$ 0160I                 40                          00656EC0 00656EA8 DFHEJC   0160I                   40
00560230 00560218 DFHEPC   0160I                 40                          006574F0 006574D8 DFHICP   0160I                   40
00563350 00563338 DFHEFC   0160I                 40                          00658440 00658428 DFHEIC   0160I                   40
00563E44 00563E28 DFHFCPDY 0160I                 4000004000                  00658E08 00658DF0 DFHSIT1$ 0160I                   00
00564020 00564008 DFHETS   0160I                 40                          00659288 00659270 DFHCICS  0160I                   40
005643BC 00564358 DFHTSP2$ 0160I                 40
005661A0 00566008 DFHDCT1$ 0160I                 0000
005662C0 005662A8 DFHETD   0160I                 40
0056655C 00566538 DFHTDP6$ 0160I                 40007A
005673C0 005673A8 DFHUEM   0160I                 40
00567C60 00567C48 DFHEMS   0160I                 40
005684F4 005684D8 DFHMCPDY 0160I                 4000000000005
00618020 00618008 DFHOCP   0160I
00619020 00619008 DFHEMB   0160I
0061A024 0061A008 DFHTEP   0160I
0061B020 0061B008 DFHFELG  0160I
0061B820 0061B808 DFHTACP  0160I
0061FA10 0061FA10 DFHTEPT  00 ≡0   0:0  58/
00620820 00620800 DFHEMTD  0160U
0062D8F0 0062D8F0 DFHEITMT 160≡0 00:14 58/ 0
0062F260 0062F240 DFHEMTP  0160U 23:56 04/23
00633CE0 00633CC8 DFHVCPDY 0160I
00637E10 00637DF8 DFHTRPD$ 0160I                 40
0063B5E0 0063B5C8 DFHETR   0160I                 40
0063B774 0063B758 DFHCMPDY 0160I                 40
END OF CICS/VS FORMATTED DUMP
```

Processing the Auxiliary Trace Data Set

```
// JOB PRTAUX
// DLBL DFHAUXT,'CICS160.AUX.TRACE.A',,SD
// EXTENT SYS001,,1,0,260,40
// ASSGN SYS001,DISK,TEMP,VOL=M3350A,SHR
// EXEC DFHTUP,SIZE=80K,PARM='DEVICE=DISK'
/*
/&

                    (This job is included in Z.DFHSPJCL)
```

Figure 21. Processing the Auxiliary Trace Data Set

<u>Notes:</u>

1. In the above sample job, the file name must be DFHAUXT.
2. You may also need to add:

   ```
   // LIBDEF CL,SEARCH=C160PCL
   ```

   before the EXEC statement.
3. The DEVICE= operand included here indicates that the auxiliary trace has been recorded on disk. Further operands can be added to indicate that trace entries are to be selected according to specified criteria (for example, those entries resulting from specified transactions or occurring between specified times). See Chapter 4.8 for details of the parameters that can be used.

Sample CICS Auxiliary Trace Output

The <u>CICS/VS Performance Guide</u> describes a trace in some detail.

```
           CUSTOMER INFORMATION CONTROL SYSTEM  -  T R A C E   U T I L I T Y   P R O G R A M
   SELECTION PARAMETERS:      DEVICE=DISK
AUXILIARY TRACE ACTIVATED AT  -  23:32:33.68
   TIME OF DAY     ID  REG 14   REQD TASK   FIELD A  FIELD B  CHARS     RESOURCE   TRACE TYPE
23:32:26.463552    F2 5053BCDA 0804 00009 00000000 00000000 ........ DFHEMB     PCP DELETE
23:32:26.463648    EA 4055ED36 0003 00009 01000300 005701C4 .......D DFHEMB     TMP PPT LOCATE
23:32:26.463808    EA 405590E8 0005 00009 01000300 00561C10 ........            TMP RETN NORMAL
23:32:26.463936    E1 40621760 00F4 00009 00000000 0000840A ........            EIP RESPONSE
23:32:26.464576    E1 50621C92 0004 00009 00572FC0 00001002 ........            EIP ASKTIME ENTRY
23:32:26.464672    F3 6065848E 1303 00009 135705D0 00000000 ........            ICP GETIME
23:32:26.464704    F3 40657626 0055 00009 00572FC0 00001002 ........            ICP RETN NORMAL
23:32:26.464736    E1 50621C92 00F4 00009 00000000 00001002 ........            EIP ASKTIME RESPONSE
23:32:26.464832    E1 50620EEC 0004 00009 00572FC0 00000404 ........            EIP SEND-TC ENTRY
23:32:26.464928    F1 405406DE C304 00009 00000066 00534020 ....... .            SCP GETMAIN INITIMG
23:32:26.465056    C8 5055A850 0004 00009 0056E9C0 85000078 ..Z.....            SCP ACQUIRED TERMINAL STORAGE
23:32:26.465120    FC 50540B70 0103 00009 00810000 00534020 ....... .            ZCP ZARQ APPL REQ ERASE WRITE
23:32:26.465184    FC 40541C3A 0105 00009 00810000 0056E9C0 ......Z.            ZCP RETN ZARQ APPL REQ ERASE WRITE
23:32:26.465216    E1 50620EEC 00F4 00009 00000000 00000404 ........            EIP SEND-TC RESPONSE
23:32:26.465280    F2 6055F3AC 1004 00009 00000000 00000000 ........ DFHEMTD    PCP RETURN
23:32:26.465312    F1 6055EFA2 4004 00009 00572FB0 01534020 ....... .            SCP FREEMAIN
23:32:26.465344    C9 5055A886 0004 00009 00572FB0 8C000248 ........            SCP RELEASED USER STORAGE
23:32:26.465376    F1 6055ED9E 4004 00009 00572F50 01534020 ...&.. .            SCP FREEMAIN
23:32:26.465440    C9 5055A886 0004 00009 00572F50 89570058 ...&....            SCP RELEASED RSA STORAGE
23:32:26.465504    F1 5053BEC0 4004 00009 00572E50 01534020 ...&.. .            SCP FREEMAIN
23:32:26.465568    C9 5055A886 0004 00009 00572E50 8C0000F8 ...&...8            SCP RELEASED USER STORAGE
23:32:26.465600    E1 5062F704 00F4 00009 00000000 00000E02 ........            EIP LINK RESPONSE
23:32:26.465664    E1 5062F736 0004 00009 005709C0 00000C04 ........            EIP FREEMAIN ENTRY
23:32:26.465696    F1 4055A4FA 0004 00009 40000000 9D5704D4  .......M            SCP FREEMAIN
23:32:26.465728    C9 5055A886 0004 00009 00570BA0 8C0021F8 .......8            SCP RELEASED USER STORAGE
23:32:26.468224    E1 5062F736 00F4 00009 00000000 00000C04 ........            EIP FREEMAIN RESPONSE
23:32:26.468416    F2 4055F3AC 1004 00009 00000000 00000000 ........ DFHEMTP    PCP RETURN
23:32:26.468448    F1 6055EFA2 4004 00009 005709B0 01534020 ....... .            SCP FREEMAIN
23:32:26.468544    C9 5055A886 0004 00009 005709B0 8C0001E8 .......Y            SCP RELEASED USER STORAGE
23:32:26.468608    F0 4055EB0C 8004 00009 00000000 00000000 ........            KCP DETACH
23:32:26.468736    F0 40536EEA 0304 00009 00000000 00000000 ........            KCP DEQALL
23:32:26.468800    D0 505382F2 0504 00009 00000000 00000000 ........            KCP DISPATCH
23:32:26.468864    D0 505377B4 0704 00009 03380400 C3C5D4E3 ....CEMT            KCP TERMINATE
23:32:26.468928    FC 50536F3E 0B03 00009 04000000 01534020 ....... .            ZCP ZISP ISC FREE DETACH
23:32:26.469024    FC 40545510 0B05 00009 04000000 00534020 ....... .            ZCP RETN ZISP ISC FREE DETACH
23:32:26.469056    F1 4053708A 4A04 KCP   00570000 00000000 ........            SCP FREEMAIN
23:32:26.499232    C9 5055A886 0004 KCP   00573200 8C0000D8 .......Q            SCP RELEASED USER STORAGE
23:32:26.500320    C9 5055A886 0004 KCP   00572DA0 8C0000A8 ........            SCP RELEASED USER STORAGE
23:32:26.501504    C9 5055A886 0004 KCP   00570000 8A0409A8 ........            SCP RELEASED TCA STORAGE
23:32:26.503456    D0 505382F2 0504 TCP   00000000 00000000 ........            KCP DISPATCH
23:32:26.510080    F0 40541660 4004 TCP   44000000 00534400 ........            KCP WAIT DCI=TCP
23:32:26.510144    D0 505382F2 0504 TCP   00000000 00000000 ........            KCP DISPATCH
23:32:26.514592    F0 40541660 4004 TCP   44000000 00534400 ........            KCP WAIT DCI=TCP
23:32:26.514688    D0 505382F2 0504 TCP   00000000 00000000 ........            KCP DISPATCH
23:32:26.519744    F0 40541660 4004 TCP   44000000 00534400 ........            KCP WAIT DCI=TCP
23:32:26.519808    D0 505382F2 0504 TCP   00000000 00000000 ........            KCP DISPATCH
23:32:26.520640    F1 50545C1C 4004 TCP   0056E9C0 80534020 ..Z... .            SCP FREEMAIN
23:32:26.520704    C9 5055A886 0004 TCP   0056E9C0 85000078 ..Z.....            SCP RELEASED TERMINAL STORAGE
23:32:26.520768    F1 40545CC0 6004 TCP   00000000 80534020 ....... .            SCP FREEMAIN ALL
23:32:26.521824    C9 5055A886 0004 TCP   0056E8E0 85000068 ..Y.....            SCP RELEASED TERMINAL STORAGE
23:32:26.521920    F1 40545B66 E504 TCP   00000052 80534020 ....... .            SCP GETMAIN CONDITIONAL INITIMG
23:32:26.521984    C8 5055A850 0004 TCP   0056E8E0 85000068 ..Y.....            SCP ACQUIRED TERMINAL STORAGE
23:32:26.526112    F0 40541660 4004 TCP   44000000 00534400 ........            KCP WAIT DCI=TCP
23:32:31.520960    D0 505382F2 0904 KCP   80000000 3A9861CB ....../.            KCP SYSTEM RESUME
23:32:31.521120    D0 505382F2 0504 TCP   00000000 00000000 ........            KCP DISPATCH
23:32:31.522432    F0 40541660 4004 TCP   44000000 00534400 ........            KCP WAIT DCI=TCP
23:32:36.527040    D0 505382F2 0904 KCP   80000000 3A98668E ........            KCP SYSTEM RESUME
23:32:36.527168    D0 505382F2 0504 TCP   00000000 00000000 ........            KCP DISPATCH
23:32:36.528512    F0 40541660 4004 TCP   44000000 00534400 ........            KCP WAIT DCI=TCP
23:32:38.862976    D0 505382F2 0904 KCP   00000000 3A986B55 .......,.            KCP SYSTEM RESUME
23:32:38.863104    D0 505382F2 0504 TCP   00000000 00000000 ........            KCP DISPATCH
23:32:38.865696    F0 50551E42 1304 TCP   C3D5E2D3 C3C5D4E3 CNSLCEMT            KCP ATTACH-CONDITIONAL
23:32:38.865792    EA 705368A2 0003 TCP   01040100 00534F08 ........ CEMT       TMP PCT LOCATE
23:32:38.866208    EA 405590E8 0005 TCP   01040100 0055CFC4 .......D            TMP RETN NORMAL
23:32:38.866304    F1 40536A00 EA04 TCP   000009A8 80534020 ....... .            SCP GETMAIN CONDITIONAL INITIMG
23:32:38.868288    C8 5055A850 0004 TCP   00570000 8A0409A8 ........            SCP ACQUIRED TCA STORAGE
23:32:38.868384    D0 505377B4 0604 00010 C3D5E2D3 C3C5D4E3 CNSLCEMT            KCP CREATE
23:32:38.868448    EA 40536D94 0003 TCP   0C000100 0055CFC4 .......D CEMT       TMP PCT TRANSFER
23:32:38.868544    EA 405590E8 0005 TCP   0C000100 00000000 ........            TMP RETN NORMAL
23:32:38.868576    D0 505382F2 0504 TCP   00000000 00000000 ........            KCP DISPATCH
23:32:38.868672    F0 40541660 4004 TCP   44000000 00534400 ........            KCP WAIT DCI=TCP
23:32:38.868736    D0 505382F2 0504 00010 00000000 00000000 ........            KCP DISPATCH
23:32:38.869120    FC 055434F8 0503 00010 00000000 00534020 ....... .            ZCP ZSUP START UP TASK
23:32:38.869504    E5 405435AA 0C03 00010 00000000 00000000 ........ CEMT.*..   XSP SECURITY
23:32:38.869632    F2 50543CE0 8804 00010 00000000 00000000 ........ DFHEMTP    PCP XCTL-CONDITIONAL
23:32:38.869664    EA 4055ED36 0003 00010 01000300 005701C4 .......D DFHEMTP    TMP PPT LOCATE
23:32:38.869792    EA 405590E8 0005 00010 01000300 00561988 ........            TMP RETN NORMAL
23:32:38.870432    F1 6053B7C4 CC04 00010 000001D8 01534020 ....... .            SCP GETMAIN INITIMG
23:32:38.870496    C8 5055A850 0004 00010 005709B0 8C0001E8 .......Y            SCP ACQUIRED USER STORAGE
23:32:38.870560    E1 5062F32E 0004 00010 005709C0 00000208 ........            EIP ASSIGN ENTRY
23:32:38.870656    E1 5062F32E 00F4 00010 00000000 00000208 ........            EIP ASSIGN RESPONSE
23:32:38.870688    E1 5062F428 0004 00010 005709C0 00000C02 ........            EIP GETMAIN ENTRY
23:32:38.870720    F1 4055A498 0004 00010 C00C0000 9D5704D4 .......M            SCP GETMAIN INITIMG
23:32:38.873760    C8 5055A850 0004 00010 00570BA0 8C0021F8 .......8            SCP ACQUIRED USER STORAGE
23:32:38.873824    E1 5062F428 00F4 00010 00000000 00000C02 ........            EIP GETMAIN RESPONSE
23:32:38.873888    E1 4062F4E8 0004 00010 005709C0 0000020A ........            EIP IGNORE-CONDITION ENTRY
23:32:38.873952    F1 6053BEC0 CC04 00010 0000009E 01534020 ....... .            SCP GETMAIN INITIMG
```

```
23:32:38.873984  C8 5055A850 0004 00010 00572DA0 8C0000A8 ........             SCP ACQUIRED USER STORAGE
23:32:38.874048  E1 4062F4E8 00F4 00010 00000000 0000020A ........             EIP IGNORE-CONDITION RESPONSE
23:32:38.874080  E1 5062F516 0004 00010 005709C0 00000402 ........             EIP RECEIVE-TC ENTRY
23:32:38.874240  E1 5062F516 00F4 00010 00000000 00000402 ........             EIP RECEIVE-TC RESPONSE
23:32:38.874336  E1 5062F6D0 0004 00010 005709C0 00000E06 ........             EIP LOAD ENTRY
23:32:38.874368  F2 60560258 8204 00010 00000000 00000000 ........ DFHEITMT  PCP LOCATE
23:32:38.874432  EA 4055ED36 0003 00010 01000300 005701C4 .......D DFHEITMT  TMP PPT LOCATE
23:32:38.874528  EA 405590E8 0005 00010 01000300 005619D0 ........             TMP RETN NORMAL
23:32:38.874560  F2 4053C144 0404 00010 00000000 00000000 ........ DFHEITMT  PCP LOAD
23:32:38.874592  EA 4055ED36 0003 00010 01000300 005701C4 .......D DFHEITMT  TMP PPT LOCATE
23:32:38.874656  EA 405590E8 0005 00010 01000300 005619D0 ........             TMP RETN NORMAL
23:32:38.874752  E1 5062F6D0 00F4 00010 00000000 00000E06 ........             EIP LOAD RESPONSE
23:32:38.874784  E1 5062F704 0004 00010 005709C0 00000E02 ........             EIP LINK ENTRY
23:32:38.886080  F2 60560258 8204 00010 00000000 00000000 ........ DFHEMTD   PCP LOCATE
23:32:38.886176  EA 4055ED36 0003 00010 01000300 005701C4 .......D DFHEMTD   TMP PPT LOCATE
23:32:38.886336  EA 405590E8 0005 00010 01000300 00561B80 ........             TMP RETN NORMAL
23:32:38.886400  F1 6053BEC0 CC04 00010 000000F0 01534020 ...0.. .             SCP GETMAIN INITIMG
23:32:38.886528  C8 5055A850 0004 00010 00572E50 8C0000F8 ...&...8             SCP ACQUIRED USER STORAGE
23:32:38.886560  F2 5053C144 8104 00010 00000000 00000000 ........ DFHEMTD   PCP LINK-CONDITIONAL
23:32:38.886624  EA 4055ED36 0003 00010 01000300 005701C4 .......D DFHEMTD   TMP PPT LOCATE
23:32:38.886688  EA 405590E8 0005 00010 01000300 00561B80 ........             TMP RETN NORMAL
23:32:38.886720  F1 4055E96E 8904 00010 00570050 01534020 ...&.. .             SCP GETMAIN
23:32:38.886752  C8 5055A850 0004 00010 00572F50 89570058 ...&....             SCP ACQUIRED RSA STORAGE
23:32:38.886944  F1 6053B7C4 CC04 00010 00000232 01534020 ...... .             SCP GETMAIN INITIMG
23:32:38.888864  C8 5055A850 0004 00010 00572FB0 8C000248 ........             SCP ACQUIRED USER STORAGE
23:32:38.892672  E1 40621760 0004 00010 00572FC0 0000840A ........             EIP ENTRY
23:32:38.892768  F2 5053BA9E 0404 00010 00000000 00000000 ........ DFHEMB    PCP LOAD
23:32:38.892832  EA 4055ED36 0003 00010 01000300 005701C4 .......D DFHEMB    TMP PPT LOCATE
23:32:38.892992  EA 405590E8 0005 00010 01000300 00561C10 ........             TMP RETN NORMAL
23:32:38.893152  F1 6053BEC0 CC04 00010 000000C4 01534020 ...D.. .             SCP GETMAIN INITIMG
23:32:38.893216  C8 5055A850 0004 00010 00573200 8C0000D8 .......Q             SCP ACQUIRED USER STORAGE
END OF AUXILIARY TRACE DATA -     TOTAL TIME ACTIVE       00:00:05.21
                                  TRACE RECORDS READ            118
                                  TRACE RECORDS SELECTED        118
```

Chapter 2.3. Sample Application Package and CICS Tables

Chapter 2.2 described the setting up of some of the CICS data sets commonly used while running a CICS system. This was done in preparation for the central task necessary to complete the installation of a working system, that is, installing and integrating your own application programs or application packages with the base distributed system whose installation was described in Part 1.

In general, installing application programs involves assembling (or translating and compiling) them and link-editing them to a private core image library other than the CICS core image library. Further, the application programs (and other entities) have to be made known to the CICS system by including their definitions in CICS control tables. This chapter continues the tutorial approach of the last chapter, and explains the steps necessary to set up a CICS system based on the sample application programs provided on the distribution tape.

Your application programs are supplied by you to perform the specific applications required by your installation. The CICS system provides the management modules that control and manage these programs and service CICS requests made by them.

A frequent requirement of an application program is to display information on a display terminal or to read information from the terminal. This involves sending to the terminal a data stream, which includes not only the information to be displayed but also the encoded layout of the information on the screen. The data stream is in the form of screen addresses, attributes of the various fields, position of the cursor, and so on. The construction of this data stream is both arduous and prone to error.

To relieve the application programmer of the need to be concerned with device data streams and, furthermore, to allow application programs to be written in a form that is independent of the characteristics of a specific device, CICS allows the layout of data on the screen to be expressed in terms of **maps**.

A map is defined independently of the application program. The application program refers to the map and to fields within it by name, when it is building up a screen of information. The defined map is then used by CICS to construct the actual data stream that is sent, at run time, to the terminal. The interpretation of the defined maps and the construction of the data stream are performed by Basic Mapping Support (BMS), a major component of CICS, which is fully described in the <u>CICS/VS Application Programmer's Reference Manual (Command Level)</u>.

CICS SYSTEM RESOURCES

Application programs and maps are examples of what are known to the CICS system as **resources**. Other examples are user files, queues (transient data and temporary storage), transaction identifiers, terminal identifiers, and so on. These, together with their characteristics, must be defined to the CICS system in the form of CICS control tables. In the case of transaction identifiers and application programs, these resources may also, or alternatively, be defined online while CICS is running (see "Chapter 2.9. Migrating Resource Definitions").

Any number of alternative versions of the control tables may be set up, defining different combinations of resources to be used for particular environments. The different tables of a particular type are distinguished by appending a 2-character suffix to the base table name. The environment for a particular

CICS run is then defined by the suffixes chosen for the various tables during CICS initialization time at start up. In addition, of course, versions of the CICS management modules must be selected that will support the resources used.

SAMPLE APPLICATION PROGRAMS

In order to demonstrate how to install application programs and how to define them and the resources they use to the CICS system, a set of sample application programs has been included with the pregenerated system. These are provided in source form as A., C., P., and R. source books in the CICS source statement library in Assembler, COBOL, PL/I, and RPG II versions respectively. The programs use the command-level interface when making CICS service requests. The function of the sets of programs in each of the four languages is identical.

The names of the programs in each set are given in Appendix E. Associated with the programs are a number of maps used by the programs to format data on an IBM 3270-series display terminal. Like the programs, the maps must be assembled and link-edited to a private core image library.

The pregenerated system contains not only the source versions of these sample application programs and maps, but also their already assembled (or compiled) and link-edited forms in the CICS core image library. This part of the installation will not, therefore, be discussed further in the present chapter. If you wish to carry out the assembly or compilation of these programs for yourself, you will need to define a private core image library and follow the instructions given in "Chapter 3.3. Installing Map Sets and Partition Sets" and "Chapter 3.4. Installing Application Programs."

CREATION OF THE USER FILE FILEA

In order to run the sample application programs (and, indeed, any typical application program), you will need to create a file (or files or data bases) of the data to be used by the programs. In general, such files can be either VSAM, ISAM, or DAM files; or they can be managed by DL/I, but this would require the installation of the DL/I DOS/VS program product on your system.

Specifically, the sample application programs require a VSAM file known as FILEA to be created. This is a key-sequenced file that you must define in VSAM space and then load with the initial data needed to run the sample application programs. The following job can be used to perform the definition:

```
// JOB DEFILEA CREATE VSAM FILEA CLUSTER
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                -
          (NAME(SAMPLE.TEST.FILEA)             -
           INDEXED                             -
           BUFFERSPACE(4096)                   -
           RECORDSIZE(80 80)                   -
           RECORDS(500 200)                    -
           KEYS(6 1)                           -
           SHAREOPTIONS(2)                     -
           VOLUMES(M3350A))                    -
        DATA                                   -
          (NAME(SAMPLE.TEST.FILEA.DATA))       -
        INDEX                                  -
          (NAME(SAMPLE.TEST.FILEA.INDEX))      -
        CATALOG(CICS160.UCAT)
 LISTCAT ENTRIES(SAMPLE.TEST.FILEA) ALL        -
        CATALOG(CICS160.UCAT)
/*
/&

                 (This job is included in Z.DFHSPJCL)
```

Figure 22. Defining the User File FILEA

The file has here been defined in VSAM space belonging to the previously-defined user catalog CICS160.UCAT. The definition is similar to that used for auxiliary temporary storage and intrapartition transient data except for the fact that those data sets are defined as nonindexed. In the case of FILEA, an index component is also defined. (If the index component name is not explicitly given in the defining job, a name will be generated by VSAM.)

Any indexed (key-sequenced) VSAM file must be loaded with at least one record before it can be used for VSAM processing. The following job loads FILEA with the data required for the sample application programs:

```
// JOB INTFILEA
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
// DLBL FILEA,'SAMPLE.TEST.FILEA',,VSAM,CAT=CICSUCT
// EXEC IDCAMS,SIZE=AUTO
 REPRO  INFILE                       -
          (SYSIPT                    -
           ENVIRONMENT               -
            (BLOCKSIZE(80)           -
             RECORDFORMAT(FIXUNB)    -
             RECORDSIZE(80)))        -
        OUTFILE(FILEA)
 000100W. DAVIS                SURREY, ENGLAND     3215677826 11 81$0100.11YES
 000102F. ALDSON               WARWICK, ENGLAND    9835618326 11 81$1111.11YES
 000104S. BOWLER               LONDON,ENGLAND      1284629326 11 81$0999.99YES
 000106B. ADAMS                CROYDON, ENGLAND    1948567326 11 81$0087.71YES
 000111GENE BARLOWE            SARATOGA,CALIFORNIA 4612075301 02 74$0111.11YES
 000762GEORGE BURROW           SAN JOSE,CALIFORNIA 2231212101 06 74$0000.00YES
 000983H. L. L. CALL           WASHINGTON, DC      3451212021 04 75$9999.99YES
 001222J.R.REYNOLDS            BOBLINGEN, GERMANY  7031555110 04 73$3349.99YES
 001781HAROLD JAMES            SINDELFINGEN,GERMANY7031999021 06 77$0009.99YES
 003210M. P. CREPIN            PARIS, FRANCE       1234567010 03 75$3349.99YES
 003214HUBERT C HERBERT        SUNNYVALE, CAL.     3411212000 06 73$0009.99NO
 003890PHILIPPE SMITH, JR      NICE, FRANCE        0000000028 05 74$0009.99NO
 004004STAN SMITH              DUBLIN, IRELAND     7111212102 11 73$1259.99NO
 004445DANIEL O'GALWAY         SOUTH BEND, S.DAK.  6121212010 10 73$0009.99NO
 004878D.C. CURRENT            SUNNYVALE, CALIF.   3221212010 06 73$5399.99NO
 005005J. S. LAVERENCE         SAN FRANCISCO, CA.  0000000101 08 73$0009.99NO
 005444JEAN LAWRENCE           SARATOGA, CALIF.    6771212020 10 74$0809.99NO
 005581JOHN ALDEN III          BOSTON, MASS.       4131212011 04 74$0259.99NO
 006016DR W. T. KAR            NEW DELHI, INDIA    7033121121 05 74$0009.88YES
 006670WILLIAM KAPP            NEW YORK, N.Y.      2121212031 01 75$3509.88NO
 006968D. CONRAD               WARWICK, ENGLAND    5671382126 11 81$0009.88YES
 007007BRIGITTE EICRN          STUTTGART, GERMANY  7031100010 10 75$5009.88NO
 007248B. C. WILLIAMSON        REDWOOD CITY, CALF. 3331212111 10 75$0009.88NO
 007779MRS. W. WELCH           SAN JOSE, CALIF.    4151212003 01 75$0009.88YES
 100000G. NEADS                TORONTO, ONTARIO    0341512126 11 81$0010.00YES
 111111C. MEARS                OTTAWA, ONTARIO     5121200326 11 81$0011.00YES
 200000A. BONFIELD             GLASGOW,  SCOTLAND  6373829026 11 81$0020.00YES
 222222J. WIEBERS              FRANKFURT, GERMANY  2003415126 11 81$0022.00YES
 300000K. TRENCHARD            NEW YORK, U.S.      6473980126 11 81$0030.00YES
 333333D. MYRING               CARDIFF, WALES      7849302026 11 81$0033.00YES
 400000W. TANNER               MILAN, ITALY        2536373826 11 81$0040.00YES
 444444A. FISHER               CALGARY, ALBERTA    7788982026 11 81$0044.00YES
 500000J. DENFORD              MADRID, SPAIN       4445464026 11 81$0000.00YES
 555555C. JARDINE              KINGSTON, N.Y.      3994442026 11 81$0005.00YES
 600000F. HUGHES               DUBLIN, IRELAND     1239878026 11 81$0010.00YES
 666666A. BROOKMAN             LA HULPE, BRUSSELS  4298384026 11 81$0016.00YES
 700000A. MACALLA              DALLAS, TEXAS       5798432026 11 81$0002.00YES
 777777D. PRYKE                WILLIAMSBURG, VIRG. 9187613126 11 81$0027.00YES
 800000H. BRISTOW              WESTEND, LONDON     2423338926 11 81$0030.00YES
 888888B. HOWARD               NORTHAMPTON, ENG.   2369163926 11 81$0038.00YES
 900000D. WOODSON              TAMPA, FLA.         3566812026 11 81$0040.00YES
 999999R. JACKSON              RALEIGH, N.Y.       8459163926 11 81$0049.00YES
/*
/&
```

(This job is included in Z.DFHSPJCL)

Figure 23. Loading the User File FILEA With Initial Data

SAMPLE TABLES

As explained above, in order to define the application programs and maps, and the other resources they require, to the CICS system, a number of control tables must be coded and assembled and link-edited to a private core image library, and then selected during system initialization at CICS start-up time.

The rules for coding control tables and the meaning of the various operands that describe the resources being defined are explained fully in the CICS/VS Resource Definition Guide. In order to simplify this part of the installation of the sample application programs, the distributed system contains copy books for the entries required. Their names and contents are listed in Appendix D. The actual copy books that are relevant to the samples are included, where required, in the text that follows.

Each table defining the resources of a particular type is coded, beginning with a TYPE=INITIAL macro statement and ending with TYPE=FINAL. The TYPE=INITIAL statement may include the SUFFIX=xx operand, where xx is a user-chosen 2-character identifier of the version of the table. In the following examples, the suffix will be S1 throughout.

The body of the table is composed of a series of macro statements, describing each resource being defined. It is these definitions for the sample applications that are included as copy books in the distributed A. library.

Before the control tables can be assembled and link-edited into your system, you must define a private core image library (if you do not already have one defined) to hold them (and application programs, maps, and so on, that you may wish to install). The following job can be used for this purpose:

```
// JOB CREATE NEW PRIVATE CORE IMAGE LIBRARY
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL',99/365
// EXTENT ,M3330A,1,0,19,57
// LIBDEF CL,NEW=U160PCL
// EXEC CORGZ
   NEWVOL CL=3(4)
/*
/&

                    (This job is included in Z.DFHSPJCL)
```

Figure 24. Defining a Private Core Image Library

The size chosen for the private core image library is such as to occupy an integral number of cylinders. The above job allocates 3 cylinders of space on an IBM 3330 Direct Access Storage Device for the private core image library, of which 4 tracks are to be used for the directory. The equivalent sizes for a private core image library on devices other than a 3330 are:

Device Type	Size	Directory Size
3340	144 tracks (12 cyls)	10 tracks
3350	57 tracks (3 cyls)	3 tracks
3310/3370	1350 blocks	94 blocks

WORK FILES

Temporary DASD space needs to be provided for three assembly work files and one input file for the linkage editor. These files can be defined in every assembly job stream, or they can be added to the system or partition standard labels. These files may already be included in the standard labels for your installation — you should check before you define them yourself.

```
// JOB CREATE WORK FILES AND LINK FILE
// OPTION STDLABEL=ADD        (or OPTION PARSTD=ADD)
// DLBL IJSYS01,'WORK.FILE.1',0,SD
// EXTENT SYS001,,1,0,418,361
// DLBL IJSYS02,'WORK.FILE.2',0,SD
// EXTENT SYS002,,1,0,779,361
// DLBL IJSYS03,'WORK.FILE.3',0,SD
// EXTENT SYS003,,1,0,1140,266
// DLBL IJSYSLN,'SYSLNK.FILE',0,SD
// EXTENT SYSLNK,,1,0,1406,76
// ASSGN SYS001,DISK,PERM,VOL=M3330A,SHR
// ASSGN SYS002,SYS001
// ASSGN SYS003,SYS001
// ASSGN SYSLNK,SYS001
/*
/&

                  (This job is included in Z.DFHSPJCL)
```

Figure 25. Adding Assembly Work Files and Link File

The equivalent sizes for these files on devices other than a 3330 are:

Device Type	SYS001	SYS002	SYS003	SYSLNK
3340	912	912	672	192
3350	285	285	228	76
3310/3370	8550	8550	6300	1800

PROCESSING PROGRAM TABLE (PPT)

This table contains definitions of the user's application programs and maps and also CICS-supplied application programs. The definitions of the CICS-supplied programs are contained in the DFHXPPT copy book that is part of DFHPPT1$, used to start up the initial system.

To this must be added one of the copy books DFHXPPTA, DFHXPPTC, DFHXPPTP, or DFHXPPTR, depending on the language you are using. The copy book for the assembler language (DFHXPPTA) is shown below:

```
         TITLE  'DFHXPPTA - COPYBOOK FOR ASSEMBLER SAMPLE PROGRAMS'
          DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHAMNU
          DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHAALL
          DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHABRW
          DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHAREN
          DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHACOM
          DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHAREP
          DFHPPT TYPE=ENTRY,MAPSET=XDFHAMA
          DFHPPT TYPE=ENTRY,MAPSET=XDFHAMB
          DFHPPT TYPE=ENTRY,MAPSET=XDFHAMC
          DFHPPT TYPE=ENTRY,MAPSET=XDFHAMD
          DFHPPT TYPE=ENTRY,MAPSET=XDFHAMK
          DFHPPT TYPE=ENTRY,MAPSET=XDFHAML
```

Thus, for example, if you will be using assembler-language application programs, you would run the following job:

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHPPTS1
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=U160PCL
// LIBDEF RL,SEARCH=C160PRL
// LIBDEF SL,SEARCH=C160PSL
// EXEC ASSEMBLY
       DFHPPT  TYPE=INITIAL,                                               *
               SUFFIX=S1
       COPY    DFHXPPT
       COPY    DFHXPPTA
       DFHPPT  TYPE=FINAL
   END
/*
// EXEC LNKEDT
/&

                        (This job is included in Z.DFHSPJCL)
```

Figure 26. Assembling and Link-Editing a Sample PPT

In order to reduce the amount of listed output from the assembly of the PPT, PCT, FCT, DCT, and TCT, you might wish to include the PRINT NOGEN option in the job stream.

PROGRAM CONTROL TABLE (PCT)

This table is used by CICS to identify incoming transaction identifiers and to initialize the transaction by invoking the appropriate program. Here again, there are a number of basic CICS-required entries that are included in the DFHXPCT copy book. To these must be added the entries in copy book DFHXPCTA, DFHXPCTC, DFHXPCTP, or DFHXPCTR, depending on the language you are using. The copy book for assembler language (DFHXPCTA) is shown below:

```
         TITLE  'DFHXPCTA - COPYBOOK FOR ASSEMBLER SAMPLE PROGRAMS'
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAMNU,TRANSID=AMNU
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAALL,TRANSID=AINQ
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAALL,TRANSID=AADD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAALL,TRANSID=AUPD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHABRW,TRANSID=ABRW
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAREN,TRANSID=AORD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHACOM,TRANSID=AORQ
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAREP,TRANSID=AREP
```

The following job should be run:

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHPCTS1
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=U160PCL
// LIBDEF RL,SEARCH=C160PRL
// LIBDEF SL,SEARCH=C160PSL
// EXEC ASSEMBLY
       DFHPCT  TYPE=INITIAL,                                          *
               SUFFIX=S1
       COPY    DFHXPCT
       COPY    DFHXPCTA
       DFHPCT  TYPE=FINAL
   END
/*
// EXEC LNKEDT
/&
```

Figure 27. Assembling and Link-Editing a Sample PCT

DESTINATION CONTROL TABLE (DCT)

The destination control table defines to CICS the transient data queues (both extrapartition and intrapartition). Certain queue names (destinations) are again required by CICS; these are included in the pregenerated DCT for the initial system startup as extrapartition destinations directed to the output printer (SYSLST).

Two further destinations are required to run the sample applications. These are defined in the DFHXDCTS copy book:

```
        TITLE  'DFHXDCTS - COPYBOOK OF DCT ENTRIES FOR SAMPLE PROGRAMS'
LOGA    DFHDCT TYPE=INDIRECT,        DESTINATION LOGA USED             X
               DESTID=LOGA,          BY CICS SAMPLE PGMS               X
               INDDEST=CPLI
L860    DFHDCT TYPE=INTRA,           DESTINATION L860 USED BY ORDER    X
               DESTID=L860,          ENTRY QUEUE PRINT SAMPLE PROGRAMS X
               DESTFAC=TERMINAL,     TERMINAL                          X
               TRANSID=AORQ,         <CHANGE IF WANT TO RUN 'PORQ','OREQ'X
               TRIGLEV=30            AORQ IS AUTO INIT'ED WHEN QUEUE=30
```

The first destination, LOGA, is directed to the output printer in the same way as the destinations in the pregenerated DCT. The other destination, L860, is an example of an intrapartition destination with a trigger level of 30 and an automatically-initiated transaction identifier AORQ (see "Intrapartition Destinations" on page 38 for a description of the trigger-level mechanism).

This means that, whenever a transaction places the thirtieth element on the L860 queue, the AORQ transaction is automatically initiated by CICS to process the queue, and the queue is effectively emptied again.

Before you assemble the sample DCT that includes these destinations, you must ensure that the AORQ transaction identifier is made to correspond to the language you are using, that is, OREQ for COBOL, PORQ for PL/I, or RORQ for RPG II.

You then assemble the DCT by running the following job:

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHDCTS1
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=U160PCL
// LIBDEF RL,SEARCH=C160PRL
// LIBDEF SL,SEARCH=C160PSL
// EXEC ASSEMBLY
         DFHDCT  TYPE=INITIAL,                                         *
                 SUFFIX=S1
         COPY    DFHXDCT1
         COPY    DFHXDCT
         COPY    DFHXDCTS
         DFHDCT  TYPE=FINAL
    END
/*
// EXEC LNKEDT
/&
```

Figure 28. Assembling and Link-Editing a Sample DCT

FILE CONTROL TABLE (FCT)

The file control table is used to define to CICS the user files or data sets required to run your application programs. (Note, however, that sequential files are defined as extrapartition destinations in the destination control table.)

The only FCT entry necessary for the running of the sample programs is for the file known as FILEA (from which the sample application package is sometimes referred to as the FILEA samples). This is a VSAM file that has been created and loaded with the initial data used for the running of the sample applications.

The DFHXFCTS copy book contains the single entry defining FILEA:

```
         TITLE 'DFHXFCTS - COPYBOOK FOR FILEA'
         DFHFCT TYPE=DATASET,DATASET=FILEA,ACCMETH=(VSAM,KSDS),        *
               SERVREQ=(GET,PUT,UPDATE,NEWREC,BROWSE,DELETE),          *
               RECFORM=(FIXED,BLOCKED),BUFND=3,BUFNI=2,STRNO=2,        *
               OPEN=INITIAL
```

The following job should then be used to assemble and link-edit the FCT:

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHFCTS1
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=U160PCL
// LIBDEF RL,SEARCH=C160PRL
// LIBDEF SL,SEARCH=C160PSL
// EXEC ASSEMBLY
         DFHFCT  TYPE=INITIAL,                                         *
                 SUFFIX=S1
         COPY    DFHXFCTS
         DFHFCT  TYPE=FINAL
   END
/*
// EXEC LNKEDT
/&
```

Figure 29. Assembling and Link-Editing a Sample FCT

TERMINAL CONTROL TABLE (TCT)

In order to run the samples, it is necessary to define at least one 3270-type display terminal and one printer in addition to the console and sequential card reader/line printer definitions already contained in the pregenerated TCT with suffix 1$ used to start up the initial system in Part 1 of this book.

The DFHXTCTC copy book defines the VSE console for use as a CICS terminal:

```
         TITLE  'DFHXTCTC-SAMPLE TCT-COPYBOOK FOR CONSOLE'
            DFHTCT TYPE=GPENTRY,                                        X
                 GPTYPE=CONSOLE,      CONSOLE SUPPORT                   X
                 TRMIDNT=CNSL,        TERMINAL NAME                     X
                 TRMUAL=32,           USER AREA LENGTH                  X
                 LININL=80            TIOA LENGTH
```

The DFHXTCTS copy book defines a pair of input and output sequential devices that can be used to simulate a terminal to CICS. In that copy book, the card reader and line printer combination simulate a terminal known to a CICS application as SAMA. Input can be submitted through the card reader (SYSRDR), and output will be sent to the line printer (SYSLST).

```
        TITLE  'DFHXTCTS - COPYBOOK TCT ENTRIES FOR SEQUENTIAL (CRLP) TX
               ERMINAL ENTRIES'
***********************************************************************
* * *                       D  T  F  S                          * * *
***********************************************************************
         DS    0D
         DC    CL32'INPUT AND OUTPUT DTFS'
         DFHTCT  TYPE=SDSCI,                                          *
               DEVADDR=SYSRDR,                                        *
               DEVICE=2540,                                           *
               DSCNAME=READER
         DFHEJECT
         DFHTCT  TYPE=SDSCI,                                          *
               DEVADDR=SYSLST,                                        *
               DEVICE=1403,                                           *
               DSCNAME=PRINTER
         DFHEJECT
***********************************************************************
* * *                  L I N E   A N D   T E R M I N A L        * * *
* * *                  D E F I N I T I O N                      * * *
***********************************************************************
         DS    0D
         DC    CL64'T E R M I N A L   A N D   L I N E   E N T R I E S'
***********************************************************************
         DS    0D
         DC    CL32'S A M A  LINE AND TERMINAL ENTRY'
         DFHTCT  TYPE=LINE,                                           *
               ACCMETH=SEQUENTIAL,                                    *
               TRMTYPE=CRLP,                                          *
               ISADSCN=READER,                                        *
               OSADSCN=PRINTER,                                       *
               INAREAL=88,                                            *
               DUMMY=DUMMY
         DFHTCT  TYPE=TERMINAL,                                       *
               TRMIDNT=SAMA,                                          *
               TRMPRTY=32,                                            *
               TRMSTAT=(TRANSCEIVE),                                  *
               DUMMY=DUMMY
***********************************************************************
* * *                  F I N A L I Z E                          * * *
***********************************************************************
         DS    0D
         DC    CL32'T C T   WAIT LIST'
```

If you are using BTAM, the DFHXTCBS copy book defines two local display terminals with symbolic units SYS020 and SYS021 and a printer with symbolic unit SYS027:

```
TITLE  'DFHXTCBS-BTAM SAMPLE TCT-COPYBOOK FOR LOCAL TERMINALS'
DFHTCT TYPE=GPENTRY,              1 LINE FOR:                   X
       GPTYPE=3270L,                 LOCAL 3270 DEVICE          X
       LININL=160,                   TIOA LENGTH                X
       LINELST=(020),                ADDRESS                    X
       TRMFEAT=(DSUA),               FEATURE                    X
       TRMSTAT=(A),                  TERMINAL STATUS            X
       TRMMODL=(2A),                 TERMINAL MODEL             X
       TRMUAL=(32),                  USER AREA LENGTH           X
       TRMIDNT=(L77A)                TERMINAL NAME
DFHTCT TYPE=GPENTRY,              1 LINE FOR:                   X
       GPTYPE=3270L,                 LOCAL 3270 DEVICES         X
       LININL=160,                   TIOA LENGTH                X
       LINELST=(021,027),            ADDRESSES                  X
       TRMFEAT=(DSUA,P),             FEATURES                   X
       TRMSTAT=(A,A),                TERMINAL STATUS            X
       TRMMODL=(2A,2C),              TERMINAL MODELS            X
       TRMUAL=(32,32),               USER AREA LENGTH           X
       TRMIDNT=(L77B,L860)           TERMINAL NAMES
```

You can use this copy book unchanged, but you must assign in your CICS startup job the addresses of the devices to the symbolic units. For example:

```
ASSGN  SYS020,0A1
```

where 0A1 is the address of a display terminal, or

```
ASSGN  SYS020,IGN
```

if commands to the symbolic unit SYS020 are to be ignored.

The following job can then be used to assemble and link-edit the DFHTCTS1 control table:

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHTCTS1
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=U160PCL      (your private core image library)
// LIBDEF RL,SEARCH=(C160PRL,BTMRLB) (include BTAM relocatable library)
// LIBDEF SL,SEARCH=(C160PSL,BTMSLB)  (include BTAM source library)
// EXEC ASSEMBLY
       DFHTCT  TYPE=INITIAL,                                    *
               ACCMETH=NONVTAM,                                 *
               SUFFIX=S1
       COPY    DFHXTCTS
       COPY    DFHXTCTC
       COPY    DFHXTCBS
       DFHTCT  TYPE=FINAL
   END
/*
// EXEC LNKEDT
/&
```

Figure 30. Assembling and Link-Editing a Sample BTAM TCT

If you are using VTAM, the copy book to be used is DFHXTCVS:

```
        TITLE  'DFHXTCVS - COPYBOOK ENTRIES FOR VTAM SAMPLE TCT'
***********************************************************************
* L77A - VTAM SNA LUTYPE2.               NETNAME - D72L301.           *
* L77D - VTAM NON-SNA 3270.              NETNAME - D72L304.           *
* L860 - VTAM NON-SNA 3270 PRINTER.      NETNAME - P42L308.           *
***********************************************************************
VTRM1    DFHTCT TYPE=TERMINAL,TRMIDNT=L77A,TRMTYPE=LUTYPE2,TRMMODL=2,   *
               TIOAL=(1500,1920),RELREQ=(NO,YES),                      *
               BUFFER=1536,                                            *
               FEATURE=(SELCTPEN,AUDALARM,UCTRAN,DCKYBD),              *
               CHNASSY=YES,                                            *
               NETNAME=D72L301,                                        *
               PRINTTO=(VTRM3),                                        *
               GMMSG=NO,                                               *
               ACCMETH=VTAM,TCTUAL=32,                                 *
               CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM2    DFHTCT TYPE=TERMINAL,TRMIDNT=L77D,TRMTYPE=3270,TRMMODL=2,      *
               CLASS=(CONV,VIDEO),TIOAL=1500,RELREQ=(NO,YES),          *
               FEATURE=(SELCTPEN,AUDALARM,UCTRAN,DCKYBD),              *
               NETNAME=D72L304,                                        *
               PRINTTO=(VTRM3),                                        *
               GMMSG=YES,                                              *
               ACCMETH=VTAM,TCTUAL=32,                                 *
               CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM3    DFHTCT TYPE=TERMINAL,TRMIDNT=L860,TRMTYPE=3270P,TRMMODL=2,     *
               NETNAME=P42L308,                                        *
               RELREQ=(YES,YES),                                       *
               CLASS=(CONV,VIDEO),TIOAL=1500,TCTUAL=32,                *
               ACCMETH=VTAM,CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
```

This defines a non-SNA local terminal (3270) with terminal name L77A and a SNA local terminal (LUTYPE2) with terminal name L77B. Here, you must ensure that the NETNAMEs for these terminals and for the printer agree with the corresponding definitions in the VTAM B. books.

If the NETNAMEs agree, you assemble and link-edit the TCT as before:

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHTCTS1
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=U160PCL      (your private core image library)
// LIBDEF RL,SEARCH=(C160PRL,VTMRLB)  (include VTAM relocatable library)
// LIBDEF SL,SEARCH=(C160PSL,VTMSLB)  (include VTAM source library)
// EXEC ASSEMBLY
       DFHTCT  TYPE=INITIAL,                                           *
               ACCMETH=(VTAM,NONVTAM)                                  *
               SUFFIX=S1
       COPY    DFHXTCTS
       COPY    DFHXTCTC
       COPY    DFHXTCVS
       DFHTCT  TYPE=FINAL
   END
/*
// EXEC LNKEDT
/&
```

Figure 31. Assembling and Link-Editing a Sample VTAM TCT

Note: If you wish to use your own TCT with the sample programs, you should ensure that uppercase translate is defined for the display terminal, and the terminal identifier for the printer must be L860.

CONSTRUCTING THE STARTUP JOB STREAM

At this point, all the definitions necessary for running the sample application programs have been completed. You are in a position to start up CICS again and try them out. In the sense that you now have a system that will perform a nontrivial piece of work, you have completed the installation of a CICS system.

In putting together the job stream to start up CICS, you need to identify (or define) to VSE, either explicitly or implicitly, all the resources external to CICS that VSE is to make available to CICS during execution. These definitions are, in many cases, performed by means of VSE DLBL and EXTENT job control statements, the filename in a DLBL statement being the symbolic identifier that CICS uses when making requests for the corresponding resource.

Other resources are defined to the system, not by explicit DLBL and EXTENT statements, but rather by any of a number of different implicit methods. We will now consider each of the various resources defined to CICS by means of CICS control tables earlier in this chapter, and examine how each is defined to the operating system.

Programs and Maps

Programs and Basic Mapping Support (BMS) maps are entities that have been link-edited as phases in the various core image libraries. As such, they are known to VSE without having to be explicitly identified. The DLBL and EXTENT statements for the core image libraries must, however, be included in the startup job stream, together with a LIBDEF statement indicating the library search order to be used by VSE when searching for a particular application program.

The CICS-provided application programs (that is, those that implement CICS-supplied transactions, for example, CEMT, CEDF, or CECI) and the pregenerated application programs have been link-edited in the CICS private core image library (C160PCL in our example). Your own application programs will have been link-edited to one or more user core image libraries (for example, U160PCL defined earlier in this chapter).

The search order in the LIBDEF statement should be such that any user libraries are searched before the CICS core image library. If you are running CICS with PLI=YES (or PL1=YES) specified in the initialization parameters, you should also include the PL/I core image library in the search chain.

Transactions

Whereas application programs and maps are physical entities (phases in a core image library), transactions are no more than identifiers (symbolic names). Insofar as the names are chosen by the user, they have to be defined to CICS in the program control table. VSE, however, is unaware of transaction identifiers, and no VSE definitions are required.

The program control table associates each transaction identifier with the corresponding application program that CICS is to invoke to satisfy the transaction request. The program control table can, therefore, be considered to be a logical extension of the program processing table, which defines the physical resources.

The program control table and the program processing table (as well as the other CICS control tables) will have been link-edited to a user core image library; the DLBL and EXTENT statements for this library, together with its LIBDEF statement, must be included in the startup job stream.

Queues or Destinations

Temporary storage queue names (DATAIDs) are dynamically created (and may be deleted) by application programs during the running of CICS. These queue names are CICS resources managed internally by CICS, which creates its own tables for this purpose. There is, of course, no question of defining these resources to VSE. The temporary storage data set, in which CICS allocates space for the queues, must be defined in the startup job stream, however.

Intrapartition transient data queue names (destinations) are similarly managed internally by CICS. The names here are predefined in the destination control table and, again, are not known to VSE. The intrapartition transient data set must, however, be included in the startup job stream.

Extrapartition transient data destinations refer to physical files that may be individually processed not only by CICS transactions, but also by other programs unconnected with the CICS system (for example, by batch programs running in a different partition).

Extrapartition transient data files must be defined to VSE in the startup job stream as sequential files, either on disk or on tape or as printer output. (The printer output is, of course, not processed by a program but instead is printed immediately or, if you are using a spooling system, is batched up and is printed at the end of the CICS run.)

The extrapartition transient data file definitions in the startup job stream must correspond to the definitions in the destination control table (DCT). For disk files, DLBL and EXTENT statements are required; the filename must be the same as the DSCNAME operand in the DCT definition. Also, the symbolic unit address, if specified in the EXTENT statement, overrides any that may have been specified in the DEVADDR operand in the DCT.

For tape files, the symbolic unit address, as specified in the DCT, should be assigned (ASSGN) in the startup job stream to the physical unit on which the tape is to reside. If standard-labeled tapes are used, a TLBL statement is required.

Finally, if extrapartition printer output is defined in the DCT by means of the DEVADDR=SYSLST operand, the VSE ASSGN for SYSLST is used and no further job control statement is required in the startup job stream.

User Files and Data Bases

User files are held on direct access storage devices and are defined as CICS resources in the file control table. The user files may be managed by any of the following access methods: VSAM, ISAM, or DAM, or they may be managed by the data base subsystem, DL/I.

User files are defined to VSE in the startup job stream in different ways, dependent on the access method used. In all cases, however, a DLBL job control statement is required, whose file name is the same as the DATASET operand in the corresponding entry in the file control table. The DLBL statement should also include the file identifier of the file, as stored on the direct access storage device.

VSAM files must have been defined previously within VSAM space and associated with an existing VSAM user catalog. The DLBL statement for such a file must include the identifier VSAM and may include the filename of the user catalog that owns the file. The DLBL statement for the user catalog must also be included in the startup job stream. Thus, for example, the VSAM file FILEA used in the sample application package would be defined by:

```
// DLBL  FILEA,'SAMPLE.TEST.FILEA',,VSAM,CAT=CICSUCT
```

ISAM and DAM files require, in addition to the DLBL statement, an EXTENT statement indicating the location of the file on the direct access storage device and an ASSGN statement specifying the physical address on which the direct access storage device is mounted. The DLBL statement must include the identifier ISE to indicate an ISAM file, or DA to indicate a DAM file.

Further examples of the job control statements needed to define user files in the startup job stream are given in "Chapter 4.11. User File Definitions."

DL/I data bases are VSAM files, and are defined in the startup job stream in exactly the same way as VSAM files processed directly by CICS. In addition, if DL/I data bases are being used, two further tables must be defined in the startup job stream. Details of how to code these definitions are given in "Chapter 4.13. DL/I Definitions."

Terminals

Not all the terminals connected to a host system are necessarily known to the CICS system. Only those terminals defined in the particular terminal control table selected during CICS initialization are made known to CICS and thereby become CICS resources. These terminals must also be made known, either explicitly or implicitly, to the operating system.

The method of defining terminals to the system and the requirement or otherwise to include job control statements for them in the startup job stream depends on whether the terminal access method used for that terminal (and reflected in the terminal control table definition) is BTAM or VTAM.

Local BTAM Terminals : Each terminal is connected (usually via a channel-attached local control unit) to a unique physical address (channel and unit address).

If you are using BTAM, you should code, in the TYPE=SDSCI macro in the terminal control table, a LINELST operand specifying a list of three-digit numbers, each of which is interpreted as the symbolic unit address of a single terminal. All the symbolic unit addresses coded in a particular LINELST operand may be considered to be related, for example referring to the terminals controlled by a single control unit.

The TYPE=SDSCI macro is followed by a single TYPE=LINE macro. (In the case of local terminals, this macro does not actually represent a physical line, but is merely required to build an internal control block.) The TYPE=LINE macro must be immediately followed by one or more TYPE=TERMINAL macros describing the actual terminals and corresponding, in order, to the symbolic unit addresses in the LINELST operand. (You may alternatively use the TYPE=GPENTRY macro as, for example, in the DFHXTCBS sample copy book described earlier; all these macro definitions are then replaced by a single macro.)

In the startup job stream, each of the symbolic unit addresses defined in the LINELST operand must be assigned to the physical address to which the corresponding terminal is connected, for example:

```
  //  ASSGN  SYS021,0A2
```

where SYS021 is the symbolic unit address and 0A2 is the physical address.

Thus, when a CICS application program refers to a terminal, which it identifies by the symbolic name in the TRMIDNT operand in the DFHTCT TYPE=TERMINAL macro, the CICS system associates the terminal with the symbolic unit address (by position in the LINELST operand) and this is finally associated with the real address by the ASSGN job control statement.

Remote BTAM Terminals: A variety of methods of connecting remote terminals with the host computer is used, depending on the characteristics of the terminals and intervening controllers. In general, however, a group of terminals may be attached to a control unit which, in turn, is connected via modems and a telecommunications line to a transmission control unit (or communications controller). The transmission control unit is attached directly to a system channel.

The transmission control unit is able to control a number of lines as described above, each of these lines corresponding to a unique physical address (channel and unit address). The way such a configuration is described in the CICS terminal control table is to code a TYPE=SDSCI macro for each transmission control unit controlling transmissions to CICS terminals.

As before, this macro contains a LINELST operand with a list of three-digit numbers; these are now interpreted as symbolic unit addresses of the lines connected to the controller. It is these lines (as compared with the actual terminals in the local case) that are associated with the physical addresses.

For each line controlled by a terminal communications unit, a TYPE=LINE macro must be coded in the same order as the corresponding LINELST entries. Each TYPE=LINE macro must be immediately followed by one or more TYPE=TERMINAL macros describing the terminals on that line.

In the startup job stream, each of the symbolic unit addresses defined in the LINELST operand is assigned to the physical address of that line. Thus:

```
//    ASSGN  SYS030,0C0
```

assigns the symbolic unit address SYS030 to the physical address 0C0.

The individual terminals on a line are further identified by polling and addressing characters that are part of the actual data stream traveling along the line. The polling and addressing characters are recognized by the control unit to which the terminals are attached, which itself delivers the data to or from the actual terminal.

The polling and addressing characters for the terminals on a line are built into the control unit to which the terminals are attached. The polling and addressing characters are made known to CICS by:

1. Preceding the corresponding TYPE=LINE macro in the terminal control table with a set of DTFTRMLST macro statements that define the characters, and

2. Associating the TYPE=LINE macro with the list of polling characters and each TYPE=TERMINAL macro with the corresponding addressing characters.

VTAM Terminals: If you are using VTAM, no definitions in the CICS startup job stream are required. VTAM, or the network control program, contains tables in which the network of terminals it controls is described, and it uses these to manage the flow of data between CICS and the terminals, using only the I/O facilities of the operating system.

The local terminals controlled by VTAM may be SNA terminals connected to a channel-attached cluster controller, or they may be non-SNA 3270 terminals connected via a local control unit (as in the BTAM configuration).

Remote terminals controlled by VTAM are attached to a SNA cluster controller, which is connected via an SDLC line with a channel-attached communications controller. The communications controller may also be loaded with code to enable remote terminals to be connected to it via a binary synchronous (BSC) line.

Terminals, controllers, and lines are defined in VTAM tables as nodes in the network. In particular, each terminal known to VTAM (or rather, each logical unit (LU) in the case of SNA terminals) has been defined in the VTAM table definitions with a VTAM node name that is unique throughout the VTAM domain.

A given VTAM terminal (or logical unit) defined in the CICS terminal control table has two names associated with it: the NETNAME, which is the same as the VTAM node name, and the TRMIDNT, which is the name used by application programs when referring to that terminal.

Before using a terminal control table with VTAM terminal definitions, you should therefore make sure that the NETNAMEs defined in the table agree with the VTAM node names in your VTAM system (as held in the B. books in the VSE libraries). If the names do not agree, it is probably most convenient to alter the NETNAMEs in the terminal control table and reassemble and link-edit it to your core image library in the usual way.

CICS Data Sets

The startup job stream must contain not only definitions of the user resources discussed up to this point, but also definitions of any CICS data sets that are to be used.

In addition to the dump and auxiliary trace data sets described in "Chapter 2.2. System Data Sets," you may also need to define:

- The restart data set, if you are using AUTO start (see "Chapter 2.4. Restart and Recovery of CICS")
- The system log, if you are running recoverable transactions (see "Chapter 2.4. Restart and Recovery of CICS")
- User journals, if you wish to record audit-type information or perform forward recovery (see "Chapter 4.5. Journal Data Sets")
- The CICS system definition file, if you wish to define certain resources interactively (see "Chapter 4.10. CICS System Definition File")
- The statistics data sets, if you intend to gather automatic statistics (as distinct from requested or termination statistics) (see "Chapter 4.7. Automatic Statistics Data Sets").

CICS Core Image Library

The CICS private core image library contains the pregenerated modules and tables; and these are used during CICS initialization to load the CICS system into the CICS partition. The CICS private core image library must be defined to VSE either in the startup job stream or, more commonly, in the partition or system standard labels (see "Starting Up CICS" on page 25).

Initialization Overrides

These should be chosen so that the CICS initialization process loads a CICS system that is suited to your precise requirements. The initialization override parameters included in the job stream supersede or supplement the initialization parameters defined in the system initialization table.

After you are satisfied that the override parameters are correctly specified for your environment, you may prefer to code them in a new system initialization table with a suffix chosen by yourself, and use that suffix as the sole override parameter in the startup job stream.

STARTING UP CICS TO RUN THE SAMPLE APPLICATION PROGRAMS

The following job stream incorporates most of the topics that have already been described earlier in this chapter, and you may wish to run the job to test the sample application programs provided with CICS.

```
// JOB CICS START UP CICS SYSTEM FOR SAMPLE RUN
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
* User VSAM Catalog
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
* Auxiliary Temporary Storage
// DLBL DFHTEMP,'CICS160.TEMP.STORAGE',,VSAM,CAT=CICSUCT
* Intrapartition Transient Data
// DLBL DFHNTRA,'CICS160.INTRA.TRANSD',,VSAM,CAT=CICSUCT
* FILEA Sample Data Set
// DLBL FILEA,'SAMPLE.TEST.FILEA',,VSAM,CAT=CICSUCT
* Dump Data Sets
// DLBL DFHDMPA,'CICS160.DUMP.DATASET.A',0,SD
// EXTENT SYS001,,1,0,210,40
// DLBL DFHDMPB,'CICS160.DUMP.DATASET.B',0,SD
// EXTENT SYS001,,1,0,250,10
* Auxiliary Trace
// DLBL DFHAUXT,'CICS160.AUX.TRACE.A',0,SD
// EXTENT SYS001,,1,0,260,40
// DLBL DFHBUXT,'CICS160.AUX.TRACE.B',0,SD
// EXTENT SYS001,,1,0,300,40
* ASSGN of DASD Containing CICS and User Files
// ASSGN SYS001,DISK,TEMP,VOL=M3350A,SHR
* Search Order for Core Image Libraries
// LIBDEF CL,SEARCH=(U160PCL,C160PCL)
* Local Terminals Defined in Sample BTAM TCT
ASSGN SYS020,0A1
ASSGN SYS021,0A2
ASSGN SYS027,0A4
* Start Up CICS With Override Parameters From SYSIPT
// EXEC DFHSIP,SIZE=1200K,PARM='SIT=1$,SI'
 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 PPT=S1,PCT=S1,DCT=S1,FCT=S1,FCP=3$,TCT=S1
 BMS=S$,PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 $END
/*
/&

                        (This job is included in Z.DFHSPJCL)
```

Figure 32. Starting Up CICS to Run the Sample Application Package

In the above job stream, the override parameters include the tables assembled previously, and also:

```
FCP=3$          Support for VSAM files
BMS=S$          Full-function BMS required to run the
                 sample transactions
PGRET=P/    )
PGPURGE=T/  )   To allow BMS paging to be performed
PGCOPY=C/   )
PGCHAIN=X/  )
```

When CICS initialization is complete and the message:

DFH1500 - CONTROL IS BEING GIVEN TO CICS

is received at the CICS system console, the CICS terminals that have been powered on are enabled for the entry of transactions. At this point, a VTAM terminal displays the following panel (you can suppress the display or change the "WELCOME TO CICS/VS" message by means of the GMMSG= operand of the DFHTCT TYPE=TERMINAL macro, and the GMTEXT= operand of the DFHTCT TYPE=INITIAL macro respectively):

```
***DFH2312   WELCOME TO CICS/VS  *** 16:18:32




          CCCCCC    IIIII    CCCCCC     SSSSSS         VVVV  VVVV    SSSSSS
        CCCCCCCC   IIIII   CCCCCCCC   SSSSSSSS         VVV  VVV    SSSSSSSS
       CCCC  CC    III    CCCC  CC   SSSS  SS          VVV  VVV   SSSS  SS
      CCC         III    CCC         SSSS       ***    VVV VVV    SSSS
     CCC         III    CCC           SSSS     ***     VVVVVV      SSSS
    CCCC  CC    III    CCCC  CC   SS  SSSS             VVVVV   SS  SSSS
   CCCCCCCC   IIIII   CCCCCCCC   SSSSSSSS              VVVV   SSSSSSSS
   CCCCCC    IIIII    CCCCCC     SSSSSS                VVV    SSSSSS
```

A BTAM terminal does not display any special message when CICS initialization is completed. However, the screen is cleared and the "SYSTEM AVAILABLE" indicator, if any, is turned on. To make sure the terminal is connected to CICS, you can press the ENTER key. If so, you will receive the message:

DFH2001 INVALID TRANSACTION IDENTIFICATION - PLEASE RESUBMIT

You can now enter transactions at the screen. You may wish to test the full-screen operation of the CICS-provided master terminal transaction CEMT. The CEMT transaction was introduced in Figure 19 on page 48, where the commands were entered from the console.

Clear the screen and enter "cemt inquire dump". The following screen is displayed:

```
INQUIRE DUMP
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Dum(DFHDMPA ) Ope















 RESPONSE: NORMAL                          TIME:      4.29   DATE:  83.071
PF: 1 HELP       3 END                      7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Return the cursor to the beginning of line 1 and overtype "INQUIRE DUMP" with "set dump close". Press ENTER. The resulting screen is:

```
SET DUMP CLOSE
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Dum(DFHDMPA ) Clo                                          NORMAL















 RESPONSE: NORMAL                          TIME:      4.57   DATE:  83.071
PF: 1 HELP       3 END                      7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

The same effect could have been achieved by moving the cursor to the start of the word "Ope" in the first screen and overtyping the letter "O" with "c".

Move the cursor either to the start of the word "CLOSE" and overtype with "open", or to the start of the word "Clo" and overtype with "o". Press ENTER. The screen display will be:

```
SET DUMP OPEN
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Dum(DFHDMPA ) Ope                                                    NORMAL






 RESPONSE: NORMAL                                TIME:      5.15   DATE:  83.071
PF: 1 HELP       3 END                            7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Return the cursor to the beginning of line 1 and overtype with "perform snap title('User''s Sample Dump')". Press ENTER. A formatted dump of the CICS partition is recorded (because FDP=(,FORMAT) in DFHSIT1$). The screen displayed is:

```
PERFORM SNAP TITLE('USER''S SAMPLE DUMP')
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Sna






 RESPONSE: NORMAL                                TIME:      6.02   DATE:  83.071
PF: 1 HELP       3 END                            7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Overtype the words "PERFORM SNAP" with "set dump switch". Press ENTER. The DFHDMPA data set has been closed automatically and is now ready to be printed (using DFHDUP), while the alternate data set DFHDMPB has been opened for output. The screen now shows:

```
SET DUMP SWITCH
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Dum(DFHDMPB ) Ope                                                  NORMAL















 RESPONSE: NORMAL                               TIME:      6.30   DATE:  83.071
PF: 1 HELP       3 END                           7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Press PF3. The following screen is now displayed:

```
CEMT SET DUMP SWITCH
 STATUS:  SESSION ENDED
```

At this point, the CEMT transaction has been completed and you may continue by typing the same or a different transaction. The following figure shows the full-screen versions of the transactions shown in "Chapter 2.2. System Data Sets." You might wish to complete these and demonstrate that the results are identical to those obtained in that chapter.

```
CEMT INQUIRE DUMP
    Shows initial status of the dump data set.  This should be OPE
SET DUMP CLOSE
    The data set is changed to CLO
SET DUMP OPEN
    The data set is opened again.
PERFORM SNAP TITLE('USER''S SAMPLE DUMP')
    This will produce a CICS system dump for later use.
SET DUMP SWITCH
    Switch the dump data set.
<PF3>

<CLEAR>

CSTT AOR
   This will put the statistics collected to date to the
    system printer or to SYSLST.
<CLEAR>

CEMT INQUIRE AUXTRACE
   Shows initial status of auxiliary trace.  Should be OFF CLO
SET AUXTRACE ON OPEN
   Opens the data set and starts auxiliary trace.
SET AUXTRACE OFF CLOSE

SET AUXTRACE SWITCH

<PF3>

<CLEAR>

CEMT PERFORM SHUTDOWN
```

You may now wish to run the sample application programs. Because FILEA is not open, you must use CEMT to open it. The full command to use is:

```
cemt set dataset(filea) open
```

After the file is open, you can run AMNU (or one of its language alternatives). Type AMNU and follow the screen prompts. A description of the sample applications is given in Appendix E and also in the CICS/VS Application Programmer's Reference Manual (Command-Level).

You might also like to test the CICS execution diagnostic facility. Type CEDF, then CLEAR the screen. Type AMNU. Each CICS request issued by the application program is displayed on the screen and the opportunity is given to alter certain of the arguments. CEDF is a very powerful online means of tracing the execution of an application program and of making online changes to EXEC CICS statements without the necessity of reassembling or recompiling the program.

Chapter 2.4. Restart and Recovery of CICS

Among the various messages displayed during CICS initialization, the message:

```
DFH1500 - CICS START-UP IS COLD
```

is of particular interest.

This message means that CICS is being started up entirely afresh and without reference to any previous run. In such a situation, it is possible that valuable information may be lost from one run to the next.

> For example, during a CICS run, records of data might be written to an intrapartition transient data queue. The queue might be defined with a trigger level, so that whenever the number of records in the queue reaches a certain value, a transaction is automatically initiated to process them. At the time CICS is shut down, it is probable that a number of records remain on the queue, waiting to be processed. If the subsequent startup is cold, intrapartition transient data is cleared and the records are lost.
>
> To prevent data being lost in this way, CICS can preserve it between one run and another and can perform what is known as a **warm restart**. A warm restart, however, relies on information having been saved, in the form of a **warm keypoint**, as part of the processing carried out is used during the following startup to re-create the data and the status of CICS resources, and to enable the system to continue largely as if the shutdown had not occurred.

REQUIREMENTS FOR A WARM KEYPOINT TO BE TAKEN

Two things are necessary to enable this saving to be carried out. First, CICS must be running with a further data set, the **restart data set**, defined. The restart data set serves as the repository of all the information that may have to be restored during the following startup.

Second, the CICS management modules must include the **keypoint program** (DFHKPP). One of the functions of the keypoint program is to record the warm keypoint on the restart data set during a normal shutdown. The information in a warm keypoint includes:

- Destination control table entries for intrapartition queues.
- Auxiliary temporary storage queue names and pointers to records in these queues. (Temporary storage queues held in main storage are lost across a warm restart.)
- Entries in the file control table, program control table, processing program table, and terminal control table (except those for BTAM switched terminals)
- Control information representing requests to initiate transactions, which have not yet been honored at the time of the shutdown.

AUTOMATIC STARTUP

After having successfully completed a warm keypoint, the keypoint program writes a further record (the **control record**) to indicate that the next time CICS is started up, a warm restart may be performed. What actually occurs at the next startup depends on the system initialization parameters (or override parameters) used.

To take full advantage of the warm keypoint, CICS must be started with the START=AUTO system initialization parameter. CICS then tests the control record during initialization and performs a warm restart if it is possible to do so.

If START=AUTO is used, it is possible to perform a warm restart on certain of the CICS facilities and not on others. This is done by specifying the (,COLD) setting for the corresponding override parameters for which a cold start is required (see "Chapter 5.2. The CICS Startup Job Stream" for details).

If, instead, the START=COLD system initialization parameter is used, a **cold start** is forced, irrespective of whether a warm keypoint was taken previously or not (the pregenerated system initialization table DFHSIT1$ includes START=COLD). Again, if the restart data set is not defined, or if the keypoint program does not exist, then, whatever the value of the START parameter, a cold start is forced by the CICS system.

You should use START=AUTO for all normal operations unless there are good reasons for forcing a specific type of start.

CREATION OF THE RESTART DATA SET

The restart data set is a key-sequenced VSAM file that you must define in VSAM space and then load with an initial record (the control record) before it can be used by CICS.

The control record contains a value such that, even with START=AUTO, CICS will recognize that the restart data set is newly created and will, therefore, perform a cold start.

The following job can be used to create the restart data set, belonging to the user catalog CICS160.UCAT:

```
// JOB DEFRSD CREATE RESTART DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                       -
          (NAME(CICS160.RSD)          -
           INDEXED                    -
           RECORDSIZE(2000 2000)      -
           RECORDS(250 100)           -
           KEYS(10 0)                 -
           FREESPACE(20 20)           -
           SHAREOPTIONS(2)            -
           VOLUMES(M3350A))           -
        DATA                          -
          (NAME(CICS160.RSD.DATA))    -
        INDEX                         -
          (NAME(CICS160.RSD.INDEX))   -
        CATALOG(CICS160.UCAT)
/*
/&
                      (This job is included in Z.DFHSPJCL)
```

Figure 33. Defining the Restart Data Set

The following job should then be used to load the initial record:

```
// JOB INITRSD INITIALISE RESTART DATA SET
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
// DLBL DFHRSD,'CICS160.RSD',,VSAM,CAT=CICSUCT
// EXEC IDCAMS,SIZE=AUTO
 REPRO  INFILE                        -
          (SYSIPT                     -
           ENVIRONMENT                -
            (RECORDFORMAT(FIXUNB)     -
             BLOCKSIZE(80)            -
             RECORDSIZE(80)))         -
         OUTFILE(DFHRSD)
ACTL 0001
/*
/&
                      (This job is included in Z.DFHSPJCL)
```

Figure 34. Initializing the Restart Data Set

DEMONSTRATION OF WARM RESTART

You may now wish to demonstrate, with the aid of a very simple example, how warm keypointing and warm restart work. You will need to start up CICS again with the restart data set defined in the startup job stream and with

```
KPP=D$
```

(the only version provided) and with START=AUTO specified in the override parameters.

The startup job is now:

```
// JOB CICS WITH WARM RESTART
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
* User VSAM Catalog
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
* Auxiliary Temporary Storage
// DLBL DFHTEMP,'CICS160.TEMP.STORAGE',,VSAM,CAT=CICSUCT
* Intrapartition Transient Data
// DLBL DFHNTRA,'CICS160.INTRA.TRANSD',,VSAM,CAT=CICSUCT
* Restart Data Set
// DLBL DFHRSD,'CICS160.RSD',,VSAM,CAT=CICSUCT
* FILEA Sample Data Set
// DLBL FILEA,'SAMPLE.TEST.FILEA',,VSAM,CAT=CICSUCT
* Dump Data Sets
// DLBL DFHDMPA,'CICS160.DUMP.DATASET.A',0,SD
// EXTENT SYS001,,1,0,210,40
// DLBL DFHDMPB,'CICS160.DUMP.DATASET.B',0,SD
// EXTENT SYS001,,1,0,250,10
* Auxiliary Trace
// DLBL DFHAUXT,'CICS160.AUX.TRACE.A',0,SD
// EXTENT SYS001,,1,0,260,40
// DLBL DFHBUXT,'CICS160.AUX.TRACE.B',0,SD
// EXTENT SYS001,,1,0,300,40
* Search Order for Core Image Libraries
// LIBDEF CL,SEARCH=(U160PCL,C160PCL)
* ASSGN of DASD containing CICS and User Files
// ASSGN SYS001,DISK,TEMP,VOL=M3350A,SHR
* Local Terminals Defined in Sample BTAM TCT
ASSGN SYS020,0A1
ASSGN SYS021,0A2
ASSGN SYS027,0A4
* Start Up CICS With Override Parameters From SYSIPT
// EXEC DFHSIP,SIZE=1200K,PARM='SIT=1$,SI'
 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 PPT=S1,PCT=S1,DCT=S1,FCT=S1,FCP=3$,TCT=S1
 BMS=S$,PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 KPP=D$
 START=AUTO,$END
/*
/&
```

Figure 35. Starting Up CICS with Warm Restart Capability

The first time you start up CICS using this job stream, a cold start will be performed, because no warm keypoint exists in the restart data set. Before shutting down CICS with a warm keypoint, and in order to have something to record, you can now use the AORD sample transaction (or any of its high-level language alternatives) to perform some processing.

For this demonstration, you should make sure that you have a local printer connected and known to CICS. If you are using BTAM, the printer should be connected on the same address as is assigned to SYS027: 0A4 in the above example. If you are using VTAM, the address should agree with that specified in the appropriate VTAM B-book.

The AORD transaction refers to the FILEA user file, which has been defined as initially closed. You must first open the file, as described in the previous chapter, by means of the command:

```
cemt set dataset(filea) open
```

You may also wish to modify the trigger level of the intrapartition transient data queue L860 from its assembled value of 30 to the value 3. The command you can use is:

```
cemt set queue(l86o) trigger(3)
```

This makes the demonstration that follows somewhat less protracted.

At a BTAM or VTAM terminal, enter the transaction AORD. A screen is displayed requesting details of a single order that is to be placed on the L860 queue.

You might reply with the following data:

```
         ORDER ENTRY


    NUMBER   : 111111
    PART NO  : 111111
    QUANTITY: 111111

 PRESS ENTER TO CONTINUE,CLEAR TO QUIT
```

The screen is redisplayed requesting a second order. Type a second screen of data with 2s in place of the 1s above. Press ENTER, then press CLEAR. The AORD transaction is now completed, with two entries having been placed on the L860 queue.

You might now wish to confirm that the two entries are actually present on the queue. There is no direct way of counting the number of elements on an intrapartition queue. Furthermore, as explained in Chapter 2.2, reading the elements from the queue also causes them to be effectively deleted from the queue.

Digression: the CEBR Transaction

Perhaps the most convenient way of demonstrating that the two entries exist on the L860 queue is to use the CEBR transaction. This transaction is normally used to browse (nondestructively) the elements on a temporary storage queue. It is, however, also possible to use this transaction to read (destructively) the elements on an intrapartition queue and to transfer them to a temporary storage queue. You can then place the elements back on the (now empty) intrapartition queue and delete them from the temporary storage queue.

You should remember, however, that this operation does not simply restore the elements into their original positions in the queue. The original elements have been deleted; the rewritten elements occupy new positions and take up new space in the intrapartition data set.

At your terminal, enter the transaction CEBR. The following screen is displayed:

```
 CEBR            TS Queue  CEBRL77A Record     1 of      0    Col     1 of      0
 ENTER COMMAND ===>
      **************************  TOP OF QUEUE  ******************************
      *************************  BOTTOM OF QUEUE  ***************************









TEMPORARY STORAGE QUEUE CEBRL77A           IS EMPTY
PF1 : HELP                PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE
PF4 : VIEW TOP            PF5 : VIEW BOTTOM         PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF    PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL    PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

Now overwrite the name of the temporary storage queue that is to be created, with the name of your choice (for example, AAAA) and type the command:

```
get l86o
```

in the command line. Press ENTER. This transfers all the entries in the L860 queue to the temporary storage queue:

```
 CEBR              TS Queue  AAAA     Record     1 of     2    Col      1 of    22
 ENTER COMMAND ===>
      *************************  TOP OF QUEUE  *****************************
00001 1111111111111111111L77A
00002 2222222222222222222L77A
      ************************  BOTTOM OF QUEUE  ****************************













PF1 : HELP                 PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE
PF4 : VIEW TOP             PF5 : VIEW BOTTOM         PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF     PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL     PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

After you have confirmed that the entries have been placed on the L860 queue, you can write them back with the command:

```
put l86o
```

and delete the entries from the temporary storage queue by entering the command:

```
purge
```

End of Digression

If you now shut down CICS using:

```
cemt perform shutdown
```

a warm keypoint will be recorded and these two entries will be saved in the restart data set. The messages displayed are:

```
2 cemt perform shutdown
F2 002 DFH1701 - C.I.C.S. IS BEING TERMINATED
F2 002 DFH1796 - KEYPOINT SUCCESSFUL
F2 002 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
F2 002 EOJ CICS

DATE 03/12/83,CLOCK 00/22/40,DURATION 00/02/57
```

Start up CICS again with START=AUTO. This time, a warm restart is performed and the following messages appear:

```
F2 002 // JOB CICS WITH WARM RESTART

DATE 03/12/83,CLOCK 00/22/43
F2 002 * User Private Core Image Library
F2 002 * User VSAM Catalog
F2 002 * Auxiliary Temporary Storage
F2 002 * Intrapartition Transient Data
F2 002 * Restart Data Set
F2 002 * FILEA Sample Data Set
F2 002 * Dump Data Sets
F2 002 * Auxiliary Trace
F2 002 * Search Order for Core Image Libraries
F2 002 * ASSGN of DASD containing CICS and User Files
F2 002 1T20I  SYS001 HAS BEEN ASSIGNED TO X'329'
F2 002 * Local Terminals Defined in Sample BTAM TCT
F2 002 * Start Up CICS With Override Parameters From SYSIPT
F2 002 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS.
F2 002 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
F2 002  TSP=2$,TDP=6$,DCP=D$,TRP=D$
F2 002  PPT=S1,PCT=S1,DCT=S1,FCT=S1,FCP=3$,TCT=S1
F2 002  BMS=S$,PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
F2 002  KPP=D$
F2 002  START=AUTO,$END
F2 002 DFH1501 - DFHSIT1$ IS BEING LOADED
F2 002 DFH1500 - LOADING CICS NUCLEUS
F2 002 DFH1500 - CICS START-UP IS WARM
F2 002 DFH1500 - OPENING INTRAPARTITION ACB
F2 002 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
F2 002 DFH1500 - DATA BASE DATASETS ARE BEING OPENED
F2 002 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
F2 002 DFH1500 - DUMP DATA SET IS BEING OPENED
F2 002 DFH1580 - NO WARM START INFORMATION FOR TSP, TSP COLD STARTED
F2 002 DFH1500 - INITIALIZING TEMPORARY STORAGE
F2 002 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS   692K
F2 002 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
F2 002 DFH1500 - STXIT MACROS ARE BEING ISSUED
F2 002 DFH1500 - PROCESSING RESIDENT PROGRAMS
F2 002 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS   626K
F2 002 DFH1500 - CONTROL IS BEING GIVEN TO CICS
```

Figure 36. Sample Output on Starting Up CICS with Warm Restart

During CICS initialization, CICS resources are restored to the same state they had at the time of the previous shutdown. In particular, the FILEA file is still open and the trigger level of the L860 queue is still set at 3.

To demonstrate this, you can run the following dialog:

```
 2 cemt inquire dataset(filea)
 F2 002   Dat(FILEA    ) Vsa Ope Ena Rea Upd Add Bro Del Exc
 F2 002  RESPONSE: NORMAL TIME:     23.46   DATE:  83.071
*F2-002
 2 cemt inquire queue(l86o)
 F2 002   Que(L860) Tri( 00003 )          Int Ena
 F2 002  RESPONSE: NORMAL TIME:     24.04   DATE:  83.071
```

Furthermore, the two entries previously placed on the L860 queue are still there. You can, if you wish, verify this by using the CEBR transaction described above to read them and then rewrite them. If you now invoke the AORD transaction again and type a further order consisting, for example, of 3s in place of the 1s and 2s used before, the arrival of the third element on the queue triggers off the AORQ transaction (see the definition of the L860 queue in "Destination Control Table (DCT)" on page 71.

This transaction processes all three entries and sends them for printing at the local printer.

You can now shut down CICS again, using the command:

```
cemt perform shutdown
```

whereupon a warm keypoint will be taken.

EMERGENCY RESTART

On rare occasions, it may happen that CICS or the VSE system suffers an unexpected failure. Such a failure may possibly occur as a result of a CICS or VSE processing error, or it may occur as a result of a hardware fault or loss of power. Alternatively, the CICS system may be deliberately abnormally terminated by the system operator, either by a request to VSE to cancel CICS or by means of the CICS command:

```
cemt perform shutdown immediate
```

Whatever the reason for the abnormal termination, CICS activity in all these situations is brought to an abrupt stop. There is no opportunity to take a warm keypoint, or even to complete any application program processing that may have been in progress.

Because no warm keypoint can be recorded after an abnormal termination, further measures must be taken to preserve data that might otherwise be lost, and to re-create it and make it available to the subsequent CICS run. These measures involve recording each change to the data at the time that the change is actually being made. The recording is performed automatically by CICS in a further CICS data set: the CICS **system log**. If CICS is terminated abnormally, the system log is scanned backward during the following startup and the correct data is extracted from it before allowing the CICS run to proceed.

The process of re-creating data from the system log after an abnormal termination is known as **emergency restart**. At startup time, CICS inspects the control record in the restart data set and determines that no warm keypoint exists. If START=AUTO is used as an initialization parameter, and if the system log is defined and available, CICS performs an emergency restart automatically. If no system log is defined, CICS performs a cold start.

The choice of which changes to data are to be recorded on the system log is made rather differently than in the case of a warm keypoint. In the case of a warm keypoint, CICS records the total environment at the time of the normal shutdown, and the user can select which (if not all) facilities are to be restored at the subsequent warm restart. In the case of system logging and emergency restart, the user has to specify explicitly which items of data are to be recorded; they are then restored automatically at the subsequent emergency restart. The data that is logged and restored in this way is known as **recoverable**.

REQUIREMENTS FOR EMERGENCY RESTART TO BE PERFORMED

Apart from the restart data set and the keypoint program, a number of other components must also be present in the CICS system to enable system logging and subsequent emergency restart to be performed.

The system log has already been mentioned. The system log is considered to be a particular case of a CICS resource (it is a special case of a journal) and, as such, it must be defined to CICS in a **journal control table**.

The CICS management module that performs the actual reading and writing to the system log is the **journal control program** (DFHJCP). This module must also be included in a CICS system to allow emergency restart to be performed.

It was mentioned earlier that, whenever a recoverable resource (intrapartition queue, temporary storage queue, or user file) is modified by an application program, the corresponding change is also recorded on the system log. This enables the data to be re-created in case an emergency restart has to be performed. In the case of transient data and temporary storage, CICS must also record on the system log the base state against which these changes are to be applied.

Such a recording of the state of these resources at a particular moment is known as a **system activity keypoint**. At least one activity keypoint must exist in the system log for emergency restart to be possible. In a practical situation, activity keypoints would be recorded with a certain frequency to reduce the amount of time spent in scanning the system log backward during emergency restart. Because, however, the default action is for no activity keypoints to be taken, you must explicitly set the frequency of activity keypoints (**AKPFREQ** system initialization parameter) to a nonzero value.

A further CICS management module must be specified explicitly if emergency restart is to be possible. This is the **transaction backout program** (DFHTBP). The other management modules that take part in emergency restart are loaded and invoked automatically without having to be specified explicitly. DFHTBP, however, can be suffixed — two versions are supplied in the pregenerated system — and the version to be used must be specified at startup time by means of a system initialization parameter.

Finally, the actual resources that are to be made recoverable must be defined as such to the CICS system. The way this is done depends on the type of resource:

- A recoverable queue must be defined in the destination control table with the appropriate DESTRCV (destination recovery) attribute.
- A recoverable (auxiliary) temporary storage queue name must be defined in the temporary storage table (DFHTST).
- A recoverable user file must be defined in the file control table with the LOG=YES attribute.

In addition, the corresponding management modules that manage these resources (DFHTDP, DFHTSP, and DFHFCP) must, where required, be chosen in the versions that can record changes to these resources onto the system log.

CREATION OF THE SYSTEM LOG

The system log is a sequential file that may be created on disk (or on tape — see Chapter 4.5). Two data sets may be defined, so that one may be receiving output while the other is being copied, for example, to archive tape, if required.

Unlike the dump and auxiliary trace data sets, the system log must be preformatted before the first time it is used. After this has been done, the system log is used repeatedly, the old data being overwritten each time a data set switch is made.

The formatting of the system log is performed by a CICS-supplied utility program (DFHJCJFP for disk, DFHFTAP for tape), which must be run as a batch job, separate from the CICS job.

The following sample jobs format two system log data sets on disk volume M3330B, each of the data sets occupying 57 tracks. In a running CICS system, the size of each data set would have to be very much larger, for example, 200 cylinders of 3330 or 3350 direct-access storage for the primary data set and 50 cylinders for the secondary data set (see "Chapter 4.5. Journal Data Sets").

```
// JOB CICSFDSK FORMAT PRIMARY JOURNAL DISK EXTENT
// DLBL JOURNAL,'CICS160.SYSTEM.LOG.A',0,SD
// EXTENT SYS011,M3330B,1,0,19,57
// ASSGN SYS011,333
// EXEC DFHJCJFP,SIZE=AUTO
/*
/&

// JOB CICSFDSK FORMAT SECONDARY JOURNAL DISK EXTENT
// DLBL JOURNAL,'CICS160.SYSTEM.LOG.B',0,SD
// EXTENT SYS011,M3330B,1,0,76,57
// ASSGN SYS011,333
// EXEC DFHJCJFP,SIZE=AUTO
/*
/&
                    (These jobs are included in Z.DFHSPJCL)
```

Figure 37. Formatting System Log Data Sets on Disk

Note: You will also need to add:

```
// LIBDEF CL,SEARCH=C160PCL
```

before each EXEC statement in the above job stream.

The formatted log data sets are then defined in the CICS startup job in the usual way for sequential files:

```
// DLBL DFHJ01A,'CICS160.SYSTEM.LOG.A',0,SD
// EXTENT SYS002,M3330B,1,0,19,57
// DLBL DFHJ01B,'CICS160.SYSTEM.LOG.B',0,SD
// EXTENT SYS002,M3330B,1,0,76,57
```

where the file names DFHJ01A and DFHJ01B are required names.

JOURNAL CONTROL TABLE (JCT)

The journal control table defines the system log as a CICS resource to the system. It is also a means of defining any user journals you may wish to employ to perform your own explicit journaling of data.

The pregenerated system does not contain any pregenerated version of the journal control table. It does contain a copy book (DFHXJCT) that defines a system log consisting of two data sets on disk:

```
TITLE 'DFHXJCT - COPYBOOK FOR SYSTEM LOG'
DFHJCT TYPE=ENTRY,JFILEID=SYSTEM,JTYPE=DISK2,BUFSIZE=4096,     *
       DEVADDR=(SYS019,SYS019),BUFSUV=2048,                    *
       JOUROPT=INPUT
```

This copy book is, in turn, referenced as an included file in the DFHJCT1$ copy book:

```
TITLE 'DFHJCT1$ SAMPLE JCT'
DFHJCT TYPE=INITIAL,                                          X
       STARTER=YES,          ALLOWS $ IN SUFFIX               X
       SUFFIX=1$
       COPY  DFHXJCT       - ENTRY FOR SYSTEM LOG
DFHJCT TYPE=FINAL
```

To assemble and link-edit a journal control table defining a system log with the above properties, you can run the following job:

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHJCT1$
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=U160PCL
// LIBDEF RL,SEARCH=C160PRL
// LIBDEF SL,SEARCH=C160PSL
// EXEC ASSEMBLY
       COPY    DFHJCT1$
   END
/*
// EXEC LNKEDT
/&
```

Figure 38. Assembling and Link-Editing a Sample JCT

DESTINATION CONTROL TABLE WITH L860 QUEUE DEFINED AS PHYSICALLY RECOVERABLE

In the demonstration that follows, the L860 queue used previously will be recovered after a system failure. To define the L860 queue as physically recoverable, you must add the DESTRCV=PH attribute to the queue definition.

You can then assemble and link-edit a new destination control table (DFHDCTS2) by running the following job:

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHDCTS2
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=U160PCL
// LIBDEF RL,SEARCH=C160PRL
// LIBDEF SL,SEARCH=C160PSL
// EXEC ASSEMBLY
        DFHDCT  TYPE=INITIAL,                                          *
                SUFFIX=S2
        COPY    DFHXDCT1
        COPY    DFHXDCT
LOGA      DFHDCT TYPE=INDIRECT,      DESTINATION LOGA USED             X
                DESTID=LOGA,         BY CICS SAMPLE PGMS               X
                INDDEST=CPLI
L860      DFHDCT TYPE=INTRA,         DESTINATION L860 USED BY ORDER    X
                DESTID=L860,         ENTRY QUEUE PRINT SAMPLE PROGRAMS X
                DESTFAC=TERMINAL,    TERMINAL                          X
                TRANSID=AORQ,        <CHANGE IF WANT TO RUN 'PORQ','OREQ'X
                TRIGLEV=30,          AORQ IS AUTO INIT'ED WHEN QUEUE=30 X
                DESTRCV=PH           PHYSICAL RECOVERY OF QUEUE
        DFHDCT  TYPE=FINAL
   END
/*
// EXEC LNKEDT
/&
```

DEMONSTRATION OF EMERGENCY RESTART

You can now start up the following job to demonstrate the working of emergency restart. The operations performed will be similar to those used when warm restart was demonstrated.

```
// JOB CICS WITH EMERGENCY RESTART CAPABILITY
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
* User VSAM Catalog
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
* Auxiliary Temporary Storage
// DLBL DFHTEMP,'CICS160.TEMP.STORAGE',,VSAM,CAT=CICSUCT
* Intrapartition Transient Data
// DLBL DFHNTRA,'CICS160.INTRA.TRANSD',,VSAM,CAT=CICSUCT
* Restart Data Set
// DLBL DFHRSD,'CICS160.RSD',,VSAM,CAT=CICSUCT
* FILEA Sample Data Set
// DLBL FILEA,'SAMPLE.TEST.FILEA',,VSAM,CAT=CICSUCT
* Dump Data Sets
// DLBL DFHDMPA,'CICS160.DUMP.DATASET.A',0,SD
// EXTENT SYS001,M3350A,1,0,210,40
// DLBL DFHDMPB,'CICS160.DUMP.DATASET.B',0,SD
// EXTENT SYS001,M3350A,1,0,250,10
* Auxiliary Trace
// DLBL DFHAUXT,'CICS160.AUX.TRACE.A',0,SD
// EXTENT SYS001,M3350A,1,0,260,40
// DLBL DFHBUXT,'CICS160.AUX.TRACE.B',0,SD
// EXTENT SYS001,M3350A,1,0,300,40
* CICS System Log
// DLBL DFHJ01A,'CICS160.SYSTEM.LOG.A',0,SD
// EXTENT SYS002,M3330B,1,0,19,57
// DLBL DFHJ01B,'CICS160.SYSTEM.LOG.B',0,SD
// EXTENT SYS002,M3330B,1,0,76,57
* Search Order for Core Image Libraries
// LIBDEF CL,SEARCH=(U160PCL,C160PCL)
* ASSGN of DASD containing CICS and User Files
// ASSGN SYS001,DISK,TEMP,VOL=M3350A,SHR
// ASSGN SYS002,DISK,TEMP,VOL=M3330B,SHR
* Local Terminals Defined in Sample BTAM TCT
ASSGN SYS020,0A1
ASSGN SYS021,0A2
ASSGN SYS027,0A4
* Start Up CICS With Override Parameters From SYSIPT
// EXEC DFHSIP,SIZE=1200K,PARM='SIT=1$,SI'
 TSP=2$,TDP=7$,DCP=D$,TRP=D$
 PPT=S1,PCT=S1,DCT=S2,FCT=S1,FCP=3$,TCT=S1
 BMS=S$,PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 KPP=D$,JCT=1$,JCP=1$,TBP=1$,AKPFREQ=200
 START=AUTO,$END
/*
/&
```

Figure 39. Starting Up CICS with Emergency Restart Capability

The differences between this job and the one used for warm restart are:

- The system log is defined.
- JCP=1$ enables changes to be recorded onto the system log.
- JCT=1$ defines the system log on disk as a CICS resource.
- TBP=1$ causes the transaction backout program to be included.
- AKPFREQ=200 causes the system to record activity keypoints.
- DCT=S2 defines the L860 queue to be physically recoverable.
- TDP=7$ causes the inclusion of a transient data program that is able to record changes to intrapartition transient data on the system log.

Because CICS was previously shut down with a warm keypoint, the above startup will again result in a warm restart. It is, therefore, unnecessary to open FILEA again or to change the trigger level of the L860 queue.

Now, using the AORD transaction, again place two entries on the queue and then end the transaction. There are many ways in which you could terminate CICS abnormally at this point; for example

```
cemt perform shutdown immediate
```

You might like, instead, to cancel CICS as a VSE job:

```
cancel f2,nodump
F2 002
AR 015 0P69I INTERV REQ  BG 00C
F2 016 0S12I SUB DFHLOADR CANCELED DUE TO MAINTASK TERMINATION
F2 017 0S12I SUB DFHJCSUB CANCELED DUE TO MAINTASK TERMINATION
F2 002 0S01I JOB CICS     CANCELED DUE TO OPERATOR INTERVENTION
F2 002 DFH0606 - ABEND X'24' HAS BEEN DETECTED
F2 002 DFH4510 - ALL OPEN JOURNALS NOW CLOSED
F2 002 DFH0611 - ABNORMAL TERMINATION COMPLETE
F2 016 0S12I SUB DFHLOADR CANCELED DUE TO MAINTASK TERMINATION
F2 002 EOJ CICS

DATE 03/12/83,CLOCK 00/30/58,DURATION 00/00/59
```

Before you start up CICS again, you may have to run a job to verify that your VSAM files were correctly closed before CICS attempts to open them again. If a system failure occurs while a VSAM file is open, the file's catalog entry may not have been updated with the correct end-of-file information. The access method services VERIFY command updates the file's catalog entry to show the file's real end-of-file indicators. (This is not necessary if you have VSE/VSAM Version 1 Release 3 installed.)

The job you should run is shown below:

```
// JOB VERIFY
// DLBL IJSYSUC,'CICS160.UCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
        VERIFY DATASET(CICS160.TEMP.STORAGE)
        VERIFY DATASET(CICS160.INTRA.TRANSD)
        VERIFY DATASET(SAMPLE.TEST.FILEA)
/*
/&
```

Figure 40. Verifying VSAM Files

Start up CICS again using the CICS startup job used previously. This time, an emergency restart is performed by the system. A major part of the initialization process is concerned with the execution of the recovery utility program and the transaction backout program to recover the data that would otherwise have been lost.

Finally, after the recovery has been completed, the system operator is given the opportunity of either proceeding or not proceeding with the CICS run. This is in case the process of recovery by the CICS system was not sufficient, by itself, to restore all the data to a consistent state. The prompt displayed at this time enables the user to perform further work, before exposing the data to a running CICS system.

Conditions that require the operator not to proceed will be very rare indeed. However, if the operator does not proceed (that is, replies CANCEL to the prompt), the system records a warm keypoint before terminating CICS.

The messages displayed during emergency restart are:

```
F2 002 // JOB CICS WITH EMERGENCY RESTART CAPABILITY

DATE 03/12/83,CLOCK 00/31/02
F2 002 * User Private Core Image Library
F2 002 * User VSAM Catalog
F2 002 * Auxiliary Temporary Storage
F2 002 * Intrapartition Transient Data
F2 002 * Restart Data Set
F2 002 * FILEA Sample Data Set
F2 002 * Dump Data Sets
F2 002 * Auxiliary Trace
F2 002 * CICS System Log
F2 002 * Search Order for Core Image Libraries
F2 002 * ASSGN of DASD containing CICS and User Files
F2 002 1T20I  SYS001 HAS BEEN ASSIGNED TO X'329'
F2 002 1T20I  SYS002 HAS BEEN ASSIGNED TO X'333'
F2 002 * Local Terminals Defined in Sample BTAM TCT
F2 002 * Start Up CICS With Override Parameters From SYSIPT
F2 002 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS.
F2 002 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
F2 002  TSP=2$,TDP=7$,DCP=D$,TRP=D$
F2 002  PPT=S1,PCT=S1,DCT=S2,FCT=S1,FCP=3$,TCT=S1
F2 002  BMS=S$,PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
F2 002  KPP=D$,JCT=1$,JCP=1$,TBP=1$,AKPFREQ=200
F2 002  START=AUTO,$END
F2 002 DFH1501 - DFHSIT1$ IS BEING LOADED
F2 002 DFH1500 - LOADING CICS NUCLEUS
F2 002 DFH1500 - CICS START-UP IS EMERGENCY
F2 002 DFH1500 - OPENING INTRAPARTITION ACB
F2 002 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
F2 002 DFH1500 - DATA BASE DATASETS ARE BEING OPENED
F2 002 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
F2 002 DFH1500 - DUMP DATA SET IS BEING OPENED
F2 002 DFH1500 - JOURNAL CONTROL SUBTASK IS BEING ATTACHED/ENTERED
F2 002 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS   682K
F2 002 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
F2 002 DFH1500 - STXIT MACROS ARE BEING ISSUED
F2 002 DFH1500 - PROCESSING RESIDENT PROGRAMS
F2 002 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS   616K
F2 002 DFH2800I - DFHRUP IN PROGRESS
F2 002 DFH2800I - DFHRUP COMPLETED
F2 002 DFH1500 - OPENING JOURNAL FILES
F2 017 DFH4508 - CICS SYSTEM LOG. PRIMARY   EXTENT NOW RECEIVING OUTPUT ON 333
F2 002 DFH4500 - 01 OF 01 JOURNALS SUCCESSFULLY OPENED
F2 002 DFH5701 - TRANSACTION BACKOUT PROGRAM IN CONTROL
F2 002 DFH5703 - NO FBO-TABLE DATA PRESENT, NO FILE BACKOUT PERFORMED
F2 002 DFH5703 - NO MBO-TABLE DATA PRESENT, NO MSG. BACKOUT PERFORMED
F2 002 DFH5790 - DFHTBP PROCESSING COMPLETE, RETURNING CONTROL TO DFHSIP.
F2 002 DFH1588 - IS START-UP TO BE CONTINUED ?
*F2 002 DFH1505 - REPLY GO OR CANCEL
2 go
F2 002 DFH1500 - CONTROL IS BEING GIVEN TO CICS
```

Figure 41. Sample Output on Starting Up CICS with Emergency Restart

Messages of the type:

DFH5703 - NO FBO-TABLE DATA PRESENT, NO FILE BACKOUT PERFORMED

do not indicate any error. Rather, they advise that backout of the named resource was not required. In this case, for example, because no changes to recoverable files had occurred in the previous run, no file backout is performed.

The entries on the L860 queue have now been recovered (you can demonstrate this by using the CEBR transaction).

Where emergency restart differs from warm restart, however, is that any changes effected by means of the CEMT transaction have not been restored. In particular, the FILEA file is now closed and the trigger level of the L860 queue has been reset to 30:

```
 2 cemt inquire dataset(filea)
 F2 002   Dat(FILEA    ) Vsa Clo Ena Rea Upd Add Bro Del Exc
*F2 002  RESPONSE: NORMAL TIME:     32.02   DATE:  83.071
 2 cemt inquire queue(l86o)
 F2-002
 F2 002   Que(L860) Tri( 00030 )          Int Ena
*F2 002  RESPONSE: NORMAL TIME:     32.18   DATE:  83.071
```

You could, at this stage, make doubly sure that the queue entries have been recovered by opening FILEA again, changing the trigger level back to 3, placing another entry on the queue, and witnessing all three elements being sent to your local printer.

DATA INTEGRITY AND BACKOUT OF DATA

In the first part of this chapter, an analogy has deliberately been drawn between warm restart and emergency restart. The transient data queue was physically recovered in both cases: in the first case, from the warm keypoint in the restart data set, in the second case, from the activity keypoints and the recordings of changes in the system log. The CICS abnormal termination was engineered to occur after the transaction that modified the L860 queue had been completed.

In practice, of course, it is unlikely that an abnormal termination of CICS will occur at a time when no transactions are running in the system. If an abnormal termination does occur, it is more likely to happen at a time when transactions are running, so that the failure will stop the transactions reaching their normal completion. A transaction that cannot complete in this way is known as an **inflight** transaction. In the case of inflight transactions, one has to consider not only physical recovery of data but also data consistency (or data integrity).

It is frequently the case that a transaction whose function is to perform a set of changes to data in the system must carry out all the changes together, or none at all. To carry out only a portion of the changes would be to leave the data in an inconsistent and unusable state.

For example, a transaction might debit an account by a certain amount and also queue a request that the same amount be paid out in cash. The two operations are separate; but they must be treated as one if data consistency is to be preserved.

A set of activities that must be completed as a unit in order to preserve consistency of data is known as a **logical unit of work**. CICS might terminate abnormally while a logical unit of work is being performed. If the logical unit of work involves changes to recoverable resources, then during the subsequent emergency restart, not only will the resources be recovered, but the changes will be backed out to the consistent state that obtained at the start of the logical unit of work.

The most common instance of a logical unit of work is the execution of a single transaction (a task). A logical unit of work is terminated by a **sync point**; and a transaction ends with an implicit sync point. An application program can, however, have within it any number of explicitly coded sync points, each sync point signaling the completion of a set of related activities. After the sync point has been executed, the set of changes carried out in the logical unit of work is said to be **committed** and will not be backed out.

At this point, it is necessary to distinguish between **physically** and **logically** recoverable resources. Only logically recoverable resources are backed out to a consistent state as described above. Physically recoverable resources are restored to the state they were in when the abnormal termination occurred, and no backout is carried out.

The L860 queue defined in the previous section as recoverable was defined as physically recoverable. This means, for example, that if a single element is placed on the queue and then CICS abnormally terminates before the AORD transaction is completed, that single element will be recovered on the following emergency restart.

If, on the other hand, the queue had been defined as logically recoverable (by coding the DESTRCV=LG parameter in the destination control table), then in the identical situation, the single element would be recovered and then backed out to return to its state at the beginning of the transaction. (In the example below, the destination control table that defines a logically recoverable queue L860 has been given the suffix S3.)

The same distinction between physically and logically recoverable resources cannot be made in the case of temporary storage queues. A temporary storage queue cannot be defined as physically recoverable. By including a temporary storage queue name (or abbreviated name) in the temporary storage table, the queue is defined to be logically recoverable and will always be restored and backed out to a consistent state.

Again, in the case of user files, either logical recovery can be defined (by means of LOG=YES in the file control table), or no recovery at all. In this case, however, "no recovery" implies physical recovery, because the files are held on permanent storage and can be accessed in the normal way after CICS restart. The only difficulty may occur when writing to a nonrecoverable VSAM file. It is possible that VSAM itself buffers up the requests and will not have completed writing to the file when the abnormal termination occurs. In such a case, data may actually be lost.

DYNAMIC TRANSACTION BACKOUT

The discussion in the previous section about data consistency was set in the context of an abnormal termination of the whole CICS system. It was seen that an abnormal termination can affect a number of inflight transactions and that data consistency has to be restored by backing out changes that had already been carried out but not yet committed.

It is also possible — and more common — for a single transaction to terminate abnormally (**abend**), while the CICS system is unaffected by the failure. The same considerations relating to data consistency apply to this case as to the more severe case of system failure. The main difference is that, because CICS is still in control after a transaction abend, it can restore consistency of data dynamically while CICS is still running and while other transactions, which do not access that data, can continue to run normally.

Theoretically, it would be possible to restore consistency of data after a transaction abend from the information stored on the system log. In practice, it is far more efficient for CICS to record the changes at the time they are made in main storage (as well as on the system log) and to use these records in main storage to back out changes in case of a transaction abend. The process of backing out these changes is known as **dynamic transaction backout.**

The concepts of logical units of work, sync points, and committed changes to resources apply equally when considering dynamic transaction backout as they do to emergency restart. During the execution of any task (that is, a particular instance of a transaction), changes to recoverable resources are recorded in the **dynamic buffer** associated with that task. If these records overflow the dynamic buffer, the contents of the dynamic buffer are written to a temporary storage queue and the dynamic buffer is cleared once more.

The combination of the dynamic buffer and any associated temporary storage records is known as the **dynamic log**.

Whether data changes are actually backed out after a transaction abend or not depends on two things:

1. Data is backed out only if it has been defined as recoverable (by means of the same definitions as those used for emergency restart recovery).

2. The transaction in which the data changes are performed must itself be defined as recoverable, that is, requiring dynamic transaction backout.

The dynamic log associated with a task is cleared whenever the corresponding transaction is completed normally or when a sync point is executed. If an inflight transaction terminates abnormally without reaching a sync point, the records in the associated dynamic log are used by the **dynamic backout program** to restore the recoverable data to the state it was in at the start of the failing transaction (or the start of the logical unit of work).

Transactions are, by default, defined as recoverable. This means that, if the dynamic backout program is present in the CICS system, recoverable data will be backed out after a transaction abend. If you wish a particular transaction to be nonrecoverable, even though it makes changes to recoverable data, you must code:

```
DTB=NO
```

in the program control table entry for that transaction.

REQUIREMENTS FOR DYNAMIC TRANSACTION BACKOUT TO BE PERFORMED

The requirements to enable data to be backed out after a transaction abend can be summarized as follows:

- DBP=1$ must be specified as a system initialization parameter. (DBP=2$ is used if DL/I backout is also required.)
- Transient data queues, temporary storage queues, or user files must be defined as recoverable using the same definitions as those used for emergency restart.
- The size of the dynamic buffer may, if required, be explicitly specified by means of the DBUFSZ system initialization parameter. (The default size is 500 bytes.)

In summary, the following job might be used to enable both emergency restart and dynamic transaction backout to be performed:

```
// JOB CICS WITH DYNAMIC TRANSACTION BACKOUT
* User Private Core Image Library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,M3330A
* User VSAM Catalog
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
* Auxiliary Temporary Storage
// DLBL DFHTEMP,'CICS160.TEMP.STORAGE',,VSAM,CAT=CICSUCT
* Intrapartition Transient Data
// DLBL DFHNTRA,'CICS160.INTRA.TRANSD',,VSAM,CAT=CICSUCT
* Restart Data Set
// DLBL DFHRSD,'CICS160.RSD',,VSAM,CAT=CICSUCT
* FILEA Sample Data Set
// DLBL FILEA,'SAMPLE.TEST.FILEA',,VSAM,CAT=CICSUCT
* Dump Datasets
// DLBL DFHDMPA,'CICS160.DUMP.DATASET.A',0,SD
// EXTENT SYS001,M3350A,1,0,210,40
// DLBL DFHDMPB,'CICS160.DUMP.DATASET.B',0,SD
// EXTENT SYS001,M3350A,1,0,250,10
* Auxiliary Trace
// DLBL DFHAUXT,'CICS160.AUX.TRACE.A',0,SD
// EXTENT SYS001,M3350A,1,0,260,40
// DLBL DFHBUXT,'CICS160.AUX.TRACE.B',0,SD
// EXTENT SYS001,M3350A,1,0,300,40
* CICS System Log
// DLBL DFHJ01A,'CICS160.SYSTEM.LOG.A',0,SD
// EXTENT SYS002,M3330B,1,0,19,57
// DLBL DFHJ01B,'CICS160.SYSTEM.LOG.B',0,SD
// EXTENT SYS002,M3330B,1,0,76,57
* Search Order for Core Image Libraries
// LIBDEF CL,SEARCH=(U160PCL,C160PCL)
* ASSGN of DASD containing CICS and User Files
// ASSGN SYS001,DISK,TEMP,VOL=M3350A,SHR
// ASSGN SYS002,DISK,TEMP,VOL=M3330B,SHR
* Local Terminals Defined in Sample BTAM TCT
ASSGN SYS020,0A1
ASSGN SYS021,0A2
ASSGN SYS027,0A4
* Start Up CICS With Override Parameters From SYSIPT
// EXEC DFHSIP,SIZE=1200K,PARM='SIT=1$,SI'
 TSP=2$,TDP=7$,DCP=D$,TRP=D$
 PPT=S1,PCT=S1,DCT=S3,FCT=S1,FCP=3$,TCT=S1
 BMS=S$,PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 KPP=D$,JCT=1$,JCP=1$,TBP=1$,AKPFREQ=200
 DBP=1$
 START=AUTO,$END
/*
/&
```

Figure 42. Starting Up CICS with Dynamic Transaction Backout Capability

For further information on recovery and restart, see the CICS/VS Recovery and Restart Guide.

Chapter 2.5. CICS System Generation

CICS consists of a set of management, service, and utility programs that have to be prepared by a process similar to VSE system generation. If you use the CICS pregenerated system, you do not need to carry out system generation. The pregenerated system provides at least one pregenerated version of every CICS program, and contains sample tables, application programs, and BMS map sets.

The CICS management programs provided with the pregenerated system are described in Appendix C. CICS provides only one unsuffixable version of certain modules. Where two or more suffixed versions of a module are supplied, Appendix C describes each version so that you can select the appropriate version for your own CICS system.

If you require support that does not exist in the pregenerated modules, you may need to generate your own version of certain CICS modules to provide that support.

WHAT IS SYSTEM GENERATION?

System generation is a process that enables you to generate your own version of a CICS module to provide the support you require. You must code a series of CICS system generation (DFHSG) macro instructions, using the detailed information given in the CICS/VS Customization Guide. When you have coded the appropriate macro instructions, you have to carry out two stages of processing:

1. Assemble the macro instructions to generate the job control and source statements (job stream) required to assemble, link-edit, and catalog the programs. This is called **stage 1** of system generation.

2. Execute the job stream. This is called **stage 2** of system generation.

The rest of this chapter explains how to process the macro instructions after they have been coded.

WHAT CAN YOU GENERATE?

Only the CICS modules listed below can be generated. See the CICS/VS Customization Guide for details of how to code the required macro instructions.

	DFHSG PROGRAM=	
ACEE	CSS	Security ID program
ACP	CSO	Abnormal condition program
ALP	KCP	Allocation program
CPY	TCP	VTAM 3270 print function support
CSA	CSA	Common systems area
DBP	DBP	Dynamic transaction backout
DSCTS	INITIAL	CICS dummy sections
EXI	TCP	VTAM 3270 print function support
FCP	FCP	File control program
GMM	TCP	VTAM Good Morning message program
ICP	ICP	Interval control program
ISP	ISC	Intercommunication program
KCP	KCP	Task control program
PCP	PCP	Program control program
PRK	TCP	VTAM 3270 print function support
P3270	TCP	3270 print function support
RKB	TCP	VTAM 3270 print function support
RTY	DBP	CICS-supplied transaction restart module

	DFHSG PROGRAM=	
SCP	SCP	Storage control program
SCR	SCP	Storage control recovery program
SPP	KCP	Sync point program
SRP	SRP	System recovery program
TACP	CSO	Terminal abnormal condition program
TBP	TBP	Transaction backout program
TCP	TCP	Terminal control program
TDP	TDP	Transient data program
TPR	BMS	BMS terminal page retrieval program
TRP	TRP	Trace program
TSP	TSP	Temporary storage program
XFP	ISC	Function shipping transformer program
XSP	CSS	Security program
ZCA	TCP	VTAM terminal control program module
ZCB	TCP	VTAM terminal control program module
ZCC	TCP	VTAM terminal control program module
ZCP	TCP	Common terminal control program module
ZCW	TCP	VTAM terminal control program module
ZCX	TCP	Common terminal control program module
ZCY	TCP	VTAM terminal control program module
ZCZ	TCP	VTAM terminal control program module
ZNAC	TCP	Node abnormal condition program
ZNEP	TCP	Node error program interface program
ZRLG	TCP	Response logging program
ZRSP	TCP	Resend program

ASSEMBLING THE DFHSG MACRO INSTRUCTIONS (STAGE 1)

The DFHSG macro instructions must be used in stage 1 of the system generation process to produce job streams for execution in stage 2.

For stage 1, the CICS private source statement library is needed. The output of stage 1 on SYSPCH can be on cards (approximately 2000 for a complete system generation), tape, or disk. This output is used as input (SYSIN) to perform stage 2.

Figure 43 shows the job stream needed to execute those instructions, using **cards** as intermediate storage between stage 1 and stage 2.

```
 Assumes SYSPCH assigned to card punch

// JOB CICSGEN
*  CICS/DOS/VS SYSTEM GENERATION - STAGE 1
// OPTION  DECK,ALIGN
// LIBDEF SL,SEARCH=(C160PSL,etc),PERM
// EXEC    ASSEMBLY
        DFHSG TYPE=INITIAL,
        .   (system generation macro instructions prepared
        .   according to instructions provided in the
        .   CICS/VS Customization Guide)
        DFHSG TYPE=FINAL
        END
/*
/&

Remove the output (from stage 1) from the punch stacker,
place in the card reader (stage 2 input), and make the
reader ready.  The stage 2 assembly and link-edit will
be performed. This completes the process for generating
CICS/DOS/VS.
```

Figure 43. Using Cards as Intermediate Storage between Stages 1 and 2

Figure 44 shows the job stream needed to execute the system generation macro instructions, using **tape** output for stage 1. The CICS source statement library is required for stage 1. A permanent assign of your other program product libraries is required before the stage 2 job is executed.

```
// JOB CICSGEN
*  CICS/DOS/VS SYSTEM GENERATION - STAGE 1
// LIBDEF SL,SEARCH=(C160PSL,etc),PERM
ASSGN SYSPCH,cuu
// MTC REW,SYSPCH
// OPTION DECK,ALIGN
// EXEC    ASSEMBLY
          DFHSG TYPE=INITIAL
          .
          .   (system generation macro instructions)
          .
         DFHSG TYPE=FINAL
         END
/*
CLOSE SYSPCH,PUNCH
// MTC REW,cuu
*  CICS/DOS/VS SYSTEM GENERATION - STAGE 2
ASSGN SYSIN,cuu
/&

After the completion of stage 2, in reply to operator
console output:

   BG 1C00A ATTN cuu

issue:

   ASSGN  SYSIN,cuu                (card reader)

This completes the process for generating CICS/DOS/VS.
```

Figure 44. Using Tape Output (Optional) for Stages 1 and 2

Figure 45 shows the job stream required to create the system generation macro instructions using the POWER internal reader to store the output from stage 1. A permanent assign of your other program product libraries is required before the stage 2 job is executed.

```
* $$ JOB JNM=CICSGEN,CLASS=A,DISP=D
* $$ PUN DISP=I,CLASS=A
// JOB CICSGEN
// LIBDEF SL,SEARCH=(C160PSL,etc),PERM
// OPTION DECK,ALIGN
// EXEC    ASSEMBLY
          DFHSG TYPE=INITIAL
         .
         .   (system generation macro instructions)
         .
         DFHSG TYPE=FINAL
         END
/*
/&
* $$ EOJ
```

Figure 45. Using POWER Internal Reader to Store the Output From Stage 1

Assembly Errors

If you use the XREF option, the stage 1 assembly cross-reference listing will point to each program generated. Check the listing carefully.

If you try to assemble a module that cannot be generated, an MNOTE will be produced.

Messages that indicate any other assembly or link-editing errors should also be investigated.

EXECUTING THE JOB STREAM (STAGE 2)

The job streams produced from stage 1 are MSHP jobs.

Each MSHP job that is executed will write the following information to the system history file:

- The date the module was generated
- A list of macros used during the assembly of that module.

If any of these macros are serviced, a message will be produced on the console saying that that module is now invalid. A flag will also be set in the system history file showing that the module is backlevel and it will remain on until the module is reassembled.

Figure 46 shows the format of the stage 1 output.

```
// JOB XXXXyyysf     (see note 1)
// OPTION CATAL,NODECK,ALIGN
// EXEC MSHP,SIZE=550K   (see note 2)
TAILOR 5746-XX-300 PHASE=DFHyyysf
RESOLVES 'USER ASSEMBLY'
EXECUTE ASSEMBLY  XREF
   .
   assembler source statements
   .
/$
EXECUTE LNKEDT
 PHASE DFHyyysf,*
/$
/*
/&
// JOB ...
   .
   .
   .
```

Figure 46. Format of Stage 1 Output

<u>**Notes:**</u>

1. In the JOB statement, "XXXX" is the job name specified in the JOBNAME operand in DFHSG TYPE=INITIAL, "yyy" is the name of the module, and "sf" the user suffix, if applicable. The jobname "XXXXyyysf" is truncated to eight characters if necessary.

2. Ensure that there is enough space in your partition to run MSHP.

For stage 2 jobs, a permanent assignment of private libraries is required before SYSIN is assigned to the intermediate storage. Stage 2 also accesses the VSE supervisor and I/O macro instructions, and related logic modules in the relocatable library.

Each MSHP job uses the CICS service information in your system history file. Part of that information is the names of your libraries that contain CICS. The MSHP jobs specify that the new CICS phases are to be cataloged into the CICS core image library. You must therefore ensure that a LIBDEF CL,TO= statement is coded before running the MSHP jobs.

Individual CICS/DOS/VS modules can be replaced, added, or included more than once (using different suffixes) by means of the system generation procedure.

Assembly Errors

If no high-level language options have been specified, DFHPCPLI and DFHPCCOB will be listed as unresolved references when DFHPCP is link-edited. If only one of PL/I or COBOL has been specified, the reference to the other language support routine will be unresolved. This type of message can be ignored.

Messages that indicate any other assembly or link-editing errors should be investigated.

REGENERATING A PREGENERATED MODULE

If you wish to reassemble a pregenerated module, you can use the job stream shown in Figure 47.

```
// JOB SYSGEN
// LIBDEF SL,SEARCH=C160PSL
// EXEC ASSEMBLY
         DFHSG TYPE=INITIAL,                                          X
               OPSYS='DOS/VSE',                                       X
               ASMBLR=ASSEMBLY,                                       X
               PRINT=(LIST,XREF,NODSECT,DSLIST),                      X
               STAGE2=FORCE,                                          X
               EJECT=YES,                                             X
               JOBNAME=DFH,                                           X
               DLI=YES,                      DL/I SUPPORT             X
               STARTER=YES,                                           X
               MOD=(DBP,2$,ISP,,TBP,2$,XFP,),                         X
               ACCTID='CICS_160'
         DFHSG PROGRAM=PREGEN
         DFHSG TYPE=FINAL
         END
/*
/&
```

Figure 47. Reassembling a Pregenerated Module

Note: In this example, stage 2 jobs will be produced for DFHDBP2$, DFHISP, DFHTBP2$, and DFHXFP.

Chapter 2.6. CICS Modules in the VSE Shared Virtual Area

Many CICS modules (both nucleus and PPT) are read-only, and you can place some or all of these modules in the shared virtual area (SVA) of VSE. Note, however, that if multiregion operation (MRO) is being used, certain modules **must** be placed in the SVA, and these are indicated later.

See Appendix B for a list of the SVA-eligible CICS modules.

Any user PPT modules that are read-only may also be placed in the SVA.

OVERVIEW

The benefits to be derived from using the SVA are:

- **Sharing** - Modules in the SVA may be shared by two or more CICS systems in the same processor, thereby achieving an overall reduction in the total working set. The CICS systems need not be using intercommunication facilities in order to benefit from the sharing of modules.
- **Integrity** - The SVA is key 0 protected, so all modules placed there are automatically protected from being overwritten by other programs such as CICS applications. This integrity feature is equally applicable to a single CICS system within the processor.

Support for CICS modules in the SVA falls into two categories:

1. Modules that are required to be in the SVA if MRO is being used. These modules are:
   - DFHIRP, the interregion communication program. For integrity reasons, this CICS nucleus module must be used from the SVA.
   - DFHSCTE, the subsystem control table extension. This module provides an anchor for the interregion communication control block structure and is **not** read-only. It is neither a nucleus module nor a PPT module.
2. Those CICS nucleus and PPT modules that are read-only and hence SVA-eligible, but are not required to be in the SVA. You can choose which modules in this category to place in the SVA. Your selection would generally include modules in the working set of the installation. Appendix B lists the CICS modules that are eligible for the SVA. If you have modified a module that is listed as SVA-eligible, it is your responsibility to verify that the read-only attribute of the module has not been destroyed if that module is to be used from the SVA.

User application programs written in **assembler** language (using either the command-level or the macro-level interface) are eligible for the SVA, provided that they are read-only. Application programs written in other languages cannot be used from the SVA, because the resulting PPT modules (phases in the core image library) are **not** read-only. BMS map sets, and programs for which RELOAD=YES is specified in the PPT, are also not SVA-eligible.

There are no special CICS table or system initialization override parameters controlling the usage of CICS modules from the SVA. The core image library search sequence in effect for the CICS execution determines whether or not modules in the SVA are used, the default being to use modules from the SVA.

The usual CICS system initialization suffixing options and overrides apply to the selection of modules that have been installed in the SVA.

Nucleus modules to be used from the SVA still require entries in the nucleus load table (NLT), although those in the CICS default table in module DFHSIB1 will suffice.

The ALIGN, FIX, ADRSPCE, PAGEIN, and PAGEOUT operands on DFHNLT TYPE=ENTRY macro instructions do not apply when a nucleus module is used from the SVA, and these will be ignored.

Application load table (ALT) and processing program table (PPT) operands specifying virtual storage attributes (ALIGN, FIX, PAGEOUT, and ADRSPCE) will be ignored for PPT modules used from the SVA.

INSTALLATION AND USE

ALLOCATING SPACE IN THE SVA

Space in the SVA for the CICS modules and their system directory list (SDL) entries must be allocated at IPL time via the SVA command. This space is additional to any required for user phases. Appendix B gives the approximate sizes (rounded up to the nearest 1K byte boundary) for the SVA-eligible CICS modules. These sizes were determined by reference to a listing of the directory for the core image library of the pregenerated system. For more accurate space estimates, you are advised to consult the library listings at the time of installation, and to verify the space estimates against actual space layout once modules have been selected for residency in the SVA.

In calculating the space required, allowance must be made for the way in which VSE loads the SVA. Starting at the low end, phases are loaded in alphameric sequence regardless of any ordering within a SET SDL command. If a phase will not fit within the current page, it is loaded at the next (2K-byte) page boundary. No back-filling takes place.

For a full-function CICS system, the space required, if all the SVA-eligible **nucleus** modules are to be installed in the SVA, is estimated to be of the order of 450K to 500K bytes.

LOADING THE SVA

Modules to be loaded into the SVA must be link-edited as SVA-eligible, that is with the SVA operand added to their PHASE statements. CICS system generation ensures that all modules considered to be SVA-eligible (as listed in Appendix B) are cataloged in this way. To load any of these modules into the SVA, a SET SDL job control command must be issued from the background (BG) partition, naming the selected modules. This command can be issued at any time after IPL, but **must** be issued before bringing up any CICS system that will be using modules from the SVA. Phases are taken from the libraries of the BG core image library search chain. Thus any private core image libraries containing modules to be loaded into the SVA must be specified in the SEARCH operand of the LIBDEF command for BG.

USING THE SVA DURING CICS EXECUTION

The core image library search sequence in effect for the CICS execution will include any private core image libraries containing CICS modules. The default library search sequence for program fetch causes the system directory list to be searched before the library search chains. Thus the default is to use modules that have been loaded into the SVA.

If you want to use a private or nonshared version of a CICS module that has been loaded into the SVA, in many cases this can be achieved by a change of module suffix accompanied by an appropriate CICS system initialization override. However, if the module is not suffixable, it will be necessary to catalog the module in a separate private core image library and to specify a temporary (Job) SEARCH chain with 'SDL' placed after this library in the chain. The same technique can be extended in order to test a service fix or a new release of CICS in one partition, where the service fix or new release has been installed into a private core image library, and where other CICS systems in the same processor are to continue to execute using modules from the SVA.

UPDATING PHASES IN THE SVA

Phases in the SVA may be updated or deleted without re-IPLing (for details, see the VSE/AF manuals listed in the Bibliography). Earlier versions of these phases remain in the SVA, but are no longer addressable from the SDL. Note that a running CICS system will continue to use the original versions of nucleus modules that were in the SVA at the time the CICS system was initialized.

The CEMT NEWCOPY function may be used to replace a PPT module that has been updated in the SVA. The core image library search sequence defined for the CICS partition will determine whether or not the updated module in the SVA is used.

Under no circumstances must the DFHSCTE module be updated in the SVA while any CICS interregion communication is in progress, because this could prove disastrous.

See VSE/Advanced Functions System Control Statements for further details on the SVA IPL command and the SET SDL and LIBDEF job control commands, and the VSE/Advanced Functions System Management Guide for general information on using the SVA.

Chapter 2.7. Adding DL/I Support

When installing CICS support for DL/I DOS/VS, you have to:

1. Define DL/I data bases and application programs during the preparation of CICS tables.

2. Specify system table macros for DL/I support as follows:

   a. Define the DL/I DOS/VS application control table (ACT). This table is required to associate online application programs with one or more DL/I data bases.

   b. Optionally define a storage layout control (SLC) table for use in an online environment, to specify the sequence in which DL/I modules are to be loaded from the core image library during DL/I initialization.

   c. Include in the CICS file control table (FCT) an entry for each data base description (DBD) corresponding to a physical data base. The name in the DATASET parameter in the FCT and the NAME parameter in the DBD must be identical.

   d. If program isolation is active or if emergency restart or dynamic transaction backout is to be used with DL/I tasks, assign the DL/I DOS/VS log to the CICS system log. The DL/I DOS/VS log is assigned by use of the VSE UPSI byte information. See "Chapter 5.3. System Initialization Parameters" on page 267 for information on how to use the UPSI byte option with CICS.

   e. Define in the processing program table an entry for the DL/I language definition table (DLZHLPI) if the execution diagnostic facility is to be used with application programs containing EXEC DLI commands. DLZHLPI is a module provided with DL/I DOS/VS.

The DL/I DOS/VS Resource Definition and Utilities book tells you how to define and generate the DL/I tables.

"Chapter 4.13. DL/I Definitions" in this book describes how to define all DL/I data bases that are to be referenced during CICS execution.

A CICS-DL/I installation tester (DFHTDLI) application program is also supplied with the pregenerated CICS system — this program is described in Appendix E.

REASSEMBLING DL/I DOS/VS MODULES

The pregenerated modules in CICS were assembled against DL/I DOS/VS Version 1 Release 6. No CICS system generation is required if you have that release of DL/I DOS/VS.

A set of DL/I DOS/VS modules will need to be reassembled if a new release of CICS/DOS/VS is produced. The DL/I DOS/VS program directory contains information on how to create the necessary job streams.

There is also a set of CICS modules that needs to be reassembled for every new release of DL/I DOS/VS. The CICS system generation macro instructions have been coded for you in a source member called DFHSGDLI supplied in the CICS source library.

```
         DFHSG TYPE=INITIAL,                                             X
               OPSYS='DOS/VSE',                                          X
               ASMBLR=ASSEMBLY,                                          X
               PRINT=(LIST,XREF,NODSECT,DSLIST),                         X
               STAGE2=FORCE,                                             X
               EJECT=YES,                                                X
               JOBNAME=DFH,                                              X
               DLI=YES,                     DL/I SUPPORT                 X
               STARTER=YES,                                              X
               MOD=(DBP,2$,ISP,,TBP,2$,XFP,),                            X
               ACCTID='CICS_160'
         DFHSG PROGRAM=PREGEN
         DFHSG TYPE=FINAL
         END
```

Figure 48. Source Member A.DFHSGDLI

When this job executes, the library search order must include your CICS, DL/I, and VSAM libraries. Four CICS assembly jobs will be produced, and these must be executed to assemble the DL/I-dependent modules.

"Chapter 2.5. CICS System Generation" has examples of executing DFHSG job streams using cards and tape as intermediate output for the assembly steps.

Chapter 2.8. Adding CICS Monitoring Facility Support

The CICS monitoring facility enables you to collect performance-related data during online processing for later offline analysis. The data is collected in three monitoring classes:

- **Accounting** class data. This is high-level information such as the number of transactions for each combination of transaction type, terminal identification, and operator identification. This data can be used for installation accounting purposes.
- **Performance** class data. This is more detailed resource-level information such as the processor and elapsed time for a transaction, or the time spent waiting for I/O. This data can be used for performance monitoring and capacity planning. As well as data for each transaction, global performance data can be collected to provide information on the entire CICS system.
- **Exception** class data. This is information on exceptional conditions raised by a transaction, such as queuing for VSAM strings or waiting for temporary storage. This data will highlight possible problems in system operation.

Each class of monitoring can be active alone or jointly with other classes, and the information provided by each class of data is described in the CICS/VS Performance Guide.

Standard CICS monitoring data is collected at predefined event monitoring points (EMPs) in the CICS code. You can specify which classes of data are to be collected at these EMPs, but you have no control over the actual data collected. Timing information in the standard CICS monitoring data can be invalidated by transactions that terminate abnormally.

As well as the standard CICS monitoring data, user application programs can contribute data to user fields in the transaction-level accounting and performance class records but not the exception class. In accounting class records, one field, a counter, is reserved for your use. In performance class records, one optional character-string field and a variable number of fields for counters and time periods can be used. See the CICS/VS Customization Guide for further information.

The monitoring data is collected in separate buffers for each monitoring class. The way in which these buffers are written to CICS user journals is controlled by the monitoring control table (MCT). It is possible to merge any two or all three classes of data onto one journal.

CONTROLLING THE MONITORING FACILITIES

The monitoring facilities are controlled by entries in the monitoring control table, the journal control table, and the system initialization table, and by the CSTT transaction. Entries are also required in the CICS system definition file if you are using resource definition online), or in the processing program table (PPT) and the program control table. You should read the information in the following sections in conjunction with the relevant information on each table in the CICS/VS Resource Definition Guide.

MONITORING CONTROL TABLE

The DFHMCT TYPE=RECORD macro is used to specify the class of monitoring data to be recorded and, for each class of data being collected, the CICS user journal to which the data is to be sent.

To specify the class of data to be collected, code the CLASS= operand with ACC, PER, or EXC. If you want to define more than one monitoring class, you have to code separate TYPE=RECORD macro instructions for the other monitoring classes you require.

Code DATASET= with the number of the journal (in the range 02 through 99) to be used (this identifier must correspond with that specified in the journal control table).

For each transaction, the monitoring facility will acquire up to three buffers (one for each class of data being collected) from the dynamic storage area. When a task terminates, the contents of each task-related buffer is written to a monitoring buffer, the size of which is controlled by the MAXBUF operand. For information on calculating the MAXBUF size, see the relevant section in the CICS/VS Customization Guide.

If you require **global performance** data, code the FREQ= operand with the time interval required between the start and stop of each interval for which global performance data is required. The number specified for FREQ= is in seconds, and can be between 1 and 43200 (12 hours).

For **performance class** data only, code CONV=YES if you want conversational tasks to have separate performance class records produced for each pair of terminal control I/O requests.

For **performance class** data only, if you want the amount of processor time and the paging activity to be recorded in the data for each transaction, code CPU=YES to specify that this facility is required for a CICS system running under VSE.

A sample monitoring control table with appropriate entries is included in the CICS source statement library as DFHMCT1$ (with copy book DFHXMCTS).

JOURNAL CONTROL TABLE

Journals used to record the monitoring output must be defined in the journal control table as user journals by specifying a JFILEID value between 02 and 99 in the DFHJCT TYPE=ENTRY macro. The JFILEID value must be identical with the DATASET= value specified in the corresponding DFHMCT TYPE=RECORD macro instruction. In addition, the journals must be specified with FORMAT=SMF.

For further information on calculating the value to use for BUFSIZE and BUFSUV, see the relevant section of the CICS/VS Customization Guide.

A sample journal control table is included in the CICS source statement library as DFHJCT1$ (with copy book DFHXJCTS).

SYSTEM INITIALIZATION TABLE

To receive any class of monitoring data, CMP=YES must be specified in the system initialization table, together with MCT=YES or xx. Monitoring classes to be active at system initialization time must be specified with the MONITOR operand (ACC, PER, or EXC). In addition, you must have a nondummy version of DFHJCP and a journal control table.

CSTT TRANSACTION

Activation and deactivation of monitoring classes during CICS execution is controlled by the CSTT transaction, which has the following format.

```
CSTT MONITOR,{ON|OFF}={([ACC][,PER][,EXC])|ALL}
```

This transaction also controls the CICS automatic statistics. Further information is given in the CICS/VS Operator's Guide.

CSD FILE OR PPT AND PCT ENTRIES

In the CSD file, the entries required for monitoring are already present in the IBM-provided DFHSTAND group.

The entries required for monitoring are included in the PPT and PCT, with other standard entries, when FN=STANDARD is specified on the DFHPPT TYPE=GROUP and DFHPCT TYPE=GROUP macro instructions.

PROCESSING OUTPUT FROM THE CICS MONITORING FACILITY

CICS monitoring facility output can be processed by the following:

1. A sample program, DFHXMOLS, which gives a basic listing of the CICS monitoring facility output. Comments within the program may help you if you wish to do your own processing of CICS monitoring facility output. For information on DFHXMOLS, see the CICS/VS Customization Guide.

2. The extended-maintenance FDP, CICSPARS (5798-DAB), which executes in conjunction with another FDP — GPAR (or GPAR-DOS). Service Level Reporter (program number 5740-DC3) is an OS/VS1 and MVS program product that can process CICS monitoring facility output presented to it on tape. For information on these products, see the relevant manuals listed in the CICS/VS Performance Guide.

3. An output-processing and analysis program provided by yourself. If you wish to code such a program, the CICS/VS Customization Guide contains relevant information about the CICS monitoring facility.

Chapter 2.9. Migrating Resource Definitions

This chapter assumes that you already have a CICS system installed with the control tables you require for normal operation. Information is given in this chapter about how to change your CICS system for resource definition online (RDO), to enable you to add new programs and transactions online using the CEDA transaction.

In order to set up RDO, you have to provide a CICS system definition (CSD) file and make it available to CICS. No special system generation options are required for RDO, but the required support must be generated in various CICS tables before you can use the CEDA transaction.

PREPARING FOR MIGRATION

1. Create a CICS system definition (CSD) file. Define a CICS system definition file to VSAM (using AMS) as described in "Job Control Statements to Define and Initialize the CICS System Definition File" on page 220. The file will hold the resource definitions when they are created.

2. Initialize the CSD file. Before it can be used, the CSD file must be initialized. The CSD file is initialized by means of the DFHCSDUP offline utility program, using the INITIALIZE command. This operation primes the CSD file with the definitions of all the IBM-supplied programs, transactions, and profiles. For details of the job stream required to run DFHCSDUP, see "Job Control Statements to Define and Initialize the CICS System Definition File" on page 220.

3. Make the CSD file available to CICS. Add the job statements for the CSD file to the CICS execution job stream. For details of the job statements, see "Job Control Statements for CICS Execution" on page 222.

4. Add entries to your existing program processing table (PPT) and program control table (PCT) to define the CEDA transaction and the programs it uses. These entries must be present in the PPT and PCT specified for use by CICS initialization.

   RDO will probably enable you to dispense with the PPT and PCT completely. However, in order to initiate the CEDA transaction for the very first time, your existing PPT and PCT have to be modified and regenerated as follows:

   a. Add the required programs to support the CEDA transaction to your program processing table (PPT). To do this, code:

      ```
      DFHPPT TYPE=GROUP,FN=SPI
      DFHPPT TYPE=GROUP,FN=BACKOUT
      DFHPPT TYPE=GROUP,FN=OPENCLSE
      ```

   b. Add the CEDA transaction definition to your program control table (PCT). To do this, code:

      ```
      DFHPCT TYPE=GROUP,FN=SPI
      ```

   c. You must provide security protection for the CEDA transaction by coding:

      ```
      DFHPCT TYPE=INITIAL,TRANSEC=(SPI(n),...),...
      ```

      where n is a security code that has been assigned to authorized users of CEDA.

The definition of the CEDA transaction must specify DTB=YES. This is the default unless you specify DTB=NO on the DFHPCT TYPE=INITIAL macro, in which case you must code a DFHPPT TYPE=ENTRY macro for the CEDA transaction, specifying DTB=YES.

Note: You cannot use the definitions from your initialized CSD file at this stage, because they are not yet included in the running CICS system.

For further details, see the CICS/VS Resource Definition Guide.

5. Check your sign-on table. The operator identification specified in the OPIDENT parameter is used by the lock mechanism of the CEDA transaction. If two or more operators with the same OPIDENT are allowed to use the CEDA transaction, one operator will be able to update groups locked to the other operator(s).

6. Add the definition of the CSD file to your file control table (FCT). To do this, code:

   ```
   DFHFCT TYPE=DATASET,DATASET=DFHCSD,...
   ```

   For further information on the FCT entry for the CSD file, see "File Control Table Definition of the CSD File" on page 220.

7. Ensure that support for dynamic transaction backout is specified in your system initialization table. To do this, code:

   ```
   DFHSIT TYPE=CSECT,DBP=xx,...
   ```

   where xx is the suffix of the required version of the dynamic backout program. The pregenerated versions have the suffixes 1$ (without DL/I support) and 2$ (with DL/I support).

8. Arrange an audit trail (optional). If you wish to have an audit trail of all the CEDA commands that modify the CSD file or the running CICS system, you should define a transient data queue named CSDL. To do this, code:

   ```
   DFHDCT TYPE=INDIRECT,DESTID=CSDL,INDDEST=CSMT
   ```

   This definition will send the audit trail of CEDA commands to the same destination as the CSMT messages. If you wish, you may direct the CSDL destination to any other transient data queue or define it as an extrapartition queue. For details of required block sizes and so on, see under "Required Entries in the DCT" in the CICS/VS Resource Definition Guide.

   If you do not want an audit trail of CEDA commands, do not define the CSDL queue in your destination control table.

9. Use CEDA. It will now be possible to initialize CICS and invoke the CEDA transaction. For details of how to use the CEDA transaction, see the CICS/VS Resource Definition Guide. It will be possible to examine the CICS-supplied DFHLIST and the contents of the groups in that list. Furthermore, it will be possible to add new resource definitions to the CSD. A useful exercise would be to run through the example session in the CICS/VS Resource Definition Guide, substituting transaction and program names of your own choice.

10. Prepare tables for migration. It is not necessary to migrate your existing PPTs and PCTs into a form accessible by the CEDA transaction. If the advantages of using RDO convince you that such a step is worthwhile, however, you should make some changes to your PPT before you carry out the migration.

    All DFHPPT TYPE=ENTRY macro instructions for map sets should be changed to specify the new MAPSET operand instead of the existing PROGRAM operand. In addition, you should ensure that existing PPT entries for programs do not specify USAGE=MAP. If you do not make these changes, the DFHCSDUP offline utility program will define some map sets as programs and some programs as map sets.

11. Migrate tables to the CSD file. The load module versions of your tables can be migrated into CSD file entries using the MIGRATE command of the DFHCSDUP offline utility program. For further details, see the next section.

MIGRATION OF PCT AND PPT ENTRIES

You can use an offline utility routine to convert the entries at present in your PCT and PPT into resource definitions on the CSD file. This is done by executing the MIGRATE command of the DFHCSDUP offline utility program, described in "DFHCSDUP Commands" on page 225.

The tables to be migrated must first be assembled for CICS 1.6, then link-edited into your CICS core image library. You should ensure that none of the tables you migrate contains internally-duplicated entries. You should not attempt to migrate a table that produced errors with a severity greater than 4 when it was assembled.

One group is created on the CSD file for each table migrated. If the migrated table is a PPT, the group will contain program, map set, and partition set definitions. If it is a PCT, the group will contain definitions of transactions and profiles. After the migration, you should reorganize the definitions into more convenient, smaller groups, by using the CEDA transaction COPY and DELETE commands.

If your tables contain entries for any IBM standard programs, profiles, or transactions, these will not normally be migrated to the group created by the MIGRATE command. This is because the definitions for these entries already exist in one of the IBM standard groups created by the INITIALIZE command, described in "DFHCSDUP Commands" on page 225.

In special cases where the properties of the resource as defined in your table differ from those of the IBM-supplied definition, the definition will be migrated. An example of this is transaction definitions with security protection, where the security key provided may not be the same as the one that was specified in the DFHPCT TYPE=INITIAL macro instruction. In this case, the definition will be migrated.

MIGRATION OF PROGRAMS, MAP SETS, AND PARTITION SETS (PPT)

One definition record is created on the CSD file for each PPT entry migrated. When you prepare the DFHPPT macros to assemble your table, you can distinguish map sets and partition sets from programs by using the MAPSET=name or PARTSET=name parameters. If you have done this, the resulting resource definitions will always have the correct resource types.

If you have not done this, and your map sets were identified by coding the PROGRAM=name parameter, the migration routine will normally create a PROGRAM definition. If however, you coded the parameter USAGE=MAP, the migration routine assumes that the entry is intended for use as a map set, and creates a MAPSET definition. (This does not happen if you specified a language other than PGMLANG=ASSEMBLER.)

Note, however, that the migrated definition of a map set does not have the properties associated with USAGE=MAP. USAGE=MAP is not supported by resource definition online.

In summary, the resource types created from PPT entries are:

DFHPPT parameter coded	CSD file resource created
PROGRAM=name (without USAGE=MAP)	PROGRAM
PROGRAM=name (with USAGE=MAP)	MAPSET
MAPSET=name	MAPSET
PARTSET=name	PARTITIONSET

MIGRATION OF PROFILES (PCT)

If your PCT contains any user-created profiles, they will be migrated to the CSD file as PROFILE definitions. Profiles can also be created automatically from transaction entries (see below). The automatically-created ones are interchangeable with the user-defined ones, both in the way they can be associated with transactions, and in the context of the EXEC CICS ALLOCATE command.

The CICS-provided profiles (DFHCICST, DFHCICSV, and so on) are created on the CSD file by the INITIALIZE command, so definitions of these will not normally be migrated from your tables.

MIGRATION OF TRANSACTIONS (PCT)

Some of the properties that, in earlier releases of CICS, were attributes of a transaction are now handled as attributes of a profile. In a system with RDO, every transaction now has an associated profile that contains most of the terminal-related properties. A profile can be used by any number of transactions.

For every transaction that is migrated, a TRANSACTION definition will be created, and possibly a PROFILE definition as well. Whether that happens depends on the values of those transaction attributes that are now handled as profile properties.

All transactions will be migrated with the following properties:

```
CLASS=LONG
PRIVATE=YES
DTB=YES
FDUMP=(ASRA,ASRB)
ANTICPG=NO
```

In addition, any PRMSIZE is ignored. There is no way, using the resource definition online facility, of changing these properties.

If the transaction being migrated has properties that differ from the list above, messages will be produced by the migration utility.

PROFILE PROPERTIES

When a transaction entry is migrated, a check is made on those fields of the PCT entry that have become 'profile' properties. They are compared with the properties of the default profile DFHCICST. If they match, DFHCICST is taken as the associated profile, and no new profile is created. Otherwise, they are compared with the properties of two other CICS-provided profiles (DFHCICSV and DFHCICSA) in a similar way. If the properties do not match any of these three standard profiles, a new profile is created automatically. It will be given the name XXXXtttt, where tttt denotes the transaction name. The profile called XXXXtttt is automatically associated with transaction tttt.

When the next transaction entry is migrated, the newly-created profile will be used for comparison, along with DFHCICST, DFHCICSV, and DFHCICSA. The properties may match any of these four profiles. As this process continues, several new profiles may be generated, and several transactions may become associated with a generated profile XXXXtttt.

In the very unlikely event of the name of a generated profile XXXXtttt being the same as that of a user-defined profile (created with a TYPE=PROFILE entry in the PCT), the latter will be migrated in the normal way. It is up to you to use the CEDA transaction afterwards to ensure that the transaction that caused this to happen is correctly defined and associated with a suitable profile, which you must define yourself.

Transaction Names

A transaction will be migrated to the CSD file only if its name conforms to the rules for a transaction name acceptable to RDO. (See the description of the DEFINE TRANSACTION command in the CICS/VS Resource Definition Guide.) Lowercase transaction names are permitted, and these will be migrated with lowercase names. Mixed uppercase and lowercase names will also be preserved.

Transactions with names containing unacceptable characters (for example, * and +) will not be migrated to the CSD file. You can use the CEDA transaction to define these with names that are valid for RDO, using the XTRANID alias to cope with special characters.

Transaction Aliases

In a system with RDO, every transaction must have a primary transaction name. This may not be the case for all the transactions currently in your PCT. Some may be referred to only by a TASKREQ key identifier. If this is the case, a name will be automatically assigned to the transaction when the PCT entry is migrated.

The name assigned to the transaction will be one of:

```
PA1, PA2, PA3     - for PA keys
PF1 through PF24  - for PF keys
OPID              - for the operator identification card reader
LPA               - for the 3270 light pen field
MSRE              - for the magnetic stripe reader
```

If a migrated transaction is assigned one of these names, it is possible that you already have another transaction in your table with the same name. If this happens, the latter will be migrated. The one with the name derived from a TASKREQ identifier will not be migrated, and you can use CEDA afterwards to define this transaction with a name that avoids the duplication.

AFTER YOU HAVE MIGRATED YOUR TABLES TO THE CSD FILE

1. Rearrange the resource definition groups.

   Bring up the CICS system with CEDA support, and use the CEDA transaction to examine the migrated resource definitions on the CSD file. The MIGRATE command will have created a group for each table. It will be found that such groups are not easily managed, and it is better to rearrange the transaction and program definitions into logical groupings. Use the COPY and DELETE commands to group related transactions, programs, and so on, into the same group. We suggest that you split groups containing many resource definitions (say, more than 200) into smaller groupings to avoid problems in the management of such large groups. Create lists that name the groups of resource definitions that must be installed together during CICS initialization.

   If you have migrated an extremely large PPT or PCT (with more than 2000 entries, say), you may experience problems when using the CEDA transaction to EXPAND or manipulate the very large group created. If this happens, you may have to split the table into smaller components and migrate the components individually.

The migration operation may have produced a number of messages indicating, for example, the certain transaction names have been created for transactions initiated by program function keys. You should read these warning messages carefully. In most cases, these messages will warn you that some parameter specified for a definition is not supported by resource definition online, and no further action is necessary. However, there may be messages that require corrective action, for example, those warning of duplication of definitions.

It may be necessary to change the name created by the migration utility by means of the RENAME command. In addition, the migration operation may have created certain profile names because a PCT TYPE=ENTRY macro was split into a transaction definition and a profile definition referenced by the transaction definition. You may wish to rename these profiles. If so, it will be necessary to alter the transaction definitions that refer to that profile.

The created profile definitions may differ from the standard CICS-provided definitions (DFHCICST) in ways that are of no significance to your implementation. For example, the DFHCICST has INBFMH=NO, whereas the PCT-derived profiles may have INBFMH=EODS (because this was the default for the PCT). If this distinction is of no significance to your installation, it would be preferable for all references to the derived profiles to be replaced by references to DFHCICST. This is most easily done by using the generic capabilities of the CEDA transaction ALTER command.

2. Change the transaction security codes.

   The following CICS-supplied transactions (shown here with their group names) are defined with a transaction security code (TRANSEC) of 2:

   - Master terminal — DFHMASTT (CSMT, CSST)
   - Operator — DFHOPER (CEMT, CEST)
   - Execution diagnostic facility — DFHEDF (CEBR, CEDF)
   - Interpreter — DFHINTER (CECI)
   - RDO — DFHSPI (CEDA)

   These transactions can be used only by a terminal operator having a security key that includes the value 2 (unless XSP=NO is coded). You will almost certainly want to change this security value by copying the appropriate groups and changing the TRANSEC value. A TRANSEC of 1 was not used because that would have enabled **all** operators to use the transactions if the CICS-supplied groups were to be installed unchanged by mistake.

3. Install CSD file definitions at CICS initialization.

   Code the GRPLIST operand on the DFHSIT macro or as an initialization override, specifying the list that contains the names of the groups to be loaded during system initialization. The contents of these groups will be merged into the PPT and PCT in storage. If you have specified either or both the PPT and PCT to have a blank suffix, it is necessary to code PPT=YES and/or PCT=YES as initialization overrides if you wish to load tables from the core image library and use the GRPLIST operand.

4. Abandon use of the PPT and PCT.

   The final step is to abandon the use of the PPT and PCT completely. Before doing so, you must ensure that the list you intend to use names all required groups, including all the CICS-provided groups that your system needs.

   To avoid loading the PPT or PCT from the core image library, delete the PPT and PCT operands from your DFHSIT macro, or code PPT=NO and PCT=NO as system initialization overrides. In this case, all transactions and program definitions will be derived from the CSD file, and no PPT or PCT will be loaded into storage.

   After you have set up appropriate back-up procedures for your CSD file and your production system is running successfully without use of the PPT and PCT load tables, you may delete your PPT and PCT (both the source form and the core image library phase form) from your libraries.

Part 3. Installing Control Tables and Application Programs

PART 3

Chapter 3.1. Introduction

This part is concerned with the **application** component of CICS. This relates to those elements of a running CICS system that are specific to the individual user and are not provided as part of the distributed system (except for the sample tables, application programs, and utilities mentioned in Appendixes D and E).

These elements will primarily consist of user-provided application programs or application packages, together with associated maps or map-set definitions. You can install these in a CICS system at any time after the CICS system itself has been installed.

In order to install application programs or packages, you will need to make changes to the CICS control tables, which define the user resources to the CICS system. This part tells you how to install modified or new tables in the CICS system, as well as how to install maps, map sets, and application programs. This information is given in the following chapters:

PART 3

- Installing CICS control tables (Chapter 3.2)
- Installing map sets and partition sets (Chapter 3.3)
- Installing application programs (Chapter 3.4).

This part does not tell you how to code the various resources. The CICS/VS Resource Definition Guide tells you how to code the CICS control tables to obtain the facilities you need. Application programming is dealt with in the CICS/VS Application Programmer's Reference Manual (Macro Level) and CICS/VS Application Programmer's Reference Manual (Command Level). The CICS/VS Recovery and Restart Guide contains guidance information on the preparation of node error programs, terminal error programs, and program error programs.

Before you start to define resources for your CICS system, you should:

- Review the pregenerated control tables described in Appendix D. Decide which of these, if any, are appropriate as they stand or if they can be used as a base for modification. Ensure that the tables and programs you select are consistent.

- For the program processing table and program control table, decide which transactions and programs should be defined in the tables, and which entries will be created dynamically by the CEDA transaction.

Chapter 3.2. Installing CICS Control Tables

CICS is a table-driven system. Before CICS can be run, the resources it controls must be defined in CICS control tables or (for certain resources) in the CICS system definition file. This chapter describes the preparation, assembly, and link-editing of CICS control tables. For information on the CICS system definition file, see "Chapter 4.10. CICS System Definition File."

CICS tables describe the data base/data communication environment and the treatment to be given to elements of that environment. All information regarding the terminals, data sets (permanent and temporary), programs, transactions, and operator identification is contained in these tables. The tables are created independently of system generation, but some of the tables are required for the system to be operational.

Each control table is created separately and can be re-created at any time prior to system initialization.

PART 3

The list below gives the names of all tables that can be installed:

Table	Name
Application load table	DFHALTxx
Destination control table	DFHDCTxx
File control table	DFHFCTxx
Journal control table	DFHJCTxx
Monitor control table	DFHMCTxx
Nucleus load table	DFHNLTxx
Program control table *	DFHPCTxx
Program list table	DFHPLTxx
Processing program table *	DFHPPTxx
System initialization table	DFHSITxx
Sign-on table	DFHSNT
System recovery table	DFHSRTxx
Terminal control table	DFHTCTxx
Terminal list table	DFHTLTxx
Temporary storage table	DFHTSTxx
Transaction list table	DFHXLTxx

Note: The resources defined by means of the tables marked with an asterisk may alternatively be defined, modified, and installed while CICS is running. This method avoids the necessity of preparing and link-editing the tables. Instead, you use:

- The CICS-provided CEDA transaction (see the CICS/VS Resource Definition Guide) and
- The CICS system definition (CSD) file (see "Chapter 4.10. CICS System Definition File").

If you wish to keep your CICS system running for an extended period of time and want to add new facilities, CEDA also allows you to change definitions without having to shut down and restart CICS — this is especially useful in the test phase of a CICS installation.

Versions of certain of these control tables are provided as samples in the pregenerated system (see Appendix D).

Apart from the sign-on table, all these tables can be created in more than one version. The different versions are distinguished by specifying a 2-character suffix when the tables are defined.

This suffix can then be used in the system initialization table or as an override parameter during system initialization to identify the version to be used for a particular CICS execution.

MINIMUM RESOURCE DEFINITIONS

Two CICS tables must exist in the load library before CICS can be run:

- System initialization table
- Terminal control table.

In addition, program and transaction definitions must exist in the system. You can do this

- By providing a processing program table (PPT) and program control table (PCT). You can supplement these with your own entries, and assemble and link-edit these tables into the core image library as described later in this chapter. You must check the security values associated with all transactions to ensure that they are available only for authorized operators.

- As an alternative to assembling and link-editing the PPT and PCT, you can run the DFHCSDUP offline utility to create and initialize the CICS system definition (CSD) file with the required entries. You can then add further program and transaction entries while running CICS, by using the CEDA transaction (see "Chapter 4.10. CICS System Definition File"). In this case, the file control table (FCT) will also be required, with a DFHCSD entry.

Other tables are needed only if you are using the corresponding CICS facilities. For full details of all CICS tables, essential and optional, see the CICS/VS Resource Definition Guide.

INSTALLING THE CICS CONTROL TABLES

The CICS/VS Resource Definition Guide tells you how to code the macro instructions for each control table, and explains the meaning of the various parameters that can be used. The rest of this chapter describes the procedures to be followed to assemble and link-edit the control tables and add them to the working system.

New tables can be assembled at any time before a CICS system is started up.

ASSEMBLING AND LINK-EDITING THE CONTROL TABLES

The steps and procedures for preparing the control tables are illustrated in Figure 49. The tables are generated by assembling the appropriate macro instruction (for example, DFHSIT macro for the system initialization table) with its associated operands. The output of each assembly is preceded by the linkage editor control statements (PHASE and INCLUDE) required to link-edit the table into the CICS core image library.

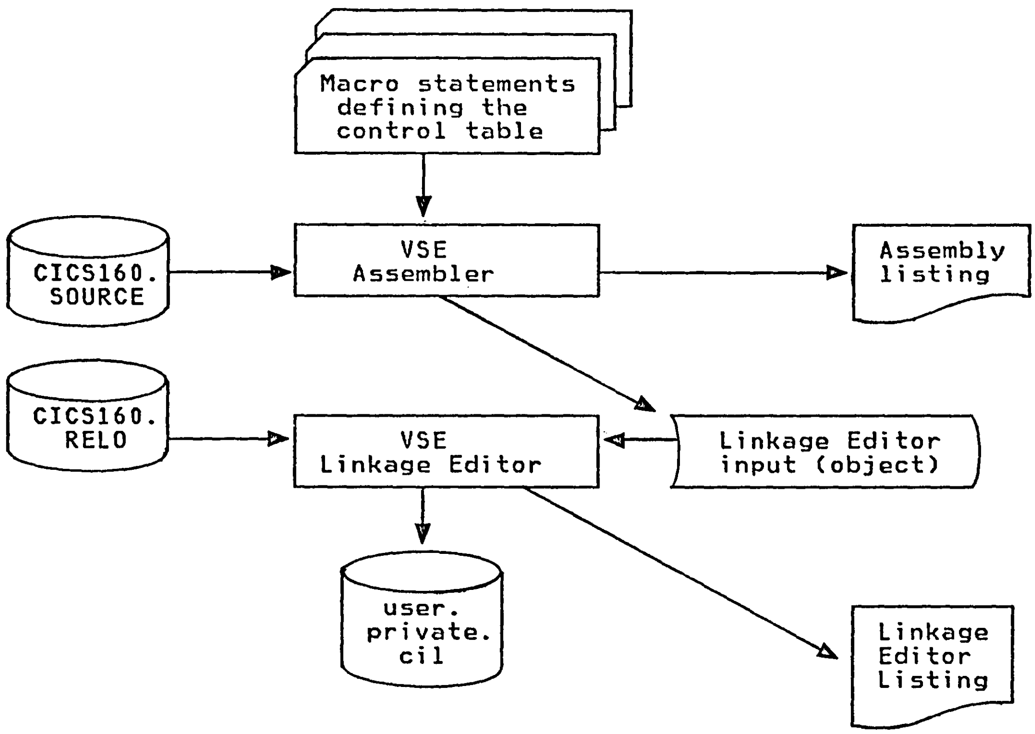


Figure 49. Installing the CICS Control Tables

Each group of source statements defining a control table must be terminated by an assembler END statement. The END statement may include the label (symbol) of the table entry point, in the form DFHxxxBA, where xxx is the name of the table (for example, TCT for the terminal control table). If this is not provided, CICS will generate the entry point address itself from the TYPE=INITIAL macro instruction for the table.

Note: The label must be coded if a user-supplied nucleus load table specifies ALIGN=ENTRY for the table. If the label is omitted in this case, no entry point is assigned by the linkage editor, and the table will be load-point aligned rather than entry-point aligned.

The following is an example of the job stream required to assemble and link-edit a CICS control table:

```
// JOB CICSTAB
*  CICS/DOS/VS CONTROL TABLE GENERATION
// OPTION  CATAL,NODECK,ALIGN
   ACTION SMAP            (VSE/AF Version 1 Release 3)
// LIBDEF CL,TO=user-cil-filename
// LIBDEF RL,SEARCH=cics-rlb-filename
// LIBDEF SL,SEARCH=cics-slb-filename
// EXEC ASSEMBLY
         .
          source statements defining a CICS control table
         .
         END
/*
// EXEC    LNKEDT
/&
```

Figure 50. Assembling and Link-Editing a CICS Control Table

Notes:

1. ACTION SMAP may be used with VSE/Advanced Functions Release 3 to produce a linkage editor map, sorted into alphameric order of CSECTs.

2. When a destination control table or a file control table addressing DAM files is link-edited, the unresolved external reference $$$$$$$$ is listed and the module will also have one unresolved address constant.

3. When a file control table addressing ISAM files is link-edited with the AUTOLINK option, one or more unresolved external references may be listed, depending on which of the CICS versions of the ISAM logic modules are present in the relocatable library. The references are of the form:

   ```
   DFHISMxy where x=R includes RPS support
                  x=N without RPS support

            and   y=C includes CORDATA support
                  y=N without CORDATA support
   ```

   In order to avoid generating unnecessarily large file control tables containing unwanted ISAM logic modules, you should specify the linkage editor NOAUTO option, and select the logic modules required, by means of INCLUDE statements.

Any other assembly and/or link-edit errors should not be ignored.

THE NEXT STEP

After you have successfully assembled and link-edited your table, you have to define it in the system initialization table which, in turn, is used when the system is initialized. Alternatively, the system operator can override the system initialization table and specify the new table explicitly in the system initialization override parameters (see "Chapter 5.3. System Initialization Parameters" for further information).

Chapter 3.3. Installing Map Sets and Partition Sets

Basic Mapping Support (BMS) enables an application program to read in device-dependent data and convert it to a device-independent standard form, or to generate device-dependent output data from this device-independent standard form. In both cases, the structure of the device-independent standard form, and the layout of the data on the display terminal are determined by a user-defined **map**. Related maps — for example, maps used in the same application program — are grouped together into a **map set**. See the CICS/VS Application Programmer's Reference Manual (Macro Level) or the CICS/VS Application Programmer's Reference Manual (Command Level) for further information on the definition and use of maps and map sets.

Some terminals, such as the IBM 8775 display terminal and the IBM 3290 display terminal, support screen division by partitions. The available display area may be split into a set of related 'logical screens' called partitions. The layout and properties of the set of partitions which can be simultaneously displayed on a terminal are defined by the BMS user in a **partition set**. See the CICS/VS Application Programmer's Reference Manual (Command Level). for further details on the definition and use of partition sets.

PART 3

CICS supports the definition of map sets and partition sets by assembler macro instructions. Map sets may also be defined interactively, using program products such as Screen Definition Facility (SDF)/CICS (program number 5740-XYF).

The remainder of this chapter describes how to assemble and link-edit map sets and partition sets for use with CICS, assuming that they are defined by the CICS-supplied macro instructions.

INSTALLING MAP SETS

Two kinds of map set must be generated:

- A **physical** map set, used by BMS to translate data from the standard device-independent form used by application programs to and from the device-dependent form required by terminals.
- A **symbolic description** map set, used in the application program to define the standard device-independent form of the user data. This is a DSECT in assembler language, a data definition in COBOL, a BASED or AUTOMATIC structure in PL/I, and a data structure in RPG.

Physical map sets must be cataloged in the core image library. Symbolic description map sets may be cataloged in the source statement library, or they may be included in the application program itself.

The map set definition macro instructions are assembled twice, once to produce the physical map set used by BMS in its formatting activities, and once to produce the symbolic storage definition that will be copied into the application program. The distinction between these two types may be made either by the TYPE operand of the DFHMSD macro instruction, or by the use of the SYSPARM parameter of the OPTION statement (SYSPARM='MAP' or SYSPARM='DSECT'), in which case the TYPE operand is ignored. The use of SYSPARM allows both the physical map set and the symbolic description map set to be generated from the same unchanged set of BMS map set definition macro instructions.

Map sets can be assembled as either **unaligned** or **aligned** (an aligned map is one in which the length field is aligned on a halfword boundary). You should use unaligned maps except in cases where an application package requires the use of aligned maps.

The distinction between aligned and unaligned map sets is made by use of the SYSPARM parameter on the OPTION statement when assembling the map set. SYSPARM='A' is used to indicate that aligned maps are to be assembled, the type of map being taken from the TYPE operand on the DFHMSD macro instruction. Alternatively both the map set type and alignment may be specified by the SYSPARM parameter. SYSPARM='AMAP' generates an aligned physical map set, and SYSPARM='ADSECT' generates an aligned symbolic description map set.

Prior to CICS 1.6, BMS was generated by the DFHSG PROGRAM=BMS macro instruction to assume that all map sets are aligned, or that all map sets are unaligned. Thus aligned and unaligned map sets could not be mixed.

In CICS 1.6, the physical map set contains information indicating whether it was assembled for aligned or unaligned maps. This information is tested at execution time, and the appropriate map alignment used. Thus aligned and unaligned map sets can be freely mixed.

The above alignment information is, of course, missing from physical map sets assembled before CICS 1.6. The BMS operand of the DFHSIT macro instruction and the associated startup override indicate whether map sets assembled before CICS 1.6 are aligned or unaligned. See the <u>CICS/VS Resource Definition Guide</u> for further information.

Offline map preparation support for the IBM 3270 Information Display System was provided under earlier CICS BMS versions through the DFHMDI and DFHMDF macro instructions, with no requirement for a DFHMSD macro instruction. Such maps are no longer supported. Their source definition must be modified to use the DFHMSD macro instruction, and the resultant map set reassembled.

Applications and maps designed for the 3270 Information Display System will run unchanged on devices supporting extensions to the 3270 data stream such as color, extended highlight, programmed symbols, and validation. To use fixed extended attributes such as color, only the physical map set need be reassembled. If dynamic attribute modification by the application program is required, both the physical and symbolic description map sets must be reassembled, and the application program reassembled or recompiled.

The following three sections describe, in turn, the preparation of physical map sets and symbolic description map sets separately, and then in one job. In these sections, it is assumed that the SYSPARM parameter is used to distinguish between the two types of map set.

INSTALLING PHYSICAL MAP SETS

The procedure for installing physical map sets, illustrated in Figure 51, is similar to that for installing assembler-language application programs. The relocatable library is not needed.

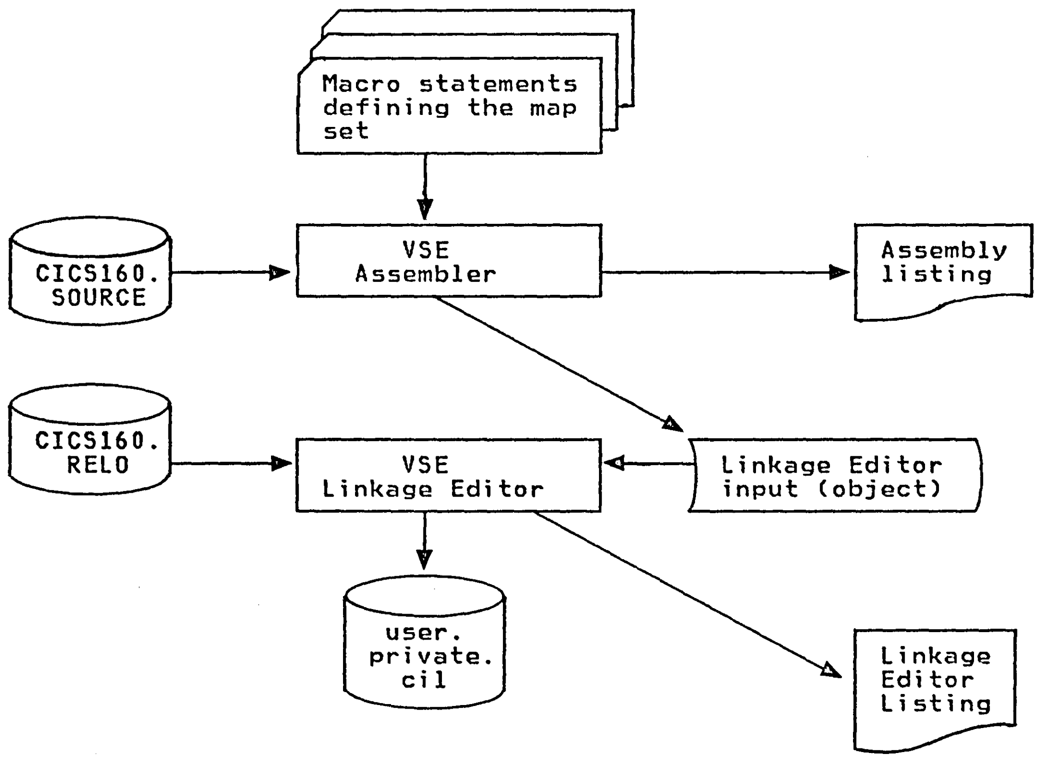


Figure 51. Installing Physical Map Sets

The following job stream is an example of the assembly and link-edit of physical map sets:

```
// JOB BMSMAP
*  CICS/DOS/VS ASSEMBLE AND LINK-EDIT BMS MAP SET
// OPTION  CATAL,NODECK,ALIGN,SYSPARM='MAP'   (see note 1)
   ACTION SMAP                        (see note 2)
   PHASE   name,*                     (see note 3)
// LIBDEF CL,TO=user-cil-filename
// LIBDEF RL,SEARCH=cics-rlb-filename
// LIBDEF SL,SEARCH=cics-slb-filename
// EXEC    ASSEMBLY,SIZE=nnK        (see note 4)
           .
           source statements defining the physical map set
           .
           END
/*
// EXEC    LNKEDT
/&
```

Figure 52. Assembling and Link-Editing a Physical Map Set

Notes:

1. If halfword-aligned length fields are required, it is necessary to specify the option SYSPARM='AMAP' instead of SYSPARM='MAP'

2. ACTION SMAP may be used with VSE/Advanced Functions Version 1 Release 3 to produce a linkage editor map, sorted into alphameric order of CSECTs.

3. The PHASE statement must be used to specify the name of the physical map set which BMS will load into storage. If the map set is device-dependent, the PHASE name should be derived by appending the device suffix to the original 1- to 7-character map set name used in the application program. The suffixes to be appended for the various terminals supported by CICS BMS depend on the parameter specified in the TERM or SUFFIX operand of the DFHMSD macro instruction that defined the map set. See the CICS/VS Application Programmer's Reference Manual (Command Level) or the CICS/VS Application Programmer's Reference Manual (Macro Level) for a complete list of map set suffixes.

4. A SIZE parameter value of up to 400K bytes will be required for the assembler under VSE.

Note that an entry in the processing program table is required for each physical map set. This is defined using the DFHPPT macro instruction or the CEDA DEFINE MAPSET command, as described in the CICS/VS Resource Definition Guide.

Programmers writing assembler-language macro level application programs have the option to assemble map sets into their application programs. In this case, no separate physical map set generation run is performed, and no processing program table entry is required. The application program passes the address of the map sets to BMS whenever a mapping operation is requested.

INSTALLING SYMBOLIC DESCRIPTION MAP SETS

Symbolic description map sets enable the application programmer to make symbolic references to fields in the physical map set. The source statements defining the symbolic description map set are assembled, and the output directed to SYSPCH (the output may be on card, tape, or disk). Where many map sets are to be used in an installation, or where there is multiple use of common map sets, you should place the symbolic description map sets in the source statement library, from which they can be copied into any application program.

When a symbolic description map set is generated under the same name for different programming languages, a separate copy must be placed on the sublibrary of the source statement library for each language. Figure 53 illustrates the preparation of symbolic description maps for BMS.

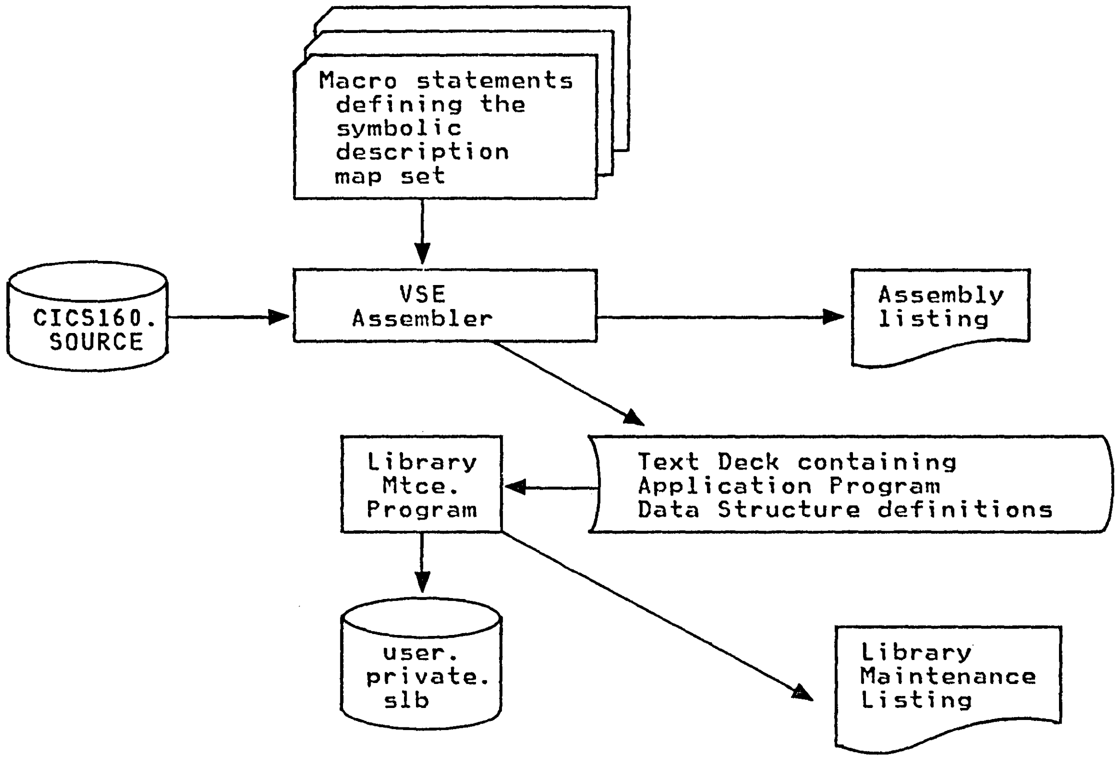


Figure 53. Installing Symbolic Description Map Sets

To use the symbolic description map set in a program, you must assemble the map set and obtain a punched copy of the storage definition through SYSPCH. Where many map sets are to be used in an installation, or where there are multiple users of common map sets, you should establish a private source statement library for each language that you use. The symbolic storage definitions for map sets should be placed in each library, from which they can be copied into application programs. When a symbolic storage definition is prepared under the same name for more than one programming language, a separate copy of the symbolic storage definition must be placed in each source statement library. You must ensure that the source statement libraries are correctly concatenated with LIBDEF.

You need only one symbolic description map set corresponding to all the different suffixed versions of the physical map set. For example, you might want to run the same application on terminals with different screen sizes. You would define two map sets each with the same fields, but positioned to suit the screen sizes. Each map set would have the same name but a different suffix, which would match the suffix specified for the terminal. You would need to assemble and link-edit the different physical map sets separately but create only one symbolic description map set, because the symbolic description map set would be the same for all physical map sets.

The following example, which applies to symbolic description map sets for any supported programming language, uses disk as intermediate storage between the assembly and the catalog in the target source statement library.

```
// JOB BMSASM
*  CICS/DOS/VS ASSEMBLE BMS SYMBOLIC DESCRIPTION MAP SET
*  USING DISK AS INTERMEDIATE OUTPUT
// DLBL IJSYSPH,'MAP ASSEMBLY',0,SD
// EXTENT SYSPCH,balance of extent information
ASSGN SYSPCH,DISK,VOL=volid,SHR
// OPTION DECK,SYSPARM='DSECT'
// LIBDEF SL,FROM=cics-slb-filename
// EXEC ASSEMBLY
   PUNCH ' CATALS sublib.bookname'         (see note 1)
   .
   source statements defining symbolic description map set
   .
   END
/*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'MAP ASSEMBLY',0,SD
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// LIBDEF SL,TO=user-slb-filename
// EXEC MAINT
CLOSE SYSIPT,SYSRDR
// EXEC SSERV                              (see note 2)
   DSPLY sublib.bookname                   (see note 1)
/*
/&
```

Figure 54. Assembling and Link-Editing a Symbolic Description Map Set

If halfword-aligned length fields are required, the option SYSPARM='ADSECT' should be specified instead of SYSPARM='DSECT'.

Notes:

1. The 'sublib' specified in the CATALS and DSPLY statements must correspond to the programming language specified in the symbolic description map definition:

   A. for assembler language
   C. for COBOL
   P. for PL/I
   R. for RPG II

2. The SSERV library service program lists the cataloged symbolic description map set.

INSTALLING PHYSICAL AND SYMBOLIC DESCRIPTION MAP SETS TOGETHER

The physical map set and the symbolic description map set can be assembled and cataloged in the same job using SYSPARM in the // OPTION job control statements for the assembler execution steps.

If unaligned length fields are required, SYSPARM='MAP' must be used to produce the physical map set, and SYSPARM='DSECT' to produce the symbolic description map set.

If halfword-aligned length fields are required, SYSPARM='AMAP' should be used to produce the physical map set, and SYSPARM='ADSECT' to produce the symbolic description map set.

The following job stream is an example of the assembly of unaligned BMS physical and symbolic description map sets in one job. Tape is used as intermediate storage for the symbolic description map set.

```
// JOB CICSBMS
*  CICS/DOS/VS ASSEMBLE AND CATALOG BMS PHYSICAL AND
*  SYMBOLIC DESCRIPTION MAP SET
// LIBDEF SL,TO=user-slb-filename
// EXEC    MAINT              STEP 1       (see note 1)
   CATALS  A.DUMMYMAP
   BKEND   A.DUMMYMAP
   .
   source statements defining physical map set
   .
   BKEND
/*
// OPTION  CATAL,NODECK,SYSPARM='MAP',ALIGN
   PHASE   mapname,*
// LIBDEF SL,SEARCH=(user-slb-filename,cics-slb-filename)
// LIBDEF CL,TO=user-cil-filename
// EXEC    ASSEMBLY,SIZE=nnK  STEP 2     (see notes 2 and 3)
   COPY    DUMMYMAP
   END
/*
// EXEC    LNKEDT             STEP 3     (see note 4)
// ASSGN   SYSPCH,cuu             (assign SYSPCH to tape)
// OPTION  DECK,SYSPARM='DSECT'
// LIBDEF SL,SEARCH=(user-slb-filename,cics-slb-filename)
// LIBDEF CL,TO=user-cil-filename
// EXEC    ASSEMBLY,SIZE=nnK  STEP 4     (see note 5)
   PUNCH   ' CATALS sublib.bookname'
   COPY    DUMMYMAP
   END
/*
// MTC     WTM,SYSPCH,2
// MTC     REW,SYSPCH
// RESET   SYSPCH
// LIBDEF SL,TO=user-slb-filename
// EXEC    MAINT              STEP 5       (see note 6)
   DELETS  A.DUMMYMAP
/*
// ASSGN   SYSIPT,cuu           (cuu from SYSPCH above)
// EXEC    MAINT              STEP 6       (see note 7)
/&
```

Figure 55. Assembling and Link-Editing a Physical and a Symbolic Description Map Set

Notes:

1. Step 1 places the BMS source statements on the source statement library using a specific bookname.

2. A SIZE parameter value of up to 400K bytes will be required for the assembler under VSE.

3. Step 2 assembles the BMS source statements to create the physical map set by specifying the system global SYSPARM='MAP' in the // OPTION job control statement.

4. Step 3 link-edits and catalogs the physical map set in the core image library.

5. Step 4 assembles the BMS source statements to create a symbolic description map set by specifying the system global SYSPARM='DSECT' in the // OPTION job control statement.

6. Step 5 deletes the BMS source statements from the source statement library.

7. Step 6 catalogs the symbolic description map set in the target source statement library.

INSTALLING PARTITION SETS

Partition sets are handled in the same way as physical map sets. There is no concept of a symbolic description partition set.

The procedure for installing partition sets is illustrated in Figure 56.

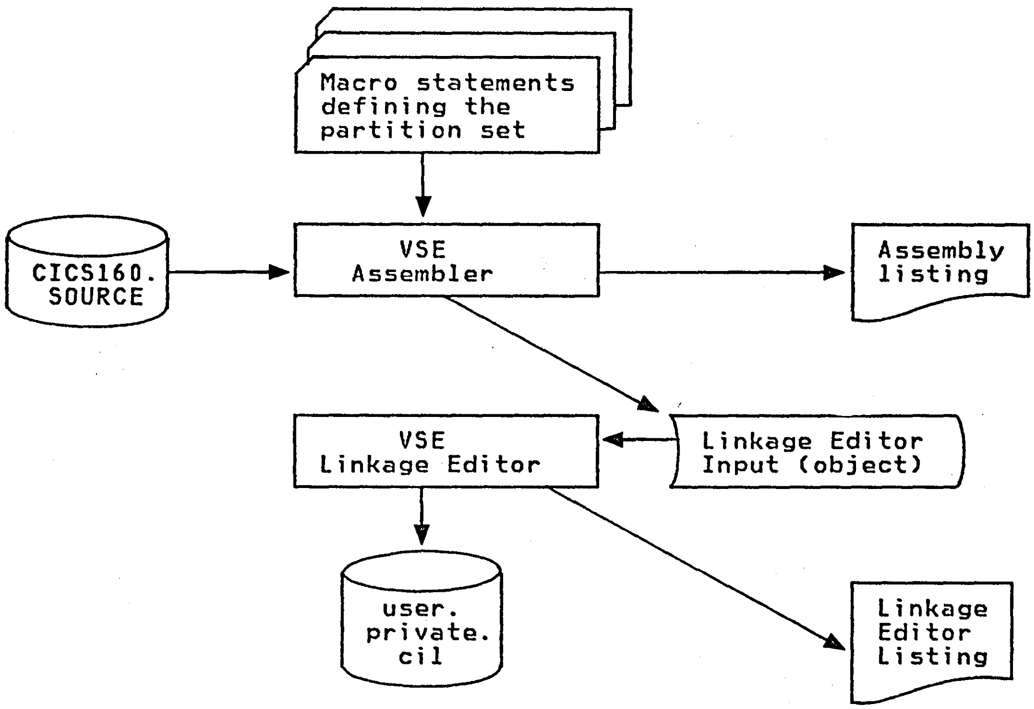


Figure 56. Installing Partition Sets

The following job stream is an example of the assembly and link-edit of partition sets:

```
// JOB BMSMAP
*  CICS/DOS/VS ASSEMBLE AND LINK-EDIT BMS PARTITION SET
// OPTION  CATAL,NODECK,ALIGN
   ACTION SMAP                                    (see note 1)
   PHASE   name,*                                 (see note 2)
// LIBDEF SL,SEARCH=cics-slb-filename
// LIBDEF CL,TO=user-cil-filename
// EXEC    ASSEMBLY,SIZE=nnK                      (see note 3)
        .
        source statements defining the partition set
        .
        END
/*
// EXEC    LNKEDT
/&
```

Figure 57. Assembling and Link-Editing a Partition Set

Notes:

1. ACTION SMAP may be used with VSE/Advanced Functions Version 1 Release 3 to produce a linkage editor map, sorted into alphameric order of CSECTs.

2. The PHASE statement must be used to specify the name of the partition set which BMS will load into storage. If the partition set is device-dependent, the PHASE name should be derived by appending the device suffix to the original 1- to 7-character partition set name used in the application program. The suffixes to be appended for the various terminals by CICS BMS depend on the parameter specified in the SUFFIX operand of the DFHPSD macro instruction that defined the partition set. See the CICS/VS Application Programmer's Reference Manual (Command Level) for a complete list of partition set suffixes.

3. A SIZE parameter value of up to 400K bytes will be required for the assembler under VSE.

Note that an entry in the processing program table is required for each partition set. This is defined using the DFHPPT macro instruction or the CEDA DEFINE PARTITIONSET command, as described in the CICS/VS Resource Definition Guide.

Chapter 3.4. Installing Application Programs

CICS application programs can be written in assembler language, ANS COBOL, PL/I, or RPG II, and can request CICS services in two ways:

1. Using the **command-level** interface:

   Assembler, COBOL, PL/I, and RPG II programs can request CICS services in the form of EXEC CICS commands. COBOL and PL/I programs can interface with DL/I with EXEC DLI commands. Each request is in the form of a single command; the application programmer need not be concerned with CICS control blocks or acquiring data areas and establishing addressability.

   For information about coding CICS application programs using EXEC CICS and EXEC DLI commands, see the CICS/VS Application Programmer's Reference Manual (Command Level) and the DL/I DOS/VS Application Programming: High Level Programming Interface or (for CICS and RQDLI commands) the CICS/VS Application Programmer's Reference Manual (RPG II).

   For information on the translator options that are available with CICS, see the CICS/VS Application Programmer's Reference Manual (Command Level).

2. Using the **macro-level** interface:

   Assembler, COBOL, and PL/I programs can include requests for CICS services in the form of CICS macro instructions. When using these macro instructions, the application programmer must be aware of certain control blocks used by the CICS management programs. Instructions to acquire data areas and establish addressability between these data areas and the CICS control blocks, must be included.

   For information about coding application programs using CICS macro instructions, see the CICS/VS Application Programmer's Reference Manual (Macro Level).

This chapter tells you how to prepare application programs written at each level, for execution under CICS.

COMMAND-LEVEL APPLICATION PROGRAMS

There are three steps in the preparation of application programs containing EXEC CICS commands:

1. Translation of the commands. Application programs written to use the command-level interface to CICS must be translated by the appropriate CICS command-level language translator before being assembled or compiled. The names of the translators supplied in the CICS core image library are:

Assembler	DFHEAP1$
COBOL	DFHECP1$
PL/I	DFHEPP1$
RPG II	DFHERP1$

   The command-level translators run in a VSE batch virtual partition size of 128K to 256K bytes, depending upon the size of the program to be translated.

   Intermediate storage for the translator output may be either tape or disk. 81-byte records are written out, but only 80 bytes are input to the following compiler or assembler.

   The options available for each of the translators are described in the CICS/VS Application Programmer's Reference Manual (Command Level).

   The DEBUG translator option is now assumed by default. This enables EDF to display the line number of each command. The line number requires 8 bytes of storage, so you might wish to suppress its production by specifying NODEBUG when your application has been fully tested.

2. Assembly or compilation.

3. Link-editing.

Figure 58 shows the steps required to install a command-level application program.

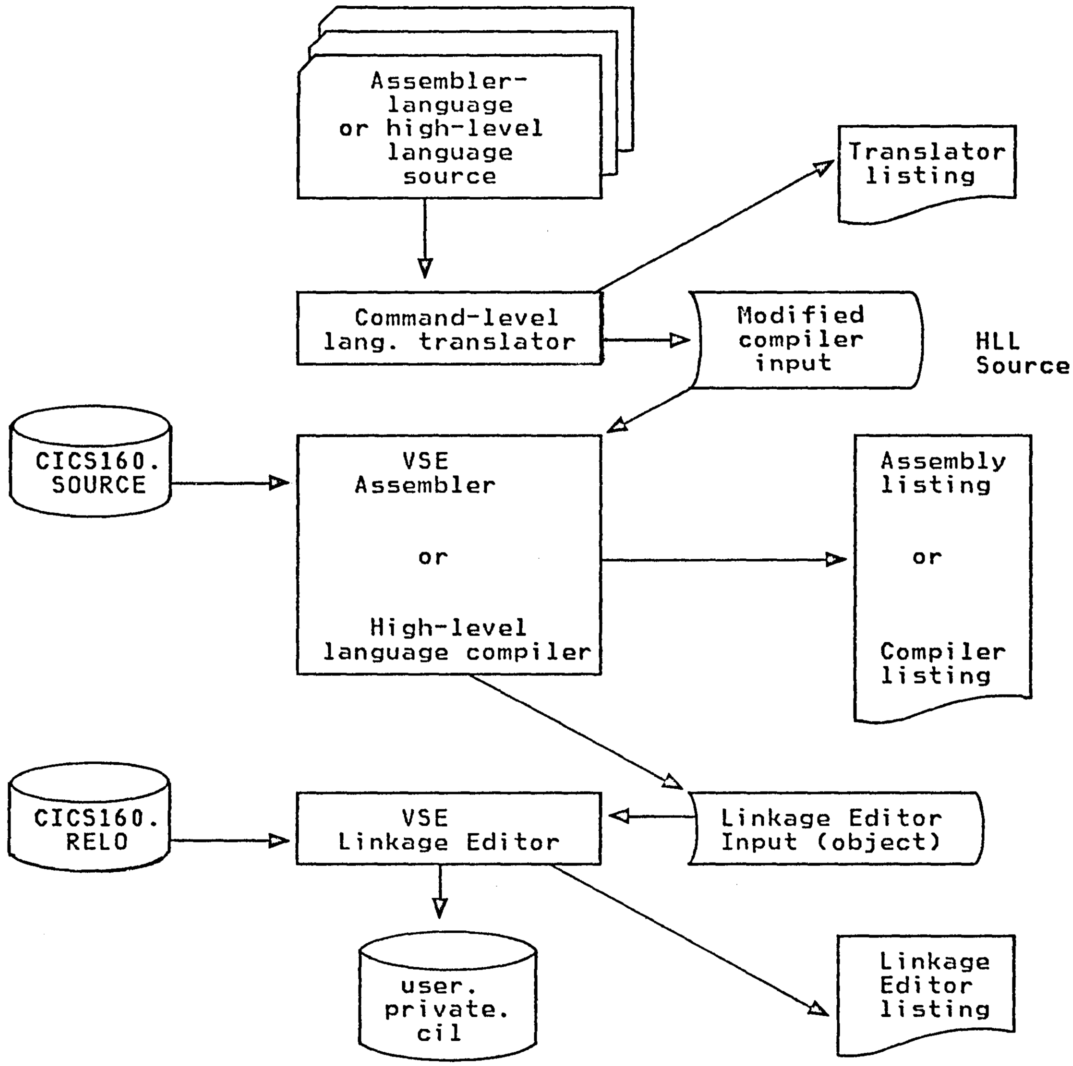


Figure 58. Installing a Command-Level Application Program

USING THE IBM-SUPPLIED SAMPLE PROCEDURES

To make it easier for the system programmer to install command-level application programs, the source code of sample procedures is supplied in the following books in the CICS source statement library:

Z.DFHEITAL for assembler language
Z.DFHEITCL for COBOL
Z.DFHEITPL for PL/I
Z.DFHEITRL for RPG II

These procedures are listed in Appendix F, and all use disk as intermediate storage for the translator output. If tape is to be used, you will need to either generate your own job stream, as described later under "Using Your Own Job Stream" on page 161, or modify the sample procedure itself.

Before putting the sample procedures into the procedure library, you must modify:

1. The extent information
2. The volume identifications.

Figure 59 through Figure 62 illustrate the use of the sample procedures.

Ignore weak external references unresolved by the linkage editor and their associated messages about unresolved address constants.

The following job stream can be used to invoke the DFHEITAL sample procedure for assembler-language application programs:

```
// JOB      jobname
// OPTION   NODECK,CATAL
   PHASE    phase-name,*
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     PROC=DFHEITAL
   .
   assembler-language source statements
   .
/*
/&
```

Figure 59. Invoking DFHEITAL Sample Procedure

The following job stream can be used to invoke the DFHEITCL sample procedure for COBOL application programs:

```
// JOB      jobname
// OPTION   NODECK,CATAL
   PHASE    phase-name,*
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     PROC=DFHEITCL
 CBL LIB
   .
   COBOL source statements
   .
/*
/&
```

Figure 60. Invoking DFHEITCL Sample Procedure

The following job stream can be used to invoke the DFHEITPL sample procedure for PL/I application programs:

```
// JOB      jobname
// OPTION   CATAL
   PHASE    phase-name,*
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     PROC=DFHEITPL
*PROCESS    INCLUDE;
   .
   PL/I source statements
   .
/*
/&
```

Figure 61. Invoking DFHEITPL Sample Procedure

The following job stream can be used to invoke the DFHEITRL sample procedure for RPG II application programs:

```
// JOB      jobname
// UPSI     00
// OPTION   CATAL
   PHASE    phase-name,*
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     PROC=DFHEITRL
   .
   RPG II source statements
   .
/*
/&
```

Figure 62. Invoking DFHEITRL Sample Procedure

USING YOUR OWN JOB STREAM

You may decide not to use the IBM-supplied sample procedures and produce your own job stream to install your command-level application programs.

Assembler-Language Application Programs

The following sample job stream can be used to install an assembler-language program, where **disk** is used as intermediate storage for translator output:

```
// JOB      jobname
// DLBL     IJSYSPH,'ASM.TRANSLATION',yy/ddd
// EXTENT   SYSPCH, balance of extent information
ASSGN  SYSPCH,DISK,VOL=volid,SHR
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHEAP1$
   .
   assembler-language source statements
   .
/*
CLOSE       SYSPCH,PUNCH
// DLBL     IJSYSIN,'ASM.TRANSLATION',yy/ddd
// EXTENT   SYSIPT
ASSGN  SYSIPT,DISK,VOL=volid,SHR
// OPTION   SYM,ERRS,NODECK,CATAL
   PHASE    phase-name,*
   INCLUDE  DFHEAI         (see notes 1 and 2)
// EXEC     ASSEMBLY
// EXEC     LNKEDT
/&
// JOB      RESET
CLOSE       SYSIPT,READER
/&
```

Figure 63. Preparing an Assembler-Language Application Program (Using Disk)

The following sample job stream can be used to install an assembler-language program, where **tape** is used as intermediate storage for translator output:

```
// JOB      jobname
// ASSGN    SYSPCH,cuu
// MTC      REW,SYSPCH
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHEAP1$
   .
   assembler-language source statements
   .
/*
// MTC      WTM,SYSPCH,2
// MTC      REW,SYSPCH
// RESET    SYSPCH
// ASSGN    SYSIPT,cuu
// OPTION   SYM,ERRS,NODECK,CATAL
   PHASE    phase-name,*
   INCLUDE  DFHEAI          (see notes 1 and 2)
// EXEC     ASSEMBLY
// EXEC     LNKEDT
/&
```

Figure 64. Preparing an Assembler-Language Application Program (Using Tape)

Notes:

1. An INCLUDE statement for DFHEAI must follow immediately after the PHASE statement and before the EXEC ASSEMBLY statement.

2. DFHEAI and DFHEAI0 must be in the relocatable library (they are supplied in the relocatable library on the CICS distribution tape).

COBOL Application Programs

The following sample job stream can be used to install a COBOL program, where **disk** is used as intermediate storage for translator output:

```
// JOB      jobname
// DLBL     IJSYSPH,'COBOL.TRANSLATION',yy/ddd
// EXTENT   SYSPCH,balance of extent information
ASSGN   SYSPCH,DISK,VOL=volid,SHR
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHECP1$
 CBL LIB            (see note 1)
   .
   COBOL source statements
   .
/*
CLOSE       SYSPCH,PUNCH
// DLBL     IJSYSIN,'COBOL.TRANSLATION',yy/ddd
// EXTENT   SYSIPT
ASSGN   SYSIPT,DISK,VOL=volid,SHR
// OPTION   SYM,ERRS,NODECK,CATAL
   PHASE    phase-name,*
   INCLUDE  DFHECI         (see notes 2 and 3)
// EXEC     FCOBOL
// EXEC     LNKEDT
/&
// JOB      RESET
CLOSE       SYSIPT,SYSRDR
/&
```

Figure 65. Preparing a COBOL Application Program (Using Disk)

The following sample job stream can be used to install a COBOL program, where **tape** is used as intermediate storage for translator output:

```
// JOB      jobname
// ASSGN    SYSPCH,cuu
// MTC      REW,SYSPCH
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHECP1$
 CBL LIB       (see note 1)
   .
   COBOL source statements
   .
/*
// MTC      WTM,SYSPCH,2
// MTC      REW,SYSPCH
// RESET    SYSPCH
// ASSGN    SYSIPT,cuu
// OPTION   SYM,ERRS,NODECK,CATAL
   PHASE    phase-name,*
   INCLUDE  DFHECI     (see notes 2 and 3)
// EXEC     FCOBOL
// EXEC     LNKEDT
/&
```

Figure 66. Preparing a COBOL Application Program (Using Tape)

__Notes:__

1. The LIB option must be specified for the compilation step so that any COPY statements in the source (for example, symbolic description maps) can be processed correctly. The COBOL translator in CICS/DOS/VS 1.6 does not generate COPY statements, but source generated by earlier versions of the translator may do so.

2. An INCLUDE statement for DFHECI must follow immediately after the PHASE statement and before the EXEC FCOBOL statement.

3. DFHECI must be in the relocatable library (this module is supplied in the relocatable library on the CICS distribution tape).

PL/I Application Programs

The following sample job stream can be used to install a PL/I program, where **disk** is used as intermediate storage for translator output:

```
// JOB      jobname
// DLBL     IJSYSPH,'PL/I.TRANSLATION',yy/ddd
// EXTENT   SYSPCH,balance of extent information
ASSGN   SYSPCH,DISK,VOL=volid,SHR
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,SEARCH=cics-cil-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHEPP1$
*PROCESS    INCLUDE;           (see note 1)
   .
   PL/I source statements
   .
/*
CLOSE       SYSPCH,PUNCH
// DLBL     IJSYSIN,'PL/I.TRANSLATION',yy/ddd
// EXTENT   SYSIPT
ASSGN   SYSIPT,DISK,VOL=volid,SHR
// OPTION   CATAL
   PHASE    phase-name,*
   INCLUDE  DFHPL1I            (see notes 2 and 3)
// EXEC     PLIOPT
// EXEC     LNKEDT
/&
// JOB      RESET
CLOSE       SYSIPT,SYSRDR
/&
```

Figure 67. Preparing a PL/I Application Program (Using Disk)

The following sample job stream can be used to install a PL/I program, where tape is used as intermediate storage for translator output:

```
// JOB      jobname
// ASSGN    SYSPCH,cuu
// MTC      REW,SYSPCH
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,SEARCH=cics-cil-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHEPP1$
*PROCESS    INCLUDE;              (see note 1)
      .
   PL/I source statements
      .
/*
// MTC      WTM,SYSPCH,2
// MTC      REW,SYSPCH
// RESET    SYSPCH
// ASSGN    SYSIPT,cuu
// OPTION   CATAL
   PHASE    phase-name,*
   INCLUDE  DFHPL1I            (see notes 2 and 3)
// EXEC     PLIOPT
// EXEC     LNKEDT
/&
```

Figure 68. Preparing a PL/I Application Program (Using Tape)

Notes:

1. The INCLUDE or MACRO option must be specified for the compilation step so that any %INCLUDE statements in the source (for example, symbolic description maps) can be processed correctly. The PL/I translator in CICS/DOS/VS 1.6 does not generate %INCLUDE statements, but source generated by earlier versions of the translator may do so.

2. An INCLUDE statement for DFHPL1I must follow immediately after the PHASE statement and before the EXEC PLIOPT statement.

3. Both DFHPL1I and DFHEPI must be in the relocatable library. DFHPL1I is supplied by PL/I. DFHEPI is supplied in the relocatable library on the CICS distribution tape.

4. The CICS translator no longer generates PL/I CALLs that prompt messages from the PL/I compiler warning of too short argument lists.

5. Weak external references unresolved by the linkage editor and their associated messages about unresolved address constants should be ignored.

The 19 declarations of the CICS entry point names are no longer required or generated by the translator. This change affects existing application programs only if they consist of included segments that do not start with valid PROCEDURE statements. If such a program is retranslated using the CICS/DOS/VS 1.6 translator, each included segment must also be retranslated. This is because previously-translated included segments will refer to the 19 entry point names that were previously declared in the outer-level procedure.

You do not have to take any action unless you retranslate the outer-level procedure.

RPG II Application Programs

The output from the RPG II translator may be directed to SYSPCH, SYS002, or SYS003, using the // UPSI job control statement. The advantage of directing the translator output to SYS002 or SYS003 is that it can be used directly as input to the following compiler (RPGII or RPGAUTO); there is no need to code job control statements to assign SYSIPT.

An INCLUDE statement for DFHERI must follow immediately after the PHASE statement and before the EXEC RPGII statement. DFHERI must be in the relocatable library (this module is supplied in the relocatable library on the CICS distribution tape).

RPG II translator work files are supported in VSAM data space managed by the VSE/VSAM space management for SAM feature.

Output on SYS002

If the leftmost two bits of the UPSI byte are set to 01, translator output is directed to SYS002. This method should be used if a translator run is to be followed immediately by an RPGII compilation. The compiler will then read its input from SYS002.

The following sample job stream can be used to install an RPG II program, where SYS002 is used as intermediate disk storage for translator output:

```
// JOB     jobname
// UPSI    01
// LIBDEF  SL,SEARCH=cics-slb-filename
// LIBDEF  RL,SEARCH=cics-rlb-filename
// LIBDEF  CL,SEARCH=cics-cil-filename
// LIBDEF  CL,TO=user-cil-filename
// EXEC    DFHERP1$
   .
   .
   RPG II source statements
   .
   .
/*
// OPTION  CATAL
   PHASE   phase-name,*
   INCLUDE DFHERI
// EXEC    RPGII
// EXEC    LNKEDT
/&
```

Figure 69. Preparing an RPG II Application Program (Using SYS002)

Output on SYS003

If the leftmost two bits of the UPSI byte are set to 10, translator output is directed to SYS003. This method should be used if a translator run is to be followed immediately by an Auto Report compilation. The compiler will then read its input directly from SYS003.

The following sample job stream can be used to install an RPG II program, where SYS003 is used as intermediate disk storage for translator output:

```
// JOB      jobname
// UPSI     10
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,SEARCH=cics-cil-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHERP1$
   .
   .
   RPG II source statements
   .
   .
/*
// OPTION   CATAL
   PHASE    phase-name,*
   INCLUDE DFHERI
// EXEC     RPGIAUTO
// EXEC     LNKEDT
/&
```

Figure 70. Preparing an RPG II Application Program (Using SYS003)

Output on SYSPCH

If the leftmost two bits of the UPSI byte are set to 00, or no UPSI statement is coded, translator output will be directed to SYSPCH.

SYSPCH can be directed to disk or tape. The following sample job stream can be used to install an RPG II program, where SYSPCH is used as intermediate **disk** storage for translator output:

```
// JOB     jobname
// UPSI    00
// DLBL    IJSYSPH,'RPG.II.TRANSLATION',yy/ddd
// EXTENT  SYSPCH,balance of extent information
ASSGN  SYSPCH,DISK,VOL=volid,SHR
// LIBDEF  SL,SEARCH=cics-slb-filename
// LIBDEF  RL,SEARCH=cics-rlb-filename
// LIBDEF  CL,SEARCH=cics-cil-filename
// LIBDEF  CL,TO=user-cil-filename
// EXEC    DFHERP1$
   .
   .
   RPG II source statements
   .
   .
/*
CLOSE      SYSPCH,PUNCH
// DLBL    IJSYSIN,'RPG.II.TRANSLATION',yy/ddd
// EXTENT  SYSIPT
ASSGN  SYSIPT,DISK,VOL=volid,SHR
// OPTION  CATAL
   PHASE   phase-name,*
   INCLUDE DFHERI
// EXEC    RPGII
// EXEC    LNKEDT
/&
// JOB     RESET
CLOSE      SYSIPT,SYSRDR
/&
```

Figure 71. Preparing an RPG II Application Program (Using SYSPCH on Disk)

The following sample job stream can be used to install an RPG II program, where SYSPCH is used as intermediate **tape** storage for translator output:

```
// JOB      jobname
// UPSI     00
// ASSGN    SYSPCH,cuu
// MTC      REW,SYSPCH
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,SEARCH=cics-cil-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHERP1$
   .
   .
   RPG II source statements
   .
   .
/*
// MTC      WTM,SYSPCH,2
// MTC      REW,SYSPCH
// RESET    SYSPCH
// ASSGN    SYSIPT,cuu
// OPTION   CATAL
   PHASE    phase-name,*
   INCLUDE  DFHERI
// EXEC     RPGII
// EXEC     LNKEDT
/&
```

Figure 72. Preparing an RPG II Application Program (Using SYSPCH on Tape)

MACRO-LEVEL APPLICATION PROGRAMS

After macro-level application programs have been coded, they must be cataloged in the core image library. The procedures for preparing and cataloging programs differ according to the programming language.

ASSEMBLER LANGUAGE

Assembler-language programs must be assembled, link-edited and cataloged in the core image library. The procedure is very similar to that used for CICS control tables, illustrated in Figure 49. The use of the relocatable library depends on whether other modules are to be link-edited with the program.

The job control statements for assembling, link-editing, and cataloging assembler-language application programs are as follows:

```
// JOB CICSASMC
*  CICS/DOS/VS ASSEMBLE AND CATALOG
// OPTION  CATAL,NODECK,ALIGN
   PHASE   name,*
// LIBDEF  SL,SEARCH=cics-slb-filename
// LIBDEF  RL,SEARCH=cics-rlb-filename
// LIBDEF  CL,SEARCH=cics-cil-filename
// LIBDEF  CL,TO=user-cil-filename
// EXEC    ASSEMBLY
   .
   source statements
   .
/*
// EXEC    LNKEDT
/&
```

If any application program contains BMS input map definitions (that is, it issues the DFHMSD and/or the DFHMDI macro with MODE=IN or MODE=OUT) and halfword-aligned length fields are required, it is necessary to specify SYSPARM='A' in the // OPTION job control statement, as follows:

```
// OPTION CATAL,NODECK,ALIGN,SYSPARM='A'
```

HIGH-LEVEL LANGUAGES

Application programs written in COBOL or PL/I that include CICS macro instructions are prepared for assembly by the CICS/DOS/VS macro-level language preprocessor (DFHPRPR). They are then assembled to convert the CICS macro instructions into COBOL or PL/I CALL statements.

The input to the preprocessor should be in 80-byte card-image format. If a stacker select character precedes each input line, it should be stripped off using the "copy and strip" function of the preprocessor, as shown in STEP 3 of Figure 74.

The standard procedure for COBOL or PL/I compilation, link-editing, and cataloging into the core image library can then be used. Figure 73 shows the complete process of installing a macro-level application program written in COBOL or PL/I:

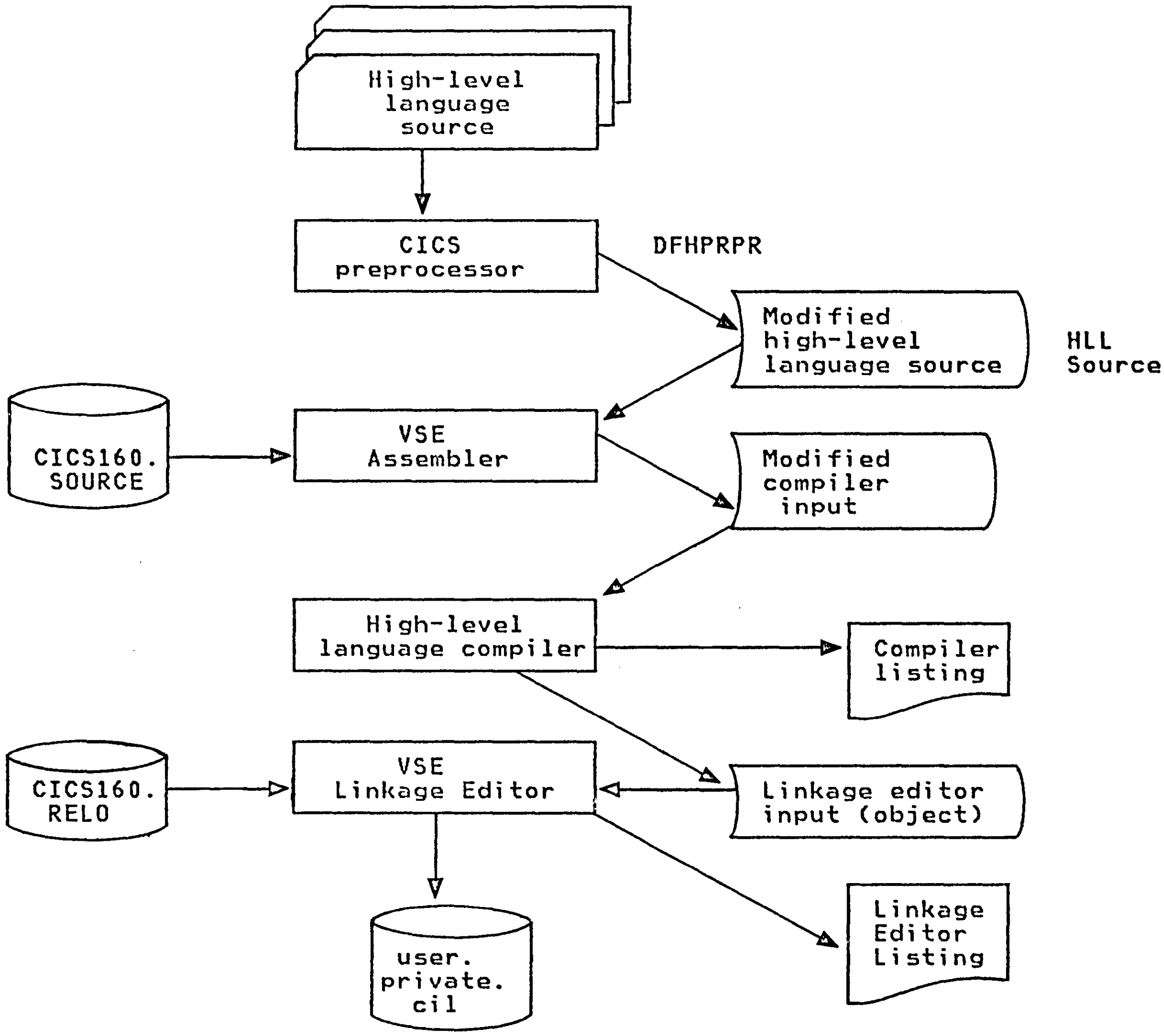


Figure 73. Installing a Macro-Level Application Program Written in COBOL or PL/I

__Notes:__

1. Any errors found during COBOL or PL/I macro expansions will cause a statement to be generated that will result in an 'invalid syntax' message at compilation time.

2. When COBOL application programs are link-edited, an unresolved external reference (DFHCBLI) is listed. Because the address constant associated with this symbol is resolved by the CICS program control program during execution of CICS, this linkage editor diagnostic message may be ignored.

3. When PL/I application programs are link-edited, several unresolved external references may appear. Among these are PLITABS, IBMBSXCA, IBMBSXCB, IBMBSIST, and SYSPRINT.

4. An INCLUDE card for DFHPL1I must exist for PL/I programs immediately after the phase name and before the EXEC LNKEDT. The link-edited output should be checked to make sure that DFHPL1N is the entry point before execution. DFHPL1N must be the first entry point and DFHIBM must be the first CSECT name.

Use of the VSE UPSI Byte

The CICS/DOS/VS macro-level language preprocessor uses the VSE UPSI byte to determine:

1. **The requested function** (bit 0)

0xxxxxxx	Preprocessor
1xxxxxxx	Copy and strip first byte of assembler output

   Note: Bit 1 is not used

2. **The language being processed** (bits 2 and 3)

xx00xxxx	ANS COBOL Compiler (not Subset) and DOS/VS COBOL Compiler
xx1xxxxx	ANS COBOL Subset Compiler
xxx1xxxx	PL/I Optimizing Compiler

3. **The intermediate storage used** (bits 4, 5, 6, and 7)

xxxx0000	Output on SYSPCH (not valid for copy function)
xxxx0001	2314 DASD
xxxx0010	3340 DASD
xxxx0011	3350 DASD
xxxx0100	Tape (no label)
xxxx1000	3330 DASD

The file names or device addresses associated with bits 4 through 7 are:

I/O Device	**Output**	**Input** (copy functions only)
2314 file names:	DFHPR14	DFHPR1I
3330 file names:	DFHPR30	DFHPR3I
3340 file names:	DFHPR40	DFHPR4I
3350 file names:	DFHPR50	DFHPR5I
Tape device addresses:	SYS012 SYS005 (for copy function only)	SYS004

For the copy and strip first byte function, both input and output must be on the same type of device.

Intermediate storage may reside on an FBA disk device. Any UPSI bit setting for a disk can be used, but the EXTENT statement "balance of extent information" must be appropriate to FBA devices (that is, in physical blocks). For intermediate storage on other DASD devices, specify 3330.

```
// JOB CONVERT
// UPSI     00000100                      USE TAPE OUTPUT
// ASSGN    SYS012,cuu    (tape 1)    PREPROCESSOR TAPE OUTPUT
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   RL,SEARCH=cics-rlb-filename
// LIBDEF   CL,SEARCH=cics-cil-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHPRPR          STEP 1
   .
   COBOL source
   .
/*
// RESET    SYS012
// ASSGN    SYSIPT,cuu    (tape 1)    ASSEMBLER INPUT
// ASSGN    SYSPCH,cuu    (tape 2)    ASSEMBLER OUTPUT
// EXEC     ASSEMBLY         STEP 2
/*
// MTC      WTM,SYSPCH,2
// MTC      REW,SYSPCH
// MTC      REW,SYSIPT
// RESET    SYSPCH
// RESET    SYSIPT
// ASSGN    SYS004,cuu    (tape 2)    (output from step 2)
// ASSGN    SYS005,cuu    (tape 1)
// UPSI     10000100
// EXEC     DFHPRPR          STEP 3
/*
// RESET    SYS005
// RESET    SYS004
// ASSGN    SYSIPT,cuu    (tape 1)    (output from step 3)
// OPTION   LISTX,SYM,ERRS,CATAL
   PHASE    COBSAMP,*
// EXEC     FCOBOL           STEP 4
// EXEC     LNKEDT           STEP 5
/&
```

Figure 74. Example in which Tape is Used as the Output from the Preprocessor and Input to the Assembler

To apply the example of Figure 74 to PL/I, a // UPSI 000101 statement must precede the first // EXEC DFHPRPR STEP 1 statement. It is also necessary to change the last four statements to the following:

```
// OPTION  CATAL
   PHASE   PL1SAMP1,*
   INCLUDE DFHPL1I
// EXEC    PLIOPT
// EXEC    LNKEDT
/&
```

Note: The DFHPL1I module is now supplied only by PL/I.

The following example is applicable only if VSE has been generated to allow system data sets to be on disk. The preprocessor writes its output on disk and this data set is used as input to the assembler.

```
// JOB CONVERT
// UPSI     00000011                 USE 3350 OUTPUT
// ASSGN    SYS010,cuu               3350 ASSIGNMENT
// DLBL     DFHPR14,'PREPROCESSOR.OUTPUT',yy/ddd    DFHPRPR OUTPUT
// EXTENT   SYS010,balance of extent information
// LIBDEF   SL,SEARCH=cics-slb-filename
// LIBDEF   CL,SEARCH=cics-cil-filename
// LIBDEF   CL,TO=user-cil-filename
// EXEC     DFHPRPR

   COBOL source

/*
// RESET    SYS010
// DLBL     IJSYSIN,'PREPROCESSOR.OUTPUT'   SYSIPT LABEL INFO
// EXTENT   SYSIPT
   ASSGN    SYSIPT,cuu               ASSIGN SYSIPT TO DISK
// EXEC     ASSEMBLY
/*
   CLOSE    SYSIPT,cuu               CLOSE SYSIPT AND RESET ASSIGNMENT
/&
```

Figure 75. Preparation of a Macro-Level COBOL or PL/I Program

Part 4. Setting Up CICS Data Sets

PART 4

Chapter 4.1. Preparing to Set Up CICS Data Sets

For execution, CICS requires certain data sets to have been defined. Some of these data sets are mandatory, others are required only if a corresponding facility is being used. In addition, data set definitions must be provided for all user files, DL/I data bases, and terminals other than VTAM terminals.

The following list gives the data sets required for each CICS facility:

Facility	Data Set Name	Chapter
Automatic statistics	Automatic statistics data set(s)	4.7
Auxiliary trace	Auxiliary trace data set(s)	4.8
DL/I access	DL/I definition statements	4.13
Dump	Dump data set(s)	4.9
Journaling	Journal data set(s) (these can be either the system log or user journals)	4.5
Recovery/restart	Restart data set	4.6
Resource definition online (RDO)	CICS system definition (CSD) file	4.10
Temporary storage	Temporary storage data set	4.2
Terminals (TCT entries)	Terminal definition statements	4.12
Transient data	Intrapartition data set and Extrapartition data set(s)	4.3 4.4
User files (FCT entries)	User file definition statements	4.11

PART 4

The rest of this part consists of a set of chapters, one for each of the facilities listed above. Each chapter describes the facility, its function and usage, and the data set(s) required to implement it on a running CICS system. If the data set(s) need to be preformatted, jobs that might be used for this purpose are shown.

Space calculations to enable you to determine the amount of space to be allocated to the data set(s) are provided, as are the data definition statements for defining them to the running CICS system.

Finally, if CICS utility programs are provided for postprocessing of the data set(s), they are described together with the syntax of any parameters that may be required.

Before you start setting up your CICS data sets, you should review the CICS programs you need, and their data set requirements.

You then have to:

- Set up the data sets and libraries that will be used by the CICS programs during execution.
- If necessary, initialize or preformat the data sets for use during CICS execution.
- Include the job control statements for the data sets in the job stream that starts CICS (see "A Typical Startup Job Stream" on page 253).

This part contains information about the file labels, space allocations, and assignments required by the data sets used by CICS. It describes the DLBL, EXTENT, and ASSGN statements required for each of these data sets.

The following must always be specified:

- CICS private core image library
- User files (see Chapter 4.11)
- BTAM, SAM, and sequential terminal devices listed in the terminal control table (see Chapter 4.12).

The following data set label information is optional. It must be specified if the related CICS facility is used:

- Transient data intrapartition and extrapartition data sets
- Temporary storage data set
- Dump data sets
- Journal data sets
- System log data sets
- Restart data set
- Automatic statistics data sets
- Auxiliary trace data sets
- CICS system definition file.

VSE/VSAM SPACE MANAGEMENT FOR SAM FEATURE

The following data sets are supported in VSAM data space managed by the VSE/VSAM Space Management for SAM feature of VSE:

- Automatic statistics data sets
- Transient data extrapartition data sets
- RPG II translator work files.

The following data sets are not supported in VSAM data space managed by the VSE/VSAM Space Management for SAM feature of VSE:

- Dump data sets
- Journal data sets
- Auxiliary trace data sets.

Chapter 4.2. Temporary Storage Data Set

The CICS temporary storage facility provides the application programmer with a means for the convenient saving of data for subsequent retrieval. The saving and retrieval may be performed by the same transaction or different transactions, the data being identified by a name, up to eight characters in length, assigned at the time it is first created. The data may be composed of a single record, or of a queue of records. At command-level, the records can be retrieved sequentially, or randomly by number. These single records or queues remain in existence until explicitly deleted. (Individual records in a temporary storage queue cannot be deleted.)

Data in temporary storage tends to be short-lived, emphasis being placed on ease of use of storing and retrieval. The data may be stored either in main storage or in VSAM-managed auxiliary storage. The type of storage used is determined by means of an option on the temporary storage write request at the time the temporary storage queue (or single record) is first created. Subsequent references to the queue or single record do not require the type of storage in which it resides to be specified.

Use of auxiliary storage requires the temporary storage program (DFHTSP) to have been generated to support auxiliary storage (the DFHTSP2$ management module in the pregenerated system supplies this function). Auxiliary storage should be used if it is required to preserve temporary storage records from one CICS run to another or if the temporary storage data are to be defined as recoverable. (The DFHTSP3$ management module in the pregenerated system supports recoverable auxiliary temporary storage.) No attempt is made to recover data in main storage.

PART 4

If DFHTSP has been generated without auxiliary storage support, auxiliary temporary storage requests do not fail but are automatically changed to use main storage.

Temporary storage requests are made by CICS application programs using the command- or macro-level interface. In addition, temporary storage is used internally by CICS for the following purposes:

- BMS paging requests
- BMS ROUTE requests
- Interval control START commands with the FROM option or with the PROTECT option
- Execution diagnostic facility (EDF) displays and control areas
- 3600 message caches following emergency restart
- Dynamic log overflow.

The first five of these functions make temporary storage requests to auxiliary storage. In the case of dynamic log overflow, you can choose explicitly whether to use main or auxiliary storage. The DFHJCP1$ management module in the pregenerated system provides main storage support; DFHJCP2$ provides auxiliary storage support.

For information about the use of temporary storage by Time Management and by Basic Mapping Support, see either the <u>CICS/VS Application Programmer's Reference Manual (Macro Level)</u> or the <u>CICS/VS Application Programmer's Reference Manual (Command Level)</u>.

Data on a temporary storage queue whose name is known can be viewed at a 3270 screen by means of the CEBR transaction. For further details of the CEBR transaction, see the <u>CICS/VS Operator's Guide</u>.

Temporary storage queues or temporary storage logical records can be defined as recoverable by including their names (DATAIDs) in the temporary storage table (TST). The TST also allows a set of temporary storage DATAIDs whose names begin with a particular set of characters to be defined as recoverable. See the CICS/VS Resource Definition Guide for a description of how to code the TST.

Auxiliary temporary storage is defined as a VSAM data set within VSAM space. CICS uses control interval processing when storing or retrieving temporary storage records in this data set. A control interval will, in general, contain a number of temporary storage records, the space occupied by each record being rounded up to a multiple of 64 or of 128, depending on whether the control interval size has been defined as less than or greater than 16384 bytes.

Temporary storage space within a control interval is reusable. If a transaction requests temporary storage and there is not sufficient space to satisfy the request, the transaction is suspended until the space becomes available. (If, however, an EXEC CICS HANDLE CONDITION NOSPACE statement was coded, control is returned to the transaction.)

JOB CONTROL STATEMENTS TO DEFINE THE TEMPORARY STORAGE DATA SET

After VSAM data space has been defined, you must define the temporary storage data set within that data space. This is an example of the coding required to define a temporary storage data set within an existing VSAM data space:

```
// JOB CRATS CREATE AUXILIARY TEMPORARY STORAGE DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                              -
          (NAME(cics160.temp.storage)                        -
           RECORDSIZE(4089,4089)                             -
           RECORDS(144)                                      -
           NONINDEXED                                        -
           CONTROLINTERVALSIZE(4096)                         -
           SHAREOPTIONS(2)                                   -
           VOLUMES(volume))                                  -
        DATA                                                 -
          (NAME(cics160.temp.storage.data))                  -
        CATALOG(user.catalog)
/*
/&
                      (This job is included in Z.DFHSPJCL)
```

You specify RECSZ and number of RECORDS required when defining the temporary storage data set. VSAM uses this information and allocates sufficient disk space to accommodate at least this number of RECORDS. The secondary allocation subparameter of the RECORDS parameter should not be specified. This is because CICS initialization will always attempt to acquire a secondary allocation, if it has been defined, thus increasing the size of the temporary storage data set on each run.

SPACE CONSIDERATIONS

The amount of space allocated to temporary storage is expressed in terms of two values that you must specify:

- The control interval size
- The number of control intervals in the data set.

THE CONTROL INTERVAL SIZE

The control interval size is specified by means of the CONTROLINTERVALSIZE parameter in the VSAM CLUSTER definition.

The control interval size should be chosen so that any normally occurring temporary storage record can be fitted within a control interval. If, infrequently, larger records are written, they will be accommodated by being split across control intervals, but the performance of the operation will be degraded. The following factors should be considered when making this choice:

- Each temporary storage record requires:

  Data length plus
  20 bytes (for the temporary storage header) plus
  32 bytes if the storage is defined as recoverable and is requested by a EXEC CICS START TRANSID (...) FROM (...) request.

  If Basic Mapping Support is installed with 3270 support, the data length of the record is at least as large as the 3270 buffer size (for 3270 terminals with the alternate screen size facility, the data length is the larger of the two sizes).

  The total number of bytes allocated for a temporary storage record is rounded up to a multiple of 64 (for control interval sizes ≤16384), or a multiple of 128 (for control interval sizes >16384).

- The control interval size should be such that, in addition to holding at least one (rounded up) temporary storage record, it allows for 64 bytes of VSAM control information for control interval sizes ≤16384, or 128 bytes of control information for control interval sizes >16384.

  You should choose the control interval size to be such that all normally occurring temporary storage records, together with the VSAM control information, can be fitted into a control interval. In particular, remember to accommodate the largest dynamic log record that might be written to temporary storage. (Note that this record can also be bigger than the dynamic buffer size.)

  The control interval size should be a multiple of 512 bytes for control interval sizes ≤8192, or a multiple of 2048 bytes for control interval sizes > 8192.

For example, if BMS is used to write a 24 x 80 character screen to temporary storage, the data length written occupies up to approximately 1920 bytes. If the temporary storage queue is defined to be recoverable, a further 32 bytes are required, and a further 20 bytes are required for the CICS temporary storage header, giving a total of 1972 bytes. This must be rounded up to a multiple of 64, resulting in 1984 bytes. Finally, a further 64 bytes of VSAM control information must be added, giving a control interval size of 2048 bytes. Typically, the control interval size will be larger than this, to accommodate several records possibly differing in size.

NUMBER OF CONTROL INTERVALS

The number of control intervals in the temporary storage data set is specified indirectly by means of the RECORDS parameter in the VSAM CLUSTER definition. (Note that the RECORDS and RECSZ parameters do not correspond to the temporary storage records as seen at the CICS temporary storage interface.) The two values of the RECSZ parameter must be coded to be the same, and must be seven bytes less than the CONTROLINTERVALSIZE. A consequence of this is that the specified number of VSAM RECORDS is the same as the number of control intervals available to temporary storage management.

The number of control intervals to be allocated depends on user and system requirements for temporary storage. Remember to consider space for dynamic transaction backout, especially the dynamic logging of DL/I data base changes, if this facility has been generated.

VSAM will allocate sufficient tracks to accommodate the required control intervals. Because a 3330 track holds three 4096-byte blocks, in the example above, 48 tracks will be allocated to satisfy the request for 144 control intervals.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

The filename applicable to the temporary storage data set is DFHTEMP. A job control statement is required, of the form:

```
// DLBL DFHTEMP,'cics160.temp.storage',,VSAM,CAT=user-catalog-filename
```

Chapter 4.3. Transient Data Intrapartition Data Set

The transient data intrapartition data set may be either a VSAM entry-sequenced data set (ESDS) or a DAM data set, and is used for queuing messages and data within the system. VSAM must be used if the intrapartition data set is to reside on a fixed-block architecture (FBA) disk device.

The data set control block for the intrapartition data set is automatically generated in the destination control table, and is opened by system initialization if intrapartition queues are being used.

The following suffixed pregenerated modules of the transient data program are supplied:

TDP= ___
in SIT

Suffix	Function
1$	Extrapartition only
2$	Extrapartition + DAM intrapartition with automatic transaction initiation (ATI), but without recovery
4$	Extrapartition + DAM intrapartition with ATI and recovery
6$	Extrapartition + VSAM intrapartition with ATI, but without recovery
7$	Extrapartition + VSAM intrapartition with ATI, and recovery.

Only one of the supported access methods can be used for intrapartition transient data during a particular execution of CICS, because the choice between the two is made by selecting a particular transient data program when CICS is initialized.

System messages produced by CICS are commonly sent to the following transient data destinations:

DESTID	Use	Data
CSCS	Message log	Sign-on, sign-off messages
CSML	Message log	Operator statistics
CSMT	Master terminal	Master terminal messages
CSTL	Terminal log	Terminal error messages
CSDL	Audit trail	Resource definition online

See the description of the DFHDCT macro in the CICS/VS Resource Definition Guide for more details.

These destinations can be defined as extrapartition output, indirect, or intrapartition output processed by a user program.

Transient data can be written directly to a local 3270 printer using the transient data write to terminal aid (DFHTDWT$). See "Appendix E. Aids and Sample Application Programs" for details.

JOB CONTROL STATEMENTS TO DEFINE A VSAM INTRAPARTITION DATA SET

An example of the code that could be used to define a VSAM transient data intrapartition data set is given below. If space is allocated in records, rather than tracks or cylinders, RECORDSIZE is required, and should be seven bytes less than the CONTROLINTERVALSIZE. For full details, see the appropriate VSAM manuals.

```
// JOB CRITD CREATE INTRAPARTITION TRANSIENT DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                          -
         (NAME(cics160.intra.transd)                     -
          RECORDSIZE(1529,1529)                          -
          RECORDS(100)                                   -
          NONINDEXED                                     -
          CONTROLINTERVALSIZE(1536)                      -
          VOLUMES(volume))                               -
       DATA                                              -
         (NAME(cics160.intra.transd.data))               -
       CATALOG(user.catalog)
/*
/&
                        (This job is included in Z.DFHSPJCL)
```

SPACE CONSIDERATIONS

Space is allocated to queues in units of a control interval at a time. The first control interval is reserved for CICS use, the remaining control intervals being available to hold data. Data records are stored in control intervals according to VSAM standards.

The first record in each control interval is a 12-byte control record, created by CICS. At the end of each control interval is the VSAM control information, which is 4 bytes long, plus 3 bytes for each record in the control interval. The length field at the start of the CICS transient data output area is removed when the record is stored, the length being held in the VSAM control information. The largest data portion that can be stored is, therefore, (CISIZE minus 22) bytes.

A data record must be contained within a single control interval, and the CISIZE should be chosen so that the longest record will fit.

JOB CONTROL STATEMENTS TO DEFINE A DAM INTRAPARTITION DATA SET

The job control statements to define a DAM intrapartition data set might be:

```
// DLBL DFHNTRA,'cics160.intra.transd',,DA
// EXTENT SYSnnn,CICSDA,1,0,relative-track-address,number-of-tracks
// ASSGN SYSnnn,cuu
```

SPACE CONSIDERATIONS

Space is allocated to intrapartition queues in units of a track at a time. The first track is reserved for CICS use, the remaining tracks being available to store data. Data records are written as separate physical blocks.

The first record on each track is a control record, created by CICS. This, together with the block overhead, uses approximately 275 bytes of each track, leaving (BLKSIZE minus 275) bytes in which to store data.

A data record must be contained within a single track, and the block size should be chosen so that the longest record will fit.

BLKSIZE would normally be set equal to the track capacity of the device.

CONSIDERATIONS INDEPENDENT OF ACCESS METHOD

In this section, "unit" means track or control interval, according to the access method being used.

- Make the data set sufficiently large to avoid a NOSPACE condition. If a queue has the reusable queue space option (REUSE=YES in the destination control table entry), a unit allocated to the queue is released for reuse when all records stored in that unit have been read by EXEC CICS READQ TD requests or DFHTD TYPE=GET macro instructions. If all available units are currently allocated to queues, further EXEC CICS WRITEQ TD requests or DFHTD TYPE=PUT macro instructions will receive a NOSPACE response until units are released by GET requests. Whether or not the queue is reusable, space is also released by a EXEC CICS DELETEQ TD request or DFHTD TYPE=PURGE macro instruction.

- The intrapartition data set should contain not less than two units.

- If the asynchronous transaction processing (ATP) facility is to be used, intrapartition transient data is required.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

The filename for the intrapartition data set is DFHNTRA.

VSAM INTRAPARTITION DATA SET

For a CICS execution using VSAM for intrapartition transient data, a job control statement is required, of the form:

```
// DLBL DFHNTRA,'cics160.intra.transd',,VSAM,CAT=user-catalog-filename
```

DAM INTRAPARTITION DATA SET

For a CICS execution using DAM for intrapartition transient data, a job control statement is required, of the form:

```
// DLBL DFHNTRA,'cics160.intra.transd',,DA
// EXTENT SYSnnn,CICSDA,1,0,relative-track-address,number-of-tracks
// ASSGN SYSnnn,cuu
```

Chapter 4.4. Transient Data Extrapartition Data Sets

Transient data extrapartition data sets are sequential files, normally on disk or tape. Each data set may be defined for input only or for output only, but not for both. Output files can be assigned to SYSLST.

You must include job control statements for each entry representing an extrapartition data set in the destination control table. The data set may be on disk or tape. The filename must be the same as that specified in the DSCNAME operand of the DFHDCT TYPE=SDSCI macro instruction.

For an extrapartition data set on **disk**, if DFHDCT TYPE=SDSCI specifies DSCNAME=DFHXTI and DEVICE=DISK, the appropriate job control statements are:

```
// DLBL    DFHXTI,'cics160.extra.a',0,SD
// EXTENT SYSnnn,,1,0,relative-track-address,number-of-tracks
// ASSGN SYSnnn,cuu
```

For an extrapartition data set on **tape**, if DFHDCT TYPE=SDSCI specifies DEVICE=TAPE and DEVADDR=SYS009, the appropriate job control statement is:

```
// ASSGN   SYS009,cuu
```

Chapter 4.5. Journal Data Sets

If journaling activity is to take place while CICS is running, each journal must be defined by means of operating system job control statements. The data sets referred to in these statements correspond to the journal definitions as provided by the entries in the journal control table (JCT).

A particular instance of a journal is the CICS system log, used by CICS modules to perform automatic logging of changes to recoverable resources. During emergency restart after an uncontrolled or immediate CICS termination, the system log is then used as the source of information that enables the resources to be restored to a self-consistent state.

In addition, user journals can be defined to record file control or message activity during CICS execution for subsequent processing and analysis by user-written programs (for example, for forward recovery). This recording is performed either by automatic journaling as specified in entries in the file control table or program control table, or by explicit user journaling. Automatic journaling and user journaling can also be directed to the system log.

Each journal is given a journal file identification by means of the JFILEID parameter in an entry in the journal control table (JCT) (see "Journal Control Table" in the CICS/VS Resource Definition Guide). This can be specified as SYSTEM to denote the system log, or a value between 02 and 99 to identify a user journal.

Each journal can be defined as residing on disk or on tape. Disk journals can consist of a single data set or a pair of data sets that are used alternately. Tape journals may be defined to reside on a single tape drive or a pair of tape drives. Tape journals must be defined on unlabeled tapes.

The above journal characteristics are specified by means of the JTYPE parameter in the JCT entry. Corresponding job control statements must be included in the CICS job stream.

The filenames of the statements defining the journal data sets in the CICS run must be of the form DFHJnnx where 'nn' is the journal file identification and 'x' is A or B corresponding to which of the pair of data sets is being defined. If the journal consists of a single data set, 'x' has the value 'A'. If the journal is the system log, the value of 'nn' is '01'.

DISK JOURNALS

Disk journals are defined as sequential data sets that are reused for output when full. This is true whether a disk journal is defined as one or two data sets. In the latter case, however, reuse will not be immediate; there will be a data-set switch first. All disk journal volumes must remain mounted at all times.

Whenever a data set of a disk journal is opened and begins to be reused for output, message DFH4508 is issued. This message is also issued when a disk journal is returned to output status after being read as input by a CICS transaction.

The system console operator can control the reuse of disk data sets if the journal was specified with the PAUSE option (see "Journal Control Table" in the CICS/VS Resource Definition Guide). If PAUSE was specified, message DFH4507 is issued, requiring a reply from the operator, before the data set is reused. This gives the operator an opportunity to copy the data set before allowing it to be overwritten. Note, however, that any transactions that cause output to the journal to occur will be held up until the copy operation is complete and the operator has replied to the message. For this reason, you should define single data set journals so that they are large enough to hold all the journal output for a given CICS run.

Journal data sets are defined to VSE as having an undefined (U) record format. CICS constructs the records in the data sets into variable blocked (VB) format.

PREPARING DISK DATA SETS FOR SYSTEM LOGS AND JOURNALS (DFHJCJFP)

Disk data sets intended to receive journal data set output must be allocated and formatted prior to use in a CICS execution. After they are formatted, data sets are reused for successive CICS executions even across cold starts.

When opening all journals for output, CICS system initialization positions disk journals so that output resumes immediately after the last record previously written to the journal. Journals created by the CICS monitoring facility are not repositioned.

Disk space of at least three tracks (four control intervals in the case of FBA devices) must be preallocated for use of a journal as either a single contiguous data set or a pair of data sets, depending on the type of disk journal (see "Journal Control Table" in the CICS/VS Resource Definition Guide).

Each data set must be preformatted by running the DFHJCJFP utility program the first time the data set is used. Failure to do so will usually adversely affect the journal, causing message DFH4501 to be issued to the system console during system initialization, indicating that the journal failed to open. However, if an unformatted disk journal does open, the positioning and the continuity of the data will be unpredictable.

After the data set has been formatted, it should **never** be reformatted unless it is to be used for another purpose. Reformatting would destroy the contents of the log or journal on which the emergency restart procedures depend. If you reformat the system log, CICS can only perform either a warm start if a warm keypoint has been taken, or a cold start.

The job stream for the DFHJCJFP journal format program is included in "Job Stream for Defining and Formatting Journals on Disk" below.

After successfully formatting an extent, a message is issued to the system console indicating the number of tracks formatted:

```
DFH4599 - JOURNAL EXTENT INITIALIZED - nnnn TRACKS AVAILABLE
```

For an explanation of the messages referred to in this chapter, see the CICS/VS Messages and Codes manual.

Job Stream for Defining and Formatting Journals on Disk

As an example, if the system log is defined as two journal data sets on disk, the job control statements to format the first journal data set might be:

```
// JOB CICSFDSK FORMAT JOURNAL DISK EXTENT
// DLBL JOURNAL,'cics160.system.log.a',0,SD    (see notes 1 & 2)
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// EXEC DFHJCJFP,SIZE=AUTO
/*
/&
                    (This job is included in Z.DFHSPJCL)
```

You may need to add:

```
// LIBDEF CL,SEARCH=cics-cil-filename
```

before the EXEC statement.

The job control statements to format the second journal data set on disk might be:

```
// JOB CICSFDSK FORMAT SECOND JOURNAL DISK EXTENT
// DLBL JOURNAL,'cics160.system.log.b',0,SD    (see notes 1 & 2)
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// EXEC DFHJCJFP,SIZE=AUTO
/*
/&
                    (This job is included in Z.DFHSPJCL)
```

Notes:

1. You may need to add:

   ```
   // LIBDEF CL,SEARCH=cics-cil-filename
   ```

   before the EXEC statement in each of the above jobs.

2. For FBA devices, CISIZE can be specified on the DLBL statement. The CISIZE specified for the journal must be at least 7 bytes greater than the value in the BUFSIZE operand in DFHJCT TYPE=ENTRY. The CISIZE value must be either a multiple of 512 bytes ≤8192 bytes or a multiple of 2048 bytes >8192 bytes, and be equal to the CISIZE value specified for CICS execution.

3. The filename for the journal DLBL statement must be JOURNAL.

SPACE CONSIDERATIONS

If disk is chosen for the system log, sufficient space must be allocated. This means that, in order to provide the minimum information needed to restart the system, the system log must be large enough to hold, at any one time, at least one complete keypoint and, for each task that performs logging, all data logged from start of task (or logical unit of work) to the moment of failure. This is an extreme case. Practically, you should allocate enough space to hold at least two to three times the amount of data logged during the processing of the longest task. You should also specify the activity keypoint frequency so that at least two or three keypoints are taken while writing each system log data set.

Ideally, you should allocate sufficient space on the combined log data sets to avoid the necessity of switching more than once during a CICS run.

Variables that affect the space requirements for the system log include:

- Duration of the longest running task; that is, how many keypoints it spans
- The frequency at which keypoints are written
- Block size of the system log
- Number of blocks actually written; that is, those written by the DFHJC TYPE=PUT macro instruction
- Track capacity of the device on which the system log resides
- The "load" on the system log. This, in turn, depends on parameters chosen during CICS table preparation, such as recoverable Destination Control Table destinations and user journaling.

You will need to monitor the CICS statistics to get an optimum value for the space required. You should start with a large value, at least for the first disk extent.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

If journaling is activated for a CICS execution, you have to assign I/O devices and supply label information for each entry of the journal control table (JCT) selected. The filename in the DLBL statement must be of the form DFHJnnx, where nn is the journal data set identification for a particular JCT entry and x is A or B. For the CICS system log, the journal data set identification is 01.

Depending on the type of disk journal data set, there must be either one DLBL statement with filename DFHJnnA, or two DLBL statements with filenames DFHJnnA and DFHJnnB, for each JCT entry. The symbolic unit (SYSnnn) specified in the DEVADDR operand of the JCT must be the same as that for the ASSGN job control statement.

It is essential that any disk logs that you may want to use for forward recovery are on a different disk device from your recoverable data sets.

If the journal data sets on disk are formatted as above, the job control statements for the CICS execution would be:

```
// DLBL DFHJ01A,'cics160.system.log.a',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// DLBL DFHJ01B,'cics160.system.log.b',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
```

For FBA devices, the CISIZE parameter can be coded on the DLBL statements defining the journals. The CISIZE value specified must be at least 7 bytes greater than the corresponding BUFSIZE value in DFHJCT TYPE=ENTRY. The CISIZE value must be either a multiple of 512 bytes ≤8192 bytes or a multiple of 2048 bytes >8192 bytes), and be equal to the CISIZE value specified when running the journal format program (DFHJCJFP).

TAPE JOURNALS

If a journal is defined as residing on tape, it will in general consist of a series of physical reels or tape volumes that are used in sequence. The volumes are mounted on either one or two tape drives in accordance with the JTYPE parameter in the JCT. Because unlabeled tapes are used, it is necessary to label each tape used with an external (paper) label and to record separately the identity and sequence of tape volumes constituting the journal.

PREPARING TAPES FOR SYSTEM LOGS AND JOURNALS (DFHFTAP)

During emergency restart, the CICS recovery utility program (DFHRUP) must find the last record written in order to process the system log from the point at which it was positioned when the system abnormally terminated. If new tapes are used in system logs and an abnormal termination causes no end-of-file mark to be written, difficulties will occur in locating the last record written, such as the tape running off the end of the reel. To avoid this, tapes used in system logs or journals must be preformatted before being used. This is accomplished by using the CICS-provided DFHFTAP utility program.

Standard Procedure for Formatting Tapes for System Log or Journal Use

The job control statements to format a system log or journal data set on unlabeled tape might be:

```
// JOB CICSFTAP FORMAT JOURNAL TAPE
// ASSGN SYS010,cuu
// EXEC DFHFTAP
/*
/&
                          (This job is included in Z.DFHSPJCL)
```

You may need to add:

```
// LIBDEF CL,SEARCH=cics-cil-filename
```

before the EXEC statement in the above job.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

If journaling is activated for a CICS execution, you have to assign I/O devices for each entry of the journal control table (JCT) selected.

For example, if the system log has been specified as:

```
JTYPE=TAPE2,DEVADDR=(SYS004,SYS005)
```

the job control statements would be:

```
// ASSGN   SYS004,cuu
// ASSGN   SYS005,cuu
```

WRITING AN EOF MARK ON A JOURNAL OR SYSTEM LOG (DFHTEOF)

The CICS-provided tape end-of-file utility program (DFHTEOF) can be used offline to reposition and close correctly a user-journal tape after an uncontrolled shutdown. For the tape system log, this function is performed by CICS during an emergency restart.

The purpose of the DFHTEOF procedure is to write an end-of-file (EOF) mark following the last valid record written prior to the failure. It does this by reading the tape forward from load point and comparing the time stamps in each record.

If the system log or any other CICS-created journal residing on tape is to be used by DL/I or by a user-written program that does not include such comparison logic, an EOF mark must be written after the last valid record. This can be done by running DFHTEOF stand-alone. This step should be taken even when a partition ABEND caused the abnormal termination, in case the ABEND routine itself failed to close the tapes.

A sample job stream that might be used to run DFHTEOF stand-alone is:

```
// JOB CICSTEOF WRITE EOF ON JOURNAL TAPE
// ASSGN SYS010,cuu
// LIBDEF CL,SEARCH=cics-cil-filename
// EXEC DFHTEOF
/*
/&
```

Ensure that the correct volume (that is, the last volume that was mounted for system log output during the previous execution) is mounted and positioned at its load point.

The restart-time execution of DFHTEOF uses the DFHJ01A job control statement (that is, the statement that exists for the system log).

JOURNALS FOR CICS MONITORING FACILITY OUTPUT

Journals used for CICS monitoring facility output must be defined as user journals in the JCT (that is, with IDs from 02 to 99), and they must have FORMAT=SMF defined.

A sample journal control table is included in the CICS source statement library as DFHJCT1$ (with copy book DFHXJCTS).

Chapter 4.6. Restart Data Set

This data set is required for automatic or emergency restart. If you need this data set, you must also code KPP=D$ in the SIT or system initialization override parameters.

The restart data set is a VSAM key-sequenced data set (KSDS) used:

- At controlled shutdown, to hold warm keypoint information for a subsequent warm start.
- During system initialization, to hold temporarily the backout information generated in the course of emergency restart.

The use of the restart data set is closely associated with the start options that are selected during CICS initialization. If the START parameter, as defined in the system initialization table (SIT) or as an override during CICS startup, has the value AUTO, CICS inspects the contents of the restart data set to determine the type of startup to perform.

Three possibilities arise:

1. The restart data set has been newly defined and initialized (as described below) before the start of the CICS run. The control record in the restart data set is thereby set to indicate that this is the first use of the data set and that a cold start is required.

2. The previous CICS run was terminated normally by means of a CEMT PERFORM SHUTDOWN or CSMT SHUT,NO command. In this case, a warm keypoint is written to the restart data set during shutdown processing, and the control record in the restart data set is marked to indicate that the following auto startup of CICS should be a warm start, restoring CICS to the same state that it had at shutdown.

   No warm keypoint is written if the shutdown is immediate.

3. The previous CICS run terminated abnormally or after an operator request to perform immediate shutdown (CEMT PERFORM SHUTDOWN IMMEDIATE or CSMT SHUT,YES). In this case, no warm keypoint is taken during shutdown, and the restart data set contains a control record indicating that an emergency restart is required during the next automatic start.

   The emergency restart procedure uses the system log at the time of the failure to recover the CICS system to a self-consistent state. During this process, it uses the restart data set temporarily to store transaction backout data for tasks that were in progress at the time of failure. This data is written by the recovery utility program, and is read by the transaction backout program and, optionally, by a user-written program executed during the program list table (PLT) processing phase of system initialization.

As an alternative to allowing CICS to determine the start option automatically from the contents of the restart data set, it is possible to force cold start or emergency restart during initialization by explicitly coding these values on the START parameter in the SIT or system initialization override parameters.

If START=COLD is coded, CICS is initialized without any use being made of the restart data set, nor is reference made to any previous CICS execution.

If START=EMER is coded, emergency restart is carried out as described above, without reference to the control record in the restart data set, but using the restart data set temporarily to store transaction backout data.

If START=AUTO is coded as a startup parameter together with KPP=NO, CICS detects this condition, displays a message to the system operator, and forces a cold start.

At the end of initialization, the control record is written to the restart data set.

INFORMATION WRITTEN TO THE RESTART DATA SET

WARM RESTART

The restart information consists of the following tables and areas:

- Program control table (PCT)
- Processing program table (PPT)
- Terminal entries (nonswitched)
- File control table (FCT)
- Selected fields from the common system area (CSA)
- Destination control table (intrapartition entries)
- Transient data intrapartition space allocation bit map
- IDs and relative byte addresses for temporary storage auxiliary destinations/queues
- Temporary storage space allocation bit map
- Interval control elements (ICEs) and automatic initiate descriptors (AIDs) at system termination time
- Batch control areas (BCAs) and their write request elements (WREs) for ATP.

EMERGENCY RESTART

The information written to the restart data set in this case includes:

- The transaction backout table, which contains details of inflight transactions and completed tasks that contain user-written log records
- The file backout table, identifying files that need to be backed out
- The message backout table, identifying terminals that require resynchronizing
- The DL/I backout table, which contains details of DL/I PSBs that need to be backed out
- Individual file records to be used for backout
- Individual DL/I records to be used for backout
- Individual temporary storage replacement records to be used for backout
- Any user-written log records that are flagged to be collected by DFHRUP.

JOB CONTROL STATEMENTS TO DEFINE AND INITIALIZE THE RESTART DATA SET

Before the restart data set can be used for the first time, it must be defined and initialized as a VSAM key sequenced data set (KSDS). The following jobs are samples that achieve this. The values of the RECORDS and CATALOG parameters may vary according to the installation's requirements. The value of the NAME parameter in the CLUSTER definition is the data set name, which must agree with the file-id in the DLBL statement for the CICS execution. The data set is initialized with one record containing a 10-byte key:

'ACTL 0001', followed by a blank.

```
// JOB DEFRSD CREATE RESTART DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                            -
          (NAME(cicsl60.rsd)                               -
           INDEXED                                         -
           RECORDSIZE(2000 2000)                           -
           RECORDS(250 100)                                -
           KEYS(10 0)                                      -
           FREESPACE(20 20)                                -
           SHAREOPTIONS(2)                                 -
           VOLUMES(volume))                                -
        DATA                                               -
          (NAME(cicsl60.rsd.data))                         -
        INDEX                                              -
          (NAME(cicsl60.rsd.index))                        -
        CATALOG(user.catalog)
/*
/&
// JOB INITRSD INITIALIZE RESTART DATASET
// DLBL user-catalog-filename,'user.catalog',,VSAM
// DLBL DFHRSD,'cicsl60.rsd',,VSAM,CAT=user-catalog-filename
// EXEC IDCAMS,SIZE=AUTO
 REPRO  INFILE                                             -
          (SYSIPT                                          -
           ENVIRONMENT                                     -
            (RECORDFORMAT(FIXUNB)                          -
             BLOCKSIZE(80)                                 -
             RECORDSIZE(80)))                              -
         OUTFILE(DFHRSD)
ACTL 0001
/*
/&
                       (These jobs are included in Z.DFHSPJCL)
```

SPACE CALCULATIONS

1. The records are up to 2000 bytes each, with a 10-byte key and 10 bytes of flags and identifiers.

2. The maximum number of bytes available in each record is 1980. The number of records required to keypoint tables and control blocks can be determined by referring to the following list:

   - Each PPT entry requires 62 bytes.
   - Each PCT entry requires 42 bytes.
   - Each FCT entry requires 58 bytes for other than VSAM, 90 bytes for VSAM.
   - Each TCT entry requires from 126 to 206 bytes, depending upon terminal type.
   - Each DCT entry requires 90 bytes.
   - Each ICE requires 34 bytes.
   - Each AID requires 50 bytes.
   - Each BCA requires 362 bytes.

- Each WRE requires 18 bytes.
- Each temporary storage destination requires 18 bytes plus 6 bytes for each auxiliary record.

Entries for different tables are written with different keys. For each table, the amount of space required to keypoint information should be calculated in 2000-byte records.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

If the restart data set has been defined using the sample job in the previous section, the job control statement for the CICS execution is:

```
// DLBL DFHRSD,'cics160.rsd',,VSAM,CAT=user-catalog-filename
```

Chapter 4.7. Automatic Statistics Data Sets

SYSTEM STATISTICS

System statistics are accumulated continually by CICS management programs in CICS system tables during the execution of CICS. System statistics can be captured and recorded, **on request**, or **automatically** at intervals, by any operator whose security code allows access to such information. In addition, system statistics are recorded on normal termination of the system.

The operator initiates the recording of requested statistics or of automatic statistics by means of the CSTT transaction with appropriate arguments. (See the CICS/VS Operator's Guide for detailed information.) The statistics are written to transient data destinations defined in the destination control table. These destinations must, in turn, correspond to data sets that receive the physical output.

REQUESTED STATISTICS

Requested system statistics are recorded in full, or in part, as variable-length, unblocked records with a minimum block size of 132 (decimal) bytes.

The operator may specify the destination for requested statistics; if the destination is omitted, the default destination CSSL is used (see "termination statistics" below).

The operator can optionally request that all selected statistics (except total number of tasks) be reset to zero after each output. Requested system statistics cannot be recorded while the recording of automatic system statistics is in effect.

AUTOMATIC STATISTICS

Automatic system statistics are recorded in full and written to destinations with system-defined names CSSM and CSSN, as variable-length blocked records with minimum block size of 304 (decimal) bytes.

The operator initiates the periodic recording of the statistics on the current transient data destination by means of the CSTT transaction, specifying the interval of time between successive recordings and the number of recordings to be placed on the primary and secondary data sets. After the specified number of intervals has been completed on the current data set, the data set is closed and a switch occurs automatically to the other data set. The first data set may be used to prepare a printed report offline by means of the DFHSTUP automatic statistics utility program (see "Processing the Automatic Statistics Data Sets (DFHSTUP)" on page 204).

The operator can switch the automatic statistics data sets explicitly by invoking the CSTT transaction. The destination currently in use is closed after the current recording, if any, is completed, and the other destination is opened and put into use for the next recording. The CSTT transaction can also be used to terminate the periodic recording of statistics.

Automatic statistics are cumulative only for the interval specified, that is, the statistics are reset to zero at the end of each interval.

TERMINATION STATISTICS

Termination statistics are recorded on normal CICS termination as if an explicit request for all requested statistics had been made at that time. They are sent to the CSSL destination and include all statistics accumulated since the counts were last set to zero. They do not include statistics that have already been recorded during automatic statistics recording.

DESTINATIONS FOR STATISTICS

The statistics destinations CSSL, CSSM, and CSSN must be defined as extrapartition transient data destinations. CSSL may refer to an output printer defined by the user.

Because system statistics are always recorded on CICS termination, the CSSL destination is a required entry in the DCT. The CSSM and CSSN destinations are required only if automatic statistics are used.

The data sets used by the CSSM and CSSN destinations are sequential data sets located on tape or direct-access storage. The filenames for these data sets are required to be DFHSTM and DFHSTN respectively. See "Chapter 4.4. Transient Data Extrapartition Data Sets" for information on defining these data sets.

The destination control table (DCT) entries required to define the CSSM and CSSN destinations are described in the CICS/VS Resource Definition Guide. You should code the DFHDCT TYPE=EXTRA macro instruction for these destinations with RESIDNT=NO. In this way, the DTF for an automatic statistics data set is reloaded each time the data set is opened. An automatic statistics data set is opened when the operator initiates automatic recording of statistics, and when a switch is made from one data set to the other.

You do not have to define both data sets to activate automatic statistics. If, however, you use only one data set, and a request is made to switch data sets, automatic statistics will be terminated. If only one data set is used, the filename must be DFHSTM.

Automatic statistics data sets are not required to be preformatted. If disk space is available, it is probably most convenient to allocate them before the very first time CICS is run.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

The following job control statements might be used in the CICS execution for the automatic statistics data sets if they are to be held on **disk**:

```
// DLBL DFHSTM,'cics160.auto.stat.a',0,SD
// EXTENT  SYSnnn,,1,0,relative-track-number,number-of-tracks
// DLBL DFHSTN,'cics160.auto.stat.b',0,SD
// EXTENT  SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
```

The following job control statements for the data sets on **tape** might be used:

```
// ASSGN   SYSmmm,cuu
// ASSGN   SYSnnn,cuu
```

where SYSmmm and SYSnnn match those specified in the DCT.

Note: The data set, DFHSTM or DFHSTN, is opened as an output work data set. Therefore, if the statistics summary data set is not expired, the VSE message:

```
4n44A OVERLAP ON UNEXPRD FILE
```

is written to the console. This is not an error. After ensuring that the overlap data set is the statistics summary data set, the operator should respond:

```
DELETE
```

The unexpired file message can be prevented by coding a date of 0 in the DLBL statement, as shown above.

SPACE CALCULATIONS

The following information will help you estimate the amount of direct-access storage needed to collect automatic statistics for your system. The space requirements for the statistics output data set are affected by:

- Duration of the automatic process of recording system statistics.
- Frequency of the interval; that is, the time specified as the interval to record system statistics.
- Block size (maximum 4100 bytes) of the statistics summarization output data set.
- Track capacity of the device (if a count-key data set) on which the data set resides.

You should allow:

- 16 bytes for the header record
- 497 bytes for both summary records
- 34 bytes x number of transactions used
- 18 bytes x number of programs used
- 40 bytes x number of terminals in system
- 12 bytes x number of transient data destinations used
- 56 bytes x number of non-DL/I files in system + 16 bytes for each file segment used
- 20 bytes for each file that uses VSAM shared resources
- 20 bytes for each journal data set in system (including the system log data set)
- 80 bytes for each system entry for a remote system
- 4 bytes of length field (LLbb) for every 300 bytes written.

Example:

You have specified an interval of two minutes to record statistics, and cancel the automatic process after two hours. The system configuration is as follows:

200 terminals in system
6 files in system - no segments used
2 journal data sets including the system log

Every two minutes, there are approximately:

10 different transactions used
10 different programs used
3 different transient data destinations used

The output data set is allocated on a 3330 disk storage device.

For this example, 60 intervals of data are recorded. Based upon the configuration and the sizes of records, approximately 9600 bytes of data would be written every interval. Assuming a block size of 4000 bytes (VB) and a 3330 disk storage device, approximately 51 tracks of disk space would be needed.

PROCESSING THE AUTOMATIC STATISTICS DATA SETS (DFHSTUP)

The automatic statistics utility program (DFHSTUP) is used to prepare and print reports offline from the data recorded by the automatic statistics program.

For automatic statistics data sets on **disk**, the job control statements to execute the automatic statistics utility program might be:

```
// JOB PRTSTATS PRINT AUTO STATS
// DLBL DTFDISK,'cics160.auto.stats.b',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// UPSI yxxxxxxx                      (see note)
// ASSGN  SYSnnn,cuu
// LIBDEF CL,SEARCH=cics-cil-filename
// EXEC   DFHSTUP,SIZE=80K
 DEVICE=DISK
/*
/&
```

Note: The file ID in the DLBL statement must be the same as that used for the CICS execution.

For automatic statistics data sets on **unlabeled tape**, the job control statements to execute the automatic statistics utility program might be:

```
// JOB CICSSTUP STATISTICS SUMMARIZATION UTILITY PROGRAM
// UPSI yxxxxxxx                        (see note)
// ASSGN   SYS011,cuu
// LIBDEF  CL,SEARCH=cics-cil-filename
// EXEC    DFHSTUP,SIZE=80K
 DEVICE=TAPE
/*
/&
```

Note: The ASSGN statement must specify SYS011. If the ASSGN statement is omitted, a tape unit will be dynamically acquired.

In the above sample jobs, the // UPSI control statement is used to indicate the following:

Bit 0
- 0 - Only the summary output report is to be printed
- 1 - Only the interval output report is to be printed

Bits 1 through 7
- 0 - 60 lines to be printed on each page of the reports (default value). Otherwise, the binary value represented by these seven bits is the number of lines to be printed on each page of the reports. Minimum page size is 30 lines.

In addition, the DEVICE= control command is used to indicate whether the automatic statistics are recorded on disk or tape.

Alternatively, all the control information may be included by means of control commands. The commands are:

DEVICE={disk|tape}
specifies whether the automatic statistics file to be processed is on disk or on tape.

STATS={int|sum}
specifies whether an interval output report or a summary output report is to be printed.

LINES=nn
specifies the number of lines (30-999) to be printed on each page of the report.

The commands may be coded in either of the following ways:

1. In the PARM operand in trhe // EXEC DFHSTUP statement:

   ```
   // EXEC DFHSTUP,SIZE=80K,PARM='command,command,....'
   ```

   If the PARM operand is used, the keywords and entries must be enclosed in quotes and separated by commas. No embedded blanks are allowed.

2. As a series of statements in the input stream following the // EXEC DFHSTUP statement:

   ```
   // EXEC DFHSTUP,SIZE=80K
        command,command,
        command
   ```

   In this case, the keywords and entries in a list must be separated by commas. Continuation to another record is allowed after any comma, provided the comma is in column 71 or is followed by a blank. Continuation records can begin in any column.

   For example, either of the following sets of statements:

```
// EXEC DFHSTUP,SIZE=80K,PARM='DEVICE=DISK,STATS=INT,LINES=060'
```

or

```
// EXEC DFHSTUP,SIZE=80K
     DEVICE=DISK,
     STATS=INT,LINES=060
```

would cause the printing of an interval output report at 60 lines per page.

The automatic statistics utility program is always generated with support for all valid input devices. The device to be used is normally specified in the job stream that executes the utility, as shown. If the device has not been specified in SYSIPT, the following messages will appear on the operator console:

```
DFH1650I   NO VALID PARAMETER CARD ON SYSIPT
DFH1651A   ENTER 'STOP' OR INPUT DEVICE TYPE
           (TAPE, 2314, 3330, 3340, 3350, FBA, DISK):
```

The operator should reply STOP to terminate the run, or enter the device type on which the output from the CICS automatic statistics program was placed during the execution of CICS. There is no default value. If an invalid parameter was read from SYSIPT, this is displayed on the console before message DFH1650I.

For details of the statistics produced, see the CICS/VS Performance Guide.

Chapter 4.8. Auxiliary Trace Data Sets

The trace management component of CICS enables you to initiate and control the recording of trace entries representing the execution of all, or selected, CICS macro- or command-level requests during execution.

Trace entries are of fixed length and defined format (see the CICS/VS Problem Determination Guide for details), recorded sequentially in an area of main storage referred to as the **trace table**. The size of the trace table is determined at system initialization by means of the TRT parameter in the SIT or as a system initialization override; if the size is specified as zero, no tracing will be performed. Entries in the trace table are written in a wraparound manner: when the end is reached, subsequent entries overwrite the entries at the beginning of the table.

In most cases, the main storage trace table will be relatively small, and entries will be overwritten. To obtain a trace of CICS activity in which trace entries are not overwritten, a further method of recording known as **auxiliary trace** can be used. The trace entries are then, in addition, recorded in a sequential data set on disk or tape. The format of trace entries in the auxiliary trace data set is identical to that in main storage.

On tape, only one auxiliary trace data set can be defined. On disk, either a single auxiliary trace data set or two data sets can be defined. Defining two data sets allows the auxiliary trace to be switched from one to the other.

When the current auxiliary trace data set is filled, CICS informs the operator by means of message DFH1401, closes the data set, and suspends auxiliary tracing.

If only one auxiliary trace data set is defined, when it is filled, CICS will turn off auxiliary tracing. Turning auxiliary tracing on with the CEMT SET AUXTRACE ON master terminal command will then reuse the data set.

If two auxiliary trace data sets are defined, you can switch to the other auxiliary trace data set with the CEMT SET AUXTRACE SWITCH command. If you switched data sets because the previous one was full, then you must turn auxiliary tracing on again with the CEMT SET AUXTRACE ON command. This is because CICS turned auxiliary tracing off when it closed the full data set.

You can switch data sets at any time with the CEMT SET AUXTRACE SWITCH command. If you switch before the data set is filled and auxiliary tracing is already turned on, then you do not need to turn auxiliary tracing on again after the switch.

The contents of the auxiliary trace data set can be printed using the DFHTUP trace utility program (see below). The program formats each trace entry to show the CICS component that made the entry and the function being performed. The entries to be printed may be selected by task or transaction identifier, by time, or by trace identifier.

The contents of the main storage trace table can be printed with the dump utility program (DFHDUP).

Main storage trace and auxiliary trace may be activated independently and in either order. Auxiliary trace entries are not recorded, however, unless main storage tracing is active at the same time. This means that auxiliary trace may be active for an extended period, but actual tracing will be controlled by activating and deactivating main storage trace.

Main storage trace is controlled by any one of three methods:

- By means of the TRP=(,ON) SIT parameter or startup override. This activates tracing of all CICS requests at CICS initialization time. The TRT parameter must have a nonzero value to allow trace entries to be recorded. Tracing continues until the end of the CICS run or until it is turned off as described below.

- By means of the CEMT master terminal transaction (see the CICS/VS Operator's Guide). This initiates the recording of all CICS requests in the trace table (and in the auxiliary trace data set if auxiliary trace is already active). The transaction can also be used to turn trace off. The CSFE transaction can also be used to control the trace facilities.

- By means of an EXEC CICS command or DFHTR macro instruction in an application program (see either the CICS/VS Application Programmer's Reference Manual (Macro Level) or the CICS/VS Application Programmer's Reference Manual (Command Level)). This method of activation allows selective recording of trace entries according to the CICS component that made the entry (for example, storage control, terminal control, or EXEC interface) as well as the recording of all entries.

Auxiliary trace is controlled by:

- The TRP=(,AUX) SIT parameter or start-up override. This activates both auxiliary trace and main storage trace.

- The CEMT master terminal transaction. If this command is used to activate auxiliary trace and the auxiliary trace data set is not open, the command causes the primary data set to be opened as well as activating auxiliary trace. Trace entries are recorded only when main storage trace is also active.

For auxiliary trace data sets on **disk**, the filenames of the auxiliary trace data sets are defined by CICS as DFHAUXT and DFHBUXT. If a single data set is to be used, its filename must be DFHAUXT.

For an auxiliary trace data set on **tape**, device SYS009 must be specified in the ASSGN job control statement.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

The following job control statements might be used in the CICS execution for the auxiliary trace data sets, if these are to be held on **disk**:

```
// DLBL DFHAUXT,'cics160.aux.trace.a',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// DLBL DFHBUXT,'cics160.aux.trace.b',0,SD
// EXTENT SYSmmm,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// ASSGN SYSmmm,cuu
```

Note: In the DLBL statement, the date value of 0 prevents the unexpired file message being generated during execution of the trace program.

If the auxiliary trace data set is to be held on **tape**, the following job control statement might be used:

```
// ASSGN SYS009,cuu
```

Note: The ASSGN statement must specify SYS009. If the ASSGN statement is omitted, a tape unit will be dynamically acquired.

A tape unit should be assigned and a tape reel mounted before the master terminal operator enters the command to turn on the auxiliary trace. Only unlabeled tapes should be used for this data set; standard labeled tapes are not supported.

SPACE CALCULATIONS

Trace entries are 32 (decimal) bytes in length. The physical record length (blocksize) cannot be modified, but is defined as 4096.

PROCESSING THE AUXILIARY TRACE DATA SET (DFHTUP)

A CICS utility program, DFHTUP, is provided to format and print trace records stored on the auxiliary trace data set. This utility program is run as a separate VSE job and extracts all or selected trace entries from the data set. The type of entries to be processed by this program are specified on parameter cards supplied as part of the input to the program. Entries that can be selected for processing include:

- All entries
- Entries written to the auxiliary trace data set within a specified period of time
- Entries written for a specified terminal
- Entries with a specified trace identifier
- Entries associated with a specified transaction identification
- Entries associated with a specific instance of a transaction identification (task).

If the auxiliary trace data sets are held on **disk**, the job control statements might be:

```
// JOB PRTAUX PRINT AUXILIARY TRACE DATASET
// DLBL DFHAUXT,'cics160.aux.trace.a',,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// EXEC DFHTUP,SIZE=80K
 DEVICE=DISK
 .
 parameter statements
 .
/*
/&
                    (This job is included in Z.DFHSPJCL)
```

You may also need to add:

```
// LIBDEF CL,SEARCH=cics-cil-filename
```

before the EXEC statement in the above job.

If the auxiliary trace data set is held on **tape**, the job control statements might be:

```
// JOB PRTAUX PRINT AUXILIARY TRACE DATASET
// ASSGN SYS009,cuu
// LIBDEF CL,SEARCH=cics-cil-filename
// EXEC DFHTUP,SIZE=80K
 DEVICE=TAPE
 .
 parameter statements
 .
/*
/&
```

Note: The ASSGN statement must specify SYS009. If the ASSGN statement is omitted, a tape unit will be dynamically acquired.

In the sample jobs above, all the control information may be included by means of control commands. The commands are:

DEVICE={disk|tape}
: specifies whether the auxiliary trace data set to be processed is on disk or on tape.

ALL
: specifies that all trace entries in the auxiliary trace data set are to be printed. This command is the default.

TASKID={id|id-id} or TASKID=({id|id-id},{id|id-id},...)
: specifies the task identifications of one or more tasks for which trace entries are to be printed. An id value can be any number up to five decimal digits in length, or any of the character strings ATP, JJJ, KCP, or TCP. A range of task identifications can be specified by use of a hyphen.

TERMID=tttt or TERMID=(tttt,tttt,...)
: specifies the terminal identifications of one or more terminals for which trace entries are to be printed.

TRANID=tttt or TRANID=(tttt,tttt,...)
: specifies the transaction identifiers of one or more transactions for which trace entries are to be printed.

TIMERG=hhmmss-hhmmss or TIMERG=(hhmmss-hhmmss,hhmmss-hhmmss,...)
: specifies the time period or periods for which trace entries are to be printed. Time periods are indicated by pairs of values represented as hours (hh), minutes (mm), and seconds (ss) separated by a hyphen. The ending value of each pair must be later than the starting value.

TYPETR={ii|ii-ii} or TYPETR=({ii|ii-ii},{ii|ii-ii},...)
: specifies the trace entry identifications for which trace entries are to be printed. An 'ii' value can be either a 3-character decimal value (000-255) or a 2-character hexadecimal value (00-FF). Decimal values are converted to hexadecimal values by the utility program. A range of trace entry identifications can be specified by use of a hyphen.

The commands may be coded in either of the following ways:

1. In the PARM operand in the // EXEC DFHTUP statement:

   ```
   // EXEC DFHTUP,SIZE=80K,PARM='command,command,....'
   ```

   If the PARM operand is used, the keywords and entries must be enclosed in quotes and separated by commas. No embedded blanks are allowed. If you need to continue the statement to another record, you must code the statement up to column 71, include a continuation character in column 72, and continue from column 16 in the next record.

2. As a series of statements in the input stream following the // EXEC DFHTUP statement:

   ```
   // EXEC DFHTUP,SIZE=80K
        command,command,
        command
   ```

   In this case, the keywords and entries in a list must be separated by commas. No embedded blanks are allowed. Continuation to another record is allowed after any comma, provided the comma is in column 71 or is followed by a blank. Continuation records can begin in any column.

For example, either of the following sets of statements:

```
// EXEC DFHTUP,SIZE=80K,PARM='DEVICE=DISK,TERMID=L77A,TRANID=(ABRW,AORD*
               ),TIMERG=(123000-150000)'
```

or

```
// EXEC DFHTUP,SIZE=80K
      DEVICE=DISK,
      TERMID=L77A,
      TRANID=(ABRW,AORD),TIMERG=(123000-150000)
```

would cause the printing of auxiliary trace entries referring to transactions ABRW or AORD, run at terminal L77A between 12.30 hrs. and 15.00 hrs.

Chapter 4.9. Dump Data Sets

Dumps may be recorded by CICS either as a consequence of a failure detected during CICS execution, or upon explicit request. CICS allows you some flexibility in specifying the content and type of dump to be produced. The choice is made by options coded in the system initialization table (SIT) or system initialization overrides, and may be affected by options specified when making an explicit request.

The occasions when dumps may be produced are:

- An explicit request issued by an application program
- A transaction abend
- Inclusion of dump option in a CICS shutdown request made by the operator
- An explicit dump request by the operator
- A storage violation detected by CICS storage control
- Abnormal termination of CICS.

When any of the above conditions occurs, the type of dump produced depends on the condition as well as the SIT options. The types of dumps that can be produced are:

- **Transaction** dump. This is recorded by CICS following any transaction abend or application program request. It contains the storage areas and CICS control blocks associated with the transaction. The application program may also request a dump of specified areas of storage.

- **Formatted** dump. This is a dump of most of the control blocks and storage areas in the CICS partition. The dump is followed by an index of the control blocks and a list of the modules with their load and entry point addresses. The formatted dump may include or invoke a dump of the CICS partition (or of the dynamic storage area, in the case of storage violation), as described below:

  - CICS partition dump. This is an unformatted dump of the entire CICS partition. It is produced by CICS from within the partition without operating system assistance.

  - Partition dump recorded by VSE. This is a dump taken as a result of a VSE PDUMP macro request issued by CICS. It includes all the CICS partition storage together with the PSW and registers.

The VSE partition dump is written by VSE either to a data set that can be on disk or on tape, or to the output printer. If the data set is on disk or tape, the filename must be DOSDMPF. If a second data set is required on disk, the filename of the second data set must be DOSDMPG. If the dump is directed to the output printer, you must assign the dump file to SYSLST. For further information on defining and processing these data sets, see the <u>VSE/Advanced Functions Serviceability Aids and Debugging Procedures</u> manual.

The remaining types of dump are recorded by CICS on a single sequential data set on tape or disk, or, on disk only, on a pair of sequential data sets. The data sets must be defined in the CICS run with filenames DFHDMPA and DFHDMPB. If a single data set is defined, its filename must be DFHDMPA. Each data set can hold a number of separate dumps.

If two data sets are being used on disk, it is possible to print down a completed set of dumps while CICS is running. In order to do this, the data sets should first be switched by means of the CEMT SET DUMP SWITCH master terminal command. CICS closes the current data set after any dump being recorded has been completed, and opens the other data set. The completed data set can be printed using the DFHDUP dump utility program (see "Processing the Dump Data Sets (DFHDUP)" on page 216).

As well as switching dump data sets explicitly, the operator can cause automatic switching to occur when the current data set becomes full, by issuing the CEMT SET DUMP AUTOSWITCH master terminal command. When the dump data set is full, the following two messages will then be displayed at the console:

```
DFH0701  THE DUMP DATASET IS FULL
DFH0702  DUMP DATASET SWITCHING TO DFHDMPx
```

where 'x' has the value A or B.

CICS then closes the current data set, opens the other one, and restarts from the beginning the dump that caused the end-of-data set condition. The operator can print the completed data set with the DFHDUP utility program and then reissue the CEMT SET DUMP AUTOSWITCH master terminal command. This will again cause the switch to be performed automatically when the current data set is full.

The dump data sets DFHDMPA and DFHDMPB can be defined as temporary data sets for each CICS run. More commonly, they will be allocated in advance, reused repeatedly, and not deleted when the CICS job has completed. The DFHDUP utility program can then be used to print the dump output at any time during or after the CICS run.

Except for the case of a transaction dump, the type of dump produced depends on the options specified in the SIT or the system initialization override parameters. The options, which may be coded in any order after the suffix on the FDP parameter, have the following meanings:

NO
: In the case of a CICS abnormal termination only, take an operating system abend dump.

PARTN
: Record a partition dump either to DFHDMPA or DFHDMPB, or to SYSLST or SYSDMP if PDUMP is also coded.

PDUMP
: If this is coded in conjunction with PARTN, record a VSE PDUMP to SYSLST or SYSDMP.

FORMAT
: Record a formatted dump to DFHDMPA or DFHDMPB.

The **FULL** option can be used to denote both PARTN and FORMAT.

The various conditions listed at the start of this section will then result in the following actions:

- Abnormal termination of CICS. A dump is taken in accordance with the SIT options or override parameters.

- Dump option on CICS shutdown request. A dump is taken in accordance with the SIT options or override parameters, except that, if NO is coded, no dump is taken.

- An explicit dump request by means of the CEMT PERFORM SNAP master terminal command. If neither FORMAT or PARTN is specified, the dump is taken in accordance with the SIT options or override parameters except that, if NO is coded, no dump is taken.

  If PARTN is specified, the SIT options are used to determine whether a dump is written to DFHDMPx or a PDUMP macro is issued. If FORMAT is specified, a formatted dump is produced.

- Storage violation. If SVD=YES is coded in the SIT, or if SVD=nn is coded and fewer than nn storage violation dumps have been taken so far, the dump is taken in accordance with the options specified on the FDP parameter in the SIT or system initialization overrides. If no partition dump is specified, however, a dump of the dynamic storage area (DSA) only, will also be produced. The partition or DSA dump will be written to the CICS dump data set or produced by issuing a PDUMP macro, depending on whether or not the PDUMP option was coded in the FDP parameter.

- Transaction abend. Unless the program control table entry for the transaction is defined with the DUMP=NO parameter, a transaction dump is recorded. In addition, if the FDUMP parameter is coded in the PCT and the transaction abend was caused by a program check (ASRA) or an operating system abend (ASRB), a dump in accordance with the SIT options or override parameters is recorded before the transaction dump.

- Explicit application program request. The type of dump taken depends on the request, and is not dependent on the options in the SIT or override parameters.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

To define two dump data sets on **disk**, the following job control statements might be used:

```
// DLBL DFHDMPA,'cics160.dump.dataset.a',0,SD    (see notes)
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// DLBL DFHDMPB,'cics160.dump.dataset.b',0,SD    (see notes)
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
```

Notes:

1. In the DLBL statement, the date should be 0, to prevent the unexpired file message during execution of the dump utility program (DFHDUP).

2. The filenames must be different, otherwise CEMT SET DUMP SWITCH will abend due to label information being deleted.

3. CISIZE can be specified in the DLBL statement if the dump data set is to reside on an FBA device. The CISIZE value must be a multiple of 512 bytes (for CI sizes ≤ 8192 bytes) or a multiple of 2048 bytes (for CI sizes > 8192 bytes).

If the single dump data set is held on an **unlabeled tape**, the job control statement might be:

```
// ASSGN SYS010,cuu
```

Note: The ASSGN statement must specify SYS010. If the ASSGN statement is omitted, a tape unit will be dynamically acquired.

SPACE CALCULATIONS

When CICS is first installed, the dump data set will probably need between five and ten cylinders of disk space. Once the CICS system has settled to normal operation, two or three cylinders may be sufficient.

When the dump data set resides on an FBA device, between 1 million and 2 million bytes of disk storage are likely to be needed when CICS is first installed. Once installed, between 400,000 and 600,000 bytes should be sufficient.

PROCESSING THE DUMP DATA SETS (DFHDUP)

The dump utility program (DFHDUP) is used to prepare the dump output for printing and to print the formatted information.

For dump data sets on **disk**, the job control statements to execute the dump utility program might be:

```
// JOB PRTDUMP PRINT DUMP DATASET
// DLBL DTFDISK,'cics160.dump.dataset.a',,SD     (see note)
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// EXEC DFHDUP,SIZE=80K,PARM='DEVICE=DISK,TRANSLATE=FOLD'
/&
                          (This job is included in Z.DFHSPJCL)
```

Notes:

1. The file ID in the DLBL statement must be the same as that used for the CICS execution.

2. You may also need to add:

   ```
   // LIBDEF CL,SEARCH=cics-cil-filename
   ```

   before the EXEC statement.

For a dump data set on **tape**, the job control statements to execute the dump utility might be:

```
// JOB PRTDUMP PRINT DUMP DATA SET
// ASSGN   SYS011,cuu
// LIBDEF  CL,SEARCH=cics-cil-filename
// EXEC    DFHDUP,SIZE=80K,PARM='DEVICE=TAPE,TRANSLATE=FOLD'
/*
/&
```

Note: The ASSGN statement must specify SYS011. If the ASSGN statement is omitted, a tape unit will be dynamically acquired.

In the sample jobs above, all the control information may be included by means of control commands. The commands are:

DEVICE={disk|tape}
specifies whether the dump data set to be processed is on disk or on tape.

{DOUBLE|SINGLE} (DOUBLE underlined)
where the options mean:

DOUBLE
For a transaction dump only, the output is printed with a blank line between each printed line. This is the default.

SINGLE
The dump output is printed single spaced.

TRANSLATE={LC|UC|FOLD} (LC underlined)
where the options mean:

LC
Both uppercase and lowercase characters are to be translated and printed. This is the default.

UC

Uppercase characters only are to be printed. The lowercase characters are to appear as periods (full stops).

FOLD

Both uppercase and lowercase characters are to printed in upper case.

The commands may be coded in either of the following ways:

1. In the PARM operand in the // EXEC DFHDUP statement:

   ```
   // EXEC DFHDUP,SIZE=80K,PARM='command,command,....'
   ```

   If the PARM operand is used, the keywords and entries must be enclosed in quotes and separated by commas. No embedded blanks are allowed.

2. As a series of statements in the input stream following the // EXEC DFHDUP statement:

   ```
   // EXEC DFHDUP,SIZE=80K command,command, command
   ```

   In this case, the keywords and entries in a list must be separated by commas. Continuation to another record is allowed after any comma, provided the comma is in column 71 or is followed by a blank. Continuation records can begin in any column.

For example, either of the following sets of statements:

```
// EXEC DFHDUP,SIZE=80K,PARM='DEVICE=DISK,TRANSLATE=FOLD,SINGLE'
```

or

```
// EXEC DFHDUP,SIZE=80K
      DEVICE=DISK,
      TRANSLATE=FOLD,
      SINGLE
```

would cause the printing of a dump single-spaced with both uppercase and lowercase characters printed in uppercase.

The dump utility program is always generated with support for all valid input devices. The device to be used is normally specified in the job stream that invokes the utility, as shown. If the device has not been specified in SYSIPT, the following messages will appear on the operator console:

```
DFH1560I  NO VALID PARAMETER CARD ON SYSIPT
DFH1561A  ENTER 'STOP' OR INPUT DEVICE TYPE
          (TAPE, 2314, 3330, 3340, 3350, FBA, DISK):
```

The operator should reply STOP to terminate the run, or enter the device type on which the output from the CICS dump control program was placed during execution of CICS. There is no default value. If an invalid parameter was read from SYSIPT, this is displayed on the console before message DFH1560I.

If the dump utility program is used to print a dump data set that is still in use by CICS, any dumps written during the current run will be printed. These may be followed by an unidentified partial dump from a previous run, whose header has been overwritten during the current run. Finally, there may be further dumps from the previous run.

Chapter 4.10. CICS System Definition File

You can define some CICS resources and add them to a running CICS system by means of **resource definition online** (RDO). This can be done without performing a restart, by using the CICS CEDA transaction running on a display screen. The CICS resources you can define in this way are those that are defined by the program control table (PCT) and the processing program table (PPT), namely, transactions, programs, map sets, partition sets, and profiles. The CEDA transaction and its display format are similar to the master terminal transaction (CEMT). Resources defined by the CEDA transaction are held on the CICS system definition (CSD) file (a VSAM file). After they have been defined on the CSD file, the resources can be installed from it onto the running CICS system.

The CSD file is a VSAM file that you define to the system. A file control table (FCT) definition of the CSD file, together with job control statements and VSAM Access Method Services control statements (IDCAMS control statements), are supplied as part of the pregenerated CICS system. If you want to create your own FCT definition of the CSD file, see "File Control Table Definition of the CSD File" on page 220.

After it is created, the CSD file must be initialized using the DFHCSDUP offline utility program. This step is described in "Processing the CICS System Definition File (DFHCSDUP)" on page 223.

When the CSD file is initialized, a basic CSD file containing the standard group entries for the program control table (PCT) and processing program table (PPT) is created. Any PCT and PPT tables that already exist can be migrated from the core image library to the CSD file before or after the CEDA transaction is used to define new entries interactively. This migration is performed using the DFHCSDUP utility program. For further information on this utility program, see below.

The commands that you can use during the CEDA transaction are described in the CICS/VS Resource Definition Guide.

FILE CONTROL TABLE DEFINITION OF THE CSD FILE

The following FCT entry (provided as copy book DFHXFCT in the source statement library) is a sample of that required to define the CSD file to your CICS system:

```
TITLE 'DFHXFCT - COPYBOOK FOR DFHCSD'
DFHFCT TYPE=DATASET,DATASET=DFHCSD,ACCMETH=(VSAM,KSDS,KEY),    *
      SERVREQ=(GET,PUT,UPDATE,NEWREC,BROWSE,DELETE),           *
      RECFORM=(VARIABLE,BLOCKED),BUFND=5,BUFNI=4,STRNO=4,      *
      OPEN=DEFERRED
```

You can vary the above FCT specification in several ways.

FCT Parameter	Description
TYPE=DATASET	Must be as in the sample.
DATASET=DFHCSD	Must be as in the sample.
ACCMETH=(VSAM,KSDS)	Used when SHAREOPTIONS(2) is used. ACCMETH=(VSAM,KSDS,KEY) is used when the CSD file is defined to VSAM as using SHAREOPTIONS(4).
SERVREQ=	Can be varied to limit access to the CSD file. For instance, SERVREQ=(GET,BROWSE) would result in a read-only CSD file definition.
RECFORM=(VARIABLE,BLOCKED)	Must be as in the sample.
BUFND= BUFNI=	Can be varied in accordance with explanations in the <u>CICS/VS Resource Definition Guide</u> and in conjunction with STRNO.
STRNO=	Must not be less than 2 for the CSD file. You can increase STRNO to allow more than one concurrent CEDA user. Insufficient STRNO value will be diagnosed by CEDA.
OPEN=DEFERRED	DEFERRED is recommended, in which case the CSD file is opened for read-only access at CICS cold and emergency start, and as specified by the FCT entry at all other times. If DEFERRED is not specified, the CSD file is opened with access as specified by your FCT entry even though this may not be required by a CICS cold or emergency start.
OPEN=INITIAL	Should be specified for the CSD file if GRPLIST is not specified in the SIT.
LOG=NO	Is the default. The CSD file is not a CICS-recoverable file, and LOG=YES should not be specified.

JOB CONTROL STATEMENTS TO DEFINE AND INITIALIZE THE CICS SYSTEM DEFINITION FILE

Before the CICS system definition file can be used for the first time, it must be defined and initialized as a VSAM KSDS data set. The sample job streams given below can be used to achieve this.

The value of the NAME parameter in the CLUSTER definition is the data set name, which must agree with the data set name in the DLBL statement for DFHCSD at CICS startup and when using DFHCSDUP, the offline CSD utility program. For considerations on the SHAREOPTIONS parameter, see "Multiple Users of the CSD File Across CICS or Batch Regions" on page 222.

The INITIALIZE command of DFHCSDUP initializes your CSD file with definitions of the IBM-supplied resources. These are contained in two supplied load modules whose names are DFHPPI and DFHPCI. After the CSD file has been initialized, you can start defining your resources interactively with the CEDA transaction. The INITIALIZE command will probably be used only once in the lifetime of the CSD file.

```
// JOB DEFCSD CREATE CICS SYSTEM DEFINITION FILE
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                                -
          (NAME(cics160.csd)                                   -
           INDEXED                                             -
           RECORDSIZE(100 200)                                 -
           RECORDS(4000 1000)                                  -
           KEYS(22 0)                                          -
           FREESPACE(0 0)                                      -
           SHAREOPTIONS(2)                                     -
           VOLUMES(volume))                                    -
        DATA                                                   -
          (NAME(cics160.csd.data))                             -
        INDEX                                                  -
          (NAME(cics160.csd.index))                            -
        CATALOG(user.catalog)
/*
/&
// JOB INITCSD INITIALIZE THE CSD
// DLBL user-catalog-filename,'user.catalog',,VSAM
// DLBL DFHCSD,'cics160.csd',,VSAM,CAT=user-catalog-filename
// EXEC DFHCSDUP,SIZE=AUTO           (see note)
 INITIALIZE
 LIST ALL
/*
/&
                      (These jobs are included in Z.DFHSPJCL)
```

Notes:

1. You may need to add the following statement:

   // LIBDEF CL,SEARCH=cics-cil-filename

   before the // EXEC statement.

2. You must use SIZE=AUTO on the EXEC statement for DFHCSDUP. The difference between the size of DFHCSDUP and the partition size as allocated is available as GETVIS storage, which is used by DFHCSDUP to load tables being migrated. GETVIS storage of 120K bytes should be adequate to migrate an 80K byte table.

SPACE CALCULATIONS

The CSD file defined by the IDCAMS statements above should be large enough to hold about 2,000 definitions. As with any VSAM file, it may be necessary to reorganize the CSD file occasionally to improve both space utilization and access performance: this can be achieved by using the EXPORT and IMPORT facilities of IDCAMS.

Each resource definition is held as a separate record on the CSD file. The size of the record depends on the type of the resource:

Type of Resource	Approximate Size
Transaction definition	100 bytes
Program definition	80 bytes
Profile definition	90 bytes
Partition set definition	80 bytes
Map set definition	80 bytes

For each group, there are control records totaling approximately 250 bytes.

The CICS-supplied definitions occupy data records totaling approximately 25K bytes (approximately 250 records averaging 100 bytes per record).

BACKUP AND RECOVERY PROCEDURES

You should maintain a backup version of your CSD file by using the IDCAMS EXPORT function. If the CSD file is destroyed for any reason, it can then be restored to its state at the last backup date by using the IDCAMS IMPORT function.

If your CSD file fills up while the CEDA transaction or an offline utility function is being processed, you should define a larger data set and use the IDCAMS EXPORT and IMPORT functions to recover the contents of the CSD file. It will usually be possible to EXPORT from the CSD file that has become full. Otherwise, a backup copy should be used.

JOB CONTROL STATEMENTS FOR CICS EXECUTION

If the CSD file has been defined using the sample job stream in the previous section, the job control statement for CICS execution is:

```
// DLBL DFHCSD,'cics160.csd',,VSAM,CAT=user-catalog-filename
```

CSD FILE AVAILABILITY

In order to conserve buffer space and to minimize conflicts of availability, you should specify OPEN=DEFERRED in the FCT entry for the CSD file (DFHCSD). If this is done, CICS initialization will cause the CSD file to be open for input only when GRPLIST=list is specified. The CSD file is left closed after the list specified by means of GRPLIST is installed. Subsequent use of the CEDA transaction will cause the CSD file to be opened in accordance with the way the FCT entry was coded.

MULTIPLE USERS OF THE CSD FILE WITHIN A CICS REGION

Several different CEDA transactions may access the CSD file concurrently. The last CEDA transaction to terminate normally causes the CSD file to be closed. See "Disposition of the CSD File After Abnormal Termination" on page 223 for the warning issued for an abnormal termination.

MULTIPLE USERS OF THE CSD FILE ACROSS CICS OR BATCH REGIONS

A SHAREOPTIONS value of 2 should be specified in the IDCAMS DEFINE CLUSTER command for the CSD file. Assuming that OPEN=DEFERRED is specified in the FCT for the CSD file, and that SHAREOPTIONS(2) is in the VSAM catalog, no availability conflict will occur when more than one CICS region is being concurrently initialized and the CSD file is being shared. Concurrent execution of CEDA in multiple CICS regions sharing the same CSD file will be prevented and diagnosed if their FCT entries have caused them to be opened for write access.

A read-only CSD file (as specified by means of the FCT or modified by CEMT) can be used. Only those commands requiring WRITE access to the CSD file will be diagnosed at the time of execution as unusable (message DFH4824).

DISPOSITION OF THE CSD FILE AFTER ABNORMAL TERMINATION

If a CEDA transaction terminates abnormally, the CSD file may remain open and, if it is accessed for output, will therefore prevent other regions (batch or CICS) from having WRITE access to the CSD file. In this case, CEMT should be used to close the CSD file. Ensure that no CEDA transaction is active at the time you close the CSD file.

If an abnormal termination of CICS occurs while the CEDA transaction or the DFHCSDUP offline utility is running, it will be necessary to execute the VERIFY command of the IDCAMS utilities to ensure that the CSD file is closed properly. If the abnormal termination occurs while any command other than LIST or EXPAND is being executed, it may be necessary to execute the VERIFY command of the DFHCSDUP offline utility to remove the internal locks established whenever the CSD file is updated.

PROCESSING THE CICS SYSTEM DEFINITION FILE (DFHCSDUP)

The DFHCSDUP offline utility program can be run to modify the CSD file when CICS is not running.

With the DFHCSDUP offline utility program, you can:

- INITIALIZE a CSD file ready for use by CEDA and create the standard IBM-supplied resource definitions.
- MIGRATE the contents of the PPT and PCT from a CICS load library to the CSD file.
- ERASE all the group names from a list or all of the resource definitions from a group.
- COPY all of the resource definitions in a group to another group from the same or another CSD file.
- APPEND all of the groups in a list to another list from the same or another CSD file.
- LIST to produce a hardcopy listing of selected resource definitions, groups, and lists on the CSD file.
- VERIFY to remove any internal locks left after abnormal termination.
- SERVICE to carry out maintenance on your CSD file between CICS releases, should it be required.

The inputs and outputs for the offline utility are shown in Figure 76.

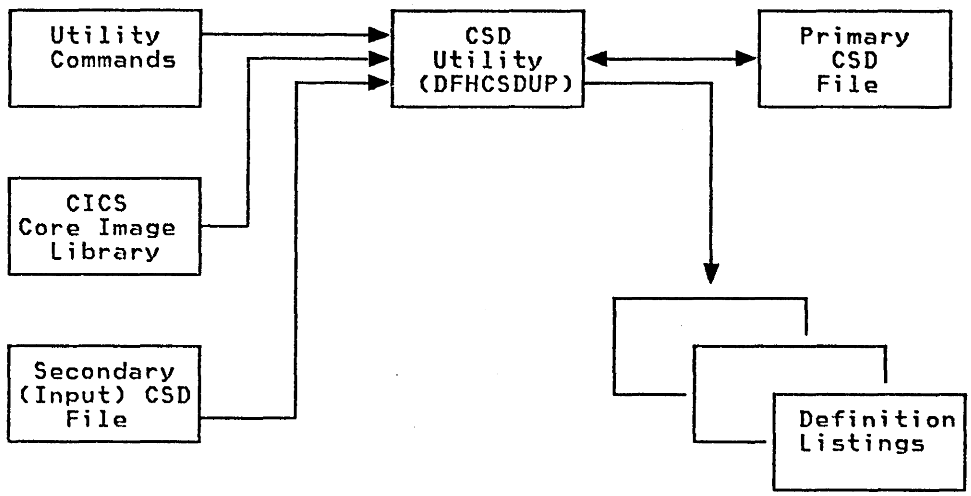


Figure 76. The DFHCSDUP Offline Utility Program

Note: The primary CSD file is always identified in the job control statements by a unique filename (DFHCSD). For this reason, its name is not specified in the utility commands. When the primary CSD file is initialized, this filename is assumed. In operations involving two CSD files, the primary file is the only one that can be updated. (For SERVICE, the new CSD file is the primary one.) You specify the filename for a secondary CSD file in those utility commands that require one.

The COPY, APPEND, and SERVICE commands refer to this secondary CSD by means of the FROMCSD or OLDCSD parameter. The value of this parameter must be the user-chosen filename defining the data set (for example, CSDF1 in the example below).

JOB CONTROL STATEMENTS TO INVOKE DFHCSDUP

The following job control statements are required to invoke the utility program:

```
// JOB CSDUP UTILITY DFHCSDUP
*
* Primary CSD file
// DLBL DFHCSD,'CICS.CSD.CLUSTER',,VSAM,CAT=user-catalog-filename
*
* Secondary CSD file examples
// DLBL CSDF1,'OTHER1.CSD.CLUSTER',,VSAM,CAT=user-catalog-filename
// DLBL FROM2,'OTHER2.CSD.CLUSTER',,VSAM,CAT=user-catalog-filename
*
* Utility Execution
// LIBDEF CL,SEARCH=(cics-cil-filename,user-cil-filename)
// EXEC DFHCSDUP,SIZE=AUTO
    .
    .
    DFHCSDUP commands
    .
    .
/*
/&
```

The commands are:

INITIALIZE

This command prepares a newly-defined data set for use as a CSD file. The standard entries for the IBM-supplied programs, transactions, and profiles are created on the CSD file. These definitions are arranged into groups, most of which correspond to the 'function groups' used by the DFHPPT and DFHPCT macros (see the CICS/VS Resource Definition Guide). The names of the IBM-supplied groups all begin with 'DFH'; for example, the resource definition online (RDO) definitions created in the RDO group DFHBMS correspond to the PCT and PPT entries in the function group identified by the macro parameter FN=BMS.

The INITIALIZE command creates the CSD file definitions for the standard resources by migrating the contents of the two load modules DFHPPI and DFHPCI. These load modules must exist in your library in assembled form (they are supplied as part of the CICS pregenerated system).

As well as setting up these definitions, a group list called DFHLIST is created. All the IBM-supplied groups are members of this list. It can be used in the same way as any group list you define yourself, as the GRPLIST parameter in the DFHSIT macro, for starting up a CICS system for use with RDO (see the CICS/VS Resource Definition Guide for information on this parameter).

The groups containing IBM-supplied resource definitions and the list (DFHLIST) cannot be modified.

Also created is a control record at the start of the CSD file. This record contains fields to identify the CICS release and the current level of service applied to the CSD file (see SERVICE command). In addition, this record has fields containing the date and time of creation of the CSD file, and the date and time the CSD was last updated. Both these fields appear on the hardcopy listing of the CSD file produced by the LIST command.

The INITIALIZE command must be executed before you can run any of the other utility commands, or the CEDA transaction. After you have initialized a CSD file, you will not need to execute this function again except, possibly, as part of the process of upgrading your CSD file for a future release of CICS.

See "Appendix D. Sample Tables and Table Entries" for a list of resource definitions in the IBM-supplied groups.

MIGRATE TABLE(tablename) {TOGROUP(groupname)}

This command is used to transfer the contents of PPT and PCT tables from a CICS core image library to the CSD file. One group is created on the CSD file for each table migrated. If the migrated table is a PPT, the group will contain program, map set, and partition set definitions. If it is a PCT, the group will contain definitions of transactions and profiles. The user core image library should be in the search chain for MIGRATE.

TABLE(tablename)

identifies the name of the table in the core image library (DFHPPTxx or DFHPCTxx).

TOGROUP(groupname)

specifies the name of the group created. No group of this name should exist on the CSD file already. If this parameter is not specified, the group created will have a name derived from the table name. For example, if the migrated table name is DFHPPT24, the name of the group created will be PPT24.

Note: When migrating large tables, remember to ensure that a sufficiently large partition has been allocated to allow the largest table to be loaded (see "Space Calculations" on page 221).

COPY GROUP(groupname1) TO(groupname2) {REPLACE|MERGE} {FROMCSD(filename)}

This command copies all the resource definitions in a group to another group. The group copied can be on the primary CSD, or it can be on another CSD file. If you are copying from another file, the secondary CSD file is identified by the value in the FROMCSD parameter.

The group named in the TO parameter (groupname2) may already exist, in which case the definitions from both groups will be included. However, the whole copy operation will fail if duplicate definitions exist in the two groups, unless the REPLACE or MERGE option is specified. REPLACE or MERGE indicates how duplicates will be handled, if they happen to occur.

GROUP(groupname1)
specifies the name of the group that is copied. You are not allowed to use a generic group name.

TO(groupname2)
specifies the name of the group to which the definitions are copied. If you are copying from another CSD file, you can give this group the same name as the one you are copying from.

REPLACE
If groupname2 already exists and duplicate definitions occur, the definitions in groupname1 replace those in groupname2.

MERGE
If groupname2 already exists and duplicate definitions occur, the original definitions in groupname2 are preserved.

FROMCSD(filename)
identifies the filename of the secondary CSD file.

Messages in the utility command listing tell you which definitions were copied, and what has happened if duplicates were found.

APPEND LIST(listname1) TO(listname2) {FROMCSD(filename)}

This command applies to group lists, and is the counterpart of COPY for groups. The names of groups that exist as elements of a list are added to another list. The list copied can be on the primary CSD file, or it can be on another CSD file. If you are appending from another file, the secondary CSD file is identified by the FROMCSD parameter.

If the list specified as the TO parameter (listname2) does not already exist, a new list of that name is created. If a list of this name already exists, the contents of listname1 are appended at the end of listname2. The ordering of the group names in both the original and appended portions of the resulting list is preserved.

No duplicate group names are allowed in a list. If any duplicate names are encountered in the APPEND operation, they are ignored. A warning message appears in the utility command listing if this happens.

LIST(listname1)
specifies the name of the list that is appended. You are not allowed to use a generic list name.

TO(listname2)
specifies the name of the list to which the group names are appended. If you are appending from another CSD file, you can give this list the same name as the one you are appending from.

FROMCSD(filename)
identifies the filename of the secondary CSD file.

ERASE GROUP(groupname)|LIST(listname)
This command erases all the resource definitions in a group, or all the group names in a list.

If you erase a LIST, the definitions of the resources within the groups contained in the list are not erased. To do this, you must also erase each GROUP individually.

You are not allowed to erase the definitions of the IBM-supplied groups and lists whose names begin with 'DFH'.

GROUP(groupname)
specifies the name of the group to be erased. You are not allowed to use a generic group name.

LIST(listname)
specifies the name of the list to be erased. You are not allowed to use a generic list name.

LIST {ALL|GROUP(groupname)|LIST(listname)} {OBJECTS}
This command produces various reports tabulating the current status of the CSD file for hardcopy printing. The listings generated are included on the output listing, along with the messages produced by the utility command processing.

Tabulations of the contents of groups can be produced at two levels of detail. Without the OBJECTS option, just the names and types of the resources in each group are tabulated. With the OBJECTS option, a listing of all the properties of each resource (or "object") is produced as well.

Generic specifications for the group name and list name are permitted for some of the LIST commands. If this is done, the contents of all the qualifying groups or lists are printed.

The listings produced by the various commands are:

- LIST ALL
  - Names of defined lists and groups
  - Summary of lists
  - Summary of groups

  This prints summaries of all the definitions of lists and groups that exist on the CSD file.

- LIST ALL OBJECTS
  - Names of defined lists and groups
  - Summary of lists
  - Summary of groups
  - Objects in groups

This prints summaries of all the definitions of lists and groups that exist on the CSD file, together with a listing of the properties of the resources in all the groups.

- LIST GROUP(groupname) (group name may be generic)

  - Summary of groups. This summarizes the names of all the resources in one or more groups. They are organized within each group into resource type categories (for example, map sets, programs, and so on).

- LIST GROUP(groupname) OBJECTS (group name may be generic)

  - Summary of groups (see above).

  - Objects in groups. This enables you to tabulate the properties of the resources, again organized according to resource type. The creation time for each resource is given, together with all its attributes, as originally set up by using DEFINE and ALTER commands in the CEDA transaction, or by migrating it from a CICS table. In the case of TRANSACTIONs and PROFILEs, their properties are arranged in the same subcategories as appear on the CEDA DEFINE screen.

- LIST LIST(listname) (list name may be generic)

  - Summary of lists. The contents of one or more group lists are tabulated. The groups appear in the same sequence as their position in the list. This order will have been determined by the commands ADD and/or APPEND, which were used in the CEDA transaction to build the list.

- LIST LIST(listname) OBJECTS (generic list name not allowed)

  - Summary of lists (see above).

  - Objects of groups in list. This enables you to tabulate the properties of all the resources that will be defined in a CICS system at startup time. These are identified by a single LIST name in the GRPLIST=listname parameter in the system initialization table. The names of each group in the list appear in the "Summary of Lists". Then, for each group contained in the list, the properties of the individual resources in the group are tabulated.

    The "Objects in Groups in Lists" tabulation has the groups arranged in the same order as they were added to the group list. This order is important in the event of duplication, when definitions of the same resource may exist in more than one group. If a list of this type is used at CICS startup time, the resource definitions used when there is duplication will be those belonging to the group occurring latest in the list.

VERIFY
This command is used to remove internal locks on groups and lists. It should be executed only when the CSD file is not in use, preferably when no CICS systems that may use it are running. The VERIFY command acts upon the whole CSD file, and is for use in the extreme condition where internal lock records have been left behind. These records are normally removed when a function that changes the CSD file has been completed. However, this may not have happened if there was a system failure when the CEDA transaction was executing, or

if the execution of an off-line utility failed to complete. As a result, these locks may prevent CEDA users from accessing certain groups and lists on the CSD file. The VERIFY command should remedy the situation if the CEDA UNLOCK command cannot be used successfully.

Note that the VERIFY command removes only the internal locks. The normal user locks applied by the LOCK command in the CEDA transaction are unaffected.

SERVICE OLDCSD(filename) LEVEL(nnn)
It might occasionally be necessary between CICS releases to apply a SERVICE routine, to carry out preventive or corrective maintenance to your CSD file. This is done by loading and running a special SERVICE program (DFHCUS1B). This program is supplied by IBM as a separately-loadable module.

The function of the SERVICE command is to create a new copy of the CSD file, from the existing CSD file. All the definitions will be preserved, with the corrections (if any) applied.

Associated with your CSD file is a current service level, initially set to 000 when the file was initialized. Applying the SERVICE routine causes the service level to be incremented in steps of 1, from a "current level" to a "target level."

OLDCSD(filename)
specifies the filename of the current CSD file, which is treated as the secondary CSD file.

LEVEL(nnn)
specifies the target service level to which the CSD file is to be upgraded. This value must be 1 higher than the current level of the OLDCSD file. It is specified as a 3-character integer, for example, LEVEL(001).

RULES FOR THE SYNTAX AND PREPARATION OF UTILITY COMMANDS

The commands are entered in columns 1-71 of 80-character input records. Leading blanks are ignored, and blanks between keywords and operands are permitted.

Comment records are permitted, and these must have an asterisk in column 1. Comment material is not permitted on a record that contains a command. Blank records are ignored.

A utility command must be confined to a single record, as there is no facility for continuation of a command over more than one record.

The conventions for the names of groups and lists must be followed in the coding of the GROUP, LIST, TO and TOGROUP parameters. (See the CICS/VS Resource Definition Guide for a description of valid group names and list names.) If you use a generic specification for the GROUP or LIST parameter in the LIST command, the symbols * and + can be used in the same way as for CEDA.

The FROMCSD and OLDCSD parameters must contain a valid filename.

Here is an example of a valid sequence of utility commands, as an illustration of these rules:

```
*                                SET UP INITIAL CSD FILE
INITIALIZE
*
LIST LIST(DFHLIST) OBJECTS
*                                    MIGRATE MAIN TABLES
*
MIGRATE TABLE(DFHPCTM1)
*
      MIGRATE TABLE(DFHPPTM1)
*
MIGRATE TABLE  (DFHPCTM2)
*
*
MIGRATE TABLE(DFHPCTM3) TOGROUP(SETMC3)
MIGRATE TABLE(DFHPPTM3) TOGROUP(SETMP3)
*
LIST GROUP(PPTM1)
LIST GROUP(SETM*)
*
*                                   CREATE GROUP PCTZ4
COPY GROUP(PCTM1)  TO(PCTZ4)
COPY GROUP(SETMP3) TO(PCTZ4) REPLACE
LIST GROUP(P++M+)
*                                   CREATE LIST MODLIST
*
APPEND LIST(TESTLIST) TO(MODLIST) FROMCSD(CSDF1)
APPEND LIST(SECLIST)  TO(MODLIST) FROMCSD(CSDF1)
APPEND LIST(DFHLIST)  TO(MODLIST)
*
LIST ALL OBJECTS
*
```

Chapter 4.11. User File Definitions

For each entry in the file control table, label information and assignment information must be included in the CICS startup job stream. (An exception is files for which OPEN=DEFERRED is specified, and which you do not intend to use during the current execution of CICS.) The file name must be the same as that specified in the DATASET operand of the DFHFCT TYPE=DATASET macro instruction. For example:

Virtual Storage Access Method - VSAM

```
// DLBL     CVSSPAC,'cics.space',exp-date,VSAM,CAT=user-catalog-filename
// DLBL     TCTVS1,'vsam.file.1',,VSAM,CAT=user-catalog-filename
// DLBL     TCTVS2,'vsam.file.2',,VSAM,CAT=user-catalog-filename
```

Indexed Sequential Access Method - ISAM

```
// DLBL     ISAMFIL,'ISAM.FILE',exp-date,ISE
// EXTENT   SYS015,CICSIS,4,1,1280,01
// EXTENT   SYS016,CICSIS,1,2,1300,60
// EXTENT   SYS017,CICSIS,2,3,1360,20
// ASSGN    SYS015,cuu
// ASSGN    SYS016,cuu
// ASSGN    SYS017,cuu
```

Direct Access Method - DAM

```
// DLBL     DAMFILE,'DAM.FILE',exp-date,DA
// EXTENT   SYS013,CICSDA,1,0,2500,50
// ASSGN    SYS013,cuu
```

Chapter 4.12. Terminal Definitions

Terminals are defined to CICS in macro instructions in a terminal control table (TCT). Only those terminals defined in the particular TCT selected during CICS initialization are made known to CICS and thereby become CICS resources.

These terminals must also be defined to VSE and to the access method as follows:

- VTAM terminals are defined to VTAM identified by node names. No job control statements are needed.
- BTAM terminals must be defined to VSE in job control statements. You may also need to code BTAM DFTRMLST macro instructions to provide polling or addressing information.
- Sequential devices also require job control statements.
- The VSE console is identified to CICS by a terminal identifier (CNSL). No additional job control statements are needed.

VTAM TERMINALS

When running CICS with VTAM, you must ensure that the application identifier (APPLID) of the CICS system is defined to VTAM in an APPL statement. The APPLID in CICS may have been specified on the DFHTCT TYPE=INITIAL macro instruction, or in the system initialization table or as a system initialization override. The APPLID supplied in the sample system initialization tables is DBDCCICS.

A VTAM terminal is defined to CICS by a DFHTCT TYPE=TERMINAL macro instruction. It specifies a TRMIDNT that is the identification used by a CICS application program, and a NETNAME that is the VTAM node name defined in the VTAM tables. VTAM terminals are controlled by the VTAM partition.

Consider the sample VTAM terminal control table DFHXTCVS copy book:

```
         TITLE  'DFHXTCVS - COPYBOOK ENTRIES FOR VTAM SAMPLE TCT'
***********************************************************************
* L77A - VTAM SNA LUTYPE2.               NETNAME - D72L301.          *
* L77D - VTAM NON-SNA 3270.              NETNAME - D72L304.          *
* L860 - VTAM NON-SNA 3270 PRINTER.      NETNAME - P42L308.          *
***********************************************************************
VTRM1    DFHTCT TYPE=TERMINAL,TRMIDNT=L77A,TRMTYPE=LUTYPE2,TRMMODL=2,  *
               TIOAL=(1500,1920),RELREQ=(NO,YES),                      *
               BUFFER=1536,                                            *
               FEATURE=(SELCTPEN,AUDALARM,UCTRAN,DCKYBD),              *
               CHNASSY=YES,                                            *
               NETNAME=D72L301,                                        *
               PRINTTO=(VTRM3),                                        *
               GMMSG=NO,                                               *
               ACCMETH=VTAM,TCTUAL=32,                                 *
               CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM2    DFHTCT TYPE=TERMINAL,TRMIDNT=L77D,TRMTYPE=3270,TRMMODL=2,     *
               CLASS=(CONV,VIDEO),TIOAL=1500,RELREQ=(NO,YES),          *
               FEATURE=(SELCTPEN,AUDALARM,UCTRAN,DCKYBD),              *
               NETNAME=D72L304,                                        *
               PRINTTO=(VTRM3),                                        *
               GMMSG=YES,                                              *
               ACCMETH=VTAM,TCTUAL=32,                                 *
               CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM3    DFHTCT TYPE=TERMINAL,TRMIDNT=L860,TRMTYPE=3270P,TRMMODL=2,    *
               NETNAME=P42L308,                                        *
               RELREQ=(YES,YES),                                       *
               CLASS=(CONV,VIDEO),TIOAL=1500,TCTUAL=32,                *
               ACCMETH=VTAM,CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
```

This example defines three VTAM terminals:

1. The macro labeled VTRM1 defines an SNA LU type 2 terminal. This terminal is known to a CICS application program by the identification L77A. It is defined to VTAM in a VTAM LU macro with the node name D72L301.

2. The macro labeled VTRM2 defines a non-SNA 3270 display terminal. This terminal is known to a CICS application program by the identification L77D. It is defined to VTAM with the node name D72L304.

3. The macro labeled VTRM3 defines a non-SNA 3270 printer. This terminal is known to a CICS application program by the identification L860. It is defined to VTAM with the node name P42L308.

BTAM TERMINALS

LOCAL AND REMOTE BTAM TERMINALS

There are two ways you can define local and remote BTAM terminals to CICS:

1. You can code three macro instructions:

   ```
   DFHTCT TYPE=SDSCI    to define the transmission control unit
   DFHTCT TYPE=LINE     to define the line
   DFHTCT TYPE=TERMINAL to define the terminals on that line
   ```

2. You can code a single DFHTCT TYPE=GPENTRY macro instruction. Consider the sample BTAM terminal control table DFHXTCBI copy book:

```
TITLE  'DFHXTCBI-BTAM TCT-COPYBOOK FOR LOCAL TERMINALS'
DFHTCT TYPE=GPENTRY,               1 LINE FOR:                  X
       GPTYPE=3270L,                  1 LOCAL 3270 DEVICE       X
       LININL=160,                    TIOA LENGTH               X
       LINELST=(020),                 ADDRESS                   X
       TRMFEAT=(DSA),                 FEATURE                   X
       TRMSTAT=(A),                   TERMINAL STATUS           X
       TRMMODL=(2A),                  TERMINAL MODEL            X
       TRMUAL=(32),                   USER AREA LENGTH          X
       TRMIDNT=(L77A)                 TERMINAL NAME
DFHTCT TYPE=GPENTRY,               1 LINE FOR:                  X
       GPTYPE=3270L,                  3 LOCAL 3270 DEVICES      X
       LININL=160,                    TIOA LENGTH               X
       LINELST=(021,022,027),         ADDRESSES                 X
       TRMFEAT=(DSA,DSA,P),           FEATURES                  X
       TRMSTAT=(A,A,A),               TERMINAL STATUS           X
       TRMMODL=(2A,2A,2C),            TERMINAL MODELS           X
       TRMUAL=(32,32,32),             USER AREA LENGTH          X
       TRMIDNT=(L77B,L77C,L86P)       TERMINAL NAMES
```

A BTAM terminal is known to a CICS application program with the identification given by the TRMIDNT operand in the DFHTCT TYPE=TERMINAL macro, or its corresponding entry in the TRMIDNT list in the DFHTCT TYPE=GPENTRY macro.

The BTAM terminal is mapped to a physical address by means of the LINELST operand in the DFHTCT TYPE=SDSCI or TYPE=GPENTRY macros. The LINELST operand has the format (nnn[,...],...) where SYSnnn is the symbolic unit address of the corresponding terminal.

You must code ASSGN job control statements to assign BTAM terminals. In the above example, four BTAM local terminals are known to CICS application programs as L77A, L77B, L77C, and L86P, with symbolic unit addresses SYS020, SYS021, SYS022, and SYS027 respectively.

You might code the following job control statements for this example:

```
// ASSGN  SYS020,0A1
// ASSGN  SYS021,0A2
// ASSGN  SYS022,0A3
// ASSGN  SYS027,0A4
```

where 0A1, 0A2, 0A3, and 0A4 are the physical addresses of the terminals.

To allow for future expansion, you may choose to define terminal entries in the TCT for terminals that are not yet physically installed. For each of these "dummy" terminals, you would code:

```
// ASSGN SYSnnn,IGN
```

where SYSnnn is the value specified in the matching LINELST parameter of the DFHTCT TYPE=SDSCI or TYPE=GPENTRY macro instruction.

The TRMSTAT operand in the DFHTCT TYPE=TERMINAL macro for the "dummy" terminal should be 'OUT OF SERVICE' (for TYPE=GPENTRY, the corresponding TRMSTAT operand should be X).

Note: The maximum number of such "dummy" terminals that may be assigned per line is **eight**. If nine or more terminals are assigned IGN in the job stream for the same line, the line will be placed irretrievably "out of service" at system startup.

Here is another example of the job control statements that might be required:

```
// ASSGN  SYS020,0A1          local 3270
// ASSGN  SYS022,021          remote 3270
// ASSGN  SYS023,022          remote 3270
// ASSGN  SYS025,025          TWX switched 2703/2
// ASSGN  SYS027,047          2740  2703/2
// ASSGN  SYS028,029          1050  2703/1
// ASSGN  SYS029,060          1050  2703/2
```

where SYSnnn is the value specified in the matching LINELST parameter of a DFHTCT macro.

REMOTE BTAM TERMINALS ONLY

For remote BTAM terminals, you may need to include BTAM DFTRMLST macro instructions in the terminal control table to define the polling and addressing characters to CICS. The <u>CICS/VS Resource Definition Guide</u> shows TCT examples containing these macros.

SEQUENTIAL DEVICES

A pair of input and output sequential data sets can be used to simulate a terminal to CICS. This can be to test an application program before the intended terminal becomes available. Again, there are two ways you can define these sequential devices:

1. You can code four macro instructions:

DFHTCT TYPE=SDSCI	to describe the input sequential data set
DFHTCT TYPE=SDSCI	to describe the output sequential data set
DFHTCT TYPE=LINE	to describe the I/O combination
DFHTCT TYPE=TERMINAL	to describe the I/O combination

2. You can code a single DFHTCT TYPE=GPENTRY macro instruction.

If you are using the first method, consider the sample terminal control table DFHXTCTS copy book:

```
       TITLE  'DFHXTCTS - COPYBOOK TCT ENTRIES FOR SEQUENTIAL (CRLP) TX
               ERMINAL ENTRIES'
***********************************************************************
* * *                          D  T  F  S                       * * *
***********************************************************************
         DS     0D
         DC     CL32'INPUT AND OUTPUT DTFS'
         DFHTCT  TYPE=SDSCI,                                          *
                DEVADDR=SYSRDR,                                       *
                DEVICE=2540,                                          *
                DSCNAME=READER
         DFHEJECT
         DFHTCT  TYPE=SDSCI,                                          *
                DEVADDR=SYSLST,                                       *
                DEVICE=1403,                                          *
                DSCNAME=PRINTER
         DFHEJECT
***********************************************************************
* * *                  L I N E   A N D   T E R M I N A L        * * *
* * *                  D E F I N I T I O N                      * * *
***********************************************************************
         DS     0D
         DC     CL64'T E R M I N A L   A N D   L I N E   E N T R I E S'
***********************************************************************
         DS     0D
         DC     CL32'S A M A  LINE AND TERMINAL ENTRY'
         DFHTCT  TYPE=LINE,                                           *
                ACCMETH=SEQUENTIAL,                                   *
                TRMTYPE=CRLP,                                         *
                ISADSCN=READER,                                       *
                OSADSCN=PRINTER,                                      *
                INAREAL=88,                                           *
                DUMMY=DUMMY
         DFHTCT  TYPE=TERMINAL,                                       *
                TRMIDNT=SAMA,                                         *
                TRMPRTY=32,                                           *
                TRMSTAT=(TRANSCEIVE),                                 *
                DUMMY=DUMMY
***********************************************************************
* * *                  F I N A L I Z E                          * * *
***********************************************************************
         DS     0D
         DC     CL32'T C T   WAIT LIST'
```

A card reader and a line printer are defined to simulate a terminal. The DEVADDR operands in the DFHTCT TYPE=SDSCI macros (or the GPSEQLU list in the TYPE=GPENTRY macro) specify the symbolic unit addresses of the devices. In this example, they are SYSRDR and SYSLST respectively.

This I/O combination simulates a "terminal" known to a CICS application program by the name SAMA. Input to the application program is submitted through the card reader (SYSRDR), and output to the "terminal" is sent to the line printer (SYSLST).

Two DASD data sets can also be used to simulate a terminal.

You must code DLBL, EXTENT, and ASSGN job control statements for these files. The file names in the DLBL statements are specified in the DSCNAME operands in the DFHTCT TYPE=SDSCI macro (or the GPNAME list in the TYPE=GPENTRY macro). Consider the following example:

```
DFHTCT TYPE=SDSCI,                                          *
       DEVADDR=SYS001,                                      *
       DEVICE=2314,                                         *
       DSCNAME=DISKIN1
DFHTCT TYPE=SDSCI,                                          *
       DEVADDR=SYS006,                                      *
       DEVICE=2314,                                         *
       DSCNAME=DISKOT1
DFHTCT TYPE=LINE,                                           *
       ACCMETH=SEQUENTIAL,                                  *
       TRMTYPE=DASD,                                        *
       ISADSCN=DISKIN1,                                     *
       OSADSCN=DISKOT1,                                     *
       INAREAL=80
DFHTCT TYPE=TERMINAL,                                       *
       TRMIDNT=SAMB,                                        *
       TRMPRTY=11,                                          *
       TRMSTAT=(TRANSCEIVE,'OUT OF SERVICE')
```

For this example, you might code the following job control statements:

```
// DLBL    DISKIN1,'SIMULATED.TERMINAL.IN',0,SD
// EXTENT  SYS001,CICSWK,,,3850,10
// ASSGN   SYS001,201

// DLBL    DISKOT1,'SIMULATED.TERMINAL.OUT',0,SD
// EXTENT  SYS006,CICSWK,,,3860,10
// ASSGN   SYS006,201
```

where 201 is the physical address of the IBM 2314 Direct Access Storage Device.

This I/O combination simulates a "terminal" known as SAMB to a CICS application program. Input from this "terminal" is submitted in the DISKIN1 data set. Output to the "terminal" is written on the DISKOT1 data set.

Note: End-of-file does not terminate sequential input. Therefore, the last transaction should be one that quiesces the terminal, such as CSSF GOODNIGHT.

VSE CONSOLE

The VSE console can be defined to CICS in one of two ways:

1. By coding DFHTCT TYPE=SDSCI, DFHTCT TYPE=LINE, and DFHTCT TYPE=TERMINAL macro instructions.

2. By coding a single DFHTCT TYPE=GPENTRY macro instruction. An example of this is provided in the sample terminal control table DFHXTCTC copy book:

```
      TITLE  'DFHXTCTC-SAMPLE TCT-COPYBOOK FOR CONSOLE'
         DFHTCT TYPE=GPENTRY,                                        X
               GPTYPE=CONSOLE,       CONSOLE SUPPORT                 X
               TRMIDNT=CNSL,         TERMINAL NAME                   X
               TRMUAL=32,            USER AREA LENGTH                X
               LININL=80             TIOA LENGTH
```

The TRMIDNT must be CNSL for VSE console support. SYSLOG is always forced as the symbolic unit address.

This second method is the one used with the sample startup job stream supplied in the CICS source statement library.

Chapter 4.13. DL/I Definitions

In the CICS startup job stream, you must define all DL/I data bases that are to be referenced during CICS execution. The job control statement to define one such data base might be:

```
// DLBL dli-filename,'database-file-id',,VSAM,CAT=user-catalog-filename
```

APPLICATION CONTROL TABLE (ACT)

The logical connection between CICS application programs and the DL/I data bases they reference is provided by the DL/I application control table (ACT). The ACT is generated by DL/I DOS/VS into the core image library as module DLZNUC. You can specify a 2-character suffix for the ACT in the DLZACT TYPE=INITIAL macro, in which case the generated module is DLZNUCxx (xx being the suffix).

To initialize CICS with DL/I, you must specify DLI=YES or DLI=xx in the DFHSIT macro, or as an initialization override. DLI=YES will cause CICS to load module DLZNUC, the unsuffixed version of the ACT. DLI=xx will cause CICS to load module DLZNUCxx, a suffixed version of the ACT (where xx is the suffix).

STORAGE LAYOUT CONTROL (SLC) TABLE

The sequence in which DL/I modules are to be loaded during initialization can be specified using the optional DL/I storage layout control (SLC) table.

If you generate an SLC table for use during CICS initialization, you must specify its phase name in the SLC operand of the DLZACT TYPE=CONFIG macro when you generate the DL/I application control table.

Further information on DL/I is given in Chapter 2.6 of this book, and in the DL/I DOS/VS Resource Definition and Utilities book.

Part 5. Running the CICS System

This part describes how to create a startup job stream to start up and run CICS. The following chapters describe specific aspects of running CICS:

- Overview of CICS operations (Chapter 5.1)
- The CICS startup job stream (Chapter 5.2)
- System initialization parameters (Chapter 5.3)
- Operating procedures (Chapter 5.4)
- Actions to be taken after an emergency restart (Chapter 5.5).

The CICS/VS Recovery and Restart Guide gives general background information on the choices available for recovery and restart, and the CICS/VS Operator's Guide gives information on the terminal transactions provided by CICS for running the system.

PART 5

Chapter 5.1. Overview of CICS Operations

The steps involved in running a CICS system include:

- Planning your system (that is, preparing CICS data sets, tables, and management modules)
- Ensuring the previous execution is complete
- Preparing the startup job stream
- Checking your VSE system
- Running the startup job stream
- Operating the CICS system
- Shutting the system down
- Completing the run by carrying out postprocessing operations.

PLANNING YOUR CICS SYSTEM

SELECTING THE FUNCTION YOU NEED

Before running a CICS system, you must decide on the function that you require to be available for your users.

After the function has been determined, there are several other preparation steps that may need to be performed. These include:

PREPARING CICS DATA SETS

Data sets are discussed in Part 4 of this book. Depending on the function that you require in your system, you may need some or all of these data sets.

Some data sets (for example, journals and the restart data set) require preparation before being used in a CICS startup. If you are using journals either on disk or on tape, you must preformat the journals once before they are used in a CICS startup. The job needed for preformatting journals is described in Chapter 4.5.

PART 5

PREPARING RESOURCE DEFINITION TABLES

For most resources, the tables are created by separate assemblies, and are stored in the core image library under the name DFHFCTxx, DFHDCTxx, and so on. For programs and transactions, you may wish to use resource definition online (RDO), in which case the definitions will be in the CSD file.

Chapter 2.3 and 3.2 tell you how to assemble control tables. See the CICS/VS Resource Definition Guide for additional information.

PREPARING MANAGEMENT MODULES

In most cases, you will be selecting from the pregenerated modules to create a CICS system that supports the function that you need. These modules are listed in Appendix C.

In some unusual cases, you may need to generate your own version of a management module that is customized to your requirement (see Chapter 2.5 and the CICS/VS Customization Guide).

The module suffixes are then coded in your SIT or as system initialization overrides. Where the modules are read-only, you should consider placing them in the SVA.

ENSURING PREVIOUS EXECUTION IS COMPLETE

After the CICS system has been run, you may have several data sets to process. In normal circumstances, these will be processed either in parallel with the CICS execution or after CICS has been shut down.

In general, these should be separate jobs, rather than new job steps in the CICS job. This means that they can be run even if the CICS job step fails.

The printing of dumps, statistics, and trace will normally be performed when the CICS system is shut down. In creating your own job stream, you have to decide what procedure you are going to follow if you have a system failure, for example.

You may have separate jobs that the operator is told to run prior to starting up a production system. A sample job for printing dumps is given in Chapter 4.9, and for printing automatic statistics in Chapter 4.7.

The areas that you will want to consider include:

- Printing or saving any dumps (see Chapter 4.9).
- Printing or saving auxiliary trace (see Chapter 4.8).
- Printing or saving statistics (see Chapter 4.7).
- Verifying the user and system VSAM files (see Chapter 2.4).
- Running DFHTEOF. If you have had a system failure on a user journal on tape, you will need to "repair" the tape by running the tape end-of-file (DFHTEOF) utility program.
- Saving disk journals. You may have your own procedure for copying disk journals to an archive tape.

PREPARING THE STARTUP JOB STREAM

This is discussed in Chapter 5.2. CICS is started up in a VSE partition like any other operating system job. The execution of the startup job stream initiates a process, referred to as **CICS system initialization**, which must be completed before any transactions can be run. The completion of CICS initialization is indicated by a message at the system console:

```
DFH1500  CONTROL IS BEING GIVEN TO CICS
```

SYSTEM INITIALIZATION PARAMETERS

To start or restart a CICS system, a startup job stream is submitted to the VSE system as a normal batch job. Flexibility is provided at start up time or system initialization time by the facility of overriding the parameters in the system initialization table (SIT). The SIT supplies the system initialization program with the information necessary to initialize CICS to suit your environment. During initialization, you can override some parameters on the operating system console. See "System Initialization Override Parameters" on page 270 for further information.

The function available in the system is based on the SIT that you provide, together with any changes that you make to the system initialization process by using system initialization overrides. There are several ways of entering system initialization overrides, and these are all described in Chapter 5.3.

In general, after a particular CICS system has become the "production system", you should perform the startup with as little operator intervention as possible.

SELECTING TYPES OF STARTUP

If CICS is being run with CICS-provided keypointing and recovery capabilities, the system initialization process can be used to ensure that the CICS environment is restored to the state it was in after a previous run, and that recoverable entities such as files and data bases are restored to a logically consistent state.

Types of startup are discussed in more detail in Chapter 2.4 and in the discussions on the restart data set and system journal in Part 4. If you are running with these functions, your normal START option will be START=AUTO.

CHECKING YOUR VSE SYSTEM

Before you run the CICS startup job stream, you will need to check that your VSE system is prepared for CICS execution.

SELECTING MODULES FOR THE SVA

You should select the modules to be included in the SVA (see Chapter 2.6 and Appendix B). If you are running a production CICS system, you are recommended to put the management modules that you need in the SVA. This gives additional protection. For a more detailed discussion of the procedure needed to load the SVA, see Chapter 2.6.

ENSURING THAT THE VTAM PARTITION IS ACTIVE

If you are running with VTAM-supported terminals, you will normally want the VTAM partition to be up before the CICS job stream is run.

RUNNING THE STARTUP JOB STREAM

As part of the preparation of your job stream, you will need to decide which partition you are going to use, and calculate its size.

SELECTING THE PARTITION

The CICS partition must have a high priority if it is to execute efficiently. The priority should be greater than any batch partitions but must be lower than VTAM.

A VSE partition is allocated for the execution of CICS. The partition is logically divided into three main storage allocations as follows:

- Nucleus
- Dynamic storage area
- GETVIS space.

A pictorial representation of the way the partition is divided might be as follows:

N U C	D S A	N U C	GETVIS

NUCLEUS AND DYNAMIC STORAGE AREA

The **nucleus** consists of the CICS modules specified in the CICS startup job stream (thus TDP=5$ specified as a SIT override will cause DFHTDP5$ to be loaded as part of the nucleus). In addition to these modules, the tables you specify (and any programs specified in the program control table as resident) are loaded, and the required control blocks are created. For the purposes of this discussion, the nucleus also holds non-VSAM buffers, access method modules, DL/I modules, and certain other items.

After the nucleus has been loaded, any storage left within the partition becomes the **dynamic storage area**. This storage is available for running transactions that are entered from user terminals, for holding nonresident applications when they are needed, and for terminal input/output buffers.

GETVIS

GETVIS space must be available for use by other VSE components that are used by CICS. In particular, GETVIS space is used by VSAM to hold control blocks for any OPEN VSAM files and also to contain VSAM buffers. In some cases, other access methods also use GETVIS space.

CALCULATING THE STORAGE FOR YOUR SYSTEM

You have to calculate the VSE ALLOC value for the CICS partition. In addition, you have to calculate a value for the EXEC DFHSIP SIZE parameter. The relationship between these two parameters can be represented as follows:

◄------------ VSE ALLOC value -----------------►

N U C	D S A	N U C	GETVIS

◄-------- DFHSIP SIZE value --------►

The <u>CICS/VS Performance Guide</u> gives detailed information on how to determine CICS storage requirements. However, when bringing up your system for the first time, you probably do not want to calculate the numbers exactly. You should start with some rough estimates and then refine them if necessary.

Size of the Nucleus

As noted earlier in this book, the nucleus of a full-function CICS/DOS/VS system requires approximately 450-500K bytes. This size will be reduced by the sizes of the modules that you wish to place in the SVA. If you plan to replace any major CICS function modules by dummy modules, the size of the nucleus will again be correspondingly smaller. The nucleus will, however, be increased by the lengths of your tables and resident application programs.

For any given combination of tables and options, you can determine the nucleus size by starting up CICS. The nucleus size is the difference between the value given for the SIZE parameter of the EXEC DFHSIP statement in the startup job stream, and the value stated in the system console message "SUBPOOL SIZE FOR THIS START-UP IS..." after the job stream has been run. (If, for example, a SIZE=1200K is specified in the job stream, and the message "SUBPOOL SIZE FOR THIS START-UP IS 636K" is sent to the console as part of the startup messages, the nucleus size will be 564K.)

Size of the DSA

Estimating the DSA requirement is more complex. A small CICS system may need as little as 100K. Systems with high message rates or applications that make heavy use of dynamic storage will require much more.

To obtain a very rough idea of the DSA size, add together the following items, which account for the largest demands on this area:

- Task storage for all tasks running at a given time. Task storage includes control blocks like the task control area and any user GETMAINs that may be issued. (A command-level transaction using BMS requires about 8.5K bytes for control blocks alone.) Allow space for the average number of concurrent tasks running, multiplied by their average storage requirements.

- Space for nonresident application programs. Estimate the sum of the sizes of all such programs that might be in use at any one time. Remember that, when one program links to another, both are active.

- Temporary storage items kept in main storage. Estimate the average number of these items in existence at any given time, and multiply by their average size.

- BTAM telecommunication buffers. Allow one buffer of INAREAL size for each remote line being polled.

- Storage cushion size (SCS) specified in the SIT.

  For example, suppose that your CICS system allows as many as ten tasks to be active at the same time (AMXT=10 in the SIT). This means that a peak of ten tasks may be active at one time, plus some suspended tasks (as many as the difference between the MXT value and the AMXT value in the SIT). For this example, allow for five suspended tasks, making a total of 15 tasks.

  Most of the applications in your hypothetical system are written at command level and use BMS — allow 8.5K in control blocks for each application. User GETMAINs average 1.5K for each task, bringing the total requirement for each task to 10K, or 150K for all tasks.

  Of these tasks, you estimate that at least two thirds will use resident programs and maps. For the remaining five, however, you need to assume the worst possible combination of transactions, in terms of programs and maps. Suppose that you estimate that, together, they may invoke ten programs and maps with an aggregate requirement of 150K.

  There are ten BTAM lines in your hypothetical system, with INAREAL=2000 specified — this requires a further 20K.

  Temporary storage (main) is not used, but the storage cushion is specified as 20K. The overall total estimate for the size of the DSA for your hypothetical system is, therefore, 340K.

One recommended procedure for determining the dynamic storage requirement is to estimate it as described above, and then to double or treble this value when calculating the value for the EXEC DFHSIP SIZE parameter. This will allow for the many other uses of the DSA, peak load effect, and storage fragmentation. In addition, specify an over-large SCS value in the SIT (100-200K), and start up the CICS system.

The size of the DSA is given in the DFH1500 message "SUBPOOL SIZE FOR THIS START-UP IS ---K" in the system console messages after the startup job stream has been run.

Now reduce the SIZE value until a short-on-storage condition is reached (see the CICS/VS Performance Guide for information on how to recognize this condition). Respecify the SCS value in the SIT to a lower value. Then reduce the SIZE value again by a similar but smaller amount (to reduce the DSA allowance). Some contingency factor must be applied: the size of this contingency factor depends on the individual installation.

Size of the GETVIS Area

The size of the GETVIS area is the difference between your actual partition size specified in the VSE ALLOC statement and your SIZE value specified in the EXEC DFHSIP statement.

The GETVIS allocation must meet all demands made both during and after CICS/DOS/VS initialization. A rough estimate can be obtained by totaling the VSAM buffer requirements for your installation.

Except where shared resources are specified, for every **key-sequenced** VSAM file specified in the file control table (FCT), CICS will allocate the **largest** of:

- The size you specify in the FCT BUFSP operand, or
- One data buffer and one index buffer per **string**, plus one additional data buffer, or
- The number of buffers specified in the FCT BUFND and BUFNI operands.

The minimum allocation is one index and two data buffers.

For **entry-sequenced** data sets, the rule is the same, except that there are no index buffers and there is just one data buffer per string.

Remember to include two buffers for temporary storage (three if you are using CICS recovery facilities) and one for transient data if these facilities are to be used.

Buffer size is equal to the control interval size. After the buffer requirements have been calculated, a contingency factor should be applied to allow for VSAM control blocks and non-VSAM demands on GETVIS.

For VTAM users with large networks, this estimate should also be adjusted to allow 0.35K for each terminal that might be in session at a given time.

OPERATING THE CICS SYSTEM

DFH1500 CONTROL IS BEING GIVEN TO CICS

After the startup process has been completed, end users will be able to enter transactions from any terminals that are connected to CICS. Typically, the first transaction to be entered will be the sign-on transaction CSSN. This is described in the CICS/VS Operator's Guide.

MASTER TERMINAL

The terminals connected to CICS will normally include the VSE system console, which you may choose to make into a "master terminal", although in many installations, the person who is responsible for controlling the CICS system will use a 3270-type display. Normally, the master terminal operator will be an individual responsible for the system. You may, if you wish, authorize more than one operator to run the CEMT transaction.

Details of the CEMT transaction and other CICS-provided transactions are given in the CICS/VS Operator's Guide.

SHUTTING DOWN THE CICS SYSTEM

CICS may be terminated from the master terminal by using the CEMT (or the CSMT) master terminal transactions.

The master terminal operator can initiate either a normal or immediate shutdown.

NORMAL
This shutdown is requested by the master terminal operator by entering:

```
CEMT PERFORM SHUTDOWN
```

It allows all transactions to complete normally, and the system is terminated in an orderly manner. System status is recorded by CICS on the restart data set in a "warm keypoint" for possible later use in a warm start. A subsequent AUTO start of CICS will result in warm start procedures being invoked automatically.

IMMEDIATE
If there is a real need to shut down the CICS system immediately, the master terminal operator can enter:

```
CEMT PERFORM SHUTDOWN IMMEDIATE
```

This command should be entered only in an unusually serious situation, because any transactions that are being run will be abnormally terminated, and no attempt will be made to quiesce the system before shutdown begins. When an immediate shutdown is forced, no "warm keypoint" is recorded. The next START=AUTO will result in an emergency restart.

After the master terminal operator has initiated a normal or immediate shutdown, the following message is displayed on the system console and on the master terminal:

```
DFH1701 - C.I.C.S. IS BEING TERMINATED
```

PART 5

UNCONTROLLED SHUTDOWN

In addition to a shutdown initiated by the master terminal operator, an **UNCONTROLLED** or abnormal system shutdown can occur. An abnormal system termination is caused by operating system or hardware failure. Transactions are stopped abruptly when this occurs. Care should be taken before the system restart can take place (see "Chapter 5.5. Actions to be Taken After an Emergency Restart" for further information).

As far as CICS is concerned, immediate and uncontrolled termination are both treated as abnormal termination. A subsequent **AUTO** start of CICS will result in emergency restart procedures being invoked automatically during CICS initialization to back out changes to recoverable resources made by tasks that were interrupted.

POSTPROCESSING

After the CICS system has been shut down, several operational procedures may be needed before a restart is performed. These could include:

- Checking the statistics
- Reevaluating the partition size
- Updating a resource definition table
- Printing and reviewing transaction dumps.

In addition, if your shutdown was as a result of an uncontrolled termination, you will need to prepare for the emergency restart that will occur when CICS is started again. This may involve running DFHTEOF if you have any user journals on tape, and verifying VSAM files.

Chapter 5.2. The CICS Startup Job Stream

A TYPICAL STARTUP JOB STREAM

The job control statements on the next page are an example of a job stream that might be used to initialize CICS. Following the typical job stream, there are notes that will enable you to create a suitable job stream for your own installation.

```
// JOB VERIFY
//*
//* Verify temporary storage, transient data, restart data set,
//*               CSD, and user files ****
//*
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
// DLBL DFHTEMP,'CICS160.TEMP.STORAGE',,VSAM,CAT=CICSUCT
// DLBL DFHNTRA,'CICS160.INTRA.TRANSD',,VSAM,CAT=CICSUCT
// DLBL DFHRSD,'CICS160.RSD',,VSAM,CAT=CICSUCT
// DLBL DFHCSD,'CICS160.CSD',,VSAM,CAT=CICSUCT
// DLBL FILEA,'SAMPLE.TEST.FILEA',,VSAM,CAT=CICSUCT
// EXEC IDCAMS,SIZE=AUTO
        VERIFY FILE(DFHTEMP)
        VERIFY FILE(DFHNTRA)
        VERIFY FILE(DFHRSD)
        VERIFY FILE(DFHCSD)
        VERIFY FILE(FILEA)
/*
//*
//*********************************************************************
//*******************  START-UP CICS  *********************************
//*********************************************************************
//*
// JOB CICS
* User private core image library
// DLBL U160PCL,'USER.PRIVATE.CIL'
// EXTENT ,TEST02
// DLBL C160PCL,'CICS160.CIL'
// EXTENT ,TEST02
* Search order for core image libraries
// LIBDEF CL,SEARCH=(U160PCL,C160PCL,PL1CLB)
* User VSAM catalog
// DLBL CICSUCT,'CICS160.UCAT',,VSAM
* Auxiliary temporary storage
// DLBL DFHTEMP,'CICS160.TEMP.STORAGE',,VSAM,CAT=CICSUCT
* Intrapartition transient data
// DLBL DFHNTRA,'CICS160.INTRA.TRANSD',,VSAM,CAT=CICSUCT
* Restart data set
// DLBL DFHRSD,'CICS160.RSD',,VSAM,CAT=CICSUCT
* CICS system definition file
// DLBL DFHCSD,'CICS160.CSD',,VSAM,CAT=CICSUCT
* FILEA sample data set
// DLBL FILEA,'SAMPLE.TEST.FILEA',,VSAM,CAT=CICSUCT
* Dump data sets
// DLBL DFHDMPA,'CICS160.DUMP.DATASET.A',0,SD
// EXTENT SYS001,,1,0,70000,1000
// DLBL DFHDMPB,'CICS160.DUMP.DATASET.B',0,SD
// EXTENT SYS001,,1,0,71000,500
* Auxiliary trace
// DLBL DFHAUXT,'CICS160.AUX.TRACE.A',0,SD
// EXTENT SYS001,,1,0,71500,1500
// DLBL DFHBUXT,'CICS160.AUX.TRACE.B',0,SD
// EXTENT SYS001,,1,0,73000,1000
* System log
// DLBL DFHJ01A,'CICS160.SYSTEM.LOG.A',0,SD
// EXTENT SYS001,,1,0,96000,1000
// DLBL DFHJ01B,'CICS160.SYSTEM.LOG.B',0,SD
// EXTENT SYS001,,1,0,97000,1000
* ASSGN of DASD containing CICS and user files
// ASSGN SYS001,DISK,TEMP,VOL=TEST02,SHR
* Local terminals defined in BTAM TCT
// ASSGN SYS020,01A
// ASSGN SYS021,IGN
// ASSGN SYS027,01E
// EXEC DFHSIP,SIZE=1200K,PARM='SIT=1$,SI',SIT=1$,TSP=2$,TDP=6$
   DCP=D$,TRP=D$,BMS=S$,PPT=S1,PCT=S1,DCT=S1,FCT=S3,FCP=3$,TCT=S1
   JCT=1$,JCP=1$,TBP=1$,KPP=D$,AKPFREQ=250,PGRET=P/,PGPURGE=T/
   PGCOPY=C/,PGCHAIN=X/,SCS=1000,AMXT=15,MXT=25
   START=AUTO
/*
/&
```

CREATING AND RUNNING YOUR OWN STARTUP JOB STREAM

When creating and running your own job stream, you will need to consider your requirement for job control statements for any of the following functions and data sets. Chapter 5.1 gives advice on data set preparation, SVA considerations, partition selection, and partition size calculation.

Verify VSAM Files
This step is required only if the previous CICS run terminated abnormally. It is not required if you have VSE/VSAM Version 1 Release 3 installed.

Private Core Image Libraries
Any private libraries that are needed for your own programs or for program products and other packages, must be in the library concatenation chain. These include PL/I and DL/I.

You should keep your installation's applications in a separate core image library from the CICS system. In addition, you may want to keep the resource definition tables in the same private core image library. If you are running multiple CICS partitions, you may prefer to have a resource definition core image library for each partition.

If you have different sign-on tables, program error programs, terminal error programs for different partitions, these must be in a different core image library for each partition.

VSAM Catalog
You may decide to have one VSAM catalog per disk pack.

Auxiliary Temporary Storage
See Chapter 4.2 for a description of the temporary storage data set and details of the AMS control statements and space calculations.

Intrapartition Transient Data
See Chapter 4.3 for a description of the transient data intrapartition data set and details of the AMS control statements and space calculations. This data set can also be a DAM file on count-key-data (CKD) disks. This is also discussed in Chapter 4.3.

Restart Data Set
See Chapter 4.6 for a description of the restart data set and details of the AMS control statements and space calculations.

CICS System Definition File
See Chapter 4.10 for relevant information.

User Data Sets
You will need to provide job control statements for files defined in the file control table that is being used for this startup. For some examples of the statements required, see Chapter 4.11.

Dump Data Sets
See Chapter 4.9 for relevant information. If you want your PDUMP dumps to go to the system dump data set, you will need to define and assign DOSDMPF and DOSDMPG on the volume assigned to SYSDMP.

Auxiliary Trace
See Chapter 4.8 for relevant information.

System Log
See Chapter 4.5 for relevant information. Note that the system log(s) on disk must have been preformatted using

DFHJCJFP before being used in a CICS startup. Similarly, if the system log is on tape, the tapes must have been preformatted using DFHFTAP.

User Journals and Monitoring Journals
These must also be preformatted before being used in a CICS startup.

Statistics Data Sets
See Chapter 4.7 for a description of the statistics data sets.

SYSLST
You will need to have SYSLST assigned if:

- Your system has any extrapartition transient data destinations defined to SYSLST
- You are using the MSGLVL=2 parameter in your SIT, or
- You have selected the PDUMP option for the formatted dump program in your SIT.

You may have SYSLST assigned to a POWER-owned printer; in this case, the CICS partition priority must be lower than the POWER partition priority.

If you do not wish to use either the POWER printer or a physical system printer, you may choose to assign SYSLST to disk.

Transient Data Extrapartition Data Sets
You must insert the required job control statements in the startup job stream for any transient data extrapartition data sets defined in the destination control table that you are using.

DL/I Data Bases
If you are using DL/I, you will need to have job control statements for your DL/I data bases. Depending on your DL/I journaling option, you may also need to specify a suitable UPSI setting.

Terminals
All BTAM terminals require assignment statements. For any card reader/line printer (CRLP) terminals, you will also require DLBL and EXTENT statements if they are defined on disk.

UPSI Settings
See Chapter 5.3 for information on using the VSE UPSI settings.

EXEC DFHSIP Statement
DFHSIP is the first program that makes up CICS.

SIZE=value
When creating your CICS system, you will need to calculate suitable values for ALLOC, ALLOCR, and the SIZE parameter. See the information given in Chapter 5.1.

PARM=value
See Chapter 5.3.

System Initialization Parameters
The characteristics and environment of the running CICS system are determined by various parameters established during startup. Flexibility is provided by allowing these parameters to be specified in a system initialization table (SIT). Several SITs can be prepared, with different suffixes. The system operator can choose the one with the parameters required for the run at startup time. Default

values will be used for any parameters not specified explicitly.

Also at startup time, the system operator can override or add to the parameters in the SIT by providing **system initialization override parameters**. The suffix of the SIT to be used, and the override parameters, can be specified in any of three ways:

- By use of the PARM field in the EXEC DFHSIP statement
- By including them in the SYSIPT data stream, with the startup job stream
- By entering them at the system operator's console during the early stages of initialization.

The syntax of the override parameters is described in "Chapter 5.3. System Initialization Parameters." Further details are given in the CICS/VS Resource Definition Guide.

START=

Types of startup are discussed in more detail in Chapter 2.4 and in the discussions on the restart data set and system journal in Part 4. If you are running with these functions, your normal START option will be START=AUTO.

SYSTEM CONSOLE MESSAGES

During the initialization and running of CICS, various messages appear on the system console. These are mainly informatory, but may require a reply or action on the part of the operator. On the following pages are reproduced typical message streams that may be displayed:

1. When running with disk journals
2. When running with tape journals.

In each case, the messages are those displayed after a START=AUTO start resulting, in turn, in a cold start, a warm start, and an emergency restart.

If you are running without keypointing or recovery, the system initialization output will be a subset of that produced for a cold start.

DISK JOURNALS

If journaling is to disk, the following messages can appear:

Cold Start (Disk Journals)

```
 F4 004 // JOB CICS
    DATE 01/24/83,CLOCK 11/19/30
 F4 004 * DEFINING CORE-IMAGE LIBRARIES
 F4 004 * DEFINING VSAM FILES
 F4 004 * DEFINING SAM FILES
 F4 004 * ASSIGNING DEVICES
 F4 004 IT20I  SYS001 HAS BEEN ASSIGNED TO X'241'
 F4 004 * STARTING CICS, ACCEPTING OVERRIDES FROM SYSIPT AND CONSOLE
 F4 004 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS
 F4 004 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
 F4 004 SIT=1$,SCS=1000,AMXT=15,MXT=25
 F4 004 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 F4 004 TCT=S1,PCT=S1,PPT=S1,DCT=S1,FCP=3$,BMS=S$
 F4 004 PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 F4 004 FCT=S2,JCT=S2,JCP=1$,START=AUTO,TBP=1$,KPP=D$,AKPFREQ=250
 F4 004 DFH1500 - END-OF-FILE ON SYSIPT
 F4 004 DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
 F4 004           AND ENTER $END WHEN COMPLETED
*F4 004
 4 $end
 F4 004 DFH1501 - DFHSIT1$ IS BEING LOADED
 F4 004 DFH1500 - LOADING CICS NUCLEUS
 F4 004 DFH1500 - CICS START-UP IS COLD                          1
 F4 004 DFH1500 - OPENING INTRAPARTITION ACB
 F4 004 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DATA BASE DATASETS ARE BEING OPENED
 F4 004 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DUMP DATA SET IS BEING OPENED
 F4 004 DFH1500 - INITIALIZING TEMPORARY STORAGE
 F4 004 DFH1500 - JOURNAL CONTROL SUBTASK IS BEING ATTACHED/ENTERED
 F4 004 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS  702K
 F4 004 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
 F4 004 DFH1500 - STXIT MACROS ARE BEING ISSUED
 F4 004 DFH1500 - PROCESSING RESIDENT PROGRAMS
 F4 004 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS  636K            2
 F4 004 DFH1500 - OPENING JOURNAL FILES
 F4 004 DFH4508 - CICS SYSTEM LOG. PRIMARY   EXTENT NOW RECEIVING
         OUTPUT ON 241                                               3
 F4 004 DFH1500 - 01 OF 01 JOURNALS SUCCESSFULLY OPENED
 F4 004 DFH1500 - CONTROL IS BEING GIVEN TO CICS                     4
  ...
  ...
  ...
  ...
 MSG F4
*F4 004
 4 cemt perform shutdown                                             5
 F4 004
 F4 004 DFH1701 - C.I.C.S. IS BEING TERMINATED
 F4 004 DFH4510 - ALL OPEN JOURNALS NOW CLOSED
 F4 004 DFH1796 - KEYPOINT SUCCESSFUL                                6
 F4 004 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
 F4 004 EOJ CICS
        DATE 01/24/83,CLOCK 11/21/52,DURATION 00/02/22
```

Warm Restart (Disk)

```
 F4 004 // JOB CICS
    DATE 01/24/83,CLOCK 11/27/15
 F4 004 * DEFINING CORE-IMAGE LIBRARIES
 F4 004 * DEFINING VSAM FILES
 F4 004 * DEFINING SAM FILES
 F4 004 * ASSIGNING DEVICES
 F4 004 IT20I  SYS001 HAS BEEN ASSIGNED TO X'241'
 F4 004 * STARTING CICS, ACCEPTING OVERRIDES FROM SYSIPT AND CONSOLE
 F4 004 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS
 F4 004 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
 F4 004 SIT=1$,SCS=1000,AMXT=15,MXT=25
 F4 004 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 F4 004 TCT=S1,PCT=S1,PPT=S1,DCT=S1,FCP=3$,BMS=S$
 F4 004 PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 F4 004 FCT=S2,JCT=S2,JCP=1$,START=AUTO,TBP=1$,KPP=D$,AKPFREQ=250
 F4 004 DFH1500 - END-OF-FILE ON SYSIPT
 F4 004 DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
 F4 004           AND ENTER $END WHEN COMPLETED
*F4 004
 4 $end
 F4 004 DFH1501 - DFHSIT1$ IS BEING LOADED
 F4 004 DFH1500 - LOADING CICS NUCLEUS
 F4 004 DFH1500 - CICS START-UP IS WARM                              7
 F4 004 DFH1500 - OPENING INTRAPARTITION ACB
 F4 004 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DATA BASE DATASETS ARE BEING OPENED
 F4 004 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DUMP DATA SET IS BEING OPENED
 F4 004 DFH1500 - NO WARM START INFORMATION FOR TSP, TSP COLD STARTED
 F4 004 DFH1500 - INITIALIZING TEMPORARY STORAGE
 F4 004 DFH1500 - JOURNAL CONTROL SUBTASK IS BEING ATTACHED/ENTERED
 F4 004 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS  702K
 F4 004 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
 F4 004 DFH1500 - STXIT MACROS ARE BEING ISSUED
 F4 004 DFH1500 - PROCESSING RESIDENT PROGRAMS
 F4 004 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS  636K
 F4 004 DFH1500 - OPENING JOURNAL FILES
 F4 004 DFH4508 - CICS SYSTEM LOG. PRIMARY   EXTENT NOW RECEIVING
         OUTPUT ON 241
 F4 004 DFH1500 - 01 OF 01 JOURNALS SUCCESSFULLY OPENED
 F4 004 DFH1500 - CONTROL IS BEING GIVEN TO CICS
*F4-004
  ...
  ...
  ...
 4 cemt perform shutdown immediate                              8
 F4 004
 F4 004 DFH1701 - C.I.C.S. IS BEING TERMINATED
 F4 004 DFH4510 - ALL OPEN JOURNALS NOW CLOSED
 F4 004 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
 F4 004 EOJ CICS
        DATE 01/24/83,CLOCK 11/28/54,DURATION 00/01/38
```

```
 F4 004 // JOB CICS
    DATE 01/24/83,CLOCK 11/29/25
 F4 004 * DEFINING CORE-IMAGE LIBRARIES
 F4 004 * DEFINING VSAM FILES
 F4 004 * DEFINING SAM FILES
 F4 004 * ASSIGNING DEVICES
 F4 004 IT20I  SYS001 HAS BEEN ASSIGNED TO X'241'
 F4 004 * STARTING CICS, ACCEPTING OVERRIDES FROM SYSIPT AND CONSOLE
 F4 004 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS
 F4 004 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
 F4 004 SIT=1$,SCS=1000,AMXT=15,MXT=25
 F4 004 TSP=2$,TDP=6$,DCP=D$,TRP=D4
 F4 004 TCT=S1,PCT=S1,PPT=S1,DCT=S1,FCP=3$,BMS=S$
 F4 004 PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 F4 004 FCT=S2,JCT=S2,JCP=1$,START=AUTO,TBP=1$,KPP=D$,AKPFREQ=250
 F4 004 DFH1500 - END-OF-FILE ON SYSIPT
 F4 004 DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
 F4 004           AND ENTER $END WHEN COMPLETED
*F4 004
 4 $end
 F4 004 DFH1501 - DFHSIT1$ IS BEING LOADED
 F4 004 DFH1500 - LOADING CICS NUCLEUS
 F4 004 DFH1500 - CICS START-UP IS EMERGENCY                      9
 F4 004 DFH1500 - OPENING INTRAPARTITION ACB
 F4 004 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DATA BASE DATASETS ARE BEING OPENED
 F4 004 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DUMP DATA SET IS BEING OPENED
 F4 004 DFH1500 - JOURNAL CONTROL SUBTASK IS BEING ATTACHED/ENTERED
 F4 004 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS  702K
 F4 004 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
 F4 004 DFH1500 - STXIT MACROS ARE BEING ISSUED
 F4 004 DFH1500 - PROCESSING RESIDENT PROGRAMS
 F4 004 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS  636K
 F4 004 DFH2800I - DFHRUP IN PROGRESS                             10
 F4 004 DFH2800I - DFHRUP COMPLETED
 F4 004 DFH1500 - OPENING JOURNAL FILES
 F4 004 DFH4508 - CICS SYSTEM LOG. PRIMARY   EXTENT NOW RECEIVING
         OUTPUT ON 241
 F4 004 DFH1500 - 01 OF 01 JOURNALS SUCCESSFULLY OPENED
 F4 004 DFH5701 - TRANSACTION BACKOUT PROGRAM IN CONTROL          11
 F4 004 DFH5703 - NO FBO-TABLE DATA PRESENT. NO FILE BACKOUT PERFORMED
 F4 004 DFH5703 - NO MBO-TABLE DATA PRESENT. NO MSG. BACKOUT PERFORMED
 F4 004 DFH5790 - DFHTBP PROCESSING COMPLETE, RETURNING CONTROL
         TO DFHSIP.
 F4 004 DFH1588 - IS START-UP TO BE CONTINUED ?
 F4 004 DFH1501 - REPLY GO OR CANCEL                              12
*F4 004
 4 go
 F4 004 DFH1500 - CONTROL IS BEING GIVEN TO CICS
  ...
  ...
  ...
 MSG F4
*F4 004
 4 cemt perform shutdown
 F4 004
 F4 004 DFH1701 - C.I.C.S. IS BEING TERMINATED
 F4 004 DFH4510 - ALL OPEN JOURNALS NOW CLOSED
 F4 004 DFH1796 - KEYPOINT SUCCESSFUL
 F4 004 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
 F4 004 EOJ CICS
        DATE 01/24/83,CLOCK 11/31/39,DURATION 00/02/14
```

In addition to the above messages, several "critical" and/or warning error messages may appear if the system initialization program detects errors that either prevent it from continuing the initialization process or cause initialization that differs from that specified by yourself. Some of these messages are self-explanatory, but a more complete explanation of the messages is given in the CICS/VS Messages and Codes manual.

Notes:

1 CICS startup is cold because the restart data set has been newly defined and initialized.

2 This message indicates the size of the dynamic storage area (DSA) remaining in the CICS partition after the resident user application programs have been loaded. See the description of the DSA in "Selecting the Partition" on page 247.

3 The DFHJ01A system log has been opened. It has become the current system log, and is being overwritten with logging information.

4 After this message, user transactions may be run.

5 Normal shutdown has been initiated by the system operator from the VSE console.

6 A warm keypoint is taken after normal shutdown.

7 CICS startup is warm because a warm keypoint exists on the restart data set.

8 An immediate shutdown has been initiated by the system operator from the VSE console. No warm keypoint is taken.

9 Because of the previous immediate shutdown, START=AUTO has caused an emergency restart to take place.

10 The recovery utility program reads the system log backward, identifying inflight tasks and copying the information to the restart data set.

11 The transaction backout program backs out any transactions that were in flight at the previous immediate shutdown, using the information on the restart data set. Messages DFH5703 are not errors. There will be no table data if no recoverable files and messages were being processed at the time of failure, or if recovery has not been requested, or if the SIT options for these resources request them to be cold started. Similar messages may occur for transaction and DL/I table data.

12 If the operator replies CANCEL to this message, CICS will be terminated with a warm keypoint.

TAPE JOURNALS

If the system log is on tape, the following are typical messages that can appear:

Cold Start (Tape)

```
 F4 004 // JOB CICS
    DATE 01/24/83,CLOCK 09/47/46
 F4 004 * DEFINING CORE-IMAGE LIBRARIES
 F4 004 * DEFINING VSAM FILES
 F4 004 * DEFINING SAM FILES
 F4 004 * ASSIGNING DEVICES
 F4 004 IT20I  SYS001 HAS BEEN ASSIGNED TO X'241'
 F4 004 * STARTING CICS, ACCEPTING OVERRIDES FROM SYSIPT AND CONSOLE
 F4 004 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS
 F4 004 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
 F4 004 SIT=1$,SCS=1000,AMXT=15,MXT=25
 F4 004 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 F4 004 TCT=S1,PCT=S1,PPT=S1,DCT=S1,FCP=3$,BMS=S$
 F4 004 PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 F4 004 FCT=S2,JCT=S1,JCP=1$,START=AUTO,TBP=1$,KPP=D$,AKPFREQ=250
 F4 004 DFH1500 - END-OF-FILE ON SYSIPT
 F4 004 DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
 F4 004           AND ENTER $END WHEN COMPLETED
*F4 004
 4 $end
 F4 004 DFH1501 - DFHSIT1$ IS BEING LOADED
 F4 004 DFH1500 - LOADING CICS NUCLEUS
 F4 004 DFH1500 - CICS START-UP IS COLD
 F4 004 DFH1500 - OPENING INTRAPARTITION ACB
 F4 004 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DATA BASE DATASETS ARE BEING OPENED
 F4 004 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DUMP DATA SET IS BEING OPENED
 F4 004 DFH1500 - INITIALIZING TEMPORARY STORAGE
 F4 004 DFH1500 - JOURNAL CONTROL SUBTASK IS BEING ATTACHED/ENTERED
 F4 004 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS  700K
 F4 004 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
 F4 004 DFH1500 - STXIT MACROS ARE BEING ISSUED
 F4 004 DFH1500 - PROCESSING RESIDENT PROGRAMS
 F4 004 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS  634K
 F4 004 DFH1500 - OPENING JOURNAL FILES
 F4 004 DFH4502 - CICS SYSTEM LOG. MOUNT ON 180 FOR OUTPUT: SCRATCH
        VOLUME                                             1
 F4 004 DFH4503 - CICS SYSTEM LOG. VOLUME NUMBER 83024/001 RUN 09.49.20
        NOW RECEIVING OUTPUT ON 180
 F4 004 DFH1500 - 01 OF 01 JOURNALS SUCCESSFULLY OPENED
 F4 004 DFH1500 - CONTROL IS BEING GIVEN TO CICS
*F4 004
  ...
  ...
  ...
 4 cemt perform shutdown
 F4 004
 F4 004 DFH1701 - C.I.C.S. IS BEING TERMINATED
 F4 004 DFH4506 - CICS SYSTEM LOG. VOLUME NUMBER 83024/001 RUN 09.49.20
        UNLOADING FROM 180                                   2
 F4 004 DFH4510 - ALL OPEN JOURNALS NOW CLOSED
 F4 004 DFH1796 - KEYPOINT SUCCESSFUL
 F4 004 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
 F4 004 EOJ CICS
        DATE 01/24/83,CLOCK 09/51/45, DURATION 00/03/59
```

```
 F4 004 // JOB CICS
    DATE 01/24/83,CLOCK 09/56/32
 F4 004 * DEFINING CORE-IMAGE LIBRARIES
 F4 004 * DEFINING VSAM FILES
 F4 004 * DEFINING SAM FILES
 F4 004 * ASSIGNING DEVICES
 F4 004 IT20I  SYS001 HAS BEEN ASSIGNED TO X'241'
 F4 004 * STARTING CICS, ACCEPTING OVERRIDES FROM SYSIPT AND CONSOLE
 F4 004 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS
 F4 004 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
 F4 004 SIT=1$,SCS=1000,AMXT=15,MXT=25
 F4 004 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 F4 004 TCT=S1,PCT=S1,PPT=S1,DCT=S1,FCP=3$,BMS=S$
 F4 004 PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 F4 004 FCT=S2,JCT=S1,JCP=1$,START=AUTO,TBP=1$,KPP=D$,AKPFREQ=250
 F4 004 DFH1500 - END-OF-FILE ON SYSIPT
 F4 004 DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
 F4 004           AND ENTER $END WHEN COMPLETED
*F4 004
 4 $end
 F4 004 DFH1501 - DFHSIT1$ IS BEING LOADED
 F4 004 DFH1500 - LOADING CICS NUCLEUS
 F4 004 DFH1500 - CICS START-UP IS WARM
 F4 004 DFH1500 - OPENING INTRAPARTITION ACB
 F4 004 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DATA BASE DATASETS ARE BEING OPENED
 F4 004 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DUMP DATA SET IS BEING OPENED
 F4 004 DFH1500 - NO WARM START INFORMATION FOR TSP, TSP COLD STARTED
 F4 004 DFH1500 - INITIALIZING TEMPORARY STORAGE
 F4 004 DFH1500 - JOURNAL CONTROL SUBTASK IS BEING ATTACHED/ENTERED
 F4 004 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS  700K
 F4 004 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
 F4 004 DFH1500 - STXIT MACROS ARE BEING ISSUED
 F4 004 DFH1500 - PROCESSING RESIDENT PROGRAMS
 F4 004 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS  634K
 F4 004 DFH1500 - OPENING JOURNAL FILES
 F4 004 DFH4502 - CICS SYSTEM LOG. MOUNT ON 180 FOR OUTPUT: SCRATCH
     VOLUME
 F4 004 DFH4503 - CICS SYSTEM LOG. VOLUME NUMBER 83024/001 RUN 09.57.42
     NOW RECEIVING OUTPUT ON 180
 F4 004 DFH1500 - 01 OF 01 JOURNALS SUCCESSFULLY OPENED
 F4 004 DFH1500 - CONTROL IS BEING GIVEN TO CICS
*F4-004
  ...
  ...
  ...
 4 cemt perform shutdown immediate
 F4 004
 F4 004 DFH1701 - C.I.C.S. IS BEING TERMINATED
 F4 025 DFH4506 - CICS SYSTEM LOG. VOLUME NUMBER 83024/001 RUN 09.57.42
     UNLOADING FROM 180
 F4 004 DFH4510 - ALL OPEN JOURNALS NOW CLOSED
 F4 004 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
 F4 004 EOJ CICS
        DATE 01/24/83,CLOCK 10/01/36,DURATION 00/05/03
```

```
 F4 004 // JOB CICS
    DATE 01/24/83,CLOCK 10/10/26
 F4 004 * DEFINING CORE-IMAGE LIBRARIES
 F4 004 * DEFINING VSAM FILES
 F4 004 * DEFINING SAM FILES
 F4 004 * ASSIGNING DEVICES
 F4 004 IT20I  SYS001 HAS BEEN ASSIGNED TO X'241'
 F4 004 * STARTING CICS, ACCEPTING OVERRIDES FROM SYSIPT AND CONSOLE
 F4 004 DFH1500 - CICS/DOS/VS VERSION 1.6, START-UP IS IN PROGRESS
 F4 004 DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
 F4 004 SIT=1$,SCS=1000,AMXT=15,MXT=25
 F4 004 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 F4 004 TCT=S1,PCT=S1,PPT=S1,DCT=S1,FCP=3$,BMS=S$
 F4 004 PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 F4 004 FCT=S2,JCT=S1,JCP=1$,START=AUTO,TBP=1$,KPP=D$,AKPFREQ=250
 F4 004 DFH1500 - END-OF-FILE ON SYSIPT
 F4 004 DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
 F4 004           AND ENTER $END WHEN COMPLETED
*F4 004
 4 $end
 F4 004 DFH1501 - DFHSIT1$ IS BEING LOADED
 F4 004 DFH1500 - LOADING CICS NUCLEUS
 F4 004 DFH1500 - CICS START-UP IS EMERGENCY
 F4 025 DFH2900I D F H T E O F  U T I L I T Y                 3

 F4 025 DFH2906I VOLUME LABEL VERIFICATION

 F4 025 DFH2907I LABEL INFORMATION-VOLUME NUMBER 83024/001 RUN 09.57.42
        BLOCK 09.57.4
 F4 025 DFH2908A IS MOUNTED VOLUME VALID - Y OR N

*F4 025
 25 y
 F4 025 DFH2914I EOD OCCURRED- LAST RECORD LABEL VERIFICATION

 F4 025 DFH2907I LABEL INFORMATION-VOLUME NUMBER 83024/001 RUN 09.57.42
        BLOCK 09.57.4
 F4 025 DFH2915A IS PROPER VOLUME MOUNTED - Y OR N

*F4 025
 25 y
 F4 025 DFH2901I SUCCESSFUL COMPLETION - DFHTEOF

 F4 004 DFH1500 - OPENING INTRAPARTITION ACB
 F4 004 DFH1500 - TRANSIENT DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DATA BASE DATASETS ARE BEING OPENED
 F4 004 DFH1500 - TERMINAL DATA SETS ARE BEING OPENED
 F4 004 DFH1500 - DUMP DATA SET IS BEING OPENED
 F4 004 DFH1500 - JOURNAL CONTROL SUBTASK IS BEING ATTACHED/ENTERED
 F4 004 DFH1500 - SUBPOOL SIZE BEFORE LOADING RESIDENT PROGRAMS IS  700K
 F4 004 DFH1500 - CPU-TERMINAL SUPPORT AVAILABLE
 F4 004 DFH1500 - STXIT MACROS ARE BEING ISSUED
 F4 004 DFH1500 - PROCESSING RESIDENT PROGRAMS
 F4 004 DFH1500 - SUBPOOL SIZE FOR THIS START-UP IS  634K
 F4 004 DFH2800I - DFHRUP IN PROGRESS                           4
 F4 025 DFH4504 - CICS SYSTEM LOG. MOUNT ON 180 FOR INPUT: LATEST
        OUTPUT VOLUME
 F4 025 DFH4505 - CICS SYSTEM LOG.  REPLY 'YES' IF VOLUME AVAILABLE,
        OR 'NO' IF NOT
*F4-025
 25 yes
 F4 025 DFH4506 - CICS SYSTEM LOG. VOLUME NUMBER 83024/001 RUN 09.57.42
        UNLOADING FROM 180                                     5
 F4 004 DFH2800I - DFHRUP COMPLETED
 F4 004 DFH1500 - OPENING JOURNAL FILES
 F4 025 DFH4502 - CICS SYSTEM LOG. MOUNT ON 180 FOR OUTPUT: SCRATCH
         VOLUME
*F4 025 4886A TAPE UNIT NOT READY  SYS004=180                  6
*F4 025
 25 ignore
```

```
 F4 025 DFH4503 - CICS SYSTEM LOG. VOLUME NUMBER 83024/001 RUN 10.13.02
          NOW RECEIVING OUTPUT ON 180
 F4 004 DFH1500 - 01 OF 01 JOURNALS SUCCESSFULLY OPENED
 F4 004 DFH5701 - TRANSACTION BACKOUT PROGRAM IN CONTROL
 F4 004 DFH5703 - NO FBO-TABLE DATA PRESENT. NO FILE BACKOUT PERFORMED
 F4 004 DFH5703 - NO MBO-TABLE DATA PRESENT. NO MSG. BACKOUT PERFORMED
 F4 004 DFH5790 - DFHTBP PROCESSING COMPLETE, RETURNING CONTROL
          TO DFHSIP.
 F4 004 DFH1588 - IS START-UP TO BE CONTINUED ?
 F4 004 DFH1501 - REPLY GO OR CANCEL
*F4 004
 4 go
 F4 004 DFH1500 - CONTROL IS BEING GIVEN TO CICS
  ...
  ...
  ...
 MSG F4
*F4 004
 4 cemt perform shutdown
 F4 004
 F4 004 DFH1701 - C.I.C.S. IS BEING TERMINATED
 F4 004 DFH4506 - CICS SYSTEM LOG. VOLUME NUMBER 83024/001 RUN 10.13.02
          UNLOADING FROM 180
 F4 004 DFH4510 - ALL OPEN JOURNALS NOW CLOSED
 F4 004 DFH1796 - KEYPOINT SUCCESSFUL
 F4 004 DFH1799 - TERMINATION OF CICS/VS IS COMPLETE
 F4 004 EOJ CICS
        DATE 01/24/83,CLOCK 10/14/49, DURATION 00/04/22
```

Notes:

The messages for a startup using a system log on tape are very similar to those when using a disk device (see earlier). The following differences can be noted.

1 The operator is prompted to mount the tape volume. CICS assigns a volume number yyddd/mmm RUN hhmmss, where yyddd is the date, mmm is a sequence number, and hhmmss is the time.

2 The tape volume is unloaded at the end of the job.

3 During an emergency restart, the tape end-of-file program (DFHTEOF) ensures that the system log has an end-of-file mark following the last valid block written prior to failure. Messages DFH2906, DFH2907, and DFH2908 ask the operator to confirm that the correct volume is mounted. Messages DFH2914, DFH2907, and DFH2915 are produced if a normal end-of-data is found; the operator is given another chance to check that the volume is correct.

4 The restart utility program (DFHRUP) prompts the operator to mount the latest output volume that was used for the system log.

5 The restart utility program (DFHRUP) unloads the tape when it has finished processing it.

6 This VSE message is produced because the tape on device 180 was unloaded by DFHRUP. The operator must reload the correct volume and reply IGNORE.

Chapter 5.3. System Initialization Parameters

The system initialization table (SIT) and the system initialization override parameters are used to determine the function of the CICS system that is to be run.

As part of the resource definition process, you will normally create one or more SITs that are suitable for your installation, and catalog these in your core image library. When you start up CICS, you can then specify the suffix of the SIT that is to be used.

You may, however, want to override the values specified in your prepared SIT. The **system initialization overrides** can be provided by means of:

- PARM options in the EXEC DFHSIP statement
- SYSIPT
- The operator console.

The choice of the method to be used is determined either by the PARM field in the EXEC DFHSIP statement or by an // UPSI job control statement. The CN (or CONSOLE) and SI (or SYSIN) parameters in the override data stream enable further parameters to be entered from the console (CN) or from SYSIPT (SI). The override process is completed by means of a $END parameter.

In general, after a particular CICS system has become the "production system", you should perform the startup with as little operator intervention as possible (that is, use PARM, SIT, or SYSIPT, and **not** the console).

PARM OPTIONS

These can consist of the system initialization override parameters listed at the end of this chapter, in addition to CN (or CONSOLE), SI (or SYSIN), and $END.

UPSI JOB CONTROL STATEMENT

As an alternative to the PARM statement, the UPSI job control statement can be used to indicate that system initialization override parameters are to included, and also that DL/I logging is to be performed.

The possible settings of the UPSI bits at system startup are:

Bit 0
0 - No parameter overrides from SYSIPT initially
1 - Read parameter overrides from SYSIPT

Bit 1
No longer allocated. The function of asynchronous loader subtask formerly determined by this bit is now standard.

Bit 2
0 - Operator will not be prompted for parameter overrides (unless UPSI bit 0 is on and the SYSIPT stream does not end with $END)
1 - Operator will be prompted for parameter overrides (after reading from SYSIPT if UPSI bit 0 is on)

Bits 3 - 5
Not currently allocated

Bit 6
0 - DL/I online log function active
1 - DL/I online log function inactive

Bit 7
0 - DL/I online log on CICS system log
1 - DL/I online log on DL/I system log device

Examples:

```
// UPSI  1    Parameter overrides (SYSIPT)
// UPSI  0    No overrides
// UPSI  001  Parameter overrides (operator console)
```

OPERATOR COMMUNICATION

In operator communication mode, you can override specific parameters in the selected system initialization table and the SYSIPT job stream when CICS displays the message:

```
DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
          AND ENTER $END WHEN COMPLETED
```

If no changes are required, the operator must reply $end or simply press the end-of-block (or ENTER) key to proceed with system initialization.

When changes are submitted from the system console, up to 80 characters can be keyed on one line. The operator should not key $end until all changes have been submitted. The following illustrates a parameter change sequence, as entered from the system console.

```
DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
          AND ENTER $END WHEN COMPLETED

          tct=a,fct=no,mxt=100             (EOB or ENTER)

DFH1500 - CONTINUE - ENTER $END WHEN COMPLETED

          tsp=2$,$end                      (EOB or ENTER)
```

If the operator makes a keystroke error when specifying a keyword, and the error string does not represent any other valid keyword, the operator can recover by typing a comma, then continuing normally. For example:

```
            tct=a,fcu,fct=no,tcq,tcp=2$        (EOB or ENTER)
DFH1500 - CONTINUE - ENTER $END WHEN COMPLETED
            tspl,tsp=2$,end                    (EOB or ENTER)
DFH1500 - CONTINUE - ENTER $END WHEN COMPLETED
            $end                               (EOB or ENTER)
```

If the operator accidentally types the wrong value for a keyword, the operator can recover by submitting the string in error, and then keying another value for the same keyword on the same or later line of input. For example:

```
            tct=1,tcp=2$                       (EOB or ENTER)
DFH1500 - CONTINUE - ENTER $END WHEN COMPLETED
            tcp=1$                             (EOB or ENTER)
DFH1500 - CONTINUE - ENTER $END WHEN COMPLETED
            $end                               (EOB or ENTER)
```

The last value keyed is always the value that is used to initialize the system.

OPERATOR AND SYSIPT OVERRIDES

The following is an example of the above interaction, with parameter changes entered from the system console.

The input stream from SYSIPT specifies an alternative temporary storage program (DFHTSP2$), and also specifies that BMS is not required. However, when the operator is prompted, the operator overrides the SYSIPT input by specifying bms=s$. In the example, lines entered by the operator are highlighted.

```
DFH1500 - CICS START-UP IS IN PROGRESS
DFH1500 - READING OVERRIDE PARAMETERS FROM SYSIPT
BMS=NO          (BMS is not required)
TSP=2$          (to include auxiliary temporary storage)
DFH1500 - END-OF-FILE ON SYSIN
DFH1500 - SPECIFY ALTERNATIVE PARAMETERS, IF ANY,
          AND ENTER $END WHEN COMPLETED
bms=s$,$end            (EOB or ENTER)
```

SYSTEM INITIALIZATION OVERRIDE PARAMETERS

The table at the end of this chapter lists all override parameters that can be entered either by means of the PARM options, or on SYSIPT, or at the console. If more that one value is specified for a parameter, they must be separated by commas and enclosed in parentheses.

Further details of these parameters are given in the CICS/VS Resource Definition Guide. The CICS/VS Performance Guide describes several of the parameters such as MXT and AMXT that can be used for adjusting the performance of CICS.

DEFAULTS

If you do not specify a parameter as a system initialization override, the system will use the value given in the SIT that you specify.

SELECTION OF MODULES AND TABLES

For many of the CICS programs and tables, you can control which version is loaded during initialization by specifying:

- A 1- or 2-character suffix that will be added to the standard name to provide a unique identification.
- YES or null, meaning that an unsuffixed version will be loaded.
- NO, meaning, for programs, that no version or a CICS-provided dummy version (suffix=DY) will be loaded or, for tables, that no version of the table will be loaded, and no version or the dummy version of the corresponding program will be loaded.

In the case of PPT and PCT, NO means that no table will be loaded, the resource definitions will be derived from the GRPLIST parameter, and versions of PCP or KCP will be loaded according to the options specified for the PCP and KCP parameters.

For example:

```
ALT=A1 to specify DFHALTA1
CMP=NO to specify DFHCMPDY
DCT=NO to specify no DCT and thus DFHTDPDY
PPT=NO to specify PPT will be derived from CSD file
```

Appendix C lists the various suffixed versions of each management module provided in the pregenerated system.

The list at the end of this chapter also shows the pregenerated suffixes that can be used. If you have generated your own version of a module, you can specify the suffix of that version instead.

START OPTIONS

For keywords that control resources, you may be able to specify whether a WARM or COLD start is to be performed for the corresponding resources irrespective of the option specified in the START parameter.

If START=(option,ALL) has been specified, all resources will be initialized according to the specified START option. You cannot override this for individual types of resource.

If START=option has been specified, then keyword=(suffix,COLD) or keyword=(suffix,WARM) can be used to override the START parameter.

In the case of DL/I, keyword=(suffix,WARM) may not be specified.

The START options (with ,ALL not specified) and their results are summarized in the following table:

START Option	Option on Individual Resource	Resulting start value
AUTO	not specified	AUTO
	COLD	COLD
	WARM	AUTO
COLD	not specified	COLD
	COLD	COLD
	WARM	WARM
EMER	not specified	EMER
	COLD	COLD
	WARM	EMER

CICS SYSTEM OVERRIDE PARAMETERS

Keyword	Meaning	Value
AKPFREQ	Activity-Keypoint trigger	0 or 200 to 65535
ALT	Application Load Table	Suffix or YES or NO
AMXT	Maximum active tasks	1 to MXT value
APPLID	Identification for VTAM connection	The name by which this CICS system is known to VTAM. It is also the name used for the system when connecting multiple CICS systems using MRO.
ATP	Asynchronous Transaction Processing support	YES or NO and WARM or COLD
ATPINS	ATP input buffer size (in bytes)	16 to 32767.
ATPOUTS	ATP output buffer size (in bytes)	16 to 32767.
ATPMB	ATP task initiation inhibitor value	1 to MXT-1.
ATPMT	Maximum number of active batched tasks	1 to ATPMB value.
BFP	Built-in Functions Program	1$, 2$, 3$, or NO
BMS	Basic Mapping Support programs	E$, A$, S$, or NO and WARM or COLD and UNALIGN or ALIGN and DDS or NODDS
CMP	CICS monitoring program	YES or NO
CMXT	Maximum number of active tasks for each transaction class used in the PCT	Up to ten numbers (one for each class) from 1 to MXT value.
CSA	Common System Area	YES and WARM or COLD
DATFORM	Date format	MMDDYY or DDMMYY or YYMMDD
DBP	Dynamic Backout Program	1$, 2$, or NO
DBUFSZ	Dynamic-log buffer size for dynamic backout (in bytes)	6 to 32000.
DCP	Dump Control Program	4$, D$, or NO
DCT	Destination Control Table (DCT=NO will suppress all transient data functions)	Suffix or YES or NO and WARM or COLD
DIP	Data Interchange Program	1$ or NO
DL1|DLI	DL/I data base support	YES or NO and COLD
DUMPDS	Dump data set suffix	DUMPDS=A|B specifies the suffix of the first data set to be used, DFHDMPA or DFHDMPB.

Keyword	Meaning	Value
EXEC	Command-level application programming interface	YES or NO.
EXITS	User exit interface	YES or NO.
EXTSEC	External security manager support	YES or NO.
FCP	File Control Program	1$, 2$, 3$, 4$, 5$, 6$, 7$, 8$, 9$, A$, B$, C$, or S$
FCT	File Control Table (FCT=NO will suppress all file control functions)	Suffix or YES or NO and WARM or COLD
FDP	Formatted Dump Program	YES followed by one or more of: FORMAT, PARTN, FULL, NO, and SNAP, in any order.
FLDSEP	Field separator characters	1 to 4 characters in quotation marks.
FLDSTRT	Field-name start character	1 character in quotation marks.
GRPLIST	Name of the LIST defining the resource groups to be installed during CICS initialization	The name of a LIST in the CSD file
ICP	Interval Control Program	YES and WARM or COLD
ICV	Maximum system exit time (in milliseconds)	100 to 327670.
ICVR	Runaway task time interval (in milliseconds)	ICV value to 2700000.
ICVS	System stall time interval (in milliseconds)	ICV value to 327670.
ICVSWT	VSE short wait interval (in milliseconds)	0 to 1000.
ICVTSD	Terminal scan delay (in milliseconds)	0 to 100000.
IRCSTRT	Interregion environment for MRO sessions is to be set up at CICS startup	YES or NO.
ISC	Intersystem Communication programs.	YES or NO
JCP	Journal Control Program	1$ or 2$
JCT	Journal Control Table (JCT=NO will suppress all journaling functions)	Suffix or YES or NO
KCP	Task Control Program	YES
KPP	Keypoint Program	D$ or NO
MCT	Monitoring Control Table (MCT=NO will force CMP=NO)	Suffix or YES or NO
MONITOR	Monitoring classes to be activated at CICS startup	NO or one or more of: ACC, PER, and EXC in any order.

Keyword	Meaning	Value
MSGLVL	Type of messages issued during CICS startup	0 or 1 or 2, where: 0 = critical I/O errors and prompting messages only 1 = all messages, at console (SYSLOG) 2 = all messages, on SYSLST as well as SYSLOG
MXT	Maximum number of tasks	2 to 999.
NLT	Nucleus Load Table	Suffix or YES or NO
PCP	Program Control Program	1$ or 2$ or 3$
PCT	Program Control Table	Suffix or YES or NO
PGCHAIN	BMS Page-Chaining code	1 to 7 characters
PGCOPY	BMS Page-Copying code	1 to 7 characters
PGPURGE	BMS Page-Purging code	1 to 7 characters
PGRET	BMS Page-Retrieving code	1 to 7 characters (see also SKRxxxx)
PLTPI	Program List Table for postinitialization	Suffix or YES or NO
PLTSD	Program List Table for preshutdown	Suffix or YES or NO
PL1|PLI	PL/I application programs will be used	YES or NO
PPT	Processing Program Table	Suffix or YES or NO
PRGDLAY	BMS purge delay time interval	hhmm, where: hh is hours from 00 to 99 mm is minutes from 00 to 59
PRINT	Method for printing the contents of a 3270 screen	PA1, PA2, PA3, YES, or NO
SCP	Storage Control Program	YES
SCS	Storage cushion size (in bytes)	20 to 524288 (the value specified is rounded up to the next higher multiple of 2048).
SIMODS	System Initialization Modules suffixes	List of suffixes.
SIT	System Initialization Table	Identifies the SIT to be used for the startup. The 1- or 2-character suffix is used exactly as you specify it; there is no special meaning for NO or YES. If you do not specify the SIT parameter, an unsuffixed name, DFHSIT, is used.
SKRxxxx	Single keystroke retrieval. xxxx specifies a key on the 3270 keyboard, and can be PA1 - PA3 or PF1 - PF36	One or more characters in quotation marks. These will be added to the PGRET value to form a complete page retrieval command

Keyword	Meaning	Value
SRP	System Recovery Program	1$
SRT	System Recovery Table (SRT=NO will force SRP=NO)	Suffix or YES or NO
START	Type of startup	AUTO or COLD or EMER and ALL
SVD	Storage-Violation recovery to be attempted; maximum number of formatted dumps for storage violations	YES or NO or 0 to 99
TBP	Transaction Backout Program	1$, 2$, or NO
TCP	Terminal Control Program for BTAM functions	1$, 2$, 3$, 4$, 5$, E$, S$, or NO
TCT	Terminal Control Table	Suffix or YES and WARM or COLD
TDP	Transient Data Program	1$, 2$, 4$, 6$, or 7$
TRP	Trace Control Program and trace startup options	4$, 5$, or D$ and ON or OFF or AUX
TRT	Trace Table size (in bytes) (TRT=0 suppresses all trace functions)	0 to 10000.
TSMGSET	Controls allocation of storage for each temporary storage queue. Specifies (in number of items) how much storage CICS is to obtain in each increment	4 to 100.
TSP	Temporary Storage Program	1$, 2$, 3$, or NO and WARM or COLD
TST	Temporary Storage Table	Suffix or YES or NO
WRKAREA	Common Work Area size (in bytes)	0 to 3584.
XLT	Transaction List Table	Suffix or YES or NO
XSP	Security Management Program	YES or NO
XTP	ISC Transaction-Routing Transformer Program	YES or NO
ZCP	Terminal Control Programs for common BTAM/VTAM and VTAM-only functions	1$, 2$, 3$, 4$, 5$, E$, or S$

Chapter 5.4. Operating Procedures

This chapter describes the procedures you should follow to start and run the online CICS system. To help you operate the system, the procedures are described on flowcharts, and additional documentation is provided on various forms and tables.

FORMS AND TABLES

Various forms and tables are illustrated to help you operate the CICS system. These are provided for your guidance; you may wish to modify them for your organization. Some of the forms are intended for your DB/DC Administration. Others are to be completed by you to document the online session.

The forms illustrated in this chapter are:

- CICS online log sheet
- CICS incident report
- Data set/data base and transaction table
- Transaction and program table.

CICS ONLINE LOG SHEET						
Job name		Date	Time	System operator	MTO name	Run no.
	Start					
	Stop					

Spec. instrns.	
SIT	
PCT	
PPT	
FCT	
DCT	
TCT	
GRPLST	

Other special instructions

Log tape		
Physical label	CICS allocated label yyddd/mmm RUN hhmmss	Status/note
	RUN	
	RUN	
	RUN	
	RUN	

Start type			Log tape(s) from previous CICS run in case of emergency restart as displayed on console			
auto	cold	emer				

Program not found	
Transaction disabled	
Terminal unavailable	
File unable to open	
Data base stopped	
Subpool size	

Time	Comments/incidents
At startup	
At shutdown	

Figure 77. CICS Online Log Sheet

CICS ONLINE LOG SHEET

This form is used to document each online session. You must complete the form every time you start up CICS, because the information is vital for restarting the system.

This CICS online log sheet will be submitted to operations by DC Administration or the job scheduler with some information already filled in. This would include the job name, date, run number, external labels of available log tapes, and special instructions for the startup.

DC Administration will indicate what start type is required (usually START=AUTO). If START=AUTO will result in an emergency restart or if emergency restart has been explicitly requested, DC Administration will write in the label(s) of the log tape(s) from the previous run. This information can be obtained from the log sheet of the previous run.

When starting up the system, you should do the following:

1. Record the start time and your name.

2. When using tape journals and the journal file is being opened for output, a scratch tape volume is requested. Mount the first tape from the available log tape list and CICS will allocate a label for it. (For details of the label information see the notes for Chart A-1.) Record the allocated label on the log sheet. Do not forget to prepare a sticker with this label for use when the tape is unloaded during or at the end of the session. You may use more than one log tape.

3. If you will be performing an emergency restart, the system log tapes from the previous session will be required. You should begin by mounting the last log tape from the previous CICS run.

   The label of the tape journal, taken from the previous log sheet, should already have been entered for you by DC Administration in the current sheet. When you mount the tape, CICS will display the first record label on the console for you to verify. See Chart F-1 for more details.

Other information you should document includes, for example, the programs not found in the library, disabled transactions, files that could not be opened, or network problems.

The bottom portion of the form, "Comments/incidents", is used to record any unusual incidents that occur at the startup, at the shutdown, or during the session. Incidents such as hardware errors on lines or terminals, or abending transactions should be noted here, and a cross-reference made to the CICS incident report where necessary. Some of these incidents will appear on the system console and others will be reported as CICS messages to a destination determined by the installation. This may be the console, another terminal, or a hard-copy printer.

CICS INCIDENT REPORT					
Incident reported to	System operator	MTO name	Date	Time	Sequence number

	Hardware	Software	Type of dump available	(x)	Time dump was printed	Time dump was saved
Component in error & symptom			CICS transaction dump			
			CICS abend dump			
			CICS formatted dump			
			VSE dump			
			Stand-alone dump			

Full description of problem	Action taken

Figure 78. CICS Incident Report

CICS INCIDENT REPORT

You should complete this form in duplicate to record any of the following error conditions:

VSE abend
CICS abend
Application abend
Line/terminal hardware problem
I/O errors on data bases
I/O errors on journal files
Response time problems reported by users.

Note the sequence number of the CICS incident report on the CICS online log sheet for that session.

In the middle block under the headings "Hardware and Software", describe the symptoms of the problem briefly. If a dump is available, indicate the type of dump and the time it was taken, and, once it has been listed, the time of printing. In the bottom part of the form, describe the problem more fully if possible and any action you have taken. This "Action" column is also used by the recipient.

You should keep one copy of the form for follow-up purposes. The other is given to the Network Control Function for line or terminal errors, to the appropriate Application Supervisor for an application problem, and to DC Administration in all other cases.

DATA SET/DATA BASE AND TRANSACTION TABLE

Data set or data base	Filename	Volume serial	* Type	Applications	Related trx id	Request

* Type = VSAM,SAM,DAM,ISAM

Date: ....

Time: ....

Name: ....

Figure 79. Data Set/Data Base and Transaction Table

DATA SET/DATA BASE AND TRANSACTION TABLE

This table is prepared initially by DC Administration. It may also be used by User Liaison, for example, to ask you to disable all transactions that use a particular data set or data base.

The table shows you the relationship between the DL/I DOS/VS data base names, CICS file names, the VSE file names, the volume serial numbers of the packs containing those data sets, the access method type, and the applications that use them. The relationship of all transactions to each data set or data base is also shown here. From this table, you can tell which transactions to disable before closing a data set or data base. The date, time, and name at the lower left-hand corner are used when you disable a related transaction if a request is made by User Liaison.

TRANSACTION AND PROGRAM TABLE

Transaction code	Tclass	Program(s)	Invoked program	Data set(s) or data base(s)	Access* intent	Application	Status / req.

Date: ....

Time: ....

Name: ....

* Access = INQ,UPD,ADD,DEL,
ALL

Figure 80. Transaction and Program Table

TRANSACTION AND PROGRAM TABLE

This table is prepared by DC Administration.

It shows you the relationship between transaction codes and programs, the data sets or data bases used, and whether the access type of that transaction to the data sets or data bases is inquiry, update, add, or delete. It also indicates the task class, the programs invoked by a transaction, the application, and the status of the program, for example, "Enabled" or "Disabled."

The date, time, and name at the lower left hand corner is used when you disable a transaction according to the request from the User Liaison.

FLOWCHARTS

The next section of this chapter contains flowcharts of the operating procedures required to run a CICS system.

These flowcharts are not complete. They provide a basis for indicating the operating considerations when using CICS with tape journaling.

HOW TO USE THE FLOWCHARTS

The flowcharts all have the same format:

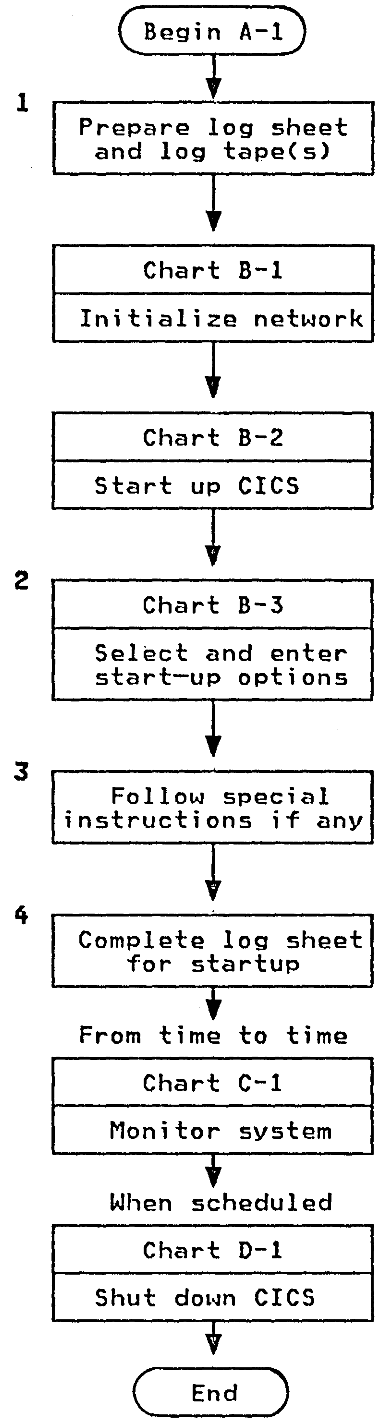
- A block diagram shows the actions and decisions to be made. If a block refers to another chart, you should turn to that chart for more details.
- Where appropriate, the actual commands entered and the responses received, appear next to the block diagram. The notation used for these commands and responses is as follows:

sc-e	Command entered at VSE console
sc-r	Response received at VSE console
mt-e	Command entered at master terminal
mt-r	Response received at master terminal

- Sometimes, additional notes about the action or decision blocks are provided on the page facing the flowchart (indicated by a number on the block diagram). You will probably only need to refer to these notes when you are unfamiliar with the procedure.

LIST OF FLOWCHARTS

Chart A-1: SEQUENCE OF OPERATIONS



Notes for Chart A-1

1. You will need a CICS online log sheet for each startup operation. You will need to prepare paper labels for the journal tapes indicating the operation date, volume sequence number for that startup date, and start time in the following format:

   ```
   yyddd/nnn RUN hhmmss
   ```

   where:

   ```
    yyddd = Today's date
      nnn = Volume sequence number
   hhmmss = Start time of CICS execution.
   ```

   The above information will be supplied by CICS during each startup.

   A procedure should be set up at your installation to control and maintain these log tapes.

2. The appropriate startup will be done by the CICS system if you have chosen the START=AUTO option.

3. For any session, you may have to follow special procedures when the system is initialized. For example, you may wish to alter the initialization parameters or there may be a transaction that is to be stopped because it is giving trouble, or a special message may need to be broadcast to all users.

4. You should fill in the following information on the CICS online log sheet at startup time:

   - Start time
   - Your name
   - Start type
   - The label allocated by CICS to the log tape(s) used for output in this run.
   - Comments and incidents (if any).

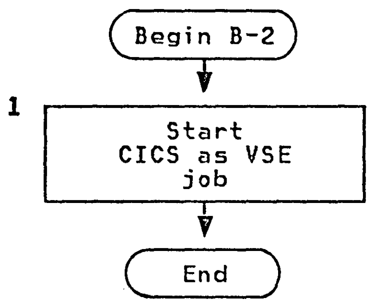
Chart B-1: INITIALIZING THE NETWORK

Put your own procedures here

Notes for Chart B-1

1. This chart should contain details of your installation procedures for initializing the data communication network. This could include such things as powering up data sets or modems, loading the 3704 or 3705 control program, and performing VTAM initialization if applicable.

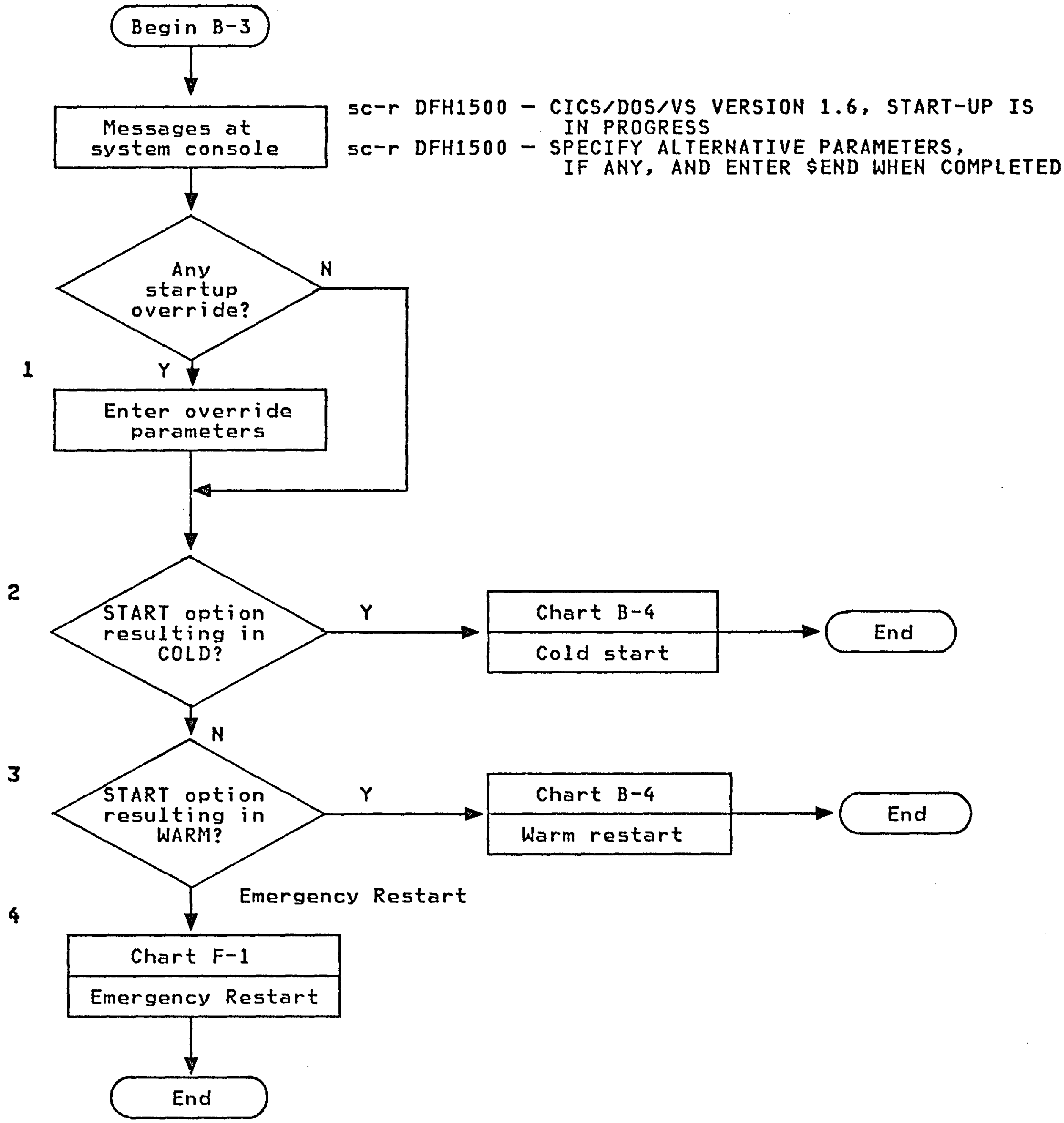
Chart B-2: START UP CICS



Notes for Chart B-2

1. There are several ways to start up CICS depending upon the system environment. You can start the CICS job from a procedure by using the START command, or you can submit the CICS job stream through the reader. This chart should be replaced by your existing normal start-up procedure.

Chart B-3: SELECTION OF CICS STARTUP OPTIONS



Notes for Chart B-3

1. There may be some override instructions for startup from the DC Administrator that you are required to perform during system initialization.

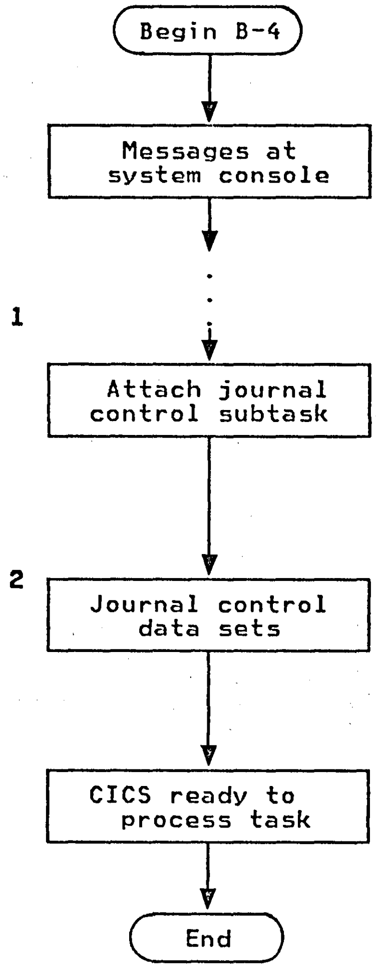
2. A cold start results if START=COLD, or if START=AUTO and the restart data set is newly initialized, or if the CICS system is generated without the keypoint program.

3. A warm start results if START=AUTO and the CICS system has previously been shut down successfully.

4. An emergency start results if START=EMER, or if START=AUTO and problems were experienced during a normal shutdown, that is, if CICS went into a loop or if the CEMT PEFORM SHUTDOWN IMMEDIATE command was specified, or after power or machine failures have occurred.

There are some exceptions to these rules, for example, if START=AUTO should have resulted in an emergency start but JCT=NO was specified. In this case, a cold start will be performed.

Chart B-4: PERFORM WARM (or COLD) START




```
sc-r DFH1501 - DFHSITS1 IS BEING LOADED
     DFH1500 - LOADING NUCLEUS
     DFH1500 - CICS START-UP IS WARM
                  or
     DFH1500 - CICS START-UP IS COLD

sc-r DFH1500 - JOURNAL CONTROL SUBTASK IS BEING ATTACHED

sc-r DFH1500 - OPENING JOURNAL FILES
sc-r DFH4502 - CICS SYSTEM LOG. MOUNT ON 660 FOR OUTPUT:
                 SCRATCH VOLUME
sc-r DFH4503 - CICS SYSTEM LOG. VOLUME NUMBER 80289/001
                 RUN 05.01.51 NOW RECEIVING OUTPUT ON 660
sc-r DFH4508 - CICS JOURNAL 02. PRIMARY  EXTENT
                 NOW RECEIVING OUTPUT ON 171
sc-r DFH4500 - 02 OF 02 JOURNALS SUCCESSFULLY OPENED

sc-r DFH1500 - CONTROL IS BEING GIVEN TO CICS
```

Notes for Chart B-4

1. Other CICS messages will be issued during startup. You should replace this section with your existing operating procedures.

2. Message DFH4502 requests that a scratch volume be mounted. The CICS-allocated label (date and time) is included as part of the message.

   After a successful open, message DFH4503 follows to indicate the label CICS has allocated to the tape volume. The label is in the form:

   ```
   yyddd/nnn RUN hhmmss
   ```

   where:

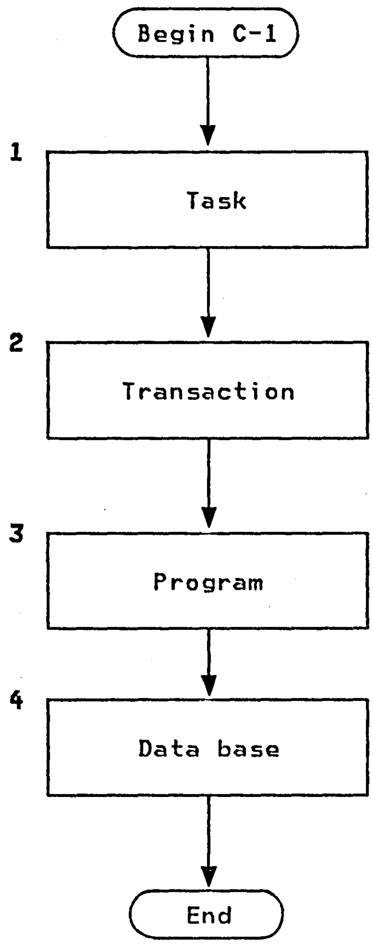
   ```
    yyddd = Today's operation date
      nnn = Volume sequence number
   hhmmss = Start time of CICS execution.
   ```

   You should prepare a physical label in this form for use later when the tape is unloaded.

   Note: In the present example, we assume that CICS monitoring data is written to "user journal" 02. If CICSPARS storage analysis support is also included, another user journal may be opened at this time.

If the master terminal operator opens the VTAM ACB for the first time, using the command CEMT SET VTAM OPEN, but CICS is not using all available VTAM function, message DFH3473 will be sent to the transient data destination called CSMT. The same message will be sent to the CSMT destination if the ACB is opened automatically during initialization, rather than by CEMT.

Chart C-1: NORMAL CICS SYSTEM MONITORING



Notes for Chart C-1

1. The monitoring of the CICS system may be made the responsibility of the VSE console operator, or, more usually, a CICS master terminal operator.

2. From time to time, you may use the CEMT transaction to query the status of tasks in the CICS system. If you notice that a user task with the same task number is suspended every time you display the task status, you may suspect that the task is in a "hard wait" condition and you should refer the problem to DC Administration.

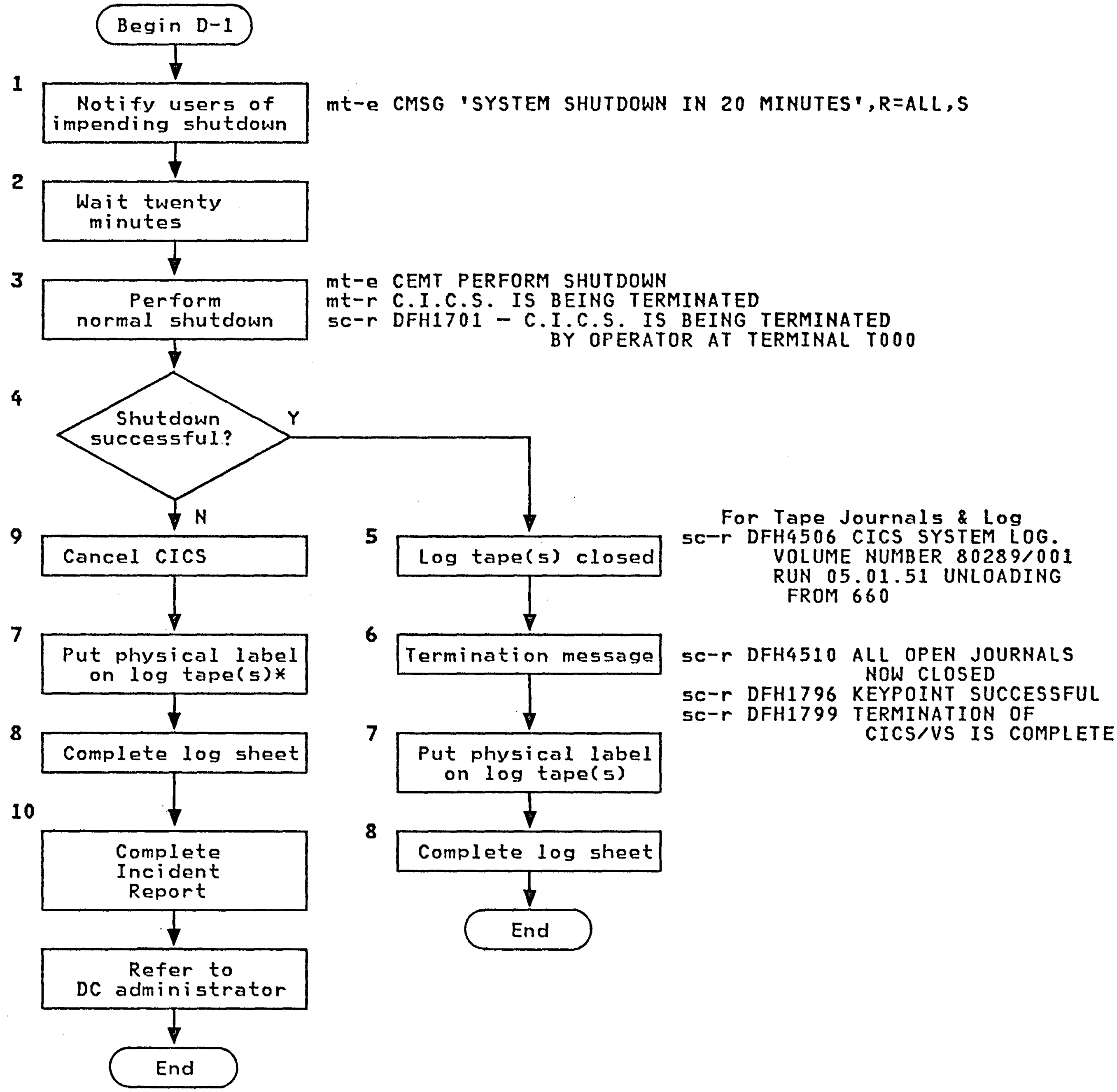
3. CEMT INQUIRE TRANSACTION will display the status of all transactions. If any transaction has a status of "Dis" (disabled) that is different from normal as reported by DC Administration on the transaction and program table, you may have to consult DC Administration.

4. CEMT INQUIRE PROGRAM displays the number of times the program is currently being used, and the number of times the program has been used. This enables you to see the load on each transaction. You may wish to increase the priority of a heavily used transaction, or adjust the maximum number of tasks by class (CEMT INQUIRE TCLASS).

   Also notice the "Ena" or "Dis" status of each program, and report to DC Administration if it is different from the transaction and program table.

5. CEMT INQUIRE DATASET displays the status of the data bases. Make sure that the data bases in the system are all opened and enabled, and report to DC Administration if you find out that a data base has been closed or disabled by the system.

Chart D-1: SHUT DOWN CICS



Notes for Chart D-1

1. Broadcast a shutdown message to users approximately 20 minutes before planned shutdown to enable users to complete their work in an orderly fashion.

2. Remember that a user who is busy with a conversational transaction will not receive this message until the conversation has been terminated.

3. When the CICS Master Terminal shutdown transaction is entered, message DFH1701 is displayed at the console to inform the system operator. This normal shutdown will prevent the users from starting any more transactions. If they attempt to do so, CICS will ignore the terminal input.

   During the early part of the shutdown process, the CICS master terminal operator can still enter CEMT transactions, for example, to display system status. But when shutdown processing reaches a certain point, CICS stops all terminal activity (including CEMT) and waits for existing transactions to halt.

4. Normal shutdown should result in message DFH4506 appearing on the system console within about 10 minutes of message DFH1701. If this message does not appear, the system operator should notify the CICS master terminal operator, because there may be a problem with shutdown.

   In rare cases, the CICS system may still be in the first part of shutdown, during which the master terminal operator can enter CEMT transactions. If so, the status of any CICS tasks that are still running can be shown and, if necessary, they can be purged. This may permit normal shutdown to proceed.

   Generally, if shutdown halts, it will be too advanced for the master terminal operator to enter CEMT. In this case the CICS system must be terminated by the system operator.

5. The message DFH4506 indicates that the specified system log tape has been closed and unloaded. The volume sequence number displayed here (see Note 1 for Chart A-1) should be that for the last log tape mounted, and should match the label displayed in the most recent console message DFH4503.

   In this example, the originally mounted tape (number 001) was sufficient for the entire run and so remained mounted throughout.

6. After message DFH1799 is displayed the CICS job step ends.

7. When the log tapes are unloaded, stick the physical labels, prepared with the volume sequence numbers allocated by CICS, on to those tapes.

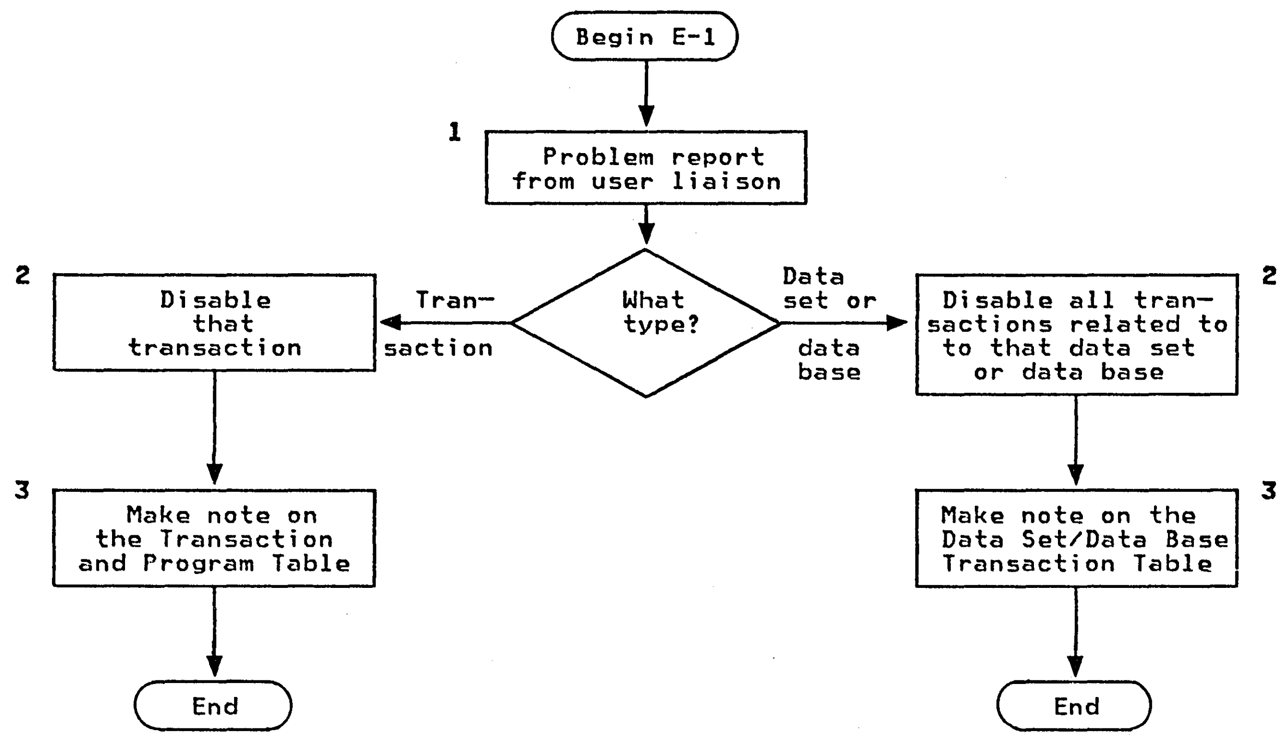
8. Complete the CICS online log sheet with stop date, time, and your name. Make sure that the allocated label column is correctly and completely filled in. If the shutdown is not normal, note this in the "Comments/incidents" area. The information on this CICS online log sheet will be used in the following CICS startup.

9. The VSE CANCEL command must be used. When this abnormal shutdown happens and if you are using nonlabeled tapes, make sure the log tape numbers are correctly recorded on your sheet, because an emergency restart will be required to start up again.

10. The CICS incident report is used here for DC Administrator follow-up.

After an immediate or uncontrolled shutdown, do not reorganize VSAM ESDS files because this will reset the delete flags.

Chart E-1: ACTION ON USER PROBLEM



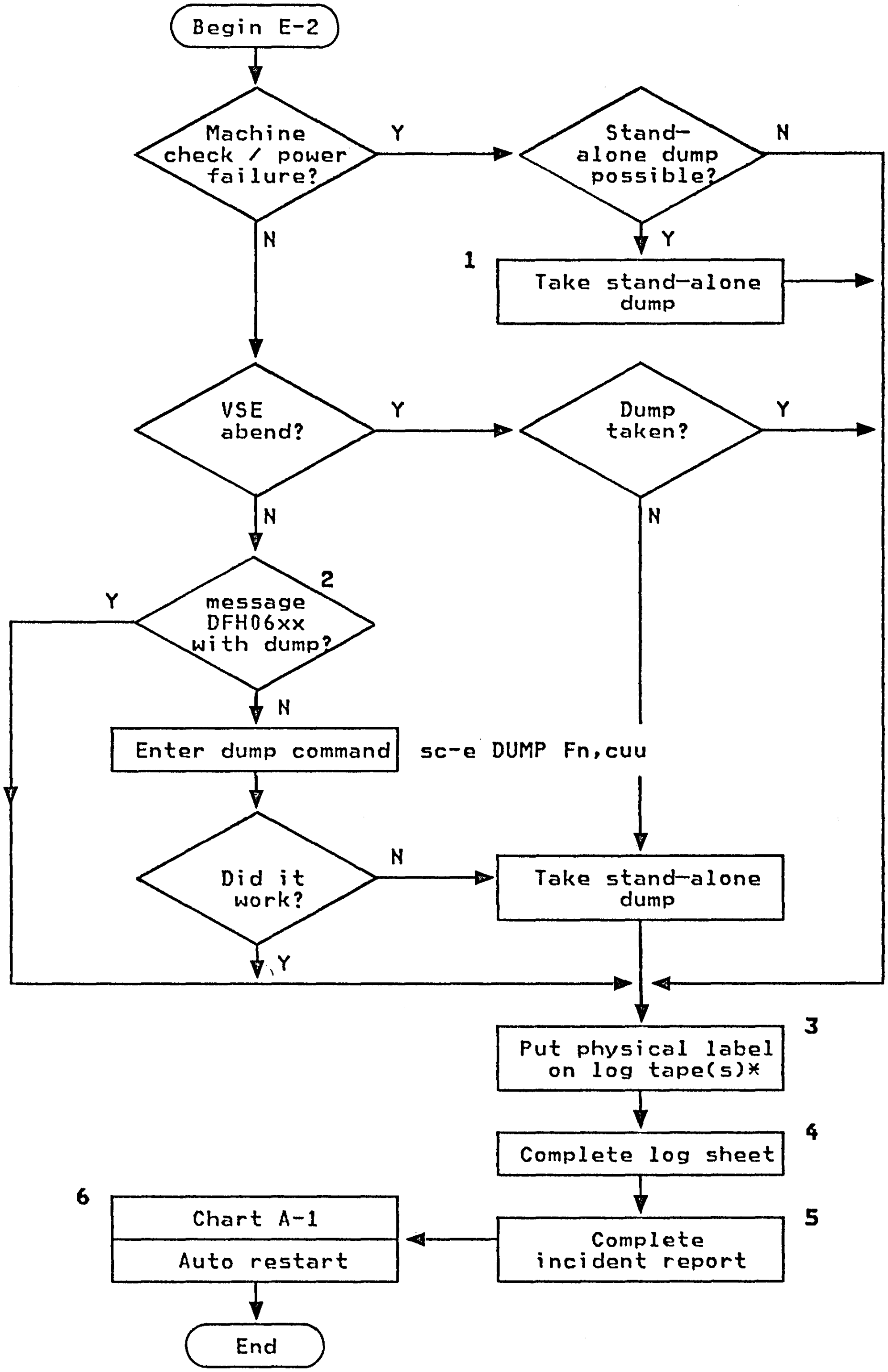
Notes for Chart E-1

1. When a remote terminal operator or end user is having problems, that person will contact user liaison. User liaison will determine the causes of the problems and decide on the action to be taken. If the problems are related to the transactions, or to the data sets or data bases, user liaison may ask you to disable some transactions by submitting a formal request. This request could be on the Transaction and Program table in the case of a transaction error, or on the Data Set/Data Base and Transaction table in the case of a data set or data base error.

2. Disable the transaction(s) according to the request instruction by entering CEMT INQUIRE TRANSACTION transaction.

3. After you have disabled the transaction(s), do not forget to complete the request by filling in the date, time, and your name on the form submitted to you.

Chart E-2: MACHINE CHECK, POWER FAILURE, VSE LOOP, OR ABEND



Notes for Chart E-2

1. In the case of a machine check, a dump will be useful to customer engineers and program service representatives to determine the cause of the problem.

2. With the message DFH06xx on the operating system console, you can check in CICS/VS Messages and Codes to see whether the dump is provided on the CICS dump data set when CICS is abnormally terminated. You should print the dump data set before starting the next CICS session.

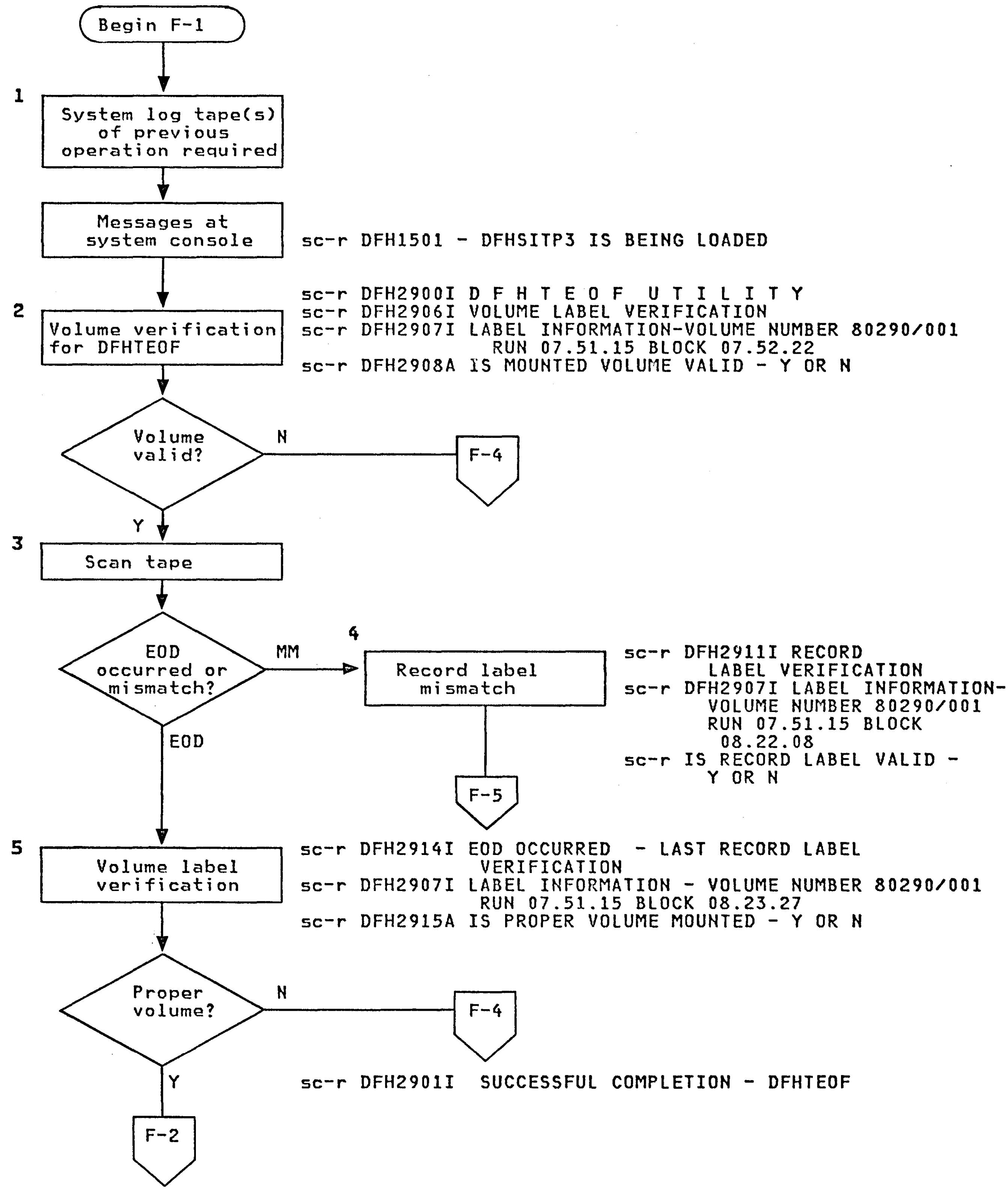
3. Stick the physical labels prepared with the volume sequence numbers allocated by CICS onto the log tapes used during this session.

4. Complete the CICS online log sheet with stop date, time, and your name. Make sure that the "allocated label" column is correctly and completely filled in. State in the "Comments/incidents" area that the next startup must be an emergency restart. The information on this CICS online log sheet will be used in the following CICS start up.

5. The CICS incident report is used here for follow-up by DC Administration.

6. If a machine check or power failure occurs, you can perform an emergency restart (START=AUTO resulting in an emergency start) as soon as the machine is up again, unless a disk failure is suspected.

Chart F-1: PERFORM EMERGENCY RESTART



Notes for Chart F-1

1. For emergency restart, you need the system log tape(s) from the previous CICS run. The keypoint records from the tape(s) are needed to restart the system.

2. CICS emergency restart invokes DFHTEOF to ensure that the last tape volume of the system log has an end-of-file mark following the last valid block prior to the failure. DFHTEOF will OPEN the log. This may cause an VSE tape mount message (not shown), unless the tape is already mounted. You must mount the last system log tape used for output in the previous run (check the CICS online log sheet for that run).

   The system will prompt for the last volume. VSE cannot check the tape, because it is unlabeled, but DFHTEOF will display the messages shown. DFH2907I lists the CICS volume sequence number, which is obtained from the first block of the mounted tape; this includes the date of the creating run. DFH2907I also shows the run start time and the time when first block was written (shown as "BLOCK...").

   You must check the sequence number against the CICS online log sheet and reply Y or N to DFH2908A. If the reply is Y, processing continues looking for the end of valid log records. If the reply is N, volume swapping will take place.

3. DFHTEOF reads the tape forward from the load point and compares the time stamps in each block until it reaches a tape mark or a mismatch between log record labels.

4. A mismatch has been found between the volume sequence numbers in successive log record labels. DFHTEOF issues messages DFHS2911, DFH2907, and DFH2912.

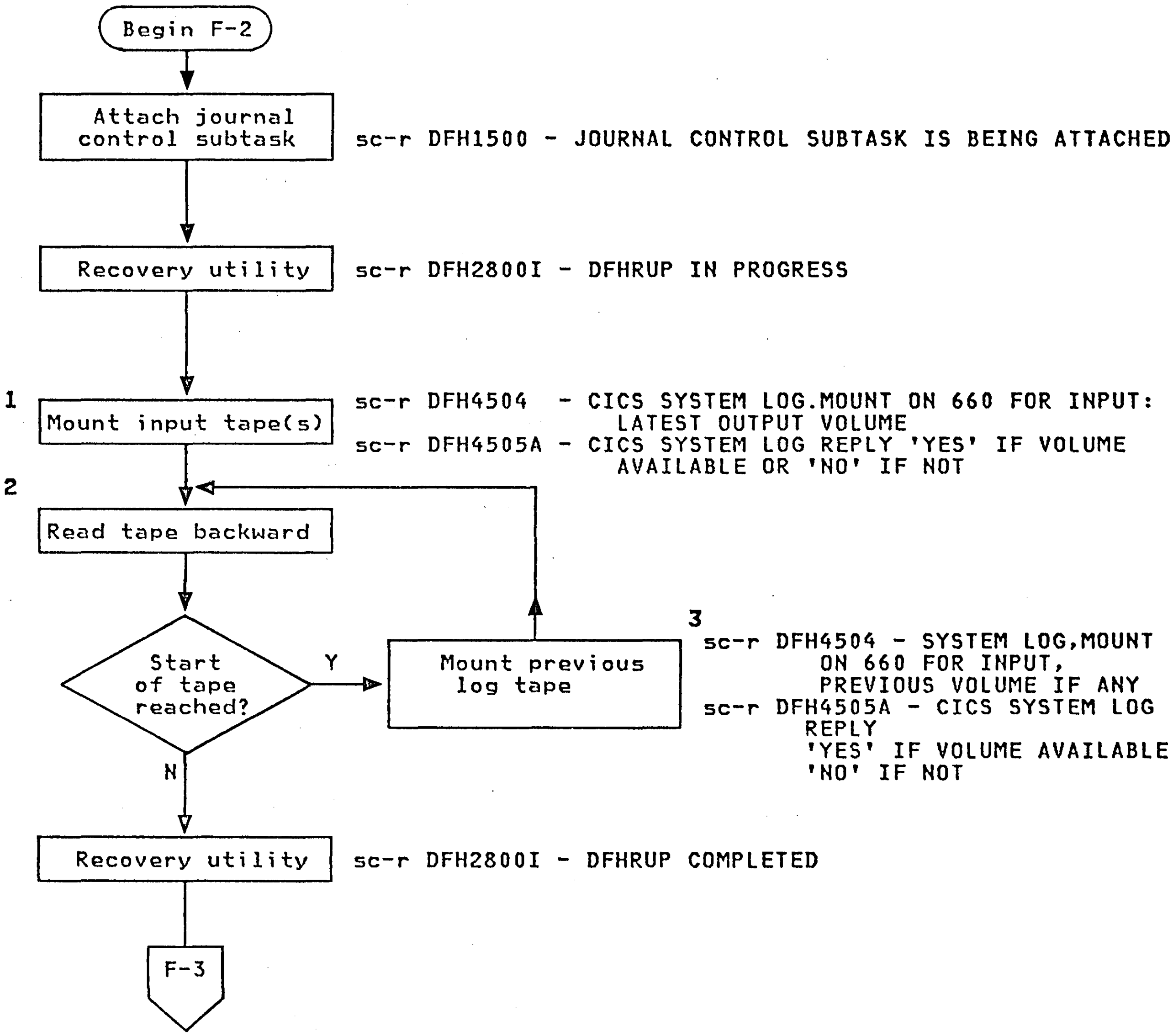
   Chart F-5 shows the action to be taken if this occurs.

5. If normal end-of-data is detected and there are no tape errors, this may mean that (a) the last CICS run closed the log successfully, (b) VSE close routines managed to write EOD even though CICS failed, or (c) the tape you loaded and confirmed for DFHTEOF scanning was possibly not the last log volume of the previous run. Therefore, DFHTEOF asks you once more to verify that the mounted volume is correct, and displays messages DFH2914, DFH2907, and DFH2915.

   Check the time stamp of the last block written ("BLOCK..." in DFH2907I). This should match the time the previous run ended/failed, as noted on your online log sheet.

   The message DFH2915A will wait for your verification. Reply Y to terminate the DFHTEOF program normally and proceed to the next step. Otherwise reply N for an option to swap volumes.

Chart F-2: PERFORM EMERGENCY RESTART (CONTINUED)



Notes for Chart F-2

1. CICS starts the recovery process by displaying message DFH4504 to mount the specified system log tape as input on the specified drive. Refer to the sequence implied by the label allocated by CICS (see Note 1 for Chart A-1). Labels are sequential by date and volume sequence number for a particular CICS execution.

   Message DFH4504 is always followed by message DFH4505A, which requires a reply of YES or NO. If the reply is YES, CICS continues by positioning the tape to the end-of-file mark and reads backward.

2. The recovery utility reads the tape(s) backward until it has collected all log information for inflight transactions. As a minimum, it reads backward to the last activity keypoint but may sometimes have to read further backward than this. If it gets as far as the start of data on the current tape, it will switch to the previous volume to continue.

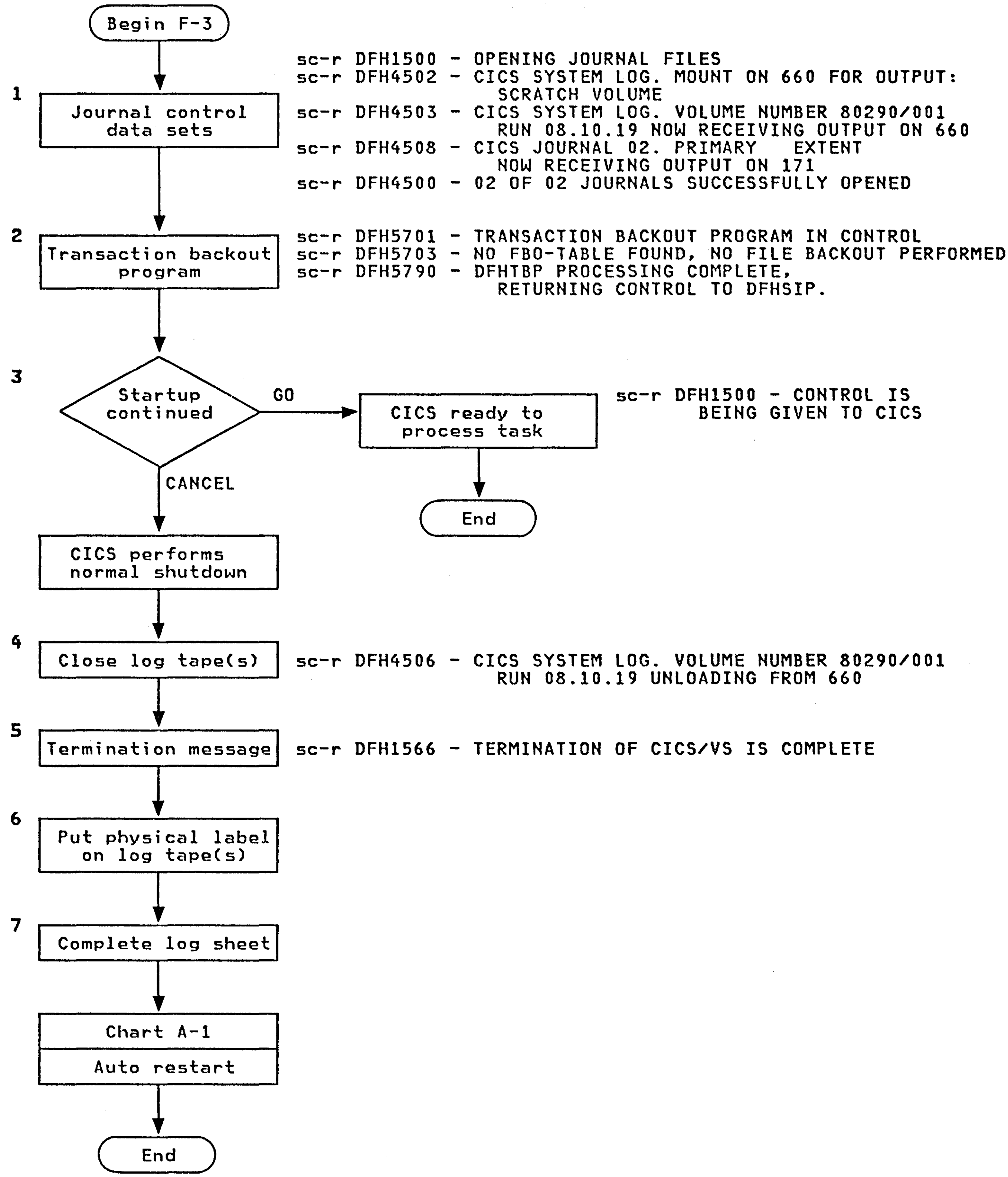
   This read-back process should always stop before running out of log volumes. This is because, immediately following each successful (re)start, CICS writes an activity keypoint on the log indicating that there are no inflight transactions at that point.

3. When the recovery utility reaches the start of the current tape, it issues messages:

   ```
   DFH4504  - CICS SYSTEM LOG. MOUNT ON 660 FOR INPUT:
                PREVIOUS VOLUME (IF ANY)
   DFH4505A - CICS SYSTEM LOG REPLY 'YES' IF VOLUME
                AVAILABLE, OR 'NO' IF NOT
   ```

   You must then mount the immediately-preceding log volume from the previous session, as indicated on the online log sheet for that session.

Chart F-3: PERFORM EMERGENCY RESTART (CONTINUED)



Notes for Chart F-3

1. When the journal file is opened, a scratch tape volume is to be mounted on the indicated address and opened to receive the output of the CICS system log.

   When the open succeeds, message DFH4503 follows to indicate the label CICS has allocated to the tape volume (see Note 1 for Chart A-1).

   You should prepare a physical label in this form for use when the tape is unloaded later in the CICS execution.

2. During the transaction backout process, message DFH5703 informs you if a certain backout cannot be performed because that table is not present. This indicates that there were no resources of this type to back out. Finally, message DFH5790 is displayed when the execution of the transaction backout program is completed.

3. After completion of the transaction backout program, the message DFH1588 will ask whether you want the startup to be continued, and will wait for your reply on message DFH1505.

   If the reply is GO, the system initialization program continues startup. If the reply is CANCEL, CICS is terminated by performing a warm keypoint.

   You may wish to perform a normal shutdown at this stage, and start the CICS session again with the warm start option.

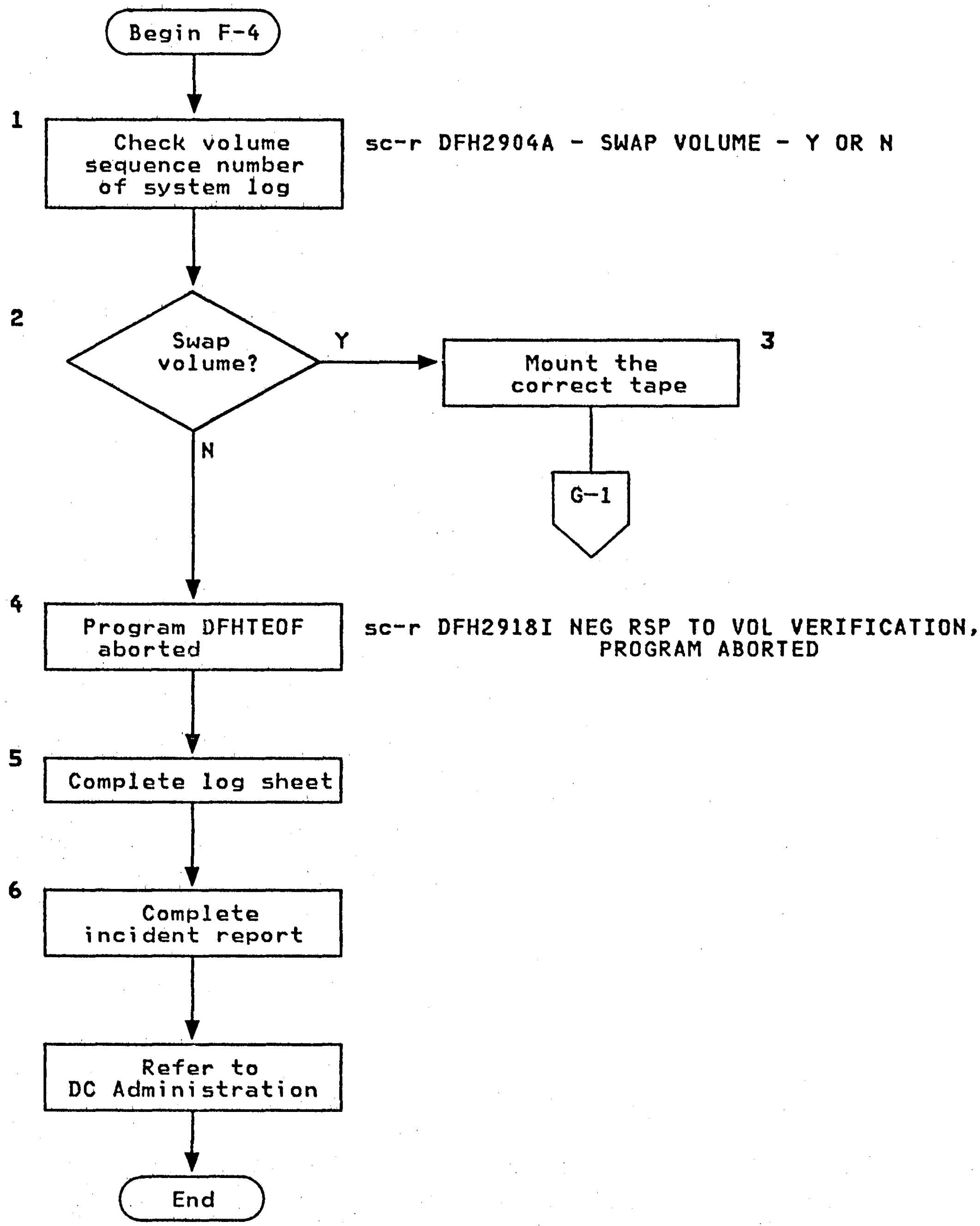
4. The message DFH4506 indicates that the specified system log tape has been closed and unloaded. A volume sequence number is provided here to check against message DFH4503 given during the opening of the journal files.

5. DFH1566 - TERMINATION OF CICS/VS IS COMPLETE is displayed and control is given back to the operating system.

6. When the log tapes are unloaded, stick the physical labels prepared with the volume sequence numbers allocated by CICS on to those tapes.

7. Complete the CICS online log sheet with stop date, time, and your name. Make sure that the allocated label column is correctly and completely filled in. The information on this CICS online log sheet will be used in the following CICS startup.

Chart F-4: PERFORM EMERGENCY RESTART (CONTINUED)



Notes for Chart F-4

1. Message DFH2904A is displayed by program DFHTEOF if, during emergency restart, you have entered N to messages DFH2908A or DFSH2915A, indicating that the wrong volume was mounted. CICS now gives you the option to change the volume or abort the restart.

   Check again the volume sequence number of the system log tape on the previous CICS online log sheet, the message display on the console, and the physical label of the mounted tape to find out what is wrong.

2. Reply Y if you find a correct tape and want to swap tapes, and N if not.

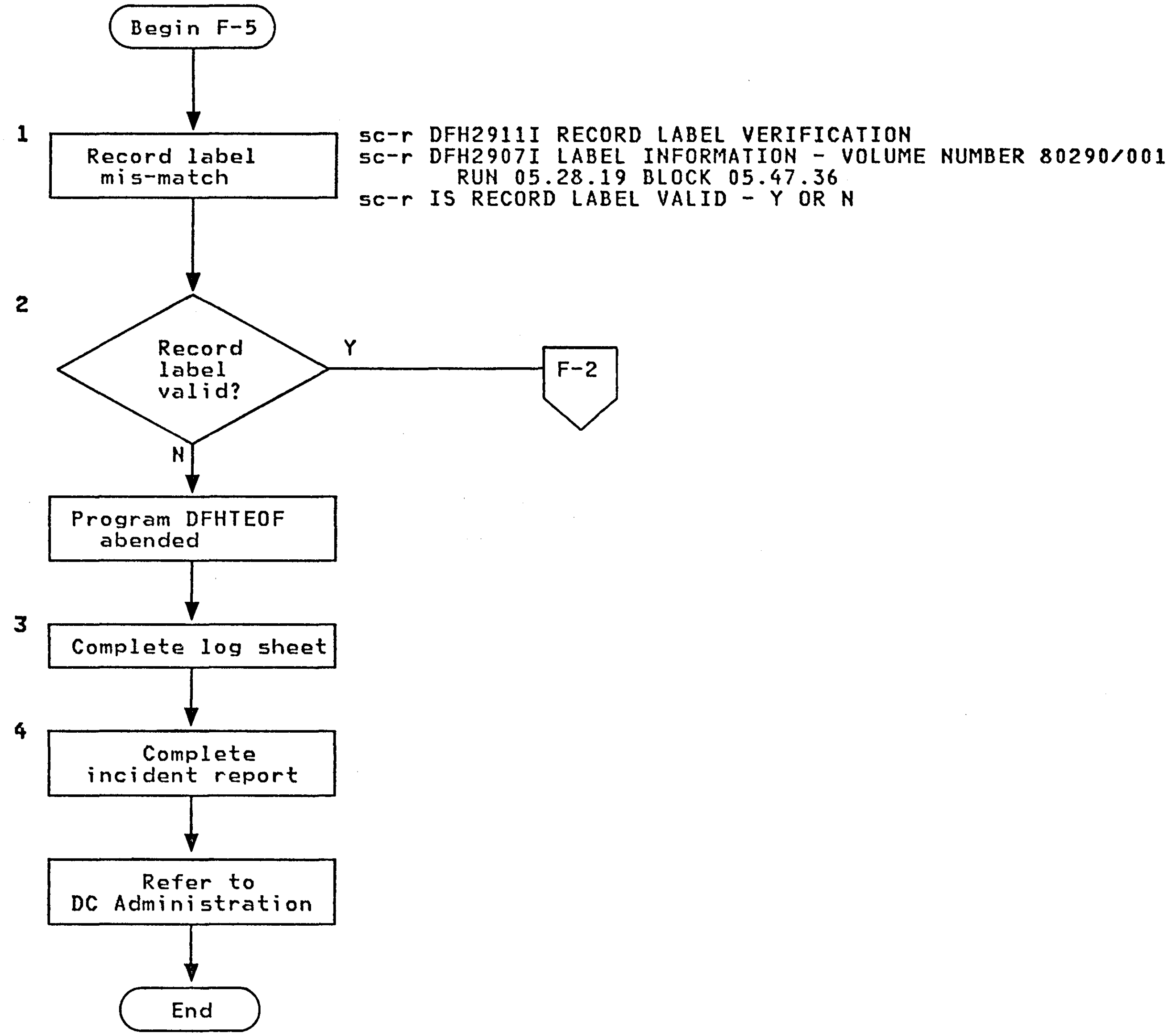
3. The program DFHTEOF will close and reopen the tape file. When you get the VSE mount message, you should demount the present tape and mount the one you require.

4. The execution of the program DFHTEOF will abnormally terminate with message DFH2918I.

5. Complete the CICS online log sheet with stop date, time, and your name. Make sure that the allocated label column is correctly and completely filled in. State in the "Comments/incidents" area why the emergency restart cannot be performed. The information on this CICS online log sheet will be used in the following CICS startup.

6. The CICS incident report is used by DC Administration to take action.

Chart F-5: PERFORM EMERGENCY RESTART (CONTINUED)



Notes for Chart F-5

1. This procedure is entered when a mismatch has been found between the volume sequence numbers in successive log records. DFHTEOF issues messages DFH2911, DFH2907, and DFH2912. The label displayed by DFH2907 is that of the last record preceding the mismatch. This is presumed to be the last valid record on the log. Check the time stamp for this record ("BLOCK..." in DFH2907). This should match the time stamp when the previous CICS run failed, as noted on the online log sheet for that run.

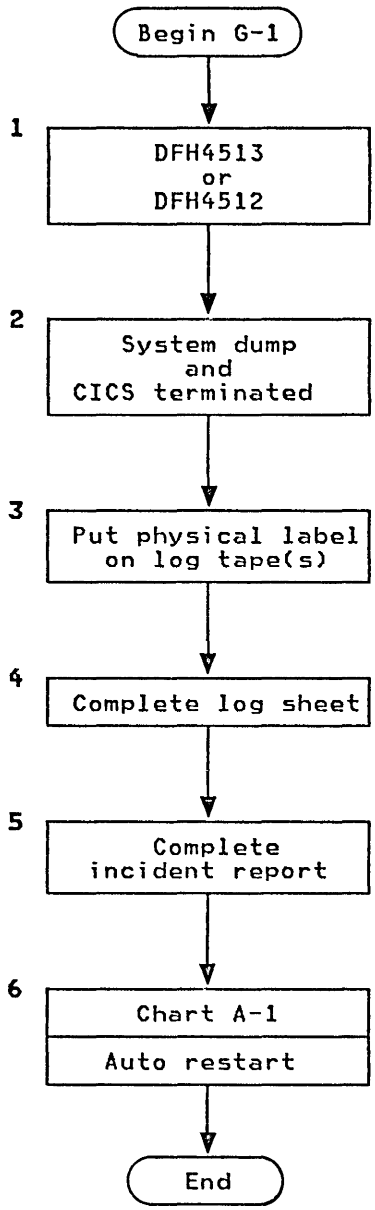
2. Message DFH2912A waits for a reply. If you reply Y, an end-of data mark is written on the tape after the last valid log record, and the program DFHTEOF terminates normally. Emergency restart continues.

   You should reply N only if (a) the time stamp of the last valid record does not match the log sheet, or (b) the log sheet indicates the run ended normally (in which case EOD should have been found). In either case, the problem must be referred to the DC Administrator for resolution. When you reply N, CICS will issue message 'DFH2913I NEG RSP TO RECORD LABEL VERIFICATION, PROGRAM ABORTED', and emergency restart halts at once.

3. Complete the CICS online log sheet with stop date, time, and your name. Make sure that the allocated label column is correctly and completely filled in. State in the 'Comments/incidents' area why the emergency restart cannot be performed. The information on this online log sheet will be used in the following CICS startup.

4. The CICS incident report is used for DC Administration to take action. The log tapes remain as they were before the restart was attempted, and can be used for another emergency restart if DC Administration so decides.

Chart G-1: LOG TAPE ERRORS




```
sc-r DFH4513 - CICS SYSTEM LOG NO LONGER AVAILABLE
                 - OUTPUT I/O ERROR
         or
sc-r DFH4512 - CICS SYSTEM LOG NO LONGER AVAILABLE
               -   OUTPUT VOLUME-SWITCH FAILURE
```

Notes for Chart G-1

1. The message DFH4513 indicates that an unrecoverable output I/O error has occurred for the specified journal data set. For message DFH4512, a bad response code was obtained by a CICS journal task while trying to perform the close/open sequence to switch automatically to a new journal output volume.

2. CICS execution is abnormally terminated with a dump.

3. Stick the physical labels prepared with the volume sequence numbers allocated by CICS on to those tapes.

4. Complete the CICS online log sheet with stop date, time, and your name. Make sure that the allocated label column is correctly and completely filled in. State in the "Comments/incidents" area the reason for this CICS abend, and that the next startup must be emergency restart. The information on this CICS online log sheet will be used in the following CICS startup.

5. The CICS incident report is used here for DC Administration follow-up.

6. If, on the emergency restart, I/O errors occur when trying to read the log tape(s), emergency restart will fail. If this happens, complete another incident report, for the restart, and refer to DC Administration.

Chapter 5.5. Actions to be Taken After an Emergency Restart

CONTROLLED SHUTDOWN FOLLOWING AN EMERGENCY RESTART

If a controlled shutdown is requested by the master terminal operator immediately following emergency restart (by replying CANCEL to message DFH1505), the warm keypoint necessary for a controlled shutdown is taken, and CICS terminates operation. The system may then be initialized at a later time by a normal warm start. You may prefer, however, to reply GO to this message, complete system startup, and perform a warm shutdown by means of the cemt perform shutdown command.

SYSTEM FAILURE DURING EMERGENCY RESTART

System failure during emergency restart represents one of the most difficult types of failures to diagnose and correct. The user must be fully aware of the functions performed during emergency restart, the sequence in which these functions are performed, and the effect that abnormal termination during this operation has on data sets and tables.

Prior to initiating emergency restart, an analysis of the failure that caused the system to terminate abnormally should be performed. It is possible that the condition that caused the system to abend will also cause emergency restart to fail. One example of this could be a physically damaged data set that caused an uncontrolled shutdown, causing the identical failure during emergency restart if the CICS transaction backout program attempts to back out modifications made to that data set.

If a data set has become physically damaged, user-provided data set recovery program(s) must recover the data set prior to the running of the CICS emergency restart transaction backout program to back out modifications to this data set. Data set recovery involves restoring the contents of that data set from some previous copy, and then applying all modifications made to it since the copy was taken. CICS automatic journaling can be used to keep track of data set modifications performed during online execution.

If the transient data intrapartition data set or temporary storage data set is physically damaged, it will not be possible for CICS to emergency restart these facilities. CICS recovery of these facilities is dependent upon the physical contents of the relevant data set as it existed prior to system failure. Therefore, if the data content of the data set has to be restored because of physical damage, CICS may not be able to reconstruct successfully the DCT or TSUT to reflect the status of the restored data set.

User journaling may be utilized, if required, to produce an audit log of all system data set activity. This audit log can be created on a user journal data set, and utilized by user programs for subsequent reconstruction of all system data sets (such as intrapartition or temporary storage) that may have been physically damaged.

CICS emergency restart is not complete until the CICS transaction backout program has successfully completed, and an activity keypoint has been taken. (Optionally, a controlled shutdown could be taken at the completion of user back out, if system execution is to be terminated.) If any failure is encountered prior to this time during emergency restart, this procedure must be followed:

1. Determine the cause of the failure. The cause of the failure of emergency restart must be determined and corrected. If the transient data intrapartition data set is damaged, that data set and the DCT must be cold started by CICS. Its contents may subsequently be restored by the user, if required, by post-initialization (PLT) program processing. (This is also true for the temporary storage data set.) If a data set is damaged it must be physically recovered by user data set recovery programs.

2. Perform an emergency restart. The emergency restart procedure is executed again using the original system log as input. The original system log is the tape or disk volume that was being used for output when the original system failure occurred.

   If tape logging is performed, the log is not written on during emergency restart. This leaves the original contents available for further attempts at restart. For disk logs, emergency restart adds records to the end of the current log. These records are ignored during a subsequent attempt at restart.

At the completion of emergency restart, the recovered status of CICS has been recorded on the new system log if system execution is to proceed, or on the system restart data set as a warm keypoint if the system is to be terminated. This status represents the predefined point to which the system is recovered; system table, system data set, and user data set status are all logically synchronized. If restart becomes necessary from this point on, the new system log must be used for restart.

If the system was terminated upon completion of emergency restart, without an intervening system failure, the system restart data set contains the fully recovered CICS status in the form of a warm keypoint. A CICS warm start may be performed using this data to initiate CICS execution with the recovered system status.

Part 6. Migrating and Applying Service to the CICS System

PART 6

Chapter 6.1. Migrating to CICS 1.6

If you are an existing CICS 1.5 user, you will have your own versions of CICS control tables and startup job streams. You may also have generated your own versions of CICS management modules.

This chapter describes changes that you must be aware of before you run your own CICS 1.6 system.

Your system must now include the Virtual Storage Access Method (VSAM) program product.

When installing a new release you should remember to perform the following checks.

1. Check your existing system initialization table (SIT) entries and overrides. They will need to be modified to reflect the CICS 1.6 options.

   You should adopt START=AUTO.

2. Check the pregenerated modules and ensure that you have specified the appropriate suffixes in the SIT, as detailed in Appendix D.

3. Check the resource definition tables for new keywords and group entries. Reassemble **all** your tables.

4. Check the CICS data sets. In particular, note that the restart data set is a VSAM data set. Note also that the trace entries are now longer; you may need to increase AUXTRACE disk space.

5. Check your job streams. They will need to be modified to reflect different data set characteristics and COBOL and PL/I changes.

6. Reassemble the user-modified CICS programs, for example, the PEP, NEP, and TEP.

7. Check your user exits and system initialization overlays.

8. Check the exceptions and restrictions mentioned later in this chapter, to ensure that your application programs are compatible with the new release.

PART 6

CHANGES TO SYSTEM GENERATION

Under this release of CICS, system generation has been almost eliminated, with most modules being supplied ready-generated. "Chapter 2.5. CICS System Generation" lists the CICS modules that can still be generated.

The CICS pregenerated system provides at least one version of every CICS module. Check the list of pregenerated versions given in Appendix C to see if you can select one of those rather than generate your own version.

- System generation stage 2 jobs run under the Maintain System History Program of VSE/Advanced Functions.

- Now only three options can be coded on the PROGRAM=FCP operand of the DFHSG macro instruction. The new options are DAM, ISAM, and VSAM.

- BMS system generation has been eliminated for CICS 1.6. You can now select one of three versions of the BMS programs during system initialization. The versions are known as the minimum, standard, and full versions, identified to DFHSIT by the suffixes E$, A$, and S$ respectively.

- You can no longer code the RECOVER operand on the DFHSG PROGRAM=SCP macro instruction. Instead, you must code the SVD operand of the DFHSIT macro instruction.

CHANGES TO CICS TABLES

These are changes that might affect your existing CICS tables. See the CICS/VS Resource Definition Guide for more information.

Note, in particular, that the formats of fields within the processing program table and program control table have changed.

- Resource Definition Online
  - "Chapter 2.9. Migrating Resource Definitions" describes how to install and initialize the resource definition online transaction (CEDA) and the CICS system definition (CSD) file.
- Program Control Table
  - The default for the DTB operand of the DFHPCT TYPE=INITIAL macro is changed from DTB=NO to DTB=YES.
  - The space required for the TCA has changed.
  - The INDEX operand is ignored for this release. A result of the new search mechanism is that you no longer improve performance by placing entries for the most frequently used programs first in the table.
  - DFHPCT TYPE=GROUP,FN=ISC now generates the transaction identifiers CSM1, CSM2, CSM3, and CSM5, as well as the existing CSMI, for function request shipping.
- Processing Program Table
  - The INDEX operand is ignored for this release. A result of the new search mechanism is that you no longer improve performance by placing entries for the most frequently used programs first in the table.
- Destination Control Table
  - The INDEX operand is ignored for this release.
- Terminal Control Table
  - CICS 1.6 does not support ISC links to CICS systems earlier than 1.5. Consequently, it no longer supports the TRMIDNT and SESTYPE operands of DFHTCT TYPE=SYSTEM.
  - The default value of the TRMSTAT operand of DFHTCT TYPE=SYSTEM is TRANSCEIVE, instead of TRANSACTION, for existing LU6 sessions.
- File Control Table
  - If your file control table names a file to be opened during CICS initialization, but VSAM indicates that the file was "not properly closed by last user", CICS disables the file. It also sends a message to the console, recommending a "VERIFY". Previous releases of CICS left such a file enabled.

RUNNING CICS 1.5 AND CICS 1.6 TOGETHER

If you wish to run CICS 1.6 and CICS 1.5 together, you should remember the following:

- An application program link-edited to run under CICS 1.6 will not necessarily run under CICS 1.5. However, an application program link-edited under CICS 1.5 will run under CICS 1.6.

- If you need to run CICS 1.5 and CICS 1.6 systems concurrently, only one release can use CICS modules from the SVA.

  Note: For multiregion operation (MRO), modules DFHIRP and DFHSCTE must be placed in the SVA. For an MRO session between different releases of CICS, the modules in the SVA must be those supplied with the later release.

CHANGES TO SYSTEM INITIALIZATION

- Under CICS 1.6, you can specify a suffix for SCP in either DFHSIT, or a system initialization override. The suffix will not apply to DFHSCR, which is now unsuffixable. DFHSCP is suffixable in case you require a static user exit.

- The meaning and use of the SVD operand on the system initialization table (SIT) has changed slightly to incorporate storage recovery.

- You can no longer code the BUFPL operand of the DFHSIT macro instruction.

- You can no longer code the ABKPOPT operand on the DFHSIT macro instruction.

- You now code the BMS option of MAPALGN on DFHSIT instead of on DFHSG.

- The WARM option on both the PPT operand and the PCT operand of the DFHSIT macro is ignored for this release of CICS.

- The HOT option of the PPT operand of the DFHSIT macro is no longer supported.

- The BMS option now has a different meaning. Specifying BMS=xx, will select BMS modules with the suffix xx.

- The options TAPE and DISK are no longer required on the JCT operand of DFHSIT.

- The transaction backout program is now suffixable. You specify the version of the program to be included during initialization by coding either the TBP option of DFHSIT or the TBP initialization override.

  The function group DFHRCVRY in the supplied CICS system definition file contains entries for programs DFHTBP1$ and DFHTBP2$. DFHTBP1$ is the version that does not support DL/I. DFHTBP2$ supports DL/I. Pregenerated versions are supplied. If you have not converted to using CEDA, you should code FN=RECOVERY on the DFHPPT macro to use the programs.

  You no longer need to rename the version of the program that you want. Under earlier releases of CICS, you had to create a version with the unsuffixed name.

- In releases prior to CICS 1.6, CICS had different rules for searching program libraries during the initialization and execution phases. During execution, the system core image library was always searched before the concatenated libraries. In CICS 1.6, CICS always searches the library in this concatenated order.

CHANGES TO THE CICS TRANSLATORS

- The PL/I command language translator has been improved for this release of CICS. Where generated arguments are dummies, the CICS translator generates STATIC dummy arguments instead of literals. Programs translated by the new translator will be more efficient than before.

REMOVAL OF THE ENTRY LEVEL SYSTEM

The reduction in the need for system generation has led to the withdrawal of the Entry Level System.

REMOVAL OF OBSOLETE FUNCTION

The following compatibility support has been removed from this release of CICS/DOS/VS:

- Compatibility support for maps assembled under pre-VS versions of CICS.
- The 2260 compatibility mode of operation of 3270 devices, and support for the FASTER language facility.

PRE-VS BMS MAPS

If you are using BMS maps that were assembled under a pre-VS version of CICS, it will be necessary, because of changes in the map definition macro instructions (CICS/DOS did not use map sets), to modify or rewrite the map definitions and to reassemble them.

Alternatively, you can redefine the maps with the licensed program product Screen Definition Facility/CICS (SDF/CICS). For more information, see the Screen Definition Facility/CICS (SDF/CICS) Program Reference Manual, SH19-6077.

Because of changes in the map field naming conventions, it will also be necessary to modify transactions that refer to the converted maps.

Chapter 6.2. Applying Service to CICS

Read this chapter if you want to apply service to CICS.

This chapter describes what is meant by "service" and the job streams required to apply its different forms. It also explains what actions have to be taken before and after service is applied.

This chapter does not address determining the cause of a problem. This is dealt with in CICS/VS Problem Determination Guide. It is intended that this chapter should be read after problem determination is complete and the symptoms of the problem have been reported to the IBM Support Center as appropriate. You may or may not be asked to submit an Authorized Program Analysis Report (APAR). For details of documentation required when submitting an APAR, see the CICS/VS Problem Determination Guide.

When a problem with CICS is experienced and the cause is found to be an error in CICS, an APAR describing the problem in detail is submitted to IBM by the customer via the IBM Support Center. IBM analyzes the problem and, if it is valid, provides a solution that is tested by the customer who first reported the problem. A record of a problem and its solution is kept by the IBM Support Center in case the same problem occurs at other customers' installations. Periodically, CICS is upgraded with batches of tested solutions before the product is supplied to the customer, in order to minimize the need for preventive service (see later).

Job streams supplied by IBM for service utilize the VSE MSHP service program. This service program enables a record of service to CICS using PTFs and APAR solutions (see later) to be kept in a history file. For introductory information on MSHP, read the VSE/Advanced Functions System Management Guide, and for details of how to use it, read the VSE/Advanced Functions Installation manual, and VSE/Advanced Functions Maintain System History Program Reference.

SIPO/E users should read the VSE System IPO/Extended: Planning Guide, GC20-1875, and the Interactive Productivity Facility VSE Feature: User's Guide, SH20-5526, for details of additional procedures to apply service.

TYPES OF SERVICE

PART 6

Service referred to in this chapter can be corrective or preventive. The term "corrective service" is used to mean applying a solution to a problem you have experienced at your installation. "Preventive service" is used to mean applying a solution to a problem that is known to exist, the intention being to prevent you from experiencing the same problem.

When you experience a problem and corrective service has to be applied, the job stream supplied by IBM to provide a solution can be one of the following.

The first possibility is a job stream that, when executed, replaces the library members affected by the problem. This kind of job stream is a Program Temporary Fix (PTF). It should be noted that library members supplied in a PTF will only be replacements for those supplied on the distribution tape. Additional CICS library members that have been generated at your own installation and that are affected by the problem have to be regenerated. See later for an example PTF.

The other possibility is a job stream that, when executed, performs source modifications (also supplied by IBM within the job). This kind of job stream to apply a fix is an APAR solution or an APAR fix. CICS library members affected by the problem have to be regenerated. APAR fixes should only be applied to a particular system if the problem exists, or if it reasonably certain that the problem will occur.

Preventive service can be performed using PTFs. IBM can also supply Program Update Tapes (PUT) for preventive service.

PROGRAM UPDATE TAPE (PUT)

The Program Update Tape (PUT) distributed to customers on a regular basis, contains maintenance fixes for all programs for which the customer holds a license, and will include CICS service. These fixes are tested and should be installed as soon as possible. The contents of the PUT should be examined to ensure that all APAR fixes that were previously applied are included. Any APAR fixes not included should be reapplied after the PUT has been installed. If changes to data base structure are anticipated, new applications are to be coded, or new functions are likely to be incorporated in the system, it is advisable to apply the current PUT.

PUTs are installed using VSE MSHP. For information on using MSHP, read the VSE/Advanced Function Maintain System History Program User's Guide.

A record of APAR fixes and PTFs contained in PUTs will be automatically generated in the MSHP history file. Users of the System IPO/E should use the Service Dialogs to apply APAR fixes and PUTs to their system.

USE OF MSHP

PTF job streams use the MSHP statement APPLY PTF which, when executed, can replace source books, relocatable modules, and phases. Phases can be replaced using the linkbook facility of MSHP. The use of link books enables phases whose names are specified in the PTF to be re-link-edited according to linkage editor control statements in specified source books. The specified source books are CICS library members. Several link-edits can be performed in a PTF according to the number of phases affected by a problem.

APAR solution job streams are supplied to provide solutions by way of a source modification to CICS for corrective service. These job streams use the MSHP statement CORRECT.

MSHP is not only used for member replacement and source modification, but also records the serial numbers of PTFs and APAR fixes in a history file. The history file contains a record of all service applied to the release of CICS at your installation, including service that is preapplied by IBM.

MSHP is also used to record the names of CICS library members generated at your installation. The job stream produced by the CICS system generation macro DFHSG uses the MSHP statements TAILOR and EXECUTE with the operand XREF.

The result of this job stream is that the appropriate phases are generated and the names of library members included in the phases are recorded in the history file. Included library member names are recorded because some future service may affect the included members and this allows a method of cross-checking.

Service job streams may be supplied which contain the MSHP statement INFLUENCES. This statement (primarily for IBM internal use) provides an additional means of indicating CICS library members that have to be regenerated because they are affected by the original problem.

SERVICE MATERIALS SUPPLIED

PTFs and APAR solutions are generally supplied in the form of a document together with one or more job streams. The job streams are generally on machine-readable media that will normally be magnetic tape.

PTFs are supplied with a cover letter document that gives details of the number of files on the tape and the contents of each file. The cover letter itself can also be on the tape in machine-readable form.

APAR solutions are supplied with brief instructions for running a job stream and any other required actions. The job stream itself, if not on magnetic tape, may be on punched cards or in the form of a document giving typing instructions.

SERVICE APPLICATION CHECKLIST

The preparation and checking needed to apply service successfully to CICS can vary according to the complexity of the modifications involved. For example, the application of a PTF can, in some cases, require little effort. However, the mass application of service such as using a PUT, can require more preparation and checking to ensure it is correctly applied.

Use the following checklist for any kind of service you apply to CICS, but evaluate each step to determine its relevance to your servicing task.

In the case of PUT tapes, use this checklist to be sure you have followed the installation instructions in the PUT documentation.

SIPO/E users need to take account of the additional SIPO/E service procedures when using this checklist.

1. As far as is reasonably possible, check that the service materials are correct and complete.

2. Read and be sure to understand the instructions supplied in the service material.

3. Consider making a plan of the steps in your own servicing task according to how your installation is organized.

4. Consider obtaining backup copies of the history file and libraries to be modified. If the application of service were to be unsuccessful, these backups would ensure that there would be no inadvertent loss of program code. To perform such backups, SIPO/E users should read the Interactive Productivity Facility VSE Feature: User's Guide, and non-SIPO/E users should read the VSE/Advanced Functions System Utilities manual.

5. Ensure that any prerequisites are satisfied and any actions required before service is applied, are carried out according to instructions supplied in the service materials. Note that MSHP can perform some prerequisite checking. MSHP helps by checking for prerequisite PTFs, for example. If necessary, read the VSE/Advanced Functions Maintain System History File User's Guide for more details.

6. Ensure that all of the CICS product necessary to apply the service exists in libraries that can be accessed by the service job stream.

7. Prepare the job stream to apply service according to how your installation is organized.

8. Execute the job stream.

9. Examine the console log (SYSLOG) and the job listing (SYSLST) for messages from the job stream, and ensure all jobs in the job streams have not ended abnormally. Be sure to understand messages written by MSHP. See VSE/Advanced Functions Messages if necessary.

10. Determine from the MSHP messages whether any further actions are necessary to complete the application of service successfully.

11. Check the listings for any link-edits with unresolved external references and satisfy yourself that there are no errors. The documentation with the service materials may indicate unresolved external references that are, in fact, acceptable and not an error.

12. If an APAR solution affects a macro, determine whether you need to edit the macro after execution of the APAR job stream, or whether it is performed for you by the APAR itself. If necessary, read the VSE/Advanced Functions Maintain System History File User's Guide for details of the MSHP statement TAILOR. For an explanation of edited macros, read the VSE/Advanced Functions System Management Guide.

13. Determine whether the application of service affects a CICS management module other than those pregenerated and supplied with the CICS product (those you have generated yourself). When the service job is executed, MSHP can write a message to inform you of phases affected other than those pregenerated.

    In the case of an APAR fix, it may be necessary to regenerate a CICS-provided version of a module. To regenerate a CICS management module, see "Chapter 2.5. CICS System Generation" for details of PROGRAM=PREGEN and of the MOD operand of DFHSG TYPE=INITIAL.

14. If instructed by the documentation supplied with the service material, prepare the necessary job stream and execute the CSD file utility DFHCSDUP using the SERVICE operand. See "Chapter 4.10. CICS System Definition File" See the instructions in the service material and perform any actions required after job stream execution.

15. In the case of corrective service, after service has been correctly applied, consider reproducing the original problem to satisfy yourself that it is solved.

16. After you have completed the application of service, consider whether you should obtain backup copies of the newly-modified history file and libraries.

17. For future reference, consider keeping a record of the actions you took to apply the service.

SAMPLE PTF

```
//   JOB UP25999
*      COPYRIGHT: 5746-XX3 COPYRIGHT IBM CORPORATION 1974, 1983
*        LICENSED MATERIAL - PROGRAM PROPERTY OF IBM
*        REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083
*      ACTION; AFTER APPLYING THIS PTF, YOU SHOULD RE-ASSEMBLE YOUR
*        OWN VERSIONS OF THE FOLLOWING:
*        DFHDBP.
// PAUSE CANCEL IF NOT APPLICABLE.
// EXEC MSHP
APPLY 5746-XX-300(N36):UP25999 RELEASE=01.55.00;
RESOLVES APARS=(PP00637,
                PP95530,
                PP00996);
AFFECTS MACROS=(DFHDBP) SUBLIB=A;
OCCUPIES SLIB=361;
DATA;
        CATALS  A.DFHDBP,15.0
        BKEND  A.DFHDBP
DBP      TITLE 'CICS/VS DYNAMIC TRANSACTION BACKOUT PROGRAM'
*************************************************************************
*                                                                       *
* MODULE NAME = DFHDBP                                                  *
*                                                                       *
* DESCRIPTIVE NAME = DYNAMIC TRANSACTION BACKOUT PROGRAM                *
    .
    .
    . ETC
    .
    .
         DFHEND DFHDBPNA
        BKEND
/$
/*
/&
```

Appendixes

APP.
A — F

Appendix A. Contents of the Distribution Libraries

The CICS/DOS/VS system, provided as machine-readable material on the (basic) distribution tape, is contained in the following libraries:

- A core image library
- A relocatable library
- A source statement library.

Note: Further source statement libraries are available on a separate (optional) tape. This appendix does not describe the contents of these optional libraries.

The contents of the CICS/DOS/VS libraries for Version 1 Release 6 correspond to VSE/Advanced Functions Version 1 Release 3.

CORE IMAGE LIBRARY

The core image library provided on the distribution tape contains the following:

- CICS programs (see Appendix C)
- Sample tables (see Appendix D)
- CICS aids (see Appendix E)
- FILEA sample application programs and their BMS map sets (see Appendix E).

RELOCATABLE LIBRARY

The relocatable library provided on the distribution tape contains the following:

- Items required to generate the contents of the core image library (see earlier).
- Modules required for link-editing with user-written programs, in particular the command-level application link-edit stubs, for example DFHECI.
- VSE logic modules, the source code for which can be found in A.DFHSTLM in the source statement library on the distribution tape.

DFHKCQ and DFHPCQ
Object modules required to generate all versions of DFHKCP and DFHPCP respectively in the core image library.

Link books and included object modules for CICS programs
Link books for all object-maintained core image library phases composed of more than one module. A link book has the same name as the generated phase, except in the few cases where one of the included object modules has the same name as the generated phase.

Link book	Phase name
DFHAMP00	DFHAMP
DFHCSDUP	DFHCSDUP
DFHDMP	DFHDMP
DFHEAP1$	DFHEAP1$
DFHECID	DFHECID
DFHECIP	DFHECIP

APP A – F

Link book	Phase name
DFHECP1$	DFHECP1$
DFHECSP	DFHECSP
DFHEDAD	DFHEDAD
DFHEDAP	DFHEDAP
DFHEDF	DFHEDFD
DFHEDFBL	DFHEDFBR
DFHEIDLI	DFHEIDLI
DFHEIGDX	DFHEIGDX
DFHEMTA	DFHEMTA
DFHEMTD	DFHEMTD
DFHEMTP	DFHEMTP
DFHEOTP	DFHEOTP
DFHEPP1$	DFHEPP1$
DFHERP1$	DFHERP1$
DFHESTP	DFHESTP

DFHEAI and DFHEAI0
Link-edit stubs for command-level assembler-language applications.

DFHECI
Link-edit stub for command-level COBOL applications.

DFHEPI
Link-edit stub for command-level PL/I applications.

DFHERI
Link-edit stub for command-level RPG II applications.

DFHPHN
Phonetic encoding module (for use by offline programs). If phonetic encoding (CALL DFHPHN) is used in offline programs to construct the data base, the phonetic encoding module must be assembled and cataloged in the relocatable library. The source statements for assembly only are:

```
// JOB PHONCODE
// LIBDEF SL,SEARCH=cics-slb-filename
// EXEC  ASSEMBLY
   COPY DFHPHN
   END
/*
/&
```

IJDVZZIW and IJDVZZZW
Printer VSE logic modules.

IJHZLGZZ
Indexed sequential logic module.

BTMODH$
BTAM logic module for 2740/41.

BTML32T and BTMODL#
BTAM logic modules for local 3270.

BTMR32T and BTMODR#
BTAM logic modules for remote 3270.

BTMLR32T and BTMODM#
BTAM logic modules for local and remote 3270.

Note: BTML32T, BTMR32T, and BTMLR32T were generated with RMSR=YES and ERLOGIC=E. See A.DFHSTLM for full details. Switched line support is not present in BTMR32T or BTMLR32T.

DFHISMNC
ISAM logic module, no RPS, "prime data in main storage" option.

DFHISMNN
ISAM logic module, no RPS, no "prime data in main storage" option.

DFHISMRC
ISAM logic module, with RPS and "prime data in main storage" option.

DFHISMRN
ISAM logic module, with RPS, no "prime data in main storage" option.

Note: To use RPS, the VSE supervisor must be generated to support RPS, and RPS must be in the VSE shared virtual area (SVA).

SOURCE STATEMENT LIBRARY

The source statement library provided on the basic distribution tape contains the following:

- Macros and copy code used by the CICS system generation process.
- Source code, macros, and copy code required to generate all user-customizable CICS modules in the core image library. These modules are identified in the CICS/VS Customization Guide.
- Macros and copy code required by the system programmer for CICS table generation.
- Macros and copy code required for application programming.
- Source code used to generate the VSE logic modules (A.DFHSTLM).
- Source code for the sample tables - as A. books (see Appendix D).
- Source code for the CICS aids - as A. books (see Appendix E).
- Source code for the sample application programs, and copy code for record descriptions used - as books appropriate to their language, namely A. books for assembler, C. books for COBOL, P. books for PL/I, and R. books for RPG II (see Appendix E).
- Copy code, where provided, for the BMS symbolic description maps used by the CICS aids and sample application programs (FILEA only) - as books appropriate to their language (see Appendix E).
- Source code for all BMS map sets and partition sets used by the CICS aids and sample application programs - as A. books (see Appendix E).
- Sample job streams - as Z. books (see Appendix F).

Appendix B. CICS Modules Eligible for the SVA

The CICS modules listed below are eligible for residence in the VSE shared virtual area (SVA).

Note: If multiregion operation (MRO) is being used, modules DFHIRP and DFHSCTE **must** be placed in the SVA.

Command-level interface modules are listed with the abbreviated name of their associated management module, if any, in parentheses after the module name. Not all of the latter are SVA-eligible.

A module type of 'N' indicates a read-only nucleus module; a module type of 'P' indicates a read-only PPT module.

Sizes are given for the SVA-eligible CICS modules as supplied pregenerated in the distribution core image library; a suffix, where given, identifies the particular version of a module. (For details of the suffixed pregenerated modules, see Appendix C). All sizes shown are rounded up to the nearest 1K byte boundary, and should be treated as approximate only. For more accurate space estimates, you are advised to consult the library listings at the time of installation, and to allow for the way in which VSE loads the SVA.

See Chapter 2.6 for information on installing and using CICS modules in the SVA.

Module	Type	Size	Suffix
DFHALP	N	6K	
DFHAMP	P	50K	
DFHCCMF	P	3K	
DFHCMON	P	6K	
DFHCMP	N	8K	
		1K	DY
DFHCRNP	P	3K	
DFHCRP	P	1K	
DFHCRQ	P	1K	
DFHCRR	P	1K	
DFHCRS	P	2K	
DFHCRSP	P	1K	
DFHDCP	N	5K	D$
		1K	DY
		6K	4$
DFHDIP	N	1K	DY
		4K	1$
DFHDMP	P	37K	
DFHDSB	N	2K	A$
		2K	S$
DFHEBF (BFP)	N	1K	
DFHEBU	N	1K	
DFHECID	P	64K	
DFHECIP	P	3K	
DFHECSP	P	3K	
DFHEDAD	P	57K	
DFHEDAP	P	3K	
DFHEDC (DCP)	N	1K	
DFHEDFBR	P	10K	
DFHEDFD	P	49K	
DFHEDFP	P	4K	
DFHEDFR	P	1K	
DFHEDFX	P	2K	
DFHEDI (DIP)	N	1K	
DFHEEI (EIP)	N	3K	
DFHEEX	N	1K	

APP A – F

Module	Type	Size	Suffix
DFHEFC (FCP)	N	3K	
DFHEGL (ZCC)	N	2K	
DFHEIC (ICP)	N	1K	
DFHEIP	N	7K	
DFHEITSP	P	4K	
DFHEJC (JCP)	N	1K	
DFHEKC (KCP)	N	1K	
DFHELR	N	2K	
DFHEMA	P	3K	
DFHEMB	P	3K	
DFHEMC	P	4K	
DFHEMD	P	4K	
DFHEME	P	3K	
DFHEMF	P	4K	
DFHEMG	P	3K	
DFHEMH	P	1K	
DFHEMI	P	2K	
DFHEMS (MCP)	N	3K	
DFHEMTA	P	4K	
DFHEMTD	P	53K	
DFHEMTP	P	4K	
DFHEOTP	P	4K	
DFHEPC (PCP)	N	2K	
DFHERM	N	2K	
DFHESC (SCP)	N	1K	
DFHESP (SPP)	N	1K	
DFHESTP	P	4K	
DFHETC (ZCP)	N	5K	
DFHETD (TDP)	N	1K	
DFHETL	N	5K	
DFHETR (TRP)	N	1K	
DFHETS (TSP)	N	1K	
DFHFCP	N	10K	A$
		9K	B$
		12K	C$
		13K	D$
		1K	DY
		14K	S$
		6K	1$
		7K	2$
		8K	3$
		8K	4$
		11K	5$
		11K	6$
		12K	7$
		7K	8$
		8K	9$
DFHFDP	N	17K	
DFHGMM	P	2K	
DFHICP	N	4K	
DFHIIP	N	2K	A$
		2K	S$
DFHIRP	N	8K	
DFHISP	N	2K	
DFHJCBSP	P	1K	
DFHJCC	P	1K	
DFHJCEOV	P	1K	
DFHJCSDJ	P	1K	
DFHKCP	N	13K	
DFHKPP	N	9K	D$
		1K	DY
DFHLFO	N	2K	
DFHMCP	N	6K	A$
		1K	DY
		5K	E$
		10K	S$
DFHMCX	N	5K	
DFHMGP	N	8K	
DFHMGT	N	31K	
DFHMIR	P	1K	
DFHML1	N	4K	

Module	Type	Size	Suffix
DFHMSP	P	11K	
DFHMXP	P	1K	
DFHM32	N	5K	A$
		5K	S$
DFHPBP	N	7K	A$
		8K	S$
DFHPCP	N	8K	1$
		9K	2$
		10K	3$
DFHPHP	N	2K	
DFHPUP	P	9K	
DFHRLR	N	2K	A$
		4K	S$
DFHRTE	P	2K	
DFHSCP	N	7K	
DFHSCR	N	4K	
DFHSCTE	-	1K	
DFHSFP	P	1K	
DFHSNP	P	5K	
DFHSPP	N	11K	
DFHSRP	N	3K	DY
		4K	1$
DFHTDP	N	1K	DY
		1K	1$
		4K	2$
		6K	4$
		4K	6$
		7K	7$
DFHTMP	N	9K	
DFHTPP	N	2K	A$
		3K	S$
DFHTPQ	P	3K	
DFHTPR	P	15K	
DFHTPS	P	4K	
DFHTSP	N	1K	DY
		4K	1$
		8K	2$
		10K	3$
DFHUEH	N	1K	
DFHUEM	N	3K	
DFHVCP	N	9K	
		1K	DY
DFHXFP	N	17K	
DFHXSP	N	1K	
		1K	DY
DFHXTP	N	4K	
DFHZCA	N	4K	
DFHZCB	N	20K	E$
		22K	S$
DFHZCC	N	22K	
DFHZCP	N	14K	E$
		17K	S$
		9K	1$
		9K	2$
		9K	3$
		9K	4$
		9K	5$
DFHZCW	N	8K	
DFHZCX	N	15K	E$
		15K	S$
		15K	1$
		15K	2$
		15K	3$
		15K	4$
		15K	5$
DFHZCY	N	19K	
DFHZCZ	N	9K	E$
		9K	S$
DFHZNAC	P	15K	
DFHZRLG	P	1K	
DFHZRSP	P	1K	

Appendix C. Pregenerated Modules

The suffixed pregenerated modules provided in the core image library are summarized below under the following two headings:

- Modules with a dummy pregenerated version
- Modules with one or more nondummy suffixed pregenerated versions.

These summaries are followed by a listing of the DFHSG macro instructions used to generate the CICS/DOS/VS system supplied on the distribution tape.

MODULES WITH A DUMMY PREGENERATED VERSION

The following modules have a dummy pregenerated version, with suffix DY, provided in the distribution core image library:

DFHBFP	Built-in functions program
DFHCMP	CICS monitoring program
DFHDCP	Dump control program
DFHDIP	Batch data interchange program
DFHFCP	File control program
DFHJCP	Journal control program
DFHKPP	Keypoint program
DFHMCP	Mapping control program
DFHSRP	System recovery program
DFHTDP	Transient data control program
DFHTRP	Trace control program
DFHTSP	Temporary storage control program
DFHVCP	Volume control manager program
DFHXSP	External security interface program

MODULES WITH NONDUMMY SUFFIXED PREGENERATED VERSIONS

The modules listed below have nondummy suffixed pregenerated versions, with suffixes and functions as indicated, provided in the distribution core image library. All other modules are provided unsuffixed even though some of these modules are also suffixable.

You should find sufficient information here to allow you to select the appropriate suffixed version of a particular CICS module, or group of modules, in order to satisfy the requirements of your installation. (For further information on the parameters that have been selected for the pregenerated CICS system, see the system generation job stream listed at the end of this appendix.)

The suffixed modules are listed under the names of their respective CICS system generation program groups, which are described in the CICS/VS Customization Guide.

Built-in Functions Program (BFP)

DFHBFP

Suffix	1$	All built-in functions
	2$	Basic functions only
	3$	Weighted retrieval function only

Basic Mapping Support Program (BMS)

DFHDSB	Data stream builder
DFHIIP	Non-3270 input mapping
DFHMCP	Mapping control program
DFHM32	3270 mapping
DFHPBP	Page build program
DFHRLR	Route list resolution
DFHTPP	Terminal page program

Suffix	E$	Minimum function	(DFHMCP only)
	A$	Standard function	(All modules listed above)
	S$	Full function	(All modules listed above)

Note: Details of the devices supported and the function provided by each version of BMS (minimum, standard, and full) are given in the CICS/VS Customization Guide.

Dynamic Transaction Backout Program (DBP)

DFHDBP

Suffix	1$	Without DL/I support
	2$	With DL/I support

Dump Control Program (DCP)

DFHDCP

Suffix	4$	For tape output device
	D$	For disk output device

Batch Data Interchange Program (DIP)

DFHDIP

Suffix 1$ -

Command (EXEC) Language Translator Program (EXP)

DFHEAP	Assembler command-level translator
DFHECP	COBOL command-level translator
DFHEPP	PL/I command-level translator
DFHERP	RPG II command-level translator

Suffix 1$ - (All modules listed above)

File Control Program (FCP)

DFHFCP

Suffix	1$	ISAM	
	2$	DAM	
	3$	VSAM	
	4$	ISAM + DAM	
	5$	ISAM + VSAM	
	6$	DAM + VSAM	
	7$	ISAM + DAM + VSAM	
	8$	ISAM	with automatic journaling
	9$	DAM	" " "
	A$	VSAM	" " "
	B$	ISAM + DAM	" " "
	C$	ISAM + VSAM	" " "
	D$	DAM + VSAM	" " "
	S$	ISAM + DAM + VSAM (full function)	" " "

Journal Control Program (JCP)

DFHJCP

Suffix	1$	With automatic journaling, dynamic logging using main temporary storage, and support for "Note" requests
	2$	With automatic journaling, dynamic logging using auxiliary temporary storage, and support for "Note" requests

Keypoint Program (KPP)

DFHKPP

Suffix D$ With activity keypointing

Program Control Program (PCP)

DFHPCP

Suffix	1$	PL/I + Assembler + RPG II	with HLL trace
	2$	COBOL + Assembler + RPG II	" " "
	3$	COBOL + PL/I + Assembler + RPG II	" " "

System Recovery Program (SRP)

DFHSRP

Suffix 1$ -

The sample system recovery table (SRT) with a suffix of 1$ can be used on your production system together with the pregenerated version of DFHSRP.

Transaction Backout Program (TBP)

DFHTBP

Suffix	1$	Without DL/I support
	2$	With DL/I support

Terminal Control Program (TCP)

DFHTCP	Non-VTAM terminals only
DFHZCB	VTAM terminals only
DFHZCP	VTAM and non-VTAM terminals
DFHZCX	VTAM and non-VTAM terminals
DFHZCZ	VTAM terminals only

The following suffixed pregenerated versions of DFHTCP, DFHZCP, and DFHZCX provide support for the devices shown, using SAM or BTAM as applicable:

Suffix	1$	Console + local 3270 (displays and printers)
	2$	Console + remote 3270 + 3275 dial-up
	3$	Console + local 3270 + remote 3270 + 3275 dial-up
	4$	Console + CRLP (card reader, line printer) + DASD
	5$	Console + 2740

The following suffixed pregenerated versions of DFHTCP, DFHZCB, DFHZCP, DFHZCX, and DFHZCZ provide support for the devices shown, using SAM, BTAM, or VTAM as applicable:

Suffix	E$	Console + VTAM 3270, 3600, and 3790
	S$	Console + all SAM devices + all BTAM devices + all VTAM devices (full function)

Transient Data Control Program (TDP)

DFHTDP

Suffix	1$	Extrapartition only
	2$	Extrapartition + DAM intrapartition with ATI, but without recovery
	4$	Extrapartition + DAM intrapartition with ATI and recovery
	6$	Extrapartition + VSAM intrapartition with ATI, but without recovery
	7$	Extrapartition + VSAM intrapartition with ATI and recovery

Trace Control Program (TRP)

DFHTRP

Suffix	4$	Main storage + auxiliary on tape device
	5$	Main storage only
	D$	Main storage + auxiliary on disk device

Temporary Storage Control Program (TSP)

DFHTSP

Suffix	1$	Main storage only
	2$	Main storage + auxiliary (VSAM) without recovery
	3$	Main storage + auxiliary (VSAM) with recovery (requires a temporary storage table)

DFHSG MACRO INSTRUCTIONS FOR THE PREGENERATED SYSTEM

The DFHSG TYPE=INITIAL macro instruction used to provide the pregenerated CICS system had the following options specified or defaulted:

DLI=YES	DL/I support	(specified)
VTAM=YES	VTAM support	(defaulted)

These specifications ensure that the CICS system supports the associated functions, and that no regeneration should be necessary if, for instance, VTAM is subsequently added to the system. Note that the specification of these options adds a very small amount of code to a few high-use CICS programs. The effect on total CICS nucleus size and performance is negligible. The CICS programs affected by the specification of these operands will work whether or not the installation uses the associated functions.

The following listing shows the DFHSG macro instructions used to generate the pregenerated system. You should check the listing to ensure that the support you require is provided by the pregenerated programs. (For a list of the programs generated by each of the DFHSG macros, see the _CICS/VS Customization Guide_).

These DFHSG macro instructions are contained in A.DFHSG04 in the source statement library on the basic distribution tape.

```
ATP      DFHSG PROGRAM=ATP
BFP1$    DFHSG PROGRAM=BFP,SUFFIX=1$,   BUILT IN FUNCTION PROGRAM      X
               BUILTIN=(BASIC,WTRET)      -ALL FUNCTIONS
BFP2$    DFHSG PROGRAM=BFP,SUFFIX=2$,                                  X
               BUILTIN=BASIC              -BASIC FUNCTIONS
BFP3$    DFHSG PROGRAM=BFP,SUFFIX=3$,                                  X
               BUILTIN=WTRET              -WEIGHTED RETRIEVAL
BMSE$    DFHSG PROGRAM=BMS,SUFFIX=E$,   3270 MINIMUM FUNCTION          X
               BMSFUNC=MINIMUM
BMSA$    DFHSG PROGRAM=BMS,SUFFIX=A$,   STANDARD FUNCTION BMS          X
               BMSFUNC=STANDARD
BMSS$    DFHSG PROGRAM=BMS,                                            *
               SUFFIX=S$,                                              *
               BMSFUNC=FULL        FULL FUNCTION BMS
CSA      DFHSG PROGRAM=CSA
* CSD    DFHSG PROGRAM=CSD         DUMMIES GENERATED LATER
CSO      DFHSG PROGRAM=CSO
CSS      DFHSG PROGRAM=CSS
CSU      DFHSG PROGRAM=CSU         UTILITIES  DUP, TUP & STUP
DBP1$    DFHSG PROGRAM=DBP,SUFFIX=1$,   DYNAMIC BACKOUT PROGRAM        X
               DLI=NO                   WITHOUT DL/I SUPPORT
DBP2$    DFHSG PROGRAM=DBP,SUFFIX=2$,   DBP WITH DL/I SUPPORT          X
               DLI=YES
DCP4$    DFHSG PROGRAM=DCP,SUFFIX=4$,                                  X
               DEVICE=TAPE
DCPD$    DFHSG PROGRAM=DCP,SUFFIX=D$,                          DISK    X
               DEVICE=DISK
DIP1$    DFHSG PROGRAM=DIP,SUFFIX=1$
EIP      DFHSG PROGRAM=EIP
EXP1$    DFHSG PROGRAM=EXP,SUFFIX=1$,   COMMAND LEVEL TRANSLATORS      X
               LANG=(COBOL,PL/I,ASM,RPG)
FCP1$    DFHSG PROGRAM=FCP,SUFFIX=1$,   ISAM                           X
               FILSERV=(ISAM),                                         X
               AUTOJRN=NO
FCP2$    DFHSG PROGRAM=FCP,SUFFIX=2$,   DAM                            X
               FILSERV=(DAM),                                          X
               AUTOJRN=NO
FCP3$    DFHSG PROGRAM=FCP,SUFFIX=3$,   VSAM                           X
               FILSERV=(VSAM),                                         X
               AUTOJRN=NO
FCP4$    DFHSG PROGRAM=FCP,SUFFIX=4$,   ISAM + DAM                     X
               FILSERV=(ISAM,DAM),                                     X
               AUTOJRN=NO
FCP5$    DFHSG PROGRAM=FCP,SUFFIX=5$,   ISAM + VSAM                    X
               FILSERV=(ISAM,VSAM),                                    X
               AUTOJRN=NO
FCP6$    DFHSG PROGRAM=FCP,SUFFIX=6$,   DAM + VSAM                     X
               FILSERV=(DAM,VSAM),                                     X
               AUTOJRN=NO
FCP7$    DFHSG PROGRAM=FCP,SUFFIX=7$,   ISAM + DAM + VSAM              X
               FILSERV=(ISAM,DAM,VSAM),                                X
               AUTOJRN=NO
FCP8$    DFHSG PROGRAM=FCP,SUFFIX=8$,   ISAM WITH AUTOJRN              X
               FILSERV=(ISAM),                                         X
               AUTOJRN=YES
FCP9$    DFHSG PROGRAM=FCP,SUFFIX=9$,   DAM WITH AUTOJRN               X
               FILSERV=(DAM),                                          X
               AUTOJRN=YES
FCPA$    DFHSG PROGRAM=FCP,SUFFIX=A$,   VSAM WITH AUTOJRN              X
               FILSERV=(VSAM),                                         X
               AUTOJRN=YES
FCPB$    DFHSG PROGRAM=FCP,SUFFIX=B$,   ISAM + DAM WITH AUTOJRN        X
               FILSERV=(ISAM,DAM),                                     X
               AUTOJRN=YES
FCPC$    DFHSG PROGRAM=FCP,SUFFIX=C$,   ISAM + VSAM WITH AUTOJRN       X
               FILSERV=(ISAM,VSAM),                                    X
               AUTOJRN=YES
FCPD$    DFHSG PROGRAM=FCP,SUFFIX=D$,   DAM + VSAM WITH AUTOJRN        X
               FILSERV=(DAM,VSAM),                                     X
               AUTOJRN=YES
FCPS$    DFHSG PROGRAM=FCP,SUFFIX=S$,   SUPPORTS ALL FILE SERVICES     X
               FILSERV=(ISAM,DAM,VSAM),                                X
               AUTOJRN=YES
```

```
HLL      DFHSG PROGRAM=HLL,        HIGH LEVEL LANGUAGE SUPPORT           X
               LANG=(COBOL,PL/I)   FOR MACRO INTERFACE
ICP      DFHSG PROGRAM=ICP
ISC      DFHSG PROGRAM=ISC              INTER SYSTEMS COUPLING
JCP1$    DFHSG PROGRAM=JCP,SUFFIX=1$,   JOURNAL CONTROL PROGRAMS         X
               DTB=MAIN,                          WITH DTB               X
               AUTOJRN=YES,                                              X
               NOTE=YES
JCP2$    DFHSG PROGRAM=JCP,SUFFIX=2$,                                    X
               DTB=AUX,            AUX TEMP STORAGE                      X
               AUTOJRN=YES,                                              X
               NOTE=YES            NOTE SUPPORT
KCP      DFHSG PROGRAM=KCP
KPPD$    DFHSG PROGRAM=KPP,SUFFIX=D$,                                    X
               AKP=YES
MTP      DFHSG PROGRAM=MTP         MASTER TERMINAL PROGRAMS
OCP      DFHSG PROGRAM=OCP         OPEN CLOSE PROGRAM
PCP1$    DFHSG PROGRAM=PCP,SUFFIX=1$,   PL/I + ASSEMBLER + RPG           X
               LANG=(PL/I),                                              X
               HLLTR=YES                TO WRITE IN TRT
PCP2$    DFHSG PROGRAM=PCP,SUFFIX=2$,   COBOL + ASSEMBLER + RPG          X
               LANG=(COBOL),COBOL=(V2,V3,SUBSET),    INCL VS             X
               HLLTR=YES                TO WRITE IN TRT
PCP3$    DFHSG PROGRAM=PCP,SUFFIX=3$,   COBOL + PL/I + ASSEMBLER + RPG X
               LANG=(COBOL,PL/I),                                        X
               COBOL=(V2,V3,SUBSET),    INCLUDES VS                      X
               HLLTR=YES                TO WRITE IN TRT
SCP      DFHSG PROGRAM=SCP         STORAGE CONTROL PROGRAM
SRP1$    DFHSG PROGRAM=SRP,             SYSTEM RECOVERY PROGRAM          X
               SUFFIX=1$
TBP1$    DFHSG PROGRAM=TBP,SUFFIX=1$,   TRANSACTION BACKOUT PROGRAM      X
               DLI=NO                   WITHOUT DL/I SUPPORT
TBP2$    DFHSG PROGRAM=TBP,SUFFIX=2$,   TBP WITH DL/I SUPPORT            X
               DLI=YES
TCP1$    DFHSG PROGRAM=TCP,SUFFIX=1$,   CONS + L3270 W/PRTR              X
               ACCMETH=(SAM,BTAM),                                       X
               DEVICE=(CONSOLE),                                         X
               BTAMDEV=(L3277,L3284,L3286),                              X
               AUTOTRN=YES,             AUTO TRANS INIT                  X
               UCTRAN=(EBCDIC),                                          X
               WRAPLST=NO,                                               X
               PUNSOL=NO                VTAM ONLY
TCP2$    DFHSG PROGRAM=TCP,SUFFIX=2$,   CONS + REMOTE 3270               X
               ACCMETH=(SAM,BTAM),                                       X
               DEVICE=(CONSOLE),                                         X
               BTAMDEV=(3275D,R3270),                                    X
               FEATURE=(AUTOPOLL,AUTOANSW),                              X
               ANSWRBK=(EXIDVER),       FOR 3275 DIAL                    X
               WRAPLST=YES,             WRAP LIST SUPPORT                X
               BSCODE=(EBCDIC),                                          X
               AUTOTRN=YES,             AUTO TRANS INIT                  X
               UCTRAN=(EBCDIC),                                          X
               PUNSOL=NO                VTAM ONLY
TCP3$    DFHSG PROGRAM=TCP,SUFFIX=3$,   CONS + LOC + REM 3270            X
               ACCMETH=(SAM,BTAM),                                       X
               DEVICE=(CONSOLE),                                         X
               BTAMDEV=(3275D,L3270,R3270),                              X
               FEATURE=(AUTOPOLL,AUTOANSW),                              X
               ANSWRBK=(EXIDVER),       FOR 3275 DIAL                    X
               WRAPLST=YES,             WRAP LIST SUPPORT                X
               BSCODE=(EBCDIC),                                          X
               AUTOTRN=YES,             AUTO TRANS INIT                  X
               UCTRAN=(EBCDIC),                                          X
               PUNSOL=NO                VTAM ONLY
TCP4$    DFHSG PROGRAM=TCP,SUFFIX=4$,   CONS + SIM TERM                  X
               ACCMETH=(SAM),                                            X
               WRAPLST=NO,                                               X
               DEVICE=(CONSOLE,CRLP,DASD),                               X
               EODI=E0,                 DEFAULT 0-2-8 PUNCH              X
               AUTOTRN=YES,                                              X
               PUNSOL=NO                VTAM ONLY
TCP5$    DFHSG PROGRAM=TCP,SUFFIX=5$,   CONS + 2740                      X
               FEATURE=(RDATT),         2740 READ ATTENTION              X
```

```
         AUTOTRN=YES,                                                 X
         ACCMETH=(SAM,BTAM),                                          X
         DEVICE=(CONSOLE),                                            X
         BTAMDEV=(2740),                                              X
         WRAPLST=NO,                NO WRAP LIST SUPPORT              X
         PUNSOL=NO                  VTAM ONLY
TCPE$    DFHSG PROGRAM=TCP,SUFFIX=E$,  CNSL + VTAM 3270/3600/3790     X
         ACCMETH=(VTAM,BSAM),                                         X
         DEVICE=CONSOLE,            CPU CONSOLE SUPPORT               X
         VTAMDEV=(3270,3600,3790),                                    X
         WRAPLST=NO,                NO BTAM WRAP LIST SUPPORT         X
         AUTOTRN=YES,               AUTO TRANSACTION INITIATION       X
         UCTRAN=EBCDIC,             UPPER CASE TRANSLATE              X
         CHNASSY=YES,               CHAIN ASSEMBLY                    X
         LOGREC=YES,                LOGICAL RECORD PRESENTATION       X
         PUNSOL=YES,                IGNORE UNSOLICITED INPUT          X
         RAQ=YES                    READ AHEAD QUEUING
TCPS$    DFHSG PROGRAM=TCP,SUFFIX=S$, SUPPORTS EVERYTHING             X
         ACCMETH=(BSAM,BTAM,SAM,VTAM),                                X
         DEVICE=(CRLP,DASD,TAPE,CONSOLE),  ALL SEQUENTIAL DEV.        X
         EODI=E0,              END BSAM INPUT WITH 0-2-8 PUNCH        X
         BTAMDEV=(1050,1050D,      ALL BTAM DEVICES                   X
         1053,                                                        X
         2260,L2260,            WILL SUPPORT 2265                     X
         2740,2740D,                                                  X
         2740-2,                                                      X
         2741C,                                                       X
         2741E,                                                       X
         2741DC,                                                      X
         2741DE,                                                      X
         2770,2770D,                                                  X
         2780,2780D,                                                  X
         2980/1,2980/2,2980/4,                                        X
         3275D,L3270,R3270,                                           X
         3600,3660,                                                   X
         3735D,                                                       X
         3740,3740D,                                                  X
         3780,3780D,                                                  X
         7770,                                                        X
         SYS/3,SYS/3D,         WILL SUPPORT S/370,BISYNC,             X
         SYS/7,SYS/7D,         S/7BSCA,S/370D,S/7BSCAD                X
         TWX,                                                         X
         TLX),                                                        X
         VTAMDEV=(3600,        ALL VTAM DEVICES                       X
         3614,                                                        X
         3650,                                                        X
         3790,                                                        X
         BCHLU,                WILL SUPPORT 3770                      X
         INTLU,                WILL SUPPORT 3767I                     X
         3270,                                                        X
         3770B,                                                       X
         3767C,                WILL SUPPORT 3770C                     X
         LUTYPE2,                                                     X
         LUTYPE3,                                                     X
         LUTYPE6,                                                     X
         3767,                                                        X
         3770I),                                                      X
         FEATURE=(AUTOANSW,AUTOPOLL,BUFFRECV,PSEUDOBIN,               X
         TRANSPARENCY,RDATT),  ALL TERMINAL FEATURES                  X
         ANSWRBK=(EXIDVER,TERMINAL,AUTOMATIC,7770TERM,7770NULL),      X
         WRAPLST=YES,                                                 X
         BSCODE=(EBCDIC,ASCII),  BOTH BISYNC CODES                    X
         AUTOTRN=YES,          AUTO. TASK INITIATION VIA DCT          X
         UCTRAN=(EBCDIC,ASCII),  UPPERCASE TRANSLATION FOR 3270       X
         TBLFIX=YES,           2980 TRANSLATE TABLES FIXED            X
         CONVTAB=(ABB,ABC,2741EU,2741EM,2741CU,2741CM),               X
         LOCKF=YES,            2848 OPTIONAL KEYBOARD LOCK FEATURE    X
         PIPELN=YES,           3650 PIPELINE SESSION INCLUDED         X
         PUNSOL=YES,                                                  X
         RAQ=YES,                                                     X
         CHNASSY=YES,                                                 X
         LOGREC=YES
TDP1$    DFHSG PROGRAM=TDP,SUFFIX=1$,   EXTRA ONLY                    X
```

```
                EXTRA=(ACQUISITION,DISPOSITION)
TDP2$     DFHSG PROGRAM=TDP,SUFFIX=2$,    EXTRA + DAM INTRA            X
                DESTRCV=NO,               NO RECOVERY                  X
                INTRA=(TRANSINIT,DAM),                                 X
                EXTRA=(ACQUISITION,DISPOSITION)
TDP4$     DFHSG PROGRAM=TDP,SUFFIX=4$,    EXTRA + DAM INTRA            X
                DESTRCV=YES,              WITH RECOVERY                X
                INTRA=(TRANSINIT,DAM),                                 X
                EXTRA=(ACQUISITION,DISPOSITION)
TDP6$     DFHSG PROGRAM=TDP,SUFFIX=6$,    EXTRA + VSAM INTRA           X
                DESTRCV=NO,               NO RECOVERY                  X
                INTRA=(TRANSINIT,VSAM),                                X
                EXTRA=(ACQUISITION,DISPOSITION)
TDP7$     DFHSG PROGRAM=TDP,SUFFIX=7$,    EXTRA + VSAM INTRA           X
                DESTRCV=YES,              WITH RECOVERY                X
                INTRA=(TRANSINIT,VSAM),                                X
                EXTRA=(ACQUISITION,DISPOSITION)
TRP4$     DFHSG PROGRAM=TRP,SUFFIX=4$,    MAIN + AUX ON TAPE           X
                AUX=YES,                  AUXILIARY TRACE              X
                DEVICE=TAPE
TRP5$     DFHSG PROGRAM=TRP,SUFFIX=5$,    MAIN STORAGE ONLY            X
                AUX=NO
TRPD$     DFHSG PROGRAM=TRP,SUFFIX=D$,    MAIN STOR + AUX ON DISK      X
                AUX=YES,                  AUXILIARY TRACE              X
                DEVICE=DISK
TSP1$     DFHSG PROGRAM=TSP,SUFFIX=1$,    MAIN STORAGE ONLY            X
                AUX=NO
TSP2$     DFHSG PROGRAM=TSP,SUFFIX=2$,    MAIN STOR + VSAM             X
                AUX=YES
TSP3$     DFHSG PROGRAM=TSP,SUFFIX=3$,    MAIN STOR + VSAM + RECOVERY  X
                AUX=REC                   REQUIRES TS TABLE
```

Appendix D. Sample Tables and Table Entries

The CICS sample table material, provided on the distribution volume, is described here under the following headings:

- Pregenerated sample tables — supporting the initial CICS system as described in Part 1 and extended in Part 2 of this book; tables satisfying VSE/ICCF requirements are also provided.
- Other sample tables — extending the initial CICS system in the areas of file control, journal control, and monitoring.
- Copy books for sample table entries — copy books included in the sample tables, and copy books for the table entries needed to run the sample application programs that operate on the sample VSAM file FILEA.

The source code used to generate the sample tables and the copy books containing sample table entries are shown in full later in this appendix.

In addition, this appendix contains a listing of the contents of the CICS system definition file after initialization.

PREGENERATED SAMPLE TABLES

The sample tables listed below are supplied pregenerated, with suffixes and functions as indicated, in the distribution core image library.

With SIT=1$ specified as a system initialization parameter, these tables support the initial CICS system as described in Part 1 of this book; the tables with a suffix of 2$ (SIT and DCT) provide a variant of this initial system intended primarily for use in the VSE/ICCF environment.

The source code used to generate the tables can be found in the source statement library as A. books with the same names as the generated tables, except for the sample sign-on table (DFHSNT), the source code for which is provided as A.DFHXSNT (to distinguish it from the corresponding table generation macro).

Most of the sample tables include copy books for the table entries, either individually or as groups. Details of these copy books are given later.

Table	Suffix	Function
Destination Control Table (DCT)	1$	Sample DCT with basic CICS facilities (extrapartition only)
	2$	1$ + ICCF intrapartition DCT entries
Program Control Table (PCT)	1$	Sample PCT with basic CICS facilities (includes ICCF PCT entries)
Processing Program Table (PPT)	1$	Sample PPT with basic CICS facilities (includes ICCF PPT entries)

Table	Suffix	Function
System Initialization Table (SIT)	1$	Sample SIT to support initial system, with TCT=1$ coded for console support. Note: Dump control, file control, journal control, monitoring, and basic mapping support are **not** provided by this SIT. (For full details, see listing of DFHSIT1$ later in this appendix).
	2$	As 1$, but with following operands: 1. DCP=D$, dump to disk 2. TDP=6$, extrapartition + VSAM intrapartition with ATI 3. DCT=2$, basic facilities + ICCF 4. PRINT=YES, TCP print requests only 5. TCT=B$, local BTAM 3270 terminals
Sign-on Table (SNT)	—	Sample SNT
System Recovery Table (SRT)	1$	Sample SRT with default entries
Terminal Control Table (TCT)	1$	Sample TCT with entries for console, and sequential (CRLP) terminal only
	B$	Sample TCT with entries for console, sequential (CRLP) terminal, and local BTAM 3270 terminals
	V$	Sample TCT with entries for console, sequential (CRLP) terminal, SNA and non-SNA VTAM 3270 terminals

OTHER SAMPLE TABLES

The following sample tables are provided in source form only. The source code can be found in the source statement library as A. books with the same names as the tables. Details of the copy books included in these tables are given later.

Table	Suffix	Function
File Control Table (FCT)	1$	Sample FCT with entry for DFHCSD
Journal Control Table (JCT)	1$	Sample JCT with entry for system log
Monitoring Control Table (MCT)	1$	Sample MCT

COPY BOOKS FOR SAMPLE TABLE ENTRIES

Copy books are provided for many of the entries in the sample tables. Further copy books are provided for the table entries needed to run the sample application programs that operate on the sample VSAM file FILEA.

The following lists summarize the functions in each of these copy books. The source code for all the copy books can be found in the A. books of the same name in the source statement library.

Destination Control Table (DCT)

```
DFHDCT1$ includes the copy books:
  DFHXDCT1 - SDSCI entry for output to SYSLST
  DFHXDCT  - entries for basic CICS facilities
```

```
DFHDCT2$ includes the copy books:
  DFHXDCT1 - SDSCI entry for output to SYSLST
  DFHXDCT  - entries for basic CICS facilities
  DFHXDCTI - ICCF DCT entries (intrapartition - L86P, etc)
```

```
The following copy book is also provided:
  DFHXDCTS - entries for Assembler FILEA samples (LOGA,L860)
```

Note: The TRANSID specified for destination L860 in copy book DFHXDCTS must be changed, as indicated in the source code, if the COBOL, PL/I, or RPG II versions of the order entry sample programs are to be run.

File Control Table (FCT)

```
DFHFCT1$ includes the copy book:
  DFHXFCT  - entry for DFHCSD
               (for Resource Definition Online support)
```

```
The following copy book is also provided:
  DFHXFCTS - entry for sample VSAM file FILEA
```

Journal Control Table (JCT)

```
DFHJCT1$ includes the copy book:
  DFHXJCT  - entry for system log (disk, 2 extents)
```

```
The following copy book is also provided:
  DFHXJCTS - entry for user journal (tape)
```

Monitoring Control Table (MCT)

```
DFHMCT1$ includes the copy book:
  DFHXMCTS - sample MCT entries
```

Program Control Table (PCT)

```
DFHPCT1$ includes the copy books:
  DFHXPCT  - basic CICS facilities
  DFHXPCTI - ICCF PCT entries
```

```
The following copy books are also provided:
  DFHXPCTA - Assembler FILEA sample transactions
  DFHXPCTC - COBOL FILEA sample transactions
  DFHXPCTP - PL/I FILEA sample transactions
  DFHXPCTR - RPG II FILEA sample transactions
```

Processing Program Table (PPT)

```
DFHPPT1$ includes the copy books:
  DFHXPPT  - basic CICS facilities
  DFHXPPTI - ICCF PPT entries
```

```
The following copy books are also provided:
  DFHXPPTA - Assembler FILEA sample programs and mapsets
  DFHXPPTC - COBOL FILEA sample programs and mapsets
  DFHXPPTP - PL/I FILEA sample programs and mapsets
  DFHXPPTR - RPG II FILEA sample programs and mapsets
```

Terminal Control Table (TCT)

```
DFHTCT1$ includes the copy books:
  DFHXTCTC - console definition (CNSL)
  DFHXTCTS - CRLP (card reader, line printer)

DFHTCTB$ includes the copy books:
  DFHXTCTC - console definition (CNSL)
  DFHXTCTS - CRLP (card reader, line printer)
  DFHXTCBI - 3 local BTAM 3270 displays
                 (L77A/B/C,SYS020/21/22)
             1 local BTAM 3270 printer
                 (L86P,SYS027)

DFHTCTV$ includes the copy books:
  DFHXTCTC - console definition (CNSL)
  DFHXTCTS - CRLP (card reader, line printer)
  DFHXTCVI - 3 SNA VTAM LUTYPE2  displays
                 (L77A/B/C, netnames D72L301/2/3)
             3 non-SNA VTAM 3270 displays
                 (L77D/E/F, netnames D72L304/5/6)
             1 non-SNA VTAM 3270 printer
                 (L86P, netname P42L308)
```

Note: L86P is the terminal identification of a printer used by ICCF; it is the name of an intrapartition destination defined in the DCT with a suffix of 2$ (see copy book DFHXDCTI).

The following copy book is provided for use with the FILEA sample programs and BTAM terminals:

```
  DFHXTCBS - 2 local BTAM 3270 displays
                 (L77A/B,SYS020/21)
             1 local BTAM 3270 printer
                 (L86O,SYS027)
```

The following copy book is provided for use with the FILEA sample programs and VTAM terminals:

```
  DFHXTCVS - 1 SNA VTAM LUTYPE2 display
                 (L77A, netname D72L301)
             1 non-SNA VTAM 3270 display
                 (L77D, netname D72L304)
             1 non-SNA VTAM 3270 printer
                 (L86O, netname P42L308)
```

Notes:

1. The terminal identification of the printer used by the sample programs must be L86O to match the intrapartition destination defined in the DCT (see copy book DFHXDCTS).

2. The DFHTCTB$ and DFHTCTV$ tables are intended primarily for use in the VSE/ICCF environment, and are not designed for use with the sample applications. If you intend to use the sample applications, see Chapter 2.3 of this book for the necessary table definitions.

SOURCE CODE FOR SAMPLE TABLES

DFHDCT1$

```
        TITLE 'DFHDCT1$ CICS SAMPLE DCT'
        DFHDCT TYPE=INITIAL,                                         X
               STARTER=YES,          ALLOWS $ IN SUFFIX              X
               SUFFIX=1$
               COPY  DFHXDCT1      - SDSCI ENTRY
               COPY  DFHXDCT       - BASIC CICS/VS FACILITIES
        DFHDCT TYPE=FINAL
```

DFHDCT2$

```
        TITLE 'DFHDCT2$ CICS ICCF DCT'
        DFHDCT TYPE=INITIAL,                                         X
               STARTER=YES,          ALLOWS $ IN SUFFIX              X
               SUFFIX=2$
               COPY  DFHXDCT1      - SDSCI ENTRY
               COPY  DFHXDCT       - BASIC CICS FACILITIES
               COPY  DFHXDCTI      - ICCF ENTRIES
        DFHDCT TYPE=FINAL
```

DFHXDCT1

```
        TITLE 'DFHXDCT1 - COPYBOOK OF SDSCI ENTRIES FOR SAMPLE DCT'
***********************************************************************
* THIS 'SDSCI' COPYBOOK MUST FOLLOW THE 'INITIAL' MACRO.              *
***********************************************************************
*
        DFHDCT TYPE=SDSCI,         OUTPUT TO SYSLST                  X
               BLKSIZE=136,                                          X
               DSCNAME=MSGUSR,                                       X
               RECFORM=VARUNB,                                       X
               DEVADDR=SYSLST,                                       X
               DEVICE=1403,                                          X
               TYPEFLE=OUTPUT
```

DFHXDCT

```
        TITLE  'DFHXDCT  - COPYBOOK ENTRIES FOR BASIC CICS FACILITIES'
CPLI    DFHDCT TYPE=EXTRA,                                           X
               DESTID=CPLI,        PL/I SYSPRT OUTPUT                X
               DSCNAME=MSGUSR
CSSL    DFHDCT TYPE=INDIRECT,                                        X
               DESTID=CSSL,        USED FOR STATISTICS, ETC.         X
               INDDEST=CPLI
CPLD    DFHDCT TYPE=INDIRECT,      PL/I DUMPS                        X
               DESTID=CPLD,                                          X
               INDDEST=CPLI
CSCS    DFHDCT TYPE=INDIRECT,      MESSAGES FROM SIGN OFF PROGRAM    X
               DESTID=CSCS,                                          X
               INDDEST=CPLI
CSDL    DFHDCT TYPE=INDIRECT,      CEDA COMMAND LOGGING TO           X
               DESTID=CSDL,        THIS DESTINATION                  X
               INDDEST=CPLI
CSML    DFHDCT TYPE=INDIRECT,      SIGN ON/OFF MESSAGES TO           X
               DESTID=CSML,        THIS DESTINATION                  X
               INDDEST=CPLI
CSMT    DFHDCT TYPE=INDIRECT,      WRITE TERM ERRORS AND ABENDS MSGS X
               DESTID=CSMT,        TO THIS DESTINATION FROM DFHTACP, X
               INDDEST=CPLI        DFHZNAC AND DFHACP
CSTL    DFHDCT TYPE=INDIRECT,                                        X
```

```
            DESTID=CSTL,         TERM I/O ERROR MSGS FROM DFHTACP    X
            INDDEST=CPLI
```

DFHXDCTI

```
************************************************************************
*   ICCF DESTINATION CONTROL TABLE (REL 3.5)  PTF0000                 *
*           82/11/22                                                   *
************************************************************************
L86P     DFHDCT TYPE=INTRA,                                             *
               DESTID=L86P,        L3286 TERMINAL PRINTER               *
               TRANSID=I$$6,                                            *
               TRIGLEV=1
PRTA     DFHDCT TYPE=INTRA,                                             *
               DESTID=PRTA,        HARDCOPY QUEUE 1                     *
               TRIGLEV=0
PRTB     DFHDCT TYPE=INTRA,                                             *
               DESTID=PRTB,        HARDCOPY QUEUE 2                     *
               TRIGLEV=0
```

DFHXDCTS

```
        TITLE  'DFHXDCTS - COPYBOOK OF DCT ENTRIES FOR SAMPLE PROGRAMS'
LOGA    DFHDCT TYPE=INDIRECT,      DESTINATION LOGA USED               X
               DESTID=LOGA,        BY CICS SAMPLE PGMS                 X
               INDDEST=CPLI
L860    DFHDCT TYPE=INTRA,         DESTINATION L860 USED BY ORDER      X
               DESTID=L860,        ENTRY QUEUE PRINT SAMPLE PROGRAMS   X
               DESTFAC=TERMINAL,   TERMINAL                            X
               TRANSID=AORQ,       <CHANGE IF WANT TO RUN 'PORQ','OREQ'X
               TRIGLEV=30          AORQ IS AUTO INIT'ED WHEN QUEUE=30
```

DFHFCT1$

```
         TITLE 'DFHFCT1$ SAMPLE FCT'
         DFHFCT TYPE=INITIAL,                                           X
               STARTER=YES,          ALLOWS $ IN SUFFIX                 X
               SUFFIX=1$
               COPY  DFHXFCT       - ENTRY FOR RDO CSD DATSET
         DFHFCT TYPE=FINAL
```

DFHXFCT

```
          TITLE 'DFHXFCT - COPYBOOK FOR DFHCSD'
          DFHFCT TYPE=DATASET,DATASET=DFHCSD,ACCMETH=(VSAM,KSDS,KEY),    *
                SERVREQ=(GET,PUT,UPDATE,NEWREC,BROWSE,DELETE),           *
                RECFORM=(VARIABLE,BLOCKED),BUFND=5,BUFNI=4,STRNO=4,      *
                OPEN=DEFERRED
```

DFHXFCTS

```
          TITLE 'DFHXFCTS - COPYBOOK FOR FILEA'
          DFHFCT TYPE=DATASET,DATASET=FILEA,ACCMETH=(VSAM,KSDS),         *
                SERVREQ=(GET,PUT,UPDATE,NEWREC,BROWSE,DELETE),           *
                RECFORM=(FIXED,BLOCKED),BUFND=3,BUFNI=2,STRNO=2,         *
                OPEN=INITIAL
```

DFHJCT1$

```
         TITLE 'DFHJCT1$ SAMPLE JCT'
         DFHJCT TYPE=INITIAL,                                          X
               STARTER=YES,          ALLOWS $ IN SUFFIX                X
               SUFFIX=1$
               COPY  DFHXJCT       - ENTRY FOR SYSTEM LOG
         DFHJCT TYPE=FINAL
```

DFHXJCT

```
          TITLE 'DFHXJCT - COPYBOOK FOR SYSTEM LOG'
          DFHJCT TYPE=ENTRY,JFILEID=SYSTEM,JTYPE=DISK2,BUFSIZE=4096,    *
                DEVADDR=(SYS019,SYS019),BUFSUV=2048,                    *
                JOUROPT=INPUT
```

DFHXJCTS

```
          TITLE 'DFHXJCTS - COPYBOOK FOR USER JOURNAL'
          DFHJCT TYPE=ENTRY,JFILEID=2,BUFSIZE=4096,DEVADDR=SYS018,      *
                JTYPE=TAPE1,FORMAT=SMF,BUFSUV=2048
```

DFHMCT1$

```
        TITLE 'DFHMCT1$ SAMPLE MCT'
        DFHMCT TYPE=INITIAL,                                           X
               STARTER=YES,          ALLOWS $ IN SUFFIX                X
               SUFFIX=1$
               COPY  DFHXMCTS      - SAMPLE MCT ENTRIES
        DFHMCT TYPE=FINAL
```

DFHXMCTS

```
         TITLE 'DFHXMCTS - SAMPLE ENTRIES'
**********************************************************************
* MONITOR CONTROL TABLE                                              *
* CLASSES ACCOUNT EXCEPTION & PERFORM ARE SEND                       *
* TO JOURNAL 02                                                      *
**********************************************************************
         DFHMCT TYPE=RECORD,                                           *
               CLASS=EXCEPT,DATASET=02,FREQ=100,MAXBUF=512
         DFHMCT TYPE=RECORD,                                           *
               CLASS=ACCOUNT,DATASET=02,FREQ=100,MAXBUF=2000
         DFHMCT TYPE=RECORD,CPU=YES,                                   *
               CLASS=PERFORM,DATASET=02,FREQ=100,MAXBUF=2000
```

DFHPCT1$

```
         TITLE 'DFHPCT1$ CICS/VS SAMPLE PROGRAM CONTROL TABLE'
         DFHPCT TYPE=INITIAL,SUFFIX=1$,STARTER=YES
               COPY  DFHXPCT            - BASIC CICS/VS FACILITIES
               COPY  DFHXPCTI           - ICCF PCT ENTRIES
         DFHPCT TYPE=FINAL
```

DFHXPCT

```
        TITLE  'DFHXPCT  -  COPYBOOK ENTRIES FOR THE SAMPLE PCT'
***********************************************************************
* NOTE THAT THIS HAS NO SECURITY. SOME OF THE TRANSACTIONS WILL NEED  *
* TO BE UPDATED TO RESTRICT THEIR USE.                                *
***********************************************************************
         DFHPCT TYPE=GROUP,FN=AKP           ACTIVITY KEYPOINT PGM
         DFHPCT TYPE=GROUP,FN=ATP           ASYNCH TRANS PROC
         DFHPCT TYPE=GROUP,FN=AUTOSTAT      AUTO STATS SUMMARY PGM
         DFHPCT TYPE=GROUP,FN=BMS           TERM PAGE RETRIEVAL..
         DFHPCT TYPE=GROUP,FN=CONSOLE       CPU CONSOLE SUPPORT
***********************************************************************
* THE NEXT TYPE=ENTRY TABLE ENTRIES TURN OFF SECURITY...              *
***********************************************************************
* I.E CECI,CECS,CEBR AND CEDF 'TYPE=ENTRY' ENTRIES SET RSLC=NO        *
* BY OVERRIDING RSLC=YES FROM 'TYPE=GROUP ENTRY.                      *
*  ... PLEASE REVIEW 'RESOURCE SECURITY' IN 'CICS/VS IOG'             *
* - APART FROM THESE POSSIBLE EXCEPTIONS 'TYPE=GROUP' ENTRIES SHOULD  *
*   BE USED, INSTEAD OF 'TYPE=ENTRY' ENTRIES, SINCE THEY GENERATE     *
*   ALL THE TRANSIDS FOR THE DEFINED FUNCTION GROUP.                  *
*   ... REFER TO 'CICS/VS RESOURCE DEFINITION'                        *
*
         DFHPCT TYPE=ENTRY,                 COMMAND INTERPRETER       X
               PROGRAM=DFHECIP,                                       X
               TRANSID=CECI,                                          X
               RSLC=NO
         DFHPCT TYPE=ENTRY,                 COMMAND INTERPRETER       X
               PROGRAM=DFHECSP,                                       X
               TRANSID=CECS,                                          X
               RSLC=NO
         DFHPCT TYPE=GROUP,FN=INTERPRETER   COMMAND INTERPRETER
*
         DFHPCT TYPE=ENTRY,                 TS BROWSE                 X
               PROGRAM=DFHEDFBR,                                      X
               TRANSID=CEBR,                                          X
               RSLC=NO
         DFHPCT TYPE=ENTRY,                 EXECUTION DIAGNOSTIC FACIL X
               PROGRAM=DFHEDFP,                                       X
               TRANSID=CEDF,                                          X
               RSLC=NO
         DFHPCT TYPE=GROUP,FN=EDF           EXECUTION DIAGNOSTIC FACIL
*
***********************************************************************
*                           FOLLOWING ENTRIES GENERATES TRANSID'S FOR:-
         DFHPCT TYPE=GROUP,FN=FE            FE TERMINAL TEST PGM
         DFHPCT TYPE=GROUP,FN=HARDCOPY      3270 PRINT BTAM & VTAM
         DFHPCT TYPE=GROUP,FN=ISC           INTERSYSTEMS COMMUNICATION
         DFHPCT TYPE=GROUP,FN=JOURNAL       JOURNALLING
         DFHPCT TYPE=GROUP,FN=MASTTERM      MASTER TERMINAL FUNCTION
         DFHPCT TYPE=GROUP,FN=MSWITCH       MESSAGE SWITCHING PGM
         DFHPCT TYPE=GROUP,FN=NUMERICS      NUMERIC SIGN ON/OFF GROUP
         DFHPCT TYPE=GROUP,FN=OPERATORS     ENH FNS-MAST,SUPV&TERM OPS
         DFHPCT TYPE=GROUP,FN=RESEND        VTAM RESEND PGM
         DFHPCT TYPE=GROUP,FN=RESPLOG       VTAM RESPONSE LOGGING PGM
         DFHPCT TYPE=GROUP,FN=SPI           SYSTEM PROGRAMMER INTERFACE
         DFHPCT TYPE=GROUP,FN=SIGNON        SIGN ON & OFF PGMS & TABLES
         DFHPCT TYPE=GROUP,FN=STANDARD      TRANSIDS REQD IN ALL CICS
         DFHPCT TYPE=GROUP,FN=TIME          TIME ADJUSTMENT PGM
         DFHPCT TYPE=GROUP,FN=VTAM          VTAM ABNORM & ERROR...PGMS
         DFHPCT TYPE=GROUP,FN=VTAMPRT       VTAM 3270 PRINT FUNCTION
```

DFHXPCTI

```
************************************************************************
*      ICCF PROGRAM CONTROL TABLE (REL 3.5)  PTF0000                    *
*           82/11/22                                                    *
************************************************************************
       DFHPCT TYPE=ENTRY,TRANSID=ICCF,PROGRAM=DTSICCF,TWASIZE=1650
       DFHPCT TYPE=ENTRY,TRANSID=iccf,PROGRAM=DTSICCF,TWASIZE=1650
       DFHPCT TYPE=ENTRY,TRANSID=I$$1,PROGRAM=DTSICCF,TWASIZE=1650
       DFHPCT TYPE=ENTRY,TRANSID=I$$2,PROGRAM=DTSICCF,TWASIZE=1650
       DFHPCT TYPE=ENTRY,TRANSID=I$$3,PROGRAM=DTSICCF,TWASIZE=1650
       DFHPCT TYPE=ENTRY,TRANSID=I$$4,PROGRAM=DTSICCF,TWASIZE=1650
       DFHPCT TYPE=ENTRY,TRANSID=I$$5,PROGRAM=DTSICCF,TWASIZE=512
       DFHPCT TYPE=ENTRY,TRANSID=I$$6,PROGRAM=DTSICCF,TWASIZE=512
       DFHPCT TYPE=ENTRY,TRANSID=I$$7,PROGRAM=DTSICCF,TWASIZE=1650
```

DFHXPCTA

```
         TITLE  'DFHXPCTA - COPYBOOK FOR ASSEMBLER SAMPLE PROGRAMS'
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAMNU,TRANSID=AMNU
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAALL,TRANSID=AINQ
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAALL,TRANSID=AADD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAALL,TRANSID=AUPD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHABRW,TRANSID=ABRW
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAREN,TRANSID=AORD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHACOM,TRANSID=AORQ
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHAREP,TRANSID=AREP
```

DFHXPCTC

```
         TITLE  'DFHXPCTC - COPYBOOK FOR COBOL SAMPLE PROGRAMS'
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHINST,TRANSID=MENU
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHCALL,TRANSID=INQY
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHCALL,TRANSID=ADDS
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHCALL,TRANSID=UPDT
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHBRWS,TRANSID=BRWS
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHOREN,TRANSID=OREN
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHCCOM,TRANSID=OREQ
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHREPT,TRANSID=REPT
```

DFHXPCTP

```
         TITLE  'DFHXPCTP - COPYBOOK FOR PL/I SAMPLE PROGRAMS'
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHPMNU,TRANSID=PMNU
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHPALL,TRANSID=PINQ
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHPALL,TRANSID=PADD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHPALL,TRANSID=PUPD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHPBRW,TRANSID=PBRW
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHPORD,TRANSID=PORD
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHPCOM,TRANSID=PORQ
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHPREP,TRANSID=PREP
```

DFHXPCTR

```
         TITLE  'DFHXPCTR - COPYBOOK FOR RPG  SAMPLE PROGRAMS'
          DFHPCT TYPE=ENTRY,PROGRAM=XDFHRMNU,TRANSID=RMNU
```

```
DFHPCT TYPE=ENTRY,PROGRAM=XDFHRALL,TRANSID=RINQ
DFHPCT TYPE=ENTRY,PROGRAM=XDFHRALL,TRANSID=RADD
DFHPCT TYPE=ENTRY,PROGRAM=XDFHRALL,TRANSID=RUPD
DFHPCT TYPE=ENTRY,PROGRAM=XDFHRBRW,TRANSID=RBRW
DFHPCT TYPE=ENTRY,PROGRAM=XDFHRORD,TRANSID=RORD
DFHPCT TYPE=ENTRY,PROGRAM=XDFHRCOM,TRANSID=RORQ
DFHPCT TYPE=ENTRY,PROGRAM=XDFHRREP,TRANSID=RREP
```

DFHPPT1$

```
         TITLE 'DFHPPT1$ CICS/VS SAMPLE PROCESSING PROGRAM TABLE'
         DFHPPT TYPE=INITIAL,SUFFIX=1$,STARTER=YES
               COPY  DFHXPPT             - BASIC CICS/VS FACILITIES
               COPY  DFHXPPTI            - ICCF PPT ENTRIES
         DFHPPT TYPE=FINAL
```

DFHXPPT

```
        TITLE  'DFHXPPT  - COPYBOOK ENTRIES FOR THE SAMPLE PPT'
         DFHPPT TYPE=ENTRY,PROGRAM=DFHNEP   SAMPLE ERROR NODE PGM
         DFHPPT TYPE=ENTRY,PROGRAM=DFHNET   SAMPLE ERROR NODE TABLE
         DFHPPT TYPE=ENTRY,PROGRAM=DFHPEP   SAMPLE PROGRAM ERROR PGM
         DFHPPT TYPE=ENTRY,PROGRAM=DFHRTY   TRANSACTION RESTART&BACKOUT
         DFHPPT TYPE=GROUP,FN=AKP           ACTIVITY KEYPOINT PROGRAM
         DFHPPT TYPE=GROUP,FN=ATP           ASYNCH TRANS PROC
         DFHPPT TYPE=GROUP,FN=AUTOSTAT      AUTO STATS SUMMARY PGM
         DFHPPT TYPE=GROUP,FN=BACKOUT       DYNAMIC BACKOUT
         DFHPPT TYPE=GROUP,FN=BMS           BASIC MAPPING SUPPORT
         DFHPPT TYPE=GROUP,FN=CONSOLE       WRITE TO CPU CONSOLE
         DFHPPT TYPE=GROUP,FN=EDF           EXECUTION DIAGNOSTIC FACIL
         DFHPPT TYPE=GROUP,FN=FE            FE TERMINAL TEST PGM
         DFHPPT TYPE=GROUP,FN=HARDCOPY      3270 PRINTER-BTAM & VTAM
         DFHPPT TYPE=GROUP,FN=INTERPRETER   COMMAND INTERPRETER
         DFHPPT TYPE=GROUP,FN=ISC           INTERSYSTEMS COMMUNICATION
         DFHPPT TYPE=GROUP,FN=JOURNAL       JOURNALLING
         DFHPPT TYPE=GROUP,FN=MASTTERM      MASTER TERMINAL FUNCTION
         DFHPPT TYPE=GROUP,FN=MSWITCH       MESSAGE SWITCHING PGM
         DFHPPT TYPE=GROUP,FN=OPENCLSE      DYNAMIC OPEN/CLOSE PGM
         DFHPPT TYPE=GROUP,FN=OPERATORS     ENH FNS-MAST,SUPV&TERM OPS
         DFHPPT TYPE=GROUP,FN=PL/I          PL/I SUPPORT IN CICS/VS
         DFHPPT TYPE=GROUP,FN=RECOVERY      RECOVERY/RESTART FACIL'Y
         DFHPPT TYPE=GROUP,FN=RESEND        VTAM RESEND PGM
         DFHPPT TYPE=GROUP,FN=RESPLOG       VTAM RESPONSE LOGGING PGM
         DFHPPT TYPE=GROUP,FN=SIGNON        SIGN ON & OFF PGMS & TABLES
         DFHPPT TYPE=GROUP,FN=SPI           SYYTEM PROGRAMMER INTERFACE
         DFHPPT TYPE=GROUP,FN=STANDARD      APPL PGMS REQD IN ALL CICS
         DFHPPT TYPE=GROUP,FN=TIME          TIME ADJUSTMENT PGM
         DFHPPT TYPE=GROUP,FN=VTAM          VTAM ABNORM & ERROR...PGMS
         DFHPPT TYPE=GROUP,FN=VTAMPRT       VTAM TERM CNTL PRINT KEY FN
```

DFHXPPTI

```
***********************************************************************
*    ICCF PROCESSING PROGRAM TABLE (REL 3.5)  PTF0000                 *
*             82/11/22                                                *
***********************************************************************
      DFHPPT  TYPE=ENTRY,RES=YES,PROGRAM=DTSICCF
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX00
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX01
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX02
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX03
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX04
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX05
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX06
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX07
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX08
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX09
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX10
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX11
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX12
      DFHPPT  TYPE=ENTRY,RES=NO,PROGRAM=DTSTX13
```

DFHXPPTA

```
TITLE  'DFHXPPTA - COPYBOOK FOR ASSEMBLER SAMPLE PROGRAMS'
 DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHAMNU
 DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHAALL
 DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHABRW
 DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHAREN
 DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHACOM
 DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=XDFHAREP
 DFHPPT TYPE=ENTRY,MAPSET=XDFHAMA
 DFHPPT TYPE=ENTRY,MAPSET=XDFHAMB
 DFHPPT TYPE=ENTRY,MAPSET=XDFHAMC
 DFHPPT TYPE=ENTRY,MAPSET=XDFHAMD
 DFHPPT TYPE=ENTRY,MAPSET=XDFHAMK
 DFHPPT TYPE=ENTRY,MAPSET=XDFHAML
```

DFHXPPTC

```
TITLE  'DFHXPPTC - COPYBOOK FOR COBOL SAMPLE PROGRAMS'
 DFHPPT TYPE=ENTRY,PGMLANG=COBOL,PROGRAM=XDFHINST
 DFHPPT TYPE=ENTRY,PGMLANG=COBOL,PROGRAM=XDFHCALL
 DFHPPT TYPE=ENTRY,PGMLANG=COBOL,PROGRAM=XDFHOREN
 DFHPPT TYPE=ENTRY,PGMLANG=COBOL,PROGRAM=XDFHCCOM
 DFHPPT TYPE=ENTRY,PGMLANG=COBOL,PROGRAM=XDFHBRWS
 DFHPPT TYPE=ENTRY,PGMLANG=COBOL,PROGRAM=XDFHREPT
 DFHPPT TYPE=ENTRY,MAPSET=XDFHCMA
 DFHPPT TYPE=ENTRY,MAPSET=XDFHCMB
 DFHPPT TYPE=ENTRY,MAPSET=XDFHCMC
 DFHPPT TYPE=ENTRY,MAPSET=XDFHCMD
 DFHPPT TYPE=ENTRY,MAPSET=XDFHCMK
 DFHPPT TYPE=ENTRY,MAPSET=XDFHCML
```

DFHXPPTP

```
TITLE  'DFHXPPTP - COPYBOOK FOR PL/I SAMPLE PROGRAMS'
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,PROGRAM=XDFHPMNU
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,PROGRAM=XDFHPALL
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,PROGRAM=XDFHPORD
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,PROGRAM=XDFHPCOM
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,PROGRAM=XDFHPBRW
 DFHPPT TYPE=ENTRY,PGMLANG=PL/I,PROGRAM=XDFHPREP
 DFHPPT TYPE=ENTRY,MAPSET=XDFHPMA
 DFHPPT TYPE=ENTRY,MAPSET=XDFHPMB
 DFHPPT TYPE=ENTRY,MAPSET=XDFHPMC
 DFHPPT TYPE=ENTRY,MAPSET=XDFHPMD
 DFHPPT TYPE=ENTRY,MAPSET=XDFHPMK
 DFHPPT TYPE=ENTRY,MAPSET=XDFHPML
```

DFHXPPTR

```
TITLE  'DFHXPPTR - COPYBOOK FOR RPG  SAMPLE PROGRAMS'
 DFHPPT TYPE=ENTRY,RELOAD=YES,PROGRAM=XDFHRREP,PGMLANG=RPG
 DFHPPT TYPE=ENTRY,RELOAD=YES,PROGRAM=XDFHRBRW,PGMLANG=RPG
 DFHPPT TYPE=ENTRY,RELOAD=YES,PROGRAM=XDFHRMNU,PGMLANG=RPG
 DFHPPT TYPE=ENTRY,RELOAD=YES,PROGRAM=XDFHRALL,PGMLANG=RPG
 DFHPPT TYPE=ENTRY,RELOAD=YES,PROGRAM=XDFHRREN,PGMLANG=RPG
 DFHPPT TYPE=ENTRY,RELOAD=YES,PROGRAM=XDFHRCOM,PGMLANG=RPG
 DFHPPT TYPE=ENTRY,MAPSET=XDRMA
 DFHPPT TYPE=ENTRY,MAPSET=XDRMB
 DFHPPT TYPE=ENTRY,MAPSET=XDRMC
 DFHPPT TYPE=ENTRY,MAPSET=XDRMD
 DFHPPT TYPE=ENTRY,MAPSET=XDRMK
 DFHPPT TYPE=ENTRY,MAPSET=XDRML
```

DFHSIT1$

```
         TITLE 'DFHSIT1$ CICS SAMPLE SIT'
         DFHSIT TYPE=CSECT,                                            X
               ALT=NO,                  NO APPLICATION LOAD TABLE      X
               AKPFREQ=0,               NO ACTIVITY KEYPOINTING        X
               AMXT=10,                 MAX ACTIVE TASKS               X
               APPLID=DBDCCICS,         APPLICATION NAME OF CICS SYSTEMX
               ATP=NO,                  NO ASYNCH TRANS PROCESSING     X
               BFP=1$,                  BUILTIN FUNCTION, BASIC + WTRETX
               BMS=NO,                  BASIC MAPPING SUPPORT          X
               CMXT=(10,10,10,10,10,10,10,10,10,10), 10 TASKS/ TRANS CLX
               DATFORM=MMDDYY,          EXTERNAL DATE DISPLAY          X
               DBP=NO,                  DYNAMIC BACKOUT                X
               DCP=NO,                  DUMP                           X
               DCT=1$,                  BASIC FACILITIES               X
               DIP=NO,                  NO BATCH DATA INTERCHANGE      X
               DLI=NO,                  NO DL/I SUPPORT                X
               EXEC=YES,                EXEC LEVEL SUPPORT             X
               EXITS=NO,                NO USER EXIT INTERFACE         X
               EXTSEC=NO,               NO RACF SUPPORT                X
               FCT=NO,                  NO FILE CONTROL TABLE          X
               FDP=(,FORMAT),           FORMATTED DUMP PROGRAM         X
               ICP=YES,                 INTERVAL CONTROL PGM           X
               ICV=5000,                INTERVAL CONTROL EXIT TIME-MS  X
               ICVR=8000,               RUNAWAY TASK TIME              X
               ICVS=20000,              DELAY BEFORE STALL PURGE       X
               ICVTSD=250,              TERMINAL SCAN DELAY            X
               ISC=NO,                  NO INTERSYSTEM COMMUNICATION   X
               JCT=NO,                  NO JOURNALLING                 X
               KCP=YES,                 TASK CTRL PGM                  X
               KPP=NO,                  NO KEY POINT PROGRAM           X
               MCT=NO,                  NO MONITOR CONTROL TABLE       X
               MSGLVL=1,                PRINT START-UP MSGS            X
               MXT=999,                 MAX NO. OF ALL CONCURRENT TASKSX
               NLT=NO,                  DEFAULT LOAD ORDER FOR NUCLEUS X
               PCP=3$,                  PL/I, ASSEMBLER AND COBOL      X
               PCT=1$,                  BASIC FACILITIES               X
               PGSIZE=2048,             SIZE OF VIRTUAL PAGING AREA    X
               PL1=YES,                 PL/I SUPPORT                   X
               PPT=1$,                  BASIC FACILITIES               X
               PRINT=PA1,               REQUEST KEY FOR 3270 PRINTOUT  X
               PRGDLAY=0,               NO TERMINAL PAGE CLEAN UP      X
               SCS=12288,               STORAGE CUSHION-MIN OF 4 PAGES X
               SRP=1$,                  SYSTEM RECOVERY PGM            X
               SRT=1$,                  DEFAULT SRT                    X
               START=COLD,              COLD START                     X
               STARTER=YES,             ALLOWS $ IN SUFFIX             X
               SUFFIX=1$,               SIT SUFFIX                     X
               SVD=YES,                 STORAGE VIOLATION DUMP&RECOVERYX
               TBP=NO,                  NO TRANSACTION BACKOUT         X
               TCP=S$,                  TERMINAL CONTROL PROGRAM       X
               TCT=1$,                  CONSOLE AND SEQUENTIAL TERMS   X
               TDP=1$,                  TRANSIENT DATA PROGRAM         X
               TRP=5$,                  MAIN ONLY                      X
               TRT=512,                 TRACE TABLE                    X
               TST=NO,                  NO TEMP STORAGE TABLE INCLUDED X
               TSMGSET=4,               4 MESSAGE SET ENTRIES          X
               TSP=1$,                  MAIN ONLY                      X
               WRKAREA=512,             COMMON WORK AREA OF THE CSA    X
               XLT=NO,                  NO TRANS LIST TABLE            X
               XSP=YES,                 EXT SECURITY PGM               X
               ZCP=S$,                  ALL ACCESS METHODS             X
               DUMMY=DUMMY              TO END MACRO
```

DFHSIT2$

```
         TITLE 'DFHSIT2$ CICS ICCF SIT'
         DFHSIT TYPE=CSECT,                                            X
               ALT=NO,                 NO APPLICATION LOAD TABLE       X
               AKPFREQ=0,              NO ACTIVITY KEYPOINTING         X
               AMXT=10,                MAX ACTIVE TASKS                X
               APPLID=DBDCCICS,        APPLICATION NAME OF CICS SYSTEMX
               ATP=NO,                 NO ASYNCH TRANS PROCESSING      X
               BFP=1$,                 BUILTIN FUNCTION, BASIC + WTRETX
               BMS=~~NO,~~ S$          BASIC MAPPING SUPPORT           X
               CMXT=(10,10,10,10,10,10,10,10,10,10), 10 TASKS/ TRANS CLX
               DATFORM=MMDDYY,         EXTERNAL DATE DISPLAY           X
               DBP=NO,                 DYNAMIC BACKOUT                 X
               DCP=D$,                 DUMP TO DISK                    X
             — DCT=2$,                 BASIC FACILITIES + ICCF         X
               DIP=NO,                 NO BATCH DATA INTERCHANGE       X
               DLI=NO,                 NO DL/I SUPPORT                 X
               EXEC=YES,               EXEC LEVEL SUPPORT              X
               EXITS=NO,               NO USER EXIT INTERFACE          X
               EXTSEC=NO,              NO RACF SUPPORT                 X
               FCT=~~NO,~~ Ø3          NO FILE CONTROL TABLE           X
               FDP=(,FORMAT),          FORMATTED DUMP PROGRAM          X
               ICP=YES,                INTERVAL CONTROL PGM            X
               ICV=5000,               INTERVAL CONTROL EXIT TIME-MS   X
               ICVR=8000,              RUNAWAY TASK TIME               X
               ICVS=20000,             DELAY BEFORE STALL PURGE        X
               ICVTSD=250,             TERMINAL SCAN DELAY             X
               ISC=NO,                 NO INTERSYSTEM COMMUNICATION    X
               JCT=NO,                 NO JOURNALLING                  X
               KCP=YES,                TASK CTRL PGM                   X
               KPP=NO,                 NO KEY POINT PROGRAM            X
               MCT=NO,                 NO MONITOR CONTROL TABLE        X
               MSGLVL=1,               PRINT START-UP MSGS             X
               MXT=999,                MAX NO. OF ALL CONCURRENT TASKSX
               NLT=NO,                 DEFAULT LOAD/ORDER FOR NUCLEUS X
               PCP=3$,                 PL/I, ASSEMBLER AND COBOL       X
               PCT=1$,                 BASIC FACILITIES                X
               PGSIZE=2048,            SIZE OF VIRTUAL PAGING AREA     X
               PL1=YES,                PL/I SUPPORT                    X
               PPT=1$,                 BASIC FACILITIES                X
               PRINT=YES,              TCP PRINT REQUESTS ONLY         X
               PRGDLAY=0,              NO TERMINAL PAGE CLEAN UP       X
               SCS=12288,              STORAGE CUSHION-MIN OF 4 PAGES X
               SRP=1$,                 SYSTEM RECOVERY PGM             X
               SRT=1$,                 DEFAULT SRT                     X
               START=COLD,             COLD START                      X
               STARTER=YES,            ALLOWS $ IN SUFFIX              X
               SUFFIX=2$,              SIT SUFFIX                      X
               SVD=YES,                STORAGE VIOLATION DUMP&RECOVERYX
               TBP=NO,                 NO TRANSACTION BACKOUT          X
               TCP=S$,                 TERMINAL CONTROL PROGRAM        X
               TCT=~~B$,~~ Ø3          BTAM BASIC FACILITIES           X
               TDP=6$,                 VSAM INTRA  + EXTRA WITH ATI    X
               TRP=5$,                 MAIN ONLY                       X
               TRT=512,                TRACE TABLE                     X
               TST=NO,                 NO TEMP STORAGE TABLE INCLUDED X
               TSMGSET=4,              4 MESSAGE SET ENTRIES           X
               TSP=1$,                 MAIN ONLY                       X
               WRKAREA=512,            COMMON WORK AREA OF THE CSA     X
               XLT=NO,                 NO TRANS LIST TABLE             X
               XSP=YES,                EXT SECURITY PGM                X
               ZCP=S$,                 ALL ACCESS METHODS              X
               DUMMY=DUMMY             TO END MACRO

               FCP=S$
```

DFHSNT

```
         TITLE 'SAMPLE SNT'
        DFHSNT TYPE=INITIAL
        DFHSNT TYPE=ENTRY,                                                *
               OPNAME='TERMINAL',        DFHSNT - OPERATOR NAME           *
               PASSWRD=MAST,        DFHSNT - PASSWORD = M A S T           *
               OPIDENT=SPR,         DFHSNT - OP IDENT = SPR               *
               SCTYKEY=(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,     *
               19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,     *
               37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,     *
               55,56,57,58,59,60,61,62,63,64),                            *
               OPPRTY=255           DFHSNT - OP PRTY = 255
        DFHSNT TYPE=ENTRY,                                                *
               OPNAME='APPLICATION',                                      *
               PASSWRD=USER,                                              *
               OPIDENT=USR,                                               *
               SCTYKEY=(1,8)
        DFHSNT TYPE=FINAL
```

DFHSRT1$

```
         TITLE 'SRT TABLE CONTAINING DEFAULT ENTRIES'
         DFHSRT TYPE=INITIAL,STARTER=YES,SUFFIX=1$
         DFHSRT TYPE=FINAL
```

DFHTCT1$

```
         TITLE 'DFHTCT1$ CICS SAMPLE TCT. CONSOLE ONLY'
         DFHTCT TYPE=INITIAL,SUFFIX=1$,STARTER=YES,                    *
               ACCMETH=(NONVTAM)
               COPY  DFHXTCTC          CONSOLE DEFINITIONS
               COPY  DFHXTCTS          CRLP    DEFINITIONS
         DFHTCT TYPE=FINAL
```

DFHTCTB$

```
         TITLE 'DFHTCTB$ CICS SAMPLE TCT. CONSOLE AND BTAM TERMINALS'
         DFHTCT TYPE=INITIAL,SUFFIX=B$,STARTER=YES,ACCMETH=NONVTAM
               COPY  DFHXTCTS          SEQUENTIAL (CRLP) DEFINITIONS
               COPY  DFHXTCTC          CONSOLE DEFINITIONS
               COPY  DFHXTCBI          BTAM DEFINITIONS
         DFHTCT TYPE=FINAL
```

DFHTCTV$

```
         TITLE 'DFHTCTV$ CICS SAMPLE TCT. CONSOLE AND VTAM TERMINALS'
         DFHTCT TYPE=INITIAL,SUFFIX=V$,STARTER=YES,                    *
               ACCMETH=(VTAM,NONVTAM)
               COPY  DFHXTCTS          SEQUENTIAL (CRLP) DEFINITIONS
               COPY  DFHXTCTC          CONSOLE DEFINITIONS
               COPY  DFHXTCVI          VTAM DEFINITIONS
         DFHTCT TYPE=FINAL
```

DFHXTCTC

```
        TITLE  'DFHXTCTC-SAMPLE TCT-COPYBOOK FOR CONSOLE'
         DFHTCT TYPE=GPENTRY,                                          X
               GPTYPE=CONSOLE,      CONSOLE SUPPORT                    X
               TRMIDNT=CNSL,        TERMINAL NAME                      X
               TRMUAL=32,           USER AREA LENGTH                   X
               LININL=80            TIOA LENGTH
```

DFHXTCTS

```
        TITLE  'DFHXTCTS - COPYBOOK TCT ENTRIES FOR SEQUENTIAL (CRLP) TX
               ERMINAL ENTRIES'
***********************************************************************
* * *                          D  T  F  S                         * * *
***********************************************************************
         DS    0D
         DC    CL32'INPUT AND OUTPUT DTFS'
         DFHTCT  TYPE=SDSCI,                                           *
               DEVADDR=SYSRDR,                                         *
               DEVICE=2540,                                            *
               DSCNAME=READER
         DFHEJECT
         DFHTCT  TYPE=SDSCI,                                           *
               DEVADDR=SYSLST,                                         *
               DEVICE=1403,                                            *
               DSCNAME=PRINTER
         DFHEJECT
***********************************************************************
* * *                    L I N E   A N D   T E R M I N A L        * * *
* * *                    D E F I N I T I O N                      * * *
***********************************************************************
         DS    0D
```

```
         DC    CL64'T E R M I N A L   A N D   L I N E   E N T R I E S'
*********************************************************************
         DS    0D
         DC    CL32'S A M A  LINE AND TERMINAL ENTRY'
         DFHTCT  TYPE=LINE,                                            *
               ACCMETH=SEQUENTIAL,                                     *
               TRMTYPE=CRLP,                                           *
               ISADSCN=READER,                                         *
               OSADSCN=PRINTER,                                        *
               INAREAL=88,                                             *
               DUMMY=DUMMY
         DFHTCT  TYPE=TERMINAL,                                        *
               TRMIDNT=SAMA,                                           *
               TRMPRTY=32,                                             *
               TRMSTAT=(TRANSCEIVE),                                   *
               DUMMY=DUMMY
*********************************************************************
* * *                  F I N A L I Z E                           * * *
*********************************************************************
         DS    0D
         DC    CL32'T C T   WAIT LIST'
```

DFHXTCBI

```
        TITLE  'DFHXTCBI-BTAM TCT-COPYBOOK FOR LOCAL TERMINALS'
        DFHTCT TYPE=GPENTRY,             1 LINE FOR:                  X
               GPTYPE=3270L,               1 LOCAL 3270 DEVICE        X
               LININL=160,                 TIOA LENGTH                X
               LINELST=(020),              ADDRESS                    X
               TRMFEAT=(DSA),              FEATURE                    X
               TRMSTAT=(A),                TERMINAL STATUS            X
               TRMMODL=(2A),               TERMINAL MODEL             X
               TRMUAL=(32),                USER AREA LENGTH           X
               TRMIDNT=(L77A)              TERMINAL NAME
        DFHTCT TYPE=GPENTRY,             1 LINE FOR:                  X
               GPTYPE=3270L,               3 LOCAL 3270 DEVICES       X
               LININL=160,                 TIOA LENGTH                X
               LINELST=(021,022,027),      ADDRESSES                  X
               TRMFEAT=(DSA,DSA,P),        FEATURES                   X
               TRMSTAT=(A,A,A),            TERMINAL STATUS            X
               TRMMODL=(2A,2A,2C),         TERMINAL MODELS            X
               TRMUAL=(32,32,32),          USER AREA LENGTH           X
               TRMIDNT=(L77B,L77C,L86P)    TERMINAL NAMES
```

DFHXTCVI

```
*********************************************************************
        TITLE  'DFHXTCVI - COPYBOOK ENTRIES FOR VTAM TCT'
*********************************************************************
* L77A - VTAM TRMTYPE LUTYPE2            NETNAME - D72L301.         *
* L77B - VTAM    ,,                      NETNAME - D72L302.         *
* L77C - VTAM    ,,                      NETNAME - D72L303.         *
* L77D - VTAM TRMTYPE 3270               NETNAME - D72L304.         *
* L77E - VTAM    ,,                      NETNAME - D72L305.         *
* L77F - VTAM    ,,                      NETNAME - D72L306.         *
* L86P - VTAM NON-SNA 3270 PRINTER.      NETNAME - P42L308.         *
*********************************************************************
VTRM1    DFHTCT TYPE=TERMINAL,TRMIDNT=L77A,TRMTYPE=LUTYPE2,TRMMODL=2,  *
               TIOAL=(1500,1920),                                      *
               BUFFER=1536,                                            *
               RELREQ=(NO,YES),                                        *
               FEATURE=(SELCTPEN,AUDALARM,DCKYBD),                     *
               CHNASSY=YES,                                            *
               NETNAME=D72L301,                                        *
               PRINTTO=(VTRM7),                                        *
               GMMSG=NO,                                               *
               ACCMETH=VTAM,TCTUAL=32,                                 *
```

```
              CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM2     DFHTCT TYPE=TERMINAL,TRMIDNT=L77B,TRMTYPE=LUTYPE2,TRMMODL=2,   *
              TIOAL=(1500,1920),                                         *
              BUFFER=1536,                                               *
              RELREQ=(NO,YES),                                           *
              FEATURE=(SELCTPEN,AUDALARM,DCKYBD),                        *
              CHNASSY=YES,                                               *
              NETNAME=D72L302,                                           *
              PRINTTO=(VTRM7),                                           *
              GMMSG=NO,                                                  *
              ACCMETH=VTAM,TCTUAL=32,                                    *
              CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM3     DFHTCT TYPE=TERMINAL,TRMIDNT=L77C,TRMTYPE=LUTYPE2,TRMMODL=2,   *
              TIOAL=(1500,1920),                                         *
              BUFFER=1536,                                               *
              RELREQ=(NO,YES),                                           *
              FEATURE=(SELCTPEN,AUDALARM,DCKYBD),                        *
              CHNASSY=YES,                                               *
              NETNAME=D72L303,                                           *
              PRINTTO=(VTRM7),                                           *
              GMMSG=NO,                                                  *
              ACCMETH=VTAM,TCTUAL=32,                                    *
              CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM4     DFHTCT TYPE=TERMINAL,TRMIDNT=L77D,TRMTYPE=3270,TRMMODL=2,      *
              CLASS=(CONV,VIDEO),TIOAL=1500,RELREQ=(NO,YES),             *
              FEATURE=(SELCTPEN,AUDALARM,DCKYBD),                        *
              NETNAME=D72L304,                                           *
              PRINTTO=(VTRM7),                                           *
              GMMSG=YES,                                                 *
              ACCMETH=VTAM,TCTUAL=32,                                    *
              CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM5     DFHTCT TYPE=TERMINAL,TRMIDNT=L77E,TRMTYPE=3270,TRMMODL=2,      *
              CLASS=(CONV,VIDEO),TIOAL=1500,RELREQ=(NO,YES),             *
              FEATURE=(SELCTPEN,AUDALARM,DCKYBD),                        *
              NETNAME=D72L305,                                           *
              PRINTTO=(VTRM7),                                           *
              GMMSG=YES,                                                 *
              ACCMETH=VTAM,TCTUAL=32,                                    *
              CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM6     DFHTCT TYPE=TERMINAL,TRMIDNT=L77F,TRMTYPE=3270,TRMMODL=2,      *
              CLASS=(CONV,VIDEO),TIOAL=1500,RELREQ=(NO,YES),             *
              FEATURE=(SELCTPEN,AUDALARM,DCKYBD),                        *
              NETNAME=D72L306,                                           *
              PRINTTO=(VTRM7),                                           *
              GMMSG=YES,                                                 *
              ACCMETH=VTAM,TCTUAL=32,                                    *
              CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM7     DFHTCT TYPE=TERMINAL,TRMIDNT=L86P,TRMTYPE=3270P,TRMMODL=2,     *
              NETNAME=P42L308,                                           *
              RELREQ=(YES,YES),                                          *
              CLASS=(CONV,VIDEO),TIOAL=1500,TCTUAL=32,                   *
              ACCMETH=VTAM,CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
```

DFHXTCBS

```
         TITLE  'DFHXTCBS-BTAM SAMPLE TCT-COPYBOOK FOR LOCAL TERMINALS'
         DFHTCT TYPE=GPENTRY,           1 LINE FOR:                      X
                GPTYPE=3270L,             LOCAL 3270 DEVICE              X
                LININL=160,               TIOA LENGTH                    X
                LINELST=(020),            ADDRESS                        X
                TRMFEAT=(DSUA),           FEATURE                        X
                TRMSTAT=(A),              TERMINAL STATUS                X
                TRMMODL=(2A),             TERMINAL MODEL                 X
                TRMUAL=(32),              USER AREA LENGTH               X
                TRMIDNT=(L77A)            TERMINAL NAME
         DFHTCT TYPE=GPENTRY,           1 LINE FOR:                      X
                GPTYPE=3270L,             LOCAL 3270 DEVICES             X
                LININL=160,               TIOA LENGTH                    X
                LINELST=(021,027),        ADDRESSES                      X
```

```
          TRMFEAT=(DSUA,P),            FEATURES                 X
          TRMSTAT=(A,A),               TERMINAL STATUS          X
          TRMMODL=(2A,2C),             TERMINAL MODELS          X
          TRMUAL=(32,32),              USER AREA LENGTH         X
          TRMIDNT=(L77B,L860)          TERMINAL NAMES
```

DFHXTCVS

```
        TITLE  'DFHXTCVS - COPYBOOK ENTRIES FOR VTAM SAMPLE TCT'
**********************************************************************
* L77A - VTAM SNA LUTYPE2.                NETNAME - D72L301.         *
* L77D - VTAM NON-SNA 3270.               NETNAME - D72L304.         *
* L860 - VTAM NON-SNA 3270 PRINTER.       NETNAME - P42L308.         *
**********************************************************************
VTRM1    DFHTCT TYPE=TERMINAL,TRMIDNT=L77A,TRMTYPE=LUTYPE2,TRMMODL=2,  *
               TIOAL=(1500,1920),RELREQ=(NO,YES),                      *
               BUFFER=1536,                                            *
               FEATURE=(SELCTPEN,AUDALARM,UCTRAN,DCKYBD),              *
               CHNASSY=YES,                                            *
               NETNAME=D72L301,                                        *
               PRINTTO=(VTRM3),                                        *
               GMMSG=NO,                                               *
               ACCMETH=VTAM,TCTUAL=32,                                 *
               CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM2    DFHTCT TYPE=TERMINAL,TRMIDNT=L77D,TRMTYPE=3270,TRMMODL=2,     *
               CLASS=(CONV,VIDEO),TIOAL=1500,RELREQ=(NO,YES),          *
               FEATURE=(SELCTPEN,AUDALARM,UCTRAN,DCKYBD),              *
               NETNAME=D72L304,                                        *
               PRINTTO=(VTRM3),                                        *
               GMMSG=YES,                                              *
               ACCMETH=VTAM,TCTUAL=32,                                 *
               CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
VTRM3    DFHTCT TYPE=TERMINAL,TRMIDNT=L860,TRMTYPE=3270P,TRMMODL=2,    *
               NETNAME=P42L308,                                        *
               RELREQ=(YES,YES),                                       *
               CLASS=(CONV,VIDEO),TIOAL=1500,TCTUAL=32,                *
               ACCMETH=VTAM,CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
```

CONTENTS OF THE CICS SYSTEM DEFINITION (CSD) FILE AFTER INITIALIZATION

All these resource definitions are "IBM-Protected" and cannot be modified. If you want to modify any of them, you must make your own copy of the group and list definitions, and amend your own copy in accordance with your requirements (see the CICS/VS Resource Definition Guide).

List Names:	DFHLIST			
Group Names:	DFHAKP	DFHATP	DFHAUTO	DFHBACK
	DFHBMS	DFHCONS	DFHEDF	DFHFE
	DFHHARDC	DFHINTER	DFHISC	DFHJRNL
	DFHMASTT	DFHMISC	DFHMSWIT	DFHNUM
	DFHOPCLS	DFHOPER	DFHPLI	DFHRCVRY
	DFHRMI	DFHRSEND	DFHSPLG	DFHSIGN
	DFHSPI	DFHSTAND	DFHTIME	DFHVTAM
	DFHVTAMP			

Group Name: DFHAKP (Activity Keypoint Program for Recovery/Restart)

Programs:	DFHAKP			
Transactions:	CSKP			

Group Name: DFHATP (Asynchronous Transaction Processing)

Programs:	DFHAQP	DFHATP	DFHRD1	DFHRD2
	DFHWT1	DFHWT2		
Transactions:	CAQP	CATP	CRDR	CWTR

Group Name: DFHAUTO (Automatic Statistics Summary Program)

Programs:	DFHSTSP			
Transactions:	CAUT			

Group Name: DFHBACK (Dynamic Backout)

Programs:	DFHDBP1$	DFHDBP2$		

Group Name: DFHBMS (Basic Mapping Support)

Programs:	DFHTPQ	DFHTPR	DFHTPS	
Transactions:	CSPG	CSPQ	CSPS	

Group Name: DFHCONS (Write to CPU Console)

Programs:	DFHCWTO			
Transactions:	CWTO			

Group Name: DFHEDF (Execution Diagnostic Facility)

Programs:	DFHDBMS	DFHEDFBR	DFHEDFD	DFHEDFF
	DFHEDFM	DFHEDFP	DFHEDFR	DFHEDFX
	DFHEIGDS	DFHEITAB		
Transactions:	CEBR	CEDF		

Group Name: DFHFE (FE Terminal Test Program)

Programs:	DFHFED1	DFHFED2	DFHFELG	DFHFEP
	DFHFERR	DFHFETX		
Transactions:	CSFE	CSFR		

Group Name: DFHHARDC (3270 Printer — BTAM and VTAM)

Programs:	DFHP3270			
Transactions:	CSPP			

Group Name: DFHINTER (Command Interpreter)

Programs:	DFHECID	DFHECIP	DFHECSP	
Transactions:	CECI	CECS		

Group Name: DFHISC (Intersystem Communication)

Profiles:	DFHCICSF	DFHCICSR	DFHCICSS	
Programs:	DFHCRNP	DFHCRP	DFHCRQ	DFHCRR
	DFHCRS	DFHCRSP	DFHLUP	DFHMIR
	DFHMXP	DFHRTE		
Transactions:	CLS1	CLS2	CMPX	CRSQ
	CRSR	CRTE	CSIR	CSMI
	CSM1	CSM2	CSM3	CSM5
	CSNC			

Group Name: DFHJRNL (Journaling)

Programs:	DFHJCBSP	DFHJCC	DFHJCEOV	DFHJCI
	DFHJCIOE	DFHJCKOJ	DFHJCO	DFHJCSDJ
Transactions:	CSJC			

X **Group Name:** DFHMASTT (Master Terminal Function)

Programs:	DFHMTPA	DFHMTPB	DFHMTPC	DFHMTPD
	DFHMTPE	DFHMTPF	DFHMTPG	
Transactions:	CSMT	CSOT	CSST	

Group Name: DFHMISC (Miscellaneous Programs)

Programs:	DFHNEP	DFHNET	DFHPEP	DFHRTY
	DFHUAKP			

Group Name: DFHMSWIT (Message Switching Program)

Programs:	DFHMSP
Transactions:	CMSG

Group Name: DFHNUM (Numeric Sign-On/Sign-Off Group)

Transactions:	8888	9999

Group Name: DFHOPCLS (Dynamic Open/Close Program)

Programs:	DFHOCP

X **Group Name:** DFHOPER (Operator Programs)

Programs:	DFHEITMT	DFHEITOT	DFHEITST	DFHEMA
	DFHEMB	DFHEMC	DFHEMD	DFHEME
	DFHEMF	DFHEMG	DFHEMH	DFHEMI
	DFHEMTA	DFHEMTD	DFHEMTP	DFHEOTP
	DFHESTP			
Trans-actions:	CEMT	CEOT	CEST	

Group Name: DFHPLI (PL/I Support in CICS)

Programs:	IBMBCCLA	IBMBCCRA	IBMBEOCA	IBMBETAA
	IBMBETBA	IBMBETCA	IBMBETIA	IBMBETOA
	IBMBETPA	IBMBETQA	IBMBETTA	IBMFEFCA
	IBMFESMA	IBMFESNA	IBMFKCSA	IBMFKMRA
	IBMFKPTA	IBMFKTBA	IBMFKTCA	IBMFKTRA
	IBMFPGDA	IBMFPMRA	IBMFSTVA	

Group Name: DFHRCVRY (Recovery/Restart Facility)

Programs:	DFHRUP	DFHTBP1$	DFHTBP2$	DFHTDRP	DFHTSRP

Group Name: DFHRMI (Resource Manager Interface)

Programs:	DFHRMSY
Transactions:	CRSY

Group Name: DFHRSEND (VTAM Resend Program)

Programs:	DFHZRSP
Transactions:	CSRS

Group Name: DFHRSPLG (VTAM Response Logging Program)

Programs:	DFHZRLG			
Transactions:	CSLG			

Group Name: DFHSIGN (Sign-On/Sign-Off Programs and Tables)

Programs:	DFHSFP	DFHSNP	DFHSNT	
Transactions:	CSSF	CSSN		

X **Group Name:** DFHSPI (Resource Definition Online)

Programs:	DFHAMP	DFHDMP	DFHEDAD	DFHEDAP
	DFHEITSP	DFHPUP		
Transactions:	CEDA			

Group Name: DFHSTAND (Standard CICS Application Programs)

Profiles:	DFHCICSA	DFHCICSE	DFHCICST	DFHCICSV
Programs:	DFHACP	DFHCCMF	DFHCMON	DFHSTKC
	DFHSTLK	DFHSTP	DFHSTPD	DFHSTTD
	DFHSTTR	DFHTACP	DFHTEP	DFHTEPT
Transactions:	CCMF	CSAC	CSTE	CSTT

Group Name: DFHTIME (Time Adjustment Program)

Programs:	DFHTAJP			
Transactions:	CSTA			

Group Name: DFHVTAM (VTAM Programs)

Programs:	DFHGMM	DFHZNAC	DFHZNEP	
Transactions:	CSGM	CSNE		

Group Name: DFHVTAMP (VTAM Terminal Control Print Key Function)

Programs:	DFHCPY	DFHEXI	DFHPRK	DFHRKB
Transactions:	CSCY	CSPK	CSRK	

Transaction Names:

Transaction	Group	Program
CCMF	DFHSTAND	DFHCCMF
CEDA	DFHSPI	DFHEDAP
CEMT	DFHOPER	DFHEMTP
CEOT	DFHOPER	DFHEOTP
CEST	DFHOPER	DFHESTP
CLS1	DFHISC	DFHLUP
CLS2	DFHISC	DFHLUP
CMPX	DFHISC	DFHMXP
CMSG	DFHMSWIT	DFHMSP
CRSQ	DFHISC	DFHCRQ
CRSR	DFHISC	DFHCRS
CRSY	DFHRMI	DFHRMSY
CRTE	DFHISC	DFHRTE
CSAC	DFHSTAND	DFHACP
CSCY	DFHVTAMP	DFHCPY
CSGM	DFHVTAM	DFHGMM
CSIR	DFHISC	DFHCRR
CSJC	DFHJRNL	DFHJCBSP
CSLG	DFHRSPLG	DFHZRLG
CSMI	DFHISC	DFHMIR
CSMT	DFHMASTT	DFHMTPA
CSM1	DFHISC	DFHMIR
CSM2	DFHISC	DFHMIR
CSM3	DFHISC	DFHMIR
CSM5	DFHISC	DFHMIR
CSNC	DFHISC	DFHCRNP
CSNE	DFHVTAM	DFHZNAC
CSOT	DFHMASTT	DFHMTPA

Transaction	Group	Program (Continued)
CSPK	DFHVTAMP	DFHPRK
CSRK	DFHVTAMP	DFHRKB
CSRS	DFHRSEND	DFHZRSP
CSSF	DFHSIGN	DFHSNP
CSSN	DFHSIGN	DFHSNP
CSST	DFHMASTT	DFHMTPA
CSTA	DFHTIME	DFHTAJP
CSTE	DFHSTAND	DFHTACP
CSTT	DFHSTAND	DFHSTKC
8888	DFHNUM	DFHSNP
9999	DFHNUM	DFHSNP

Appendix E. Aids and Sample Application Programs

This appendix describes the CICS aids and the sample application programs that are provided on the distribution tape.

CICS AIDS

The following aids are provided:

DFHSDR$
BTAM error statistics recorder.

DFHTDWT$
A program to write transient data direct to a terminal.

DFHTDLI
A sample application that can be used to test a CICS-DL/I installation.

DFHXMOLS
A sample program to list the journal data set produced by the CICS monitoring facility.

The source code for all the aids can be found in the A. books of the same name in the source statement library. DFHSDR$, DFHTDWT$, DFHTDLI, and DFHXMOLS are in the core image library of the pregenerated system ready for use.

BTAM ERROR STATISTICS RECORDER (DFHSDR$)

This aid is provided for users who wish to log all errors and BTAM statistics to the VSE IJSYSRC file. While CICS is running, recording is done if the BTAM logic module was generated with RMSR support (RMSR=YES and ERLOGIC=E specified), and your TCT includes an RMSRTAB macro and also has ERROPT=E and LERBADR='the name on the RMSRTAB macro' specified on the DFHTCT TYPE=SDSCI macro instruction. During system termination the residual BTAM statistics are written by this program to the VSE IJSYSRC file, then a successful completion message is written to the computer operator's console. If RMSR support or the IJSYSRC file is not included or available, a message is printed on the computer operator's console stating that statistics were not logged.

It is invoked by specifying the name of this program in a program list table (PLT) used for non-immediate shutdown (CEMT PERFORM SHUTDOWN). This shutdown PLT is specified in the SIT or system initialization override statements by PLTSD='the table suffix'. Both this program and the shutdown PLT each require an entry in the PPT.

An example follows of a terminal control table supporting a local BTAM 3277 display and also exhibiting the functions needed for DFHSDR$:

```
* T IN COLUMN 70 FLAGS LINES RELATING TO RMSR SUPPORT AND BTAM TERMTST
         DFHTCT TYPE=INITIAL,                                          X
               ACCMETH=NONVTAM,                                        X
               MODNAME=BTML32T      TERMTST RMSR ERROPT=E           T
L77S     DFHTCT TYPE=SDSCI,         LOCAL 3277-2                       X
               CU=3272,                                                X
               DSCNAME=DFH002,                                         X
               ERROPT=E,                                            T  X
               LERBADR=RMSR1,       RMSR SUPPORT                    T  X
               LINELST=(011),                                          X
               SWITCH=NO,                                              X
               TERMTST=YES,         BTAM TERM TEST                  T  X
               DEVICE=L3277
```

```
         DFHTCT TYPE=LINE,                                                 X
                ACCMETH=BTAM,                                              X
                DSCNAME=DFH002,                                            X
                INAREAL=120,                                               X
                POOLADR=G002T002,                                          X
                TRMTYPE=L3277
G002T002 DFHTCT TYPE=TERMINAL,                                             X
                COMPAT=NO,                                                 X
                FEATURE=(DCKYBD,UCTRAN,SELCTPEN),                          X
                LASTTRM=POOL,                                              X
                LVUNIT=1,                                                  X
                PGESIZE=(24,80),                                           X
                PGESTAT=PAGE,                                              X
                TCTUAL=0,                                                  X
                TRMIDNT=L77S,                                              X
                TRMMODL=2,                                                 X
                TRMPRTY=0,                                                 X
                TRMSTAT=TRANSCEIVE,                                        X
                TRMTYPE=L3277
RMSR1    RMSRTAB 1                       RMSR MACRO                      T
* THE FOLLOWING IS THE EQUIVALENT GPENTRY FOR THIS TCT                    *
*        DFHTCT TYPE=GPENTRY,            LOCAL 3277                       *
*               GPTYPE=3270L,                                             *
*               LININL=120,                                               *
*               LINELST=011,             SYS011                           *
*               TRMMODL=2A,              1920 CHAR                        *
*               TRMFEAT=(DUS),           DCKYBD,UCTRAN,LIGHT PEN          *
*               TRMIDNT=L77S,                                             *
*               LERBADR=RMSR1,           RMSR SUPPORT                    T*
*               TERMTST=YES,             BTAM TERMINAL TEST              T*
*               TRMSTAT=A                TRANSCEIVE                       *
         DFHTCT TYPE=FINAL
```

CAUTION: Large users of remote BTAM terminals may find too much activity on the IJSYSRC file because even polling is logged when the counter reaches a threshold value. These users should investigate adopting a different approach.

TRANSIENT DATA WRITE TO TERMINAL (DFHTDWT$)

This aid prints messages on a local 3270 printer as they occur. In the destination control table, the user can specify that messages, such as those from the abnormal condition program (DFHACP) and the sign-on and sign-off messages, are to be sent to indirect destinations. These indirect destinations are, in turn, directed to an intrapartition destination specified as the terminal identification in the TCT for the local 3270 printer. The DCT entry for the intrapartition destination should specify the transaction identifier for this aid, with a trigger level of 1 if the messages are to be printed immediately. When a message arrives at the destination and the terminal is available, CICS uses automatic transaction initiation to start this aid program that sends the message to the terminal.

To use this aid, the CICS system must include automatic transaction initiation (ATI) and the transient data intrapartition data set.

Sample table entries for this technique are as follows:

```
         DFHDCT TYPE=INDIRECT,DESTID=CSML,INDDEST=LPRT
         DFHDCT TYPE=INDIRECT,DESTID=CSMT,INDDEST=LPRT
         DFHDCT TYPE=INDIRECT,DESTID=CSTL,INDDEST=LPRT
```

Note: The above are examples of some message destinations.

```
         DFHDCT TYPE=INTRA,                                                *
               DESTFAC=TERMINAL,                                           *
               DESTID=LPRT,        .. same name as TRMIDNT on TCT entry *
               TRANSID=TDWT,                                               *
               TRIGLEV=1
```

```
DFHPCT TYPE=ENTRY,TRANSID=TDWT,PROGRAM=DFHTDWT$

DFHPPT TYPE=ENTRY,PROGRAM=DFHTDWT$

DFHTCT TYPE=TERMINAL,                                        *
      TRMIDNT=LPRT,                                          *
      TRMTYPE=3270P,    .. a local non-SNA VTAM 3270 printer *
      .....
```

CICS-DL/I INSTALLATION TESTER (DFHTDLI)

This is a macro-level assembler-language program that can be used to test CICS-DL/I installations. Before DFHTDLI can be run, however, a DL/I application control table (ACT) must be generated including an entry for the test program. The program requires a data base to work on. For information on the creation of data bases, see the appropriate DL/I publications. Further requirements are listed at the end of this description.

The transaction identifier is TDLI, and the core image library phase name is DFHTDLI.

The program uses Basic Mapping Support to format a 1920-character 3270 screen, requesting the user to build a DL/I CALL on the screen. Any of the DL/I calls may be issued, using up to five qualified or unqualified segment search arguments. System calls and calls that include command codes are not supported. DL/I segments up to a length of 512 bytes only are supported by DFHTDLI. If larger segments are accessed, unpredictable results may occur.

To use the program, key in the transaction code and press ENTER. The first screen returned will request the entry of a PSB name. Any PSB name defined in the application control table specified for use by the program may be entered. If a PSB name is not specified (just the ENTER key pressed), the program will use the first default PSB defined in the ACT. If the PSB named cannot be found, an error message will be displayed. This message displays in hexadecimal notation the DL/I error codes returned in TCAFCTR and TCAFCTR+1. The _DL/I DOS/VS Application Programming Reference Manual_ should be used to determine the meaning of these codes.

ENTER Screen: This screen is the first screen returned. It allows a DL/I call to be built to the user's requirements.

Calls will be executed using the DBD name of the first PCB in the PSB unless the name of a DBD is specified on the screen. The program will search from 1 to 8 PCBs for the DBD name entered and, if found, will use that PCB for the call. Some syntax editing is carried out on the input data, and any errors found are indicated by a message displayed at the bottom of the screen asking the operator to correct the error and press the ENTER key.

If a function code is not entered, the program assumes that a GN call is required on the PCB last used. Pressing the CLEAR key terminates the program at any point.

GOUT (Good Output) Screen: If the call is successful, the segment requested is returned using this screen.

This screen displays the parameters used for the call, the PCB key feedback fully-concatenated key of the path, and the segment requested. If another call using the same PCB is required, for example GN or REPL, the call should be keyed into the function code and the ENTER key pressed. The program will execute that call. If a new call is required to be built, the ENTER screen is obtained by simply pressing the ENTER key.

ERMP (Error) Screen: If DL/I finds an error in the call, the program will display the parameters that were used in the call, the PCB information, and up to 256 bytes of the PCB key feedback field. Press the ENTER key to return to the ENTER screen, or the CLEAR key to end the program.

Further requirements for DFHTDLI include:

PCT Entry:

TRANSID=TDLI
PROGRAM=DFHTDLI
TWASIZE=1200 bytes
(long conversational)

PPT Entries:

PROGRAM=DFHTDLI	
PROGRAM=TDLIMS	BMS map set

No PPT entry is required for the PSB for DL/I; the PSB name is specified in the DL/I application control table (ACT) entry for the DFHTDLI program.

FCT Entries:

Each data base must be specified in the CICS file control table.

CAUTION:

1. This program will issue **any** call entered on the screen except system or command code calls.
2. If a PSB with a PROCOPT of other than G is used, an intent scheduling conflict in DL/I may be encountered, causing other tasks to wait for access to the data base being used. This program is a long conversation which keeps its PSB scheduled from the start until terminated. Refer to the DL/I manuals for more details on intent scheduling.
3. This program must **not** be used to read or write segments greater than 512 bytes long, or unpredictable results may occur.

The source code for DFHTDLI and the BMS map set TDLIMS can be found in the following source statement library books:

A.DFHTDLI	Program
A.TDLIMS	Symbolic description map (copy code)
A.DFHXTDLM	BMS map set source code for TDLIMS

CICS MONITORING FACILITY OUTPUT LISTING PROGRAM (DFHXMOLS)

The version of DFHXMOLS provided in the core image library is intended for batch execution with tape input. A version suitable for batch execution with disk input, and another version suitable for online execution as a CICS transaction, may be generated by assembling the source code with appropriate SYSPARM values. Details of this and how to run the monitoring sample program are provided in the CICS/VS Customization Guide, and also in the source code for DFHXMOLS, which may be printed down from the source statement library. The CICS/VS Performance Guide tells you how to interpret the output from CICS monitoring. DFHXMOLS reads, formats, and prints the contents of the monitoring data set; it does not perform any analysis of the data.

SAMPLE APPLICATION PROGRAMS

Sample applications written in command-level assembler language, COBOL, PL/I, and RPG II are provided. A macro-level assembler language program to test CICS-DL/I installations is also provided, and this is described under "CICS-DL/I Installation Tester (DFHTDLI)" on page 379.

The following sample application programs are provided:

- Four sets of sample command-level application programs, one set for each of the programming languages supported (assembler, COBOL, PL/I, and RPG II), that operate on the sample VSAM file FILEA. Both source code and object code are supplied. Copy books containing the table entries needed for running these programs are also provided.

- A sample application, consisting of two command-level PL/I programs, for sending a message to a remote CICS destination across an intersystem link. Source code only is supplied.

- Sample command-level applications programs, written in assembler language, illustrating distributed transaction processing (synchronous processing) between two CICS systems and between a CICS system and an IMS/VS system. Source code only is supplied.

- Two sets of sample command-level application programs, one set each in COBOL and PL/I, illustrating BMS partition support. Source code only is supplied.

FILEA SAMPLE APPLICATIONS

Four sets of sample applications using the command-level interface are provided, one set for each of the programming languages supported (assembler, COBOL, PL/I, and RPG II). The applications are for use with IBM 3270 display terminals. There are six programs in each set, and these programs cover eight functions as follows (file inquiry, add, and update being combined into one program):

menu (displays other functions)
file inquiry
file add
file update
file browse
order entry
order entry queue print
report

To use the PL/I sample programs, the DOS PL/I optimizing compiler transient library must be installed. The assembler language programs can be used by all users. The COBOL programs may need to be recompiled against the particular COBOL compiler installed. (FCOBOL was used to generate the sample programs provided in the distribution core image library.)

The CICS system must include VSAM file control support, full-function BMS, auxiliary temporary storage, and the transient data intrapartition data set.

All sets of sample application programs operate on a sample VSAM file, FILEA, which must first be created. Sample job control statements and the data needed to create FILEA are provided in a source statement library book, details of which are given in Appendix F. The file consists of records containing details of individual bank accounts, and specific account numbers have to be entered. These numbers include 100000, 111111, 200000, 222222, 300000, 333333, 400000, 444444, 500000, 555555, 600000, 666666, 700000, 777777, 800000, 888888, 900000, and 999999. The programs allow you to display, add, update, or browse through entries.

The logic of the assembler language, COBOL, and PL/I sample programs is given in the CICS/VS Application Programmer's Reference Manual (Command Level). The logic of the RPG II programs is given in the CICS/VS Application Programmer's Reference Manual (RPG II).

Both source code and object code for the sample application programs and their BMS map sets are supplied on the distribution volume. The naming convention used for the source statement library books is the following:

```
Programs -
  x.DFHXaaaa (x=A,C,P,R)        XDFHaaaa

Record descriptions (copy code) -
  x.FILEA (x=A,C,P,R)           FILEA
  x.LOGA                        LOGA
  x.L860                        L860

Symbolic description maps (copy code) -
  A.XDFHAMy (y=A,B,C,D,K,L)     XDFHAMy   Assembler
  C.XDFHCMy                     XDFHCMy   COBOL
  P.XDFHPMy                     XDFHPMy   PL/I
  R.XDRMy                       XDRMy     RPG II

BMS map set source code -
  A.DFHXAMy (y=A,B,C,D,K,L)     XDFHAMy   Assembler
  A.DFHXCMy                     XDFHCMy   COBOL
  A.DFHXPMy                     XDFHPMy   PL/I
  A.DFHXRMy                     XDRMy     RPG II
```

Note: Source code for the sample programs, and copy code for the record descriptions and symbolic description maps used, are held as books appropriate to their language, namely A. books for assembler, C. books for COBOL, P. books for PL/I, and R. books for RPG II. Source code for all the BMS map sets used is held as A. books.

Before bringing up CICS to run the sample applications, you must ensure that suitable tables are provided and either named in the SIT or specified as system initialization overrides. The following copy books are provided for the table entries needed to run the FILEA sample applications (see Appendix D for details):

```
DFHXDCTS
DFHXFCTS
DFHXPCTy (y=A,C,P,R)
DFHXPPTy (y=A,C,P,R)
DFHXTCBS
DFHXTCVS
```

It is not necessary to sign on to CICS (using CSSN) before running the sample applications, because all the sample transactions have a security code of 1 (the default).

When CICS is running, any of the transaction identifiers listed below for the sample applications may be entered.

When the sample applications are finished with, they can be deleted from the core image library by using the MAINT utility, and requesting it to DELETC XDFH.ALL. (All the pregenerated sample application programs start with the 4-character group XDFH.)

Operator Instruction Sample Program

The instruction program displays a map containing operator instructions. This map lists some of the sample application functions provided, and the transaction identifiers that can be used to invoke them for the programming language being used. To initiate the file browse, inquiry, add, or update programs, the appropriate transaction identifier must be entered in the menu map.

The program names, maps, and transaction identifiers are:

	Program	Map	Trans. ID
Assembler	XDFHAMNU	XDFHAMA	AMNU
COBOL	XDFHINST	XDFHCMA	MENU
PL/I	XDFHPMNU	XDFHPMA	PMNU
RPG II	XDFHRMNU	XDRMA	RMNU

Update Sample Program

The update program includes file update, file add, and file inquiry operations.

The program names, maps, and transaction identifiers are:

	Program	Maps	Trans. IDs
Assembler	XDFHAALL	XDFHAMA,XDFHAMB	AINQ,AADD,AUPD
COBOL	XDFHCALL	XDFHCMA,XDFHCMB	INQY,ADDS,UPDT
PL/I	XDFHPALL	XDFHPMA,XDFHPMB	PINQ,PADD,PUPD
RPG II	XDFHRALL	XDRMA,XDRMB	RINQ,RADD,RUPD

Browse Sample Program

The browse program sequentially retrieves a page or set of records for display, starting at a point specified by the terminal operator, and continuing either forwards or backwards through the file.

The program names, maps, and transaction identifiers are:

	Program	Maps	Trans. ID
Assembler	XDFHABRW	XDFHAMA,XDFHAMC	ABRW
COBOL	XDFHBRWS	XDFHCMA,XDFHCMC	BRWS
PL/I	XDFHPBRW	XDFHPMA,XDFHPMC	PBRW
RPG II	XDFHRBRW	XDRMA,XDRMC	RBRW

Order Entry Sample Program

The order entry sample application program accepts input simulating the ordering of parts. The data entered will be rejected if the number entered in the "number" field does not match the number of one of the records in the sample file. The entries are sent to an intrapartition transient data queue called L860. When 30 order entries have accumulated, the list is automatically sent to another location to be processed by the Order Entry Queue Print sample program. (The trigger level for queue L860 can be changed dynamically via the CEMT SET QUEUE TRIGGER command.)

The program names, maps, and transaction identifiers are:

	Program	Map	Trans. ID
Assembler	XDFHAREN	XDFHAMK	AORD
COBOL	XDFHOREN	XDFHCMK	OREN
PL/I	XDFHPORD	XDFHPMK	PORD
RPG II	XDFHRREN	XDRMK	RORD

Order Entry Queue Print Sample Program

This program reads the order entry queue when the number of entries in the queue reaches 30 - the trigger level defined in the DCT entry for L860, and prints the queue items on a 3270 printer.

If the transaction identifier is entered by the terminal operator, the program will read the order entry queue at hourly intervals. If there are no items on the queue, a message is printed indicating that the queue is empty.

The program names, maps, and transaction identifiers are:

	Program	Map	Trans. ID
Assembler	XDFHACOM	XDFHAML	AORQ
COBOL	XDFHCCOM	XDFHCML	OREQ
PL/I	XDFHPCOM	XDFHPML	PORQ
RPG II	XDFHRCOM	XDRML	RORQ

Note: If the COBOL, PL/I, or RPG II version of this program is to be used, the operand TRANSID=AORQ specified for destination L860 in copy book DFHXDCTS must be changed to indicate 'OREQ', 'PORQ', or 'RORQ' respectively.

Report Sample Program

The report sample program produces a list of all entries in the sample file that are less than or equal to a specified amount.

This program illustrates page-building techniques and use of the terminal-paging facilities of full-function BMS. To use the program, suitable character-string values for the terminal-paging commands must be defined either in the system initialization table (SIT) or as overrides at CICS system initialization. For example:

```
PGCHAIN=X/
PGCOPY=C/
PGPURGE=T/
PGRET=P/
```

The program names, maps, and transaction identifiers are:

	Program	Map	Trans. ID
Assembler	XDFHAREP	XDFHAMD	AREP
COBOL	XDFHREPT	XDFHCMD	REPT
PL/I	XDFHPREP	XDFHPMD	PREP
RPG II	XDFHRREP	XDRMD	RREP

INTERSYSTEM COMMUNICATION MESSAGE ROUTING SAMPLE APPLICATION

This sample application consists of two command-level PL/I programs, for which the source code only is supplied on the distribution tape - as books P.DFHXPXP and P.DFHXPRP in the source statement library. To use these programs, they must first be translated, compiled, and link-edited, and the DOS PL/I transient library must have been installed. Recommended core image library phase names are XDFHPXP and XDFHPRP respectively, matching the PCT and PPT entries given here.

The ISC message-routing sample application enables an operator on one CICS system to send a message to a specified destination on another CICS system, across an intersystem link. Messages can be sent to any CICS system in the network, and to any transient data destination defined in the remote system. The initiating transaction identifier is XPXP.

The application uses an unformatted screen, and is coded for a maximum message length of 160 characters. Only IBM 3270 display terminals are supported.

The following table entries are required:

PCT Entries:

```
DFHPCT TYPE=ENTRY,TRANSID=XPXP,PROGRAM=XDFHPXP,                        *
      SPURGE=YES,TPURGE=YES
DFHPCT TYPE=ENTRY,TRANSID=XPRP,PROGRAM=XDFHPRP,                        *
      SPURGE=YES,TPURGE=YES
```

PPT Entries:

```
DFHPPT TYPE=ENTRY,PROGRAM=XDFHPXP,PGMLANG=PL/I
DFHPPT TYPE=ENTRY,PROGRAM=XDFHPRP,PGMLANG=PL/I
```

DCT Entry (Optional):

```
DFHDCT TYPE=INTRA,DESTFAC=TERMINAL,DESTID=xxxx,                        *
      TRANSID=XPRP,TRIGLEV=1
```

Note: These table entries are not included in the sample tables.

If the transient data destination for the message is a terminal, the owning CICS system can use ATI to display the message at the destination terminal by having a DCT entry coded as above.

The operator can specify DESTID and SYSIDNT, in which case no DCT entry for the remote destination is required. (This could be used to check the network nodes (SYSIDNTs) from a terminal.) Alternatively, the operator may specify only the DESTID, in which case CICS will use the definitions of remote DESTIDs in the DCT to route the message to the correct system. DESTID and SYSIDNT are assumed by the application to be four characters long, so you should pad with blanks as necessary.

DISTRIBUTED TRANSACTION PROCESSING SAMPLE APPLICATIONS

Sample applications are provided to illustrate distributed transaction processing (synchronous processing) between two CICS systems and between a CICS system and an IMS/VS system.

There are five sample applications:

1. A CICS to CICS sample showing a remote browsing operation

2. A CICS to CICS or IMS/VS sample illustrating a simple conversation

3. A CICS to CICS sample illustrating the transfer of a temporary storage queue from one system to another

4. A CICS to IMS/VS conversational sample

5. A CICS to IMS/VS sample illustrating demand paging

The CICS-to-CICS samples are suitable for both intersystem communication and multiregion operation.

Full descriptions of each of these samples are given in the CICS/VS Intercommunication Facilities Guide.

The sample programs are written in assembler language using the command-level interface, and must be translated, assembled, and link-edited before they can be run. All the programs use BMS map sets, which must first be prepared.

Source code for the programs and their map sets is supplied in the following source statement library books (the numbers in parentheses identify the sample applications):

```
Programs -
  A.DFHXAIBL, A.DFHXAIBR  (1)
  A.DFHXAI1A              (2)
  A.DFHXAI2A              (3)
  A.DFHXAI4A              (4)
  A.DFHXAI4B              (5)

Map sets -
  A.DFHXAMC, A.DFHXAIB    Used by (1)
  A.DFHXAI1                       (2)
  A.DFHXAI2                       (3)
  A.DFHXAI4                       (4) and (5)
```

Note: DFHXAMC is also used by the FILEA browse sample program (transaction identifier ABRW); the symbolic description map and the physical map are both supplied pregenerated.

The following entries are required in the using system's PCT and PPT to enable the samples to be run:

PCT Entries:

```
DFHPCT TYPE=ENTRY,TRANSID=AIBL,PROGRAM=XDFHAIBL,                       *
      SPURGE=YES,TPURGE=YES,INBFMH=ALL
DFHPCT TYPE=ENTRY,TRANSID=AIBR,PROGRAM=XDFHAIBR,                       *
      SPURGE=YES,TPURGE=YES
DFHPCT TYPE=ENTRY,TRANSID=AICC,PROGRAM=XDFHAI1A,                       *
      SPURGE=YES,TPURGE=YES
DFHPCT TYPE=ENTRY,TRANSID=AISC,PROGRAM=XDFHAI2A,                       *
      SPURGE=YES,TPURGE=YES
DFHPCT TYPE=ENTRY,TRANSID=AISR,PROGRAM=XDFHAI2A,                       *
      SPURGE=YES,TPURGE=YES,INBFMH=ALL
DFHPCT TYPE=ENTRY,TRANSID=AICO,PROGRAM=XDFHAI4A,                       *
      SPURGE=YES,TPURGE=YES,INBFMH=ALL
DFHPCT TYPE=ENTRY,TRANSID=AICD,PROGRAM=XDFHAI4B,                       *
      SPURGE=YES,TPURGE=YES,INBFMH=ALL
```

PPT Entries:

```
DFHPPT TYPE=ENTRY,PROGRAM=XDFHAIBL
DFHPPT TYPE=ENTRY,PROGRAM=XDFHAIBR
DFHPPT TYPE=ENTRY,PROGRAM=XDFHAI1A
DFHPPT TYPE=ENTRY,PROGRAM=XDFHAI2A
DFHPPT TYPE=ENTRY,PROGRAM=XDFHAI4A
DFHPPT TYPE=ENTRY,PROGRAM=XDFHAI4B
DFHPPT TYPE=ENTRY,MAPSET=XDFHAMC
DFHPPT TYPE=ENTRY,MAPSET=XDFHAIB
DFHPPT TYPE=ENTRY,MAPSET=XDFHAI1
DFHPPT TYPE=ENTRY,MAPSET=XDFHAI2
DFHPPT TYPE=ENTRY,MAPSET=XDFHAI4
```

Note: These table entries are not included in the sample tables.

BMS PARTITION SUPPORT SAMPLE APPLICATIONS

Sample applications are provided to illustrate two possible uses for BMS partition support:

1. Keystroke Overlap - Two copies of a data entry map are displayed. Having entered a completed map in one partition, the operator may switch to another partition and start entering data there while the data from the first partition is being processed.

2. Look-Aside Query - Two partitions are used, one for data entry and one for look-aside queries. The operator may enter data into either partition at any time.

Two sets of these sample programs are provided, one set written in COBOL, the other in PL/I. The programs must be translated, compiled, and link-edited before they can be run. All the programs use a BMS map set and a partition set, which must first be prepared. To use the PL/I programs, the DOS PL/I transient library must have been installed.

Further details of the samples are provided in the source code, which may be printed down from the source statement library.

The source code for the programs, map set, and partition set is contained in the following books:

```
Programs -
  C.DFHXCDE, P.DFHXPDE   Keystroke Overlap (COBOL, PL/I)
  C.DFHXCLA, P.DFHXPLA   Look-Aside Query (COBOL, PL/I)

Map set -
  A.DFHXAPM       Used by all programs

Partition set -
  A.DFHXAPP       Used by all programs
```

Note: The map set source code A.DFHXAPM must be edited to indicate the desired language, by specifying LANG=COBOL or LANG=PLI as appropriate on the DFHMSD TYPE=DSECT macro.

The following entries are required in the using system's PCT and PPT to enable the samples to be run:

PCT Entries:

```
DFHPCT TYPE=ENTRY,TRANSID=XCDE,PROGRAM=DFHXCDE,                  *
      SPURGE=YES,TPURGE=YES
DFHPCT TYPE=ENTRY,TRANSID=XCLA,PROGRAM=DFHXCLA,                  *
      SPURGE=YES,TPURGE=YES
DFHPCT TYPE=ENTRY,TRANSID=XPDE,PROGRAM=DFHXPDE,                  *
      SPURGE=YES,TPURGE=YES
DFHPCT TYPE=ENTRY,TRANSID=XPLA,PROGRAM=DFHXPLA,                  *
      SPURGE=YES,TPURGE=YES
```

PPT Entries:

```
DFHPPT TYPE=ENTRY,PROGRAM=DFHXCDE
DFHPPT TYPE=ENTRY,PROGRAM=DFHXCLA
DFHPPT TYPE=ENTRY,PROGRAM=DFHXPDE
DFHPPT TYPE=ENTRY,PROGRAM=DFHXPLA
DFHPPT TYPE=ENTRY,MAPSET=DFHXAPM
DFHPPT TYPE=ENTRY,PARTSET=XAPP
```

Notes:

1. These table entries are not included in the sample tables.

2. The partitioned display terminal to be used must be defined in the CICS terminal control table.

Appendix F. Sample Job Streams

SAMPLE PROCEDURES

To make it easier for the system programmer to install command-level application programs, the source code of sample procedures is supplied in the following books in the CICS source statement library:

Z.DFHEITAL for assembler language Z.DFHEITCL for COBOL
Z.DFHEITPL for PL/I Z.DFHEITRL for RPG II.

"Chapter 3.4. Installing Application Programs" tells you how to use these procedures, each of which consists of three steps:

- Translate
- Assemble/compile
- Link-edit.

Note: These procedures are valid only if the output from the translation step is written to a disk other than that on which the procedure library resides.

Before putting the sample procedures into the procedure library, you must modify:

1. The extent information
2. The volume identifications.

DFHEITAL — TRANSLATE, ASSEMBLE, AND LINK-EDIT ASSEMBLER-LANGUAGE APPLICATION PROGRAMS

```
// DLBL     IJSYSPH,'ASM.TRANSLATION',99/365
// EXTENT   SYSPCH,,1,0,399,19
ASSGN   SYSPCH,DISK,VOL=DB0007,SHR
// EXEC     DFHEAP1$
CLOSE       SYSPCH,PUNCH
// DLBL     IJSYSIN,'ASM.TRANSLATION',99/365
// EXTENT   SYSIPT
ASSGN   SYSIPT,DISK,VOL=DB0007,SHR
   INCLUDE DFHEAI
// EXEC     ASSEMBLY
CLOSE       SYSIPT,READER
// EXEC     LNKEDT
```

Note: Disk is used as intermediate storage for translator output.

DFHEITCL — TRANSLATE, COMPILE, AND LINK-EDIT COBOL APPLICATION PROGRAMS

```
// DLBL     IJSYSPH,'COBOL.TRANSLATION',99/365
// EXTENT   SYSPCH,,1,0,399,19
ASSGN   SYSPCH,DISK,VOL=DB0007,SHR
// EXEC     DFHECP1$
CLOSE       SYSPCH,PUNCH
// DLBL     IJSYSIN,'COBOL.TRANSLATION',99/365
// EXTENT   SYSIPT
ASSGN   SYSIPT,DISK,VOL=DB0007,SHR
   INCLUDE DFHECI
// EXEC     FCOBOL
CLOSE       SYSIPT,READER
// EXEC     LNKEDT
```

Note: Disk is used as intermediate storage for translator output.

DFHEITPL — TRANSLATE, COMPILE, AND LINK-EDIT PL/I APPLICATION PROGRAMS

```
// DLBL     IJSYSPH,'PL/I.TRANSLATION',99/365
// EXTENT   SYSPCH,,1,0,399,19
ASSGN  SYSPCH,DISK,VOL=DB0007,SHR
// EXEC     DFHEPP1$
CLOSE       SYSPCH,PUNCH
// DLBL     IJSYSIN,'PL/I.TRANSLATION',99/365
// EXTENT   SYSIPT
ASSGN  SYSIPT,DISK,VOL=DB0007,SHR
   INCLUDE DFHPL1I
// EXEC     PLIOPT
CLOSE       SYSIPT,READER
// EXEC     LNKEDT
```

Note: Disk is used as intermediate storage for translator output.

DFHEITRL — TRANSLATE, COMPILE, AND LINK-EDIT RPG II APPLICATION PROGRAMS

```
// DLBL     IJSYSPH,'RPG.II.TRANSLATION',99/365
// EXTENT   SYSPCH,,1,0,399,19
ASSGN  SYSPCH,DISK,VOL=DB0007,SHR
// EXEC     DFHERP1$
CLOSE       SYSPCH,PUNCH
// DLBL     IJSYSIN,'RPG.II.TRANSLATION',99/365
// EXTENT   SYSIPT
ASSGN  SYSIPT,DISK,VOL=DB0007,SHR
   INCLUDE DFHERI
// EXEC     RPGII
CLOSE       SYSIPT,READER
// EXEC     LNKEDT
```

Note: The output from the translator is directed to SYSPCH (on disk).

GENERAL-PURPOSE JOB STREAM

The following listing is of the source code of a general-purpose set of jobs for creating VSAM file FILEA, bringing up CICS, and so on. The fields included in lowercase should be replaced with your values before each job is run. The source code is contained in Z.DFHSPJCL.

DEFINE VSAM MASTER CATALOG

```
// JOB CREATE VSAM MASTER CATALOG AND SPACE
// DLBL IJSYSCT,'vsam.master.catalog',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEFINE MASTERCATALOG                                     -
          (NAME(vsam.master.catalog)                      -
           VOLUME(volume)                                 -
           ORIGIN(origin)                                 -
           RECORDS(records))
 LISTCAT ALL
/*
/&
```

DEFINE VSAM USER CATALOG

```
// JOB DEFUCAT DEFINE VSAM USER CATALOG
*
* if you are using CKD dasd then you must replace the
* blocks parameter with TRACKS(tracks)
*
// EXEC IDCAMS,SIZE=AUTO
 DEFINE USERCATALOG                                       -
          (NAME(user.catalog)                             -
           ORIGIN(origin)                                 -
           VOLUME(volume)                                 -
           BLOCKS(blocks))                                -
       CATALOG(vsam.master.catalog)
/*
/&
```

DEFINE VSAM SPACE

```
// JOB DEFSPACE DEFINE VSAM SPACE
// EXEC IDCAMS,SIZE=AUTO
 DEFINE SPACE                                             -
          (VOLUME(volume)                                 -
           ORIGIN(origin)                                 -
           BLOCKS(blocks))                                -
       CATALOG(user.catalog)
/*
/&
```

DEFINE AUXILIARY TEMPORARY STORAGE DATA SET

```
// JOB CRATS CREATE AUXILIARY TEMPORARY STORAGE DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                           -
       (NAME(cics160.temp.storage)                        -
        RECORDSIZE(4089,4089)                             -
        RECORDS(144)                                      -
        NONINDEXED                                        -
        CONTROLINTERVALSIZE(4096)                         -
        SHAREOPTIONS(2)                                   -
        VOLUMES(volume))                                  -
       DATA                                               -
          (NAME(cics160.temp.storage.data))               -
       CATALOG(user.catalog)
/*
/&
```

DEFINE TRANSIENT DATA INTRAPARTITION DATA SET

```
// JOB CRITD CREATE INTRAPARTITION TRANSIENT DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                                -
           (NAME(cics160.intra.transd)                         -
            RECORDSIZE(1529,1529)                              -
            RECORDS(100)                                       -
            NONINDEXED                                         -
            CONTROLINTERVALSIZE(1536)                          -
            VOLUMES(volume))                                   -
        DATA                                                   -
          (NAME(cics160.intra.transd.data))                    -
        CATALOG(user.catalog)
/*
/&
```

JOB TO START UP CICS WITH DATA SETS

```
// JOB CICS BRING UP CICS SYSTEM WITH DATA SETS
* USER VSAM CATALOG
// DLBL user-catalog-filename,'user.catalog',,VSAM
* AUXILIARY TEMPORARY STORAGE
// DLBL DFHTEMP,'cics160.temp.storage',,VSAM,CAT=user-catalog-filename
* INTRAPARTITION TRANSIENT DATA
// DLBL DFHNTRA,'cics160.intra.transd',,VSAM,CAT=user-catalog-filename
* DUMP DATA SETS
// DLBL DFHDMPA,'cics160.dump.dataset.a',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// DLBL DFHDMPB,'cics160.dump.dataset.b',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
* AUXILIARY TRACE
// DLBL DFHAUXT,'cics160.aux.trace.a',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// DLBL DFHBUXT,'cics160.aux.trace.b',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
* ASSGN OF DASD CONTAINING CICS AND USER FILES
// ASSGN SYSnnn,DISK,TEMP,VOL=volume,SHR
* SEARCH ORDER FOR CORE IMAGE LIBRARIES
// LIBDEF CL,SEARCH=cics-cil-filename
* START UP CICS WITH OVERRIDE PARAMETERS FROM SYSIPT
// EXEC DFHSIP,SIZE=1200K,PARM='SIT=1$,SI'
 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 $END
/*
/&
```

PRINT DUMP DATA SET

```
// JOB PRTDUMP PRINT DUMP DATA SET
// DLBL DTFDISK,'cics160.dump.dataset.a',,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// EXEC DFHDUP,SIZE=80K,PARM='DEVICE=DISK,translate=fold,single'
/*
/&
```

PRINT AUXILIARY TRACE DATA SET

```
// JOB PRTAUX PRINT AUXILIARY TRACE DATA SET
// DLBL DFHAUXT,'cics160.aux.trace.a',,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// EXEC DFHTUP,SIZE=80K
 DEVICE=DISK
/*
/&
```

DEFINE VSAM FILEA CLUSTER

```
// JOB DEFILEA CREATE VSAM FILEA CLUSTER
// EXEC IDCAMS,SIZE=AUTO
 DEFINE  CLUSTER                                    -
           (NAME(sample.test.filea)                 -
            INDEXED                                 -
            BUFFERSPACE(4096)                       -
            RECORDSIZE(80 80)                       -
            RECORDS(500 200)                        -
            KEYS(6 1)                               -
            SHAREOPTIONS(2)                         -
            VOLUMES(VOLUME))                        -
         DATA                                       -
           (NAME(sample.test.filea.data))           -
         INDEX                                      -
           (NAME(sample.test.filea.index))          -
         CATALOG(user.catalog)
 LISTCAT ENTRIES(sample.test.filea) ALL             -
         CATALOG(user.catalog)
/*
/&
```

INITIALIZE SAMPLE VSAM FILEA FILE

```
  // JOB INTFILEA INITIALISE SAMPLE VSAM FILE FILEA
  // DLBL user-catalog-filename,'user.catalog',,VSAM
  // DLBL FILEA,'sample.test.filea',,VSAM,CAT=user-catalog-filename
  // EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE                                             -
            (SYSIPT                                         -
             ENVIRONMENT                                    -
              (RECORDFORMAT(FIXUNB)                         -
               BLOCKSIZE(80)                                -
               RECORDSIZE(80)))                             -
         OUTFILE (FILEA)
   000100W. DAVIS              SURREY, ENGLAND     3215677826 11 81$0100.11YES
   000102F. ALDSON             WARWICK, ENGLAND    9835618326 11 81$1111.11YES
   000104S. BOWLER             LONDON,ENGLAND      1284629326 11 81$0999.99YES
   000106B. ADAMS              CROYDON, ENGLAND    1948567326 11 81$0087.71YES
   000111GENE BARLOWE          SARATOGA,CALIFORNIA 4612075301 02 74$0111.11YES
   000762GEORGE BURROW         SAN JOSE,CALIFORNIA 2231212101 06 74$0000.00YES
   000983H. L. L. CALL         WASHINGTON, DC      3451212021 04 75$9999.99YES
   001222J.R.REYNOLDS          BOBLINGEN, GERMANY  7031555110 04 73$3349.99YES
   001781HAROLD JAMES          SINDELFINGEN,GERMANY7031999021 06 77$0009.99YES
   003210M. P. CREPIN          PARIS, FRANCE       1234567010 03 75$3349.99YES
   003214HUBERT C HERBERT      SUNNYVALE, CAL.     3411212000 06 73$0009.99NO
   003890PHILIPPE SMITH, JR    NICE, FRANCE        0000000028 05 74$0009.99NO
   004004STAN SMITH            DUBLIN, IRELAND     7111212102 11 73$1259.99NO
   004445DANIEL O'GALWAY       SOUTH BEND, S.DAK.  6121212010 10 73$0009.99NO
   004878D.C. CURRENT          SUNNYVALE, CALIF.   3221212010 06 73$5399.99NO
   005005J. S. LAVERENCE       SAN FRANCISCO, CA.  0000000101 08 73$0009.99NO
   005444JEAN LAWRENCE         SARATOGA, CALIF.    6771212020 10 74$0809.99NO
   005581JOHN ALDEN III        BOSTON, MASS.       4131212011 04 74$0259.99NO
   006016DR W. T. KAR          NEW DELHI, INDIA    7033121121 05 74$0009.88YES
   006670WILLIAM KAPP          NEW YORK, N.Y.      2121212031 01 75$3509.88NO
   006968D. CONRAD             WARWICK, ENGLAND    5671382126 11 81$0009.88YES
   007007BRIGITTE EICRN        STUTTGART, GERMANY  7031100010 10 75$5009.88NO
   007248B. C. WILLIAMSON      REDWOOD CITY, CALF. 3331212111 10 75$0009.88NO
   007779MRS. W. WELCH         SAN JOSE, CALIF.    4151212003 01 75$0009.88YES
   100000G. NEADS              TORONTO, ONTARIO    0341512126 11 81$0010.00YES
   111111C. MEARS              OTTAWA, ONTARIO     5121200326 11 81$0011.00YES
   200000A. BONFIELD           GLASGOW,  SCOTLAND  6373829026 11 81$0020.00YES
   222222J. WIEBERS            FRANKFURT, GERMANY  2003415126 11 81$0022.00YES
   300000K. TRENCHARD          NEW YORK, U.S.      6473980126 11 81$0030.00YES
   333333D. MYRING             CARDIFF, WALES      7849302026 11 81$0033.00YES
   400000W. TANNER             MILAN, ITALY        2536373826 11 81$0040.00YES
   444444A. FISHER             CALGARY, ALBERTA    7788982026 11 81$0044.00YES
   500000J. DENFORD            MADRID, SPAIN       4445464026 11 81$0000.00YES
   555555C. JARDINE            KINGSTON, N.Y.      3994442026 11 81$0005.00YES
   600000F. HUGHES             DUBLIN, IRELAND     1239878026 11 81$0010.00YES
   666666A. BROOKMAN           LA HULPE, BRUSSELS  4298384026 11 81$0016.00YES
```

```
  700000A. MACALLA          DALLAS, TEXAS        5798432026 11 81$0002.00YES
  777777D. PRYKE            WILLIAMSBURG, VIRG.  9187613126 11 81$0027.00YES
  800000H. BRISTOW          WESTEND, LONDON      2423338926 11 81$0030.00YES
  888888B. HOWARD           NORTHAMPTON, ENG.    2369163926 11 81$0038.00YES
  900000D. WOODSON          TAMPA, FLA.          3566812026 11 81$0040.00YES
  999999R. JACKSON          RALEIGH, N.Y.        8459163926 11 81$0049.00YES
 /*
 /&
```

CREATE NEW PRIVATE CORE IMAGE LIBRARY

```
// JOB CREATE NEW PRIVATE CORE IMAGE LIBRARY
* USER PRIVATE CORE IMAGE LIBRARY
// DLBL user-cil-filename,'user.private.cil',99/365
// EXTENT ,pcil-volume,1,0,relative-track-number,number-of-tracks
// LIBDEF CL,NEW=user-cil-filename
// EXEC CORGZ
   NEWVOL CL=number-of-cylinders(directory-tracks)
/*
/&
```

ASSEMBLE AND LINK-EDIT SAMPLE DFHPPTS1

```
// JOB CICSTAB ASSEMBLE AND LINK-EDIT SAMPLE DFHPPTS1
* USER PRIVATE CORE IMAGE LIBRARY
// DLBL user-cil-filename,'user.private.cil'
// EXTENT ,pcil-volume
// OPTION CATAL,NODECK,ALIGN
// LIBDEF CL,TO=user-cil-filename
// LIBDEF RL,SEARCH=cics-rlb-filename
// LIBDEF SL,SEARCH=cics-slb-filename
// EXEC ASSEMBLY
       DFHPPT  TYPE=INITIAL,                                       *
               SUFFIX=S1
       COPY    DFHXPPT
       COPY    DFHXPPTA
       DFHPPT  TYPE=FINAL
   END
/*
// EXEC LNKEDT
/&
```

START UP CICS SYSTEM FOR SAMPLE RUN

```
  // JOB CICS START UP CICS SYSTEM FOR SAMPLE RUN
  * USER PRIVATE CORE IMAGE LIBRARY
  // DLBL user-cil-filename,'user.private.cil'
  // EXTENT ,pcil-volume
  * USER VSAM CATALOG
  // DLBL user-catalog-filename,'user.catalog',,VSAM
  * AUXILIARY TEMPORARY STORAGE
  // DLBL DFHTEMP,'cics160.temp.storage',,VSAM,CAT=user-catalog-filename
  * INTRAPARTITION TRANSIENT DATA
  // DLBL DFHNTRA,'cics160.intra.transd',,VSAM,CAT=user-catalog-filename
  * FILEA SAMPLE DATA SET
  // DLBL FILEA,'sample.test.filea',,VSAM,CAT=user-catalog-filename
  * DUMP DATA SETS
  // DLBL DFHDMPA,'cics160.dump.dataset.a',0,SD
  // EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
  // DLBL DFHDMPB,'cics160.dump.dataset.b',0,SD
  // EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
  * AUXILIARY TRACE
  // DLBL DFHAUXT,'cics160.aux.trace.a',0,SD
  // EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
  // DLBL DFHBUXT,'cics160.aux.trace.b',0,SD
  // EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
  * ASSGN OF DASD CONTAINING CICS AND USER FILES
  // ASSGN SYSnnn,DISK,TEMP,VOL=volume,SHR
  * SEARCH ORDER FOR CORE IMAGE LIBRARIES
  // LIBDEF CL,SEARCH=(user-cil-filename,cics-cil-filename,etc)
```

```
* LOCAL TERMINALS DEFINED IN SAMPLE BTAM TCT
ASSGN SYS020,cuu
ASSGN SYS021,cuu
ASSGN SYS027,cuu
* START UP CICS WITH OVERRIDE PARAMETERS FROM SYSIPT
// EXEC DFHSIP,SIZE=1200K,PARM='SIT=1$,SI'
 TSP=2$,TDP=6$,DCP=D$,TRP=D$
 PPT=S1,PCT=S1,DCT=S1,FCT=S1,FCP=3$,TCT=S1
 BMS=S$,PGRET=P/,PGPURGE=T/,PGCOPY=C/,PGCHAIN=X/
 $END
/*
/&
```

DEFINE RESTART DATA SET

```
// JOB DEFRSD CREATE RESTART DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                               -
         (NAME(cics160.rsd)                                   -
          INDEXED                                             -
          RECORDSIZE(2000 2000)                               -
          RECORDS(250 100)                                    -
          KEYS(10 0)                                          -
          FREESPACE(20 20)                                    -
          SHAREOPTIONS(2)                                     -
          VOLUMES(volume))                                    -
       DATA                                                   -
         (NAME(cics160.rsd.data))                             -
       INDEX                                                  -
         (NAME(cics160.rsd.index))                            -
       CATALOG(user.catalog)
/*
/&
```

INITIALIZE RESTART DATA SET

```
// JOB INITRSD INITIALISE RESTART DATA SET
// DLBL user-catalog-filename,'user.catalog',,VSAM
// DLBL DFHRSD,'cics160.rsd',,VSAM,CAT=user-catalog-filename
// EXEC IDCAMS,SIZE=AUTO
 REPRO  INFILE                                                -
          (SYSIPT                                             -
           ENVIRONMENT                                        -
            (RECORDFORMAT(FIXUNB)                             -
             BLOCKSIZE(80)                                    -
             RECORDSIZE(80)))                                 -
        OUTFILE(DFHRSD)
ACTL 0001
/*
/&
```

FORMAT JOURNAL DISK EXTENT

```
// JOB CICSFDSK FORMAT JOURNAL DISK EXTENT
// DLBL JOURNAL,'cics160.system.log.a',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// EXEC DFHJCJFP,SIZE=AUTO
/*
/&
```

FORMAT SECOND JOURNAL DISK EXTENT

```
// JOB CICSFDSK FORMAT SECOND JOURNAL DISK EXTENT
// DLBL JOURNAL,'cics160.system.log.b',0,SD
// EXTENT SYSnnn,,1,0,relative-track-number,number-of-tracks
// ASSGN SYSnnn,cuu
// EXEC DFHJCJFP,SIZE=AUTO
/*
/&
```

FORMAT JOURNAL TAPE

```
// JOB CICSFTAP FORMAT JOURNAL TAPE
// ASSGN SYS010,cuu
// EXEC DFHFTAP
/*
/&
```

VERIFY VSAM FILES

```
// JOB VERIFY VSAM FILES
// DLBL IJSYSUC,'user.catalog',,VSAM
// EXEC IDCAMS,SIZE=AUTO
       VERIFY DATASET(cics160.temp.storage)
       VERIFY DATASET(cics160.intra.transd)
       VERIFY DATASET(cics160.rsd)
       VERIFY DATASET(cics160.csd)
       VERIFY DATASET(sample.test.filea)
/*
/&
```

DEFINE CICS SYSTEM DEFINITION FILE

```
// JOB DEFCSD CREATE CICS SYSTEM DEFINITION FILE
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                                                  -
          (NAME(cics160.csd)                                     -
           INDEXED                                               -
           RECORDSIZE(100 200)                                   -
           RECORDS(4000 1000)                                    -
           KEYS(22 0)                                            -
           FREESPACE(0 0)                                        -
           SHAREOPTIONS(4)                                       -
           VOLUMES(volume))                                      -
        DATA                                                     -
          (NAME(cics160.csd.data))                               -
        INDEX                                                    -
          (NAME(cics160.csd.index))                              -
        CATALOG(user.catalog)
/*
/&
```

INITIALIZE CICS SYSTEM DEFINITION FILE

```
// JOB INITCSD INITIALISE THE CSD
// DLBL user-catalog-filename,'user.catalog',,VSAM
// DLBL DFHCSD,'cics160.csd',,VSAM,CAT=user-catalog-filename
// EXEC DFHCSDUP,SIZE=AUTO
 INITIALISE
 LIST ALL
/*
/&
```

DELETE CICS SYSTEM DEFINITION FILE

```
// JOB DELETE PURGE CICS SYSTEM DEFINITION FILE
// EXEC IDCAMS,SIZE=AUTO
 DELETE cics160.csd                              -
        PURGE                                    -
        CLUSTER                                  -
        CATALOG(user.catalog)
/*
/&
```

DELETE RESTART DATA SET

```
// JOB DELETE PURGE RESTART DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DELETE cics160.rsd                              -
        PURGE                                    -
        CLUSTER                                  -
        CATALOG(user.catalog)
/*
/&
```

DELETE FILEA FILE

```
// JOB DELETE PURGE FILEA FILE
// EXEC IDCAMS,SIZE=AUTO
 DELETE sample.test.filea                        -
        PURGE                                    -
        CLUSTER                                  -
        CATALOG(user.catalog)
/*
/&
```

DELETE INTRAPARTITION TRANSIENT DATA SET

```
// JOB DELETE PURGE INTRAPARTITION TRANSIENT DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DELETE cics160.intra.transd                     -
        PURGE                                    -
        CLUSTER                                  -
        CATALOG(user.catalog)
/*
/&
```

DELETE AUXILIARY TEMPORARY STORAGE DATA SET

```
// JOB DELETE PURGE AUXILIARY TEMPORARY STORAGE DATA SET
// EXEC IDCAMS,SIZE=AUTO
 DELETE cics160.temp.storage                     -
        PURGE                                    -
        CLUSTER                                  -
        CATALOG(user.catalog)
/*
/&
```

DELETE VSAM SPACE

```
// JOB DELETE VSAM SPACE
// EXEC IDCAMS,SIZE=AUTO
 DELETE VOLUME                   -
        SPACE                    -
        CATALOG(user.catalog)
/*
/&
```

DELETE VSAM USER CATALOG

```
// JOB DELETE USER CATALOG
// EXEC IDCAMS,SIZE=AUTO
  DELETE (USER.CATALOG)  -
          USERCATALOG
/*
/&
```

Glossary

This glossary defines special CICS terms used in the library and words used with other than their everyday meaning. In some cases, a definition may not be the only one applicable to a term, but gives the particular sense in which it is used in this book.

This glossary includes terms and definitions from the IBM Vocabulary for Data Processing, Telecommunications, and Office Systems, GC20-1699.

American National Standards Institute (ANSI) definitions are preceded by an asterisk.

The symbol "(ISO)" at the beginning of a definition indicates that it has been discussed and agreed on at meetings of the International Organization for Standardization, Technical Committee 97/Subcommittee 1, and has been approved by ANSI for inclusion in the American National Dictionary for Information Processing.

abend. Abnormal end of task.

ACB. Access method control block. (VTAM and VSAM.)

access method. A technique for moving data between main storage and input/output devices.

ACF. Advanced Communication Function.

activity keypoint. A keypoint written to the system log during normal operation of CICS. In the event of an uncontrolled shutdown and subsequent emergency restart, activity keypoints can shorten the process of backward scanning through the system log. Activity keypoints are written automatically by the system (system activity keypoints) or by the user (user activity keypoints). (See also **keypoint.**)

address space. The complete range of addresses that is available to the programmer.

addressing. In data communication, the means whereby the originator or control station selects the unit to which it is going to send a message.

ALT. Application load table.

ANSI. American National Standards Institute.

application program. (1) A program written for or by a user that applies to the user's work. (2) In data communication, a program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

APAR. Authorized program analysis report.

*** ASCII.** American National Standard Code for Information Interchange.

assembler language. A source language that includes symbolic machine language statements in which there is a one-to-one correspondence with the instruction formats and data formats of the computer.

audit trail. A manual or computerized means for tracing the transactions affecting the contents of a record.

auxiliary storage. Data storage other than main storage; for example, storage on magnetic tape or direct access devices.

backout. See **dynamic transaction backout.**

Basic Mapping Support. A facility that handles data streams to and from a terminal. It provides device independence and format independence for application programs.

batch. An accumulation of data to be processed.

BDAM. Basic Direct Access Method. An access method used to retrieve or update particular blocks of a data set on a direct access device.

blocking. The process of combining two or more records into one block.

BMS. See **Basic Mapping Support.**

BSC. Binary synchronous communication.

BTAM. Basic telecommunications access method.

byte. In System/370, a sequence of eight adjacent binary digits that are operated on as a unit.

cataloged procedure. A set of control statements that has been placed in a library and can be retrieved by name.

CEDA. The resource definition online transaction.

CEMT. The master terminal transaction.

CEST. The supervisory terminal transaction.

CICS. Customer Information Control System.

CKD. Count-key-data. A disk storage device for storing data in the format: count field normally followed by a key field followed by the actual data of a record. The count field contains, besides other information, the address of the record in the format: CCHHR (where CC is the two-digit cylinder number, HH is the two-digit head number, and R is the record number) and the length of the data. The key field contains the record's key (search argument).

COBOL. Common business-oriented language. An English-like programming language designed for business data processing applications.

command. In CICS, an instruction similar in format to a high-level programming language statement. (Contrast with macro.) CICS commands invariably include the verb EXECUTE (or EXEC). They may be issued by an application program to make use of CICS facilities.

command-language statement. In CICS, synonym for command.

common system area (CSA). A major CICS storage control block.

* **concurrent.** Pertaining to the occurrence of two or more activities within a given interval of time.

control area. Synonym for control block.

control block. In CICS, a storage area used to hold dynamic data during the execution of control programs and application programs. Synonym for control area. Contrast with control table.

control table. In CICS, a storage area used to define or describe the configuration or operation of the system in a relatively permanent way. Contrast with control block.

conversational. Pertaining to a program or a system that carries on a dialogue with a terminal user, alternately accepting input and then responding to the input quickly enough for the user to maintain a train of thought.

CSA. See **common system area**.

CSD. CICS system definition file.

DAM. Direct access method.

DASD. Direct access storage device.

* **data base.** A collection of data fundamental to a system.

* **data communication.** The transmission and reception of data.

data independence. In CICS, the ability to request data by a high-level data-management method without concern as to the mechanics of data storage or retrieval. DL/I provides application programs with greater data independence.

data integrity. The quality of data that exists as long as accidental or malicious destruction, alteration, or loss of data are prevented.

data link protocol. A set of rules for data communication over a data link in terms of a transmission code, a transmission mode, and control and recovery procedures.

data security. The protection of data against unauthorized disclosure, transfer, modifications, or destruction, whether accidental or intentional.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data stream. All data transmitted through a data channel in a single read or write operation.

DB/DC. Data-base/data-communication.

DCT. Destination control table.

deadlock. (1) Unresolved contention for the use of a resource. (2) An error condition in which processing cannot continue because each of two elements of the process is waiting for an action by, or a response from, the other.

deblocking. The process of removing each logical record from a block.

despatch. To allocate time on a processor to jobs or tasks that are ready for execution.

destination control table. A table describing each of the transient data destinations used in the system, or in connected CICS systems.

device independence. The capability to write application programs so that they do not depend on the physical characteristics of devices. BMS provides a measure of device independence.

direct access storage. (1) * A storage device in which the access time is in effect independent of the location of the data. (2) A storage device that provides direct access to data.

distributed transaction processing (DTP). The distribution of processing between transactions that communicate synchronously with one another over intersystem or interregion links.

DL/I. Data Language/I. An IBM data base management facility provided by DOS DL/I and IMS data base program products.

DMB. Data management block (DL/I).

DOS. Disk Operating System.

DPCX. Distributed Processing Control Executive on IBM 8100 Series.

DSA. Dynamic storage area.

DTB. See **dynamic transaction backout.**

DTP. See **distributed transaction processing.**

DTR. Distribution tape reels.

dump control. The CICS element that provides storage dumps for help during testing.

dynamic log. An area in main storage used (by the journal control program) for storing copies of all changes to recoverable resources that might be required for dynamic backout of an LUW. Every execution of a transaction that has dynamic transaction backout specified has an associated dynamic log area.

dynamic transaction backout. The process of canceling changes made to stored data by a transaction following the failure of that transaction for whatever reason.

* **EBCDIC.** Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

EDF. Execution (command-level) diagnostic facility for testing command-level programs interactively at a terminal.

EIB. EXEC interface block.

emergency restart. The CICS facility for use following a system failure. It restores the data files of all interrupted transactions to the condition they were in when they started.

end user. In CICS, typically anyone using CICS to do a non-CICS job, typically by interacting with an application program (transaction) with a terminal.

enqueued. The state of a task scheduled to update a physical segment of a data base when another task is currently accessing that segment.

ESDS. Entry-sequenced data set.

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file.

EXEC. EXECUTE (as used in a CICS command).

FAQE. Free area queue element.

FBA. Fixed-block-architecture. A disk storage device that stores data in blocks of fixed size. These blocks are addressed by block number relative to the beginning of the particular file.

FCT. See **file control table.**

FERS. Facility Error Recognition System. A communications problem determination aid.

* **file.** (ISO) A set of related records treated as a unit, for example, in stock control, a file could consist of a set of invoices.

file control. The CICS element that controls all CICS file operations.

file control table. A table containing the characteristics of the files accessed by file control.

* **format.** The arrangement or layout of data on a data medium. In CICS, the data medium is usually a display screen.

format independence. The ability to send data to a device without having to be concerned with the format in which the data will be displayed. The same data may appear in different formats on different devices.

formatted dump. A dump of the CICS partition separating out the control blocks and storage areas.

function management header (FMH). In SNA, one or more headers optionally present in the the leading request unit (RU) of an RU chain. It allows one session partner in a LU-LU session to send function management information to the other.

function request shipping. The process, transparent to the application program, by which CICS accesses resources when those resources are actually held on another CICS system.

GLOS

GDDM. Graphical Data Display Manager.

* **generate.** To produce a computer program by selection of subsets from skeletal code under the control of parameters.

*** hardware.** (ISO) Physical equipment used in data processing, as opposed to computer programs, procedures, rules, and associated documentation. Contrast with software.

host processor. The primary or controlling computer in a multiple computer installation.

ICR. Independent component release.

inflight task. A task which, at the time of an abnormal termination of the system, had caused records to be written to the system log, but whose processing was only part of the way through an LUW.

initialization. Actions performed by the CICS system to construct the environment in the CICS partition to enable CICS applications to be run.

inquiry. A request for information from storage; for example, a request for the number of available airline seats.

installation. (1) A particular computing system, in terms of the work it does and the people who manage it, operate it, apply it to problems, service it and use the work it produces. (2) The task of making a program ready to do useful work. This task includes generating a program, initializing it, and applying PTFs to it.

integrity. See **data integrity**.

intent scheduling. Ensuring that a particular segment type of a data base is only accessible for potential update by one task at a time.

interactive. Pertaining to an application in which each entry calls forth a response from a system or program, as in an inquiry system or an airline reservation system. An interactive system may also be conversational, implying a continuous dialogue between the user and the system.

intercommunication facilities. A generic term covering intersystem communication (ISC) and multiregion operation (MRO).

interregion communication (IRC). The method by which CICS provides communication between a CICS region and another region in the same processor. Used for multiregion operation.

intersystem communication (ISC). Communication between separate systems by means of SNA networking facilities or by means of the application-to-application facilities of an SNA access method. Contrast with MRO.

interval control. The CICS element that provides time-dependent facilities.

intrapartition destination. A queue of transient data used subsequently as input data to another task within the CICS partition or region.

*** I/O.** Input/Output.

IPL. Initial Program Load.

IRC. See **interregion communication.**

ISAM. Indexed Sequential Access Method.

ISC. See **intersystem communication.**

JC. Journal control.

JCT. Journal control table.

journal. A set of one or more data sets to which records are written during a CICS run:

1. By CICS to implement user-defined resource protection (logging to the system log)
2. By CICS to implement user-defined automatic journaling (to any journal, including the system log)
3. Explicitly by JOURNAL command (or macro) from an application program (user journaling to any journal, including the system log).

journaling. The recording of information onto any journal (including the system log), for processing by the user.

keypoint. A set of records that describes the status of the system at a particular moment in time. Keypoint information includes extracts from system tables and control blocks such as: TCAs, FCT, PCT, DCT, TST. (See also **activity keypoint**, and **warm keypoint.**)

keyword. (1) A symbol that identifies a parameter. (2) A part of a command operand that consists of a specific character string.

KSDS. Key-sequenced data set.

line. (1) On a terminal, one or more characters entered before a return to the first printing or display position. (2) A string of characters accepted by the system as a single block of input from a terminal, for example, all characters entered before a carriage return or all characters entered before the terminal user hits the attention key.

linkage editor. (ISO) A computer program used to create one load module from one or more independently-translated object modules or load modules by resolving cross references among the modules.

*** loader.** A routine, commonly a computer program, that reads data into main storage.

local. In data communication, pertaining to devices that are attached to a controlling unit by cables, rather than data links.

local device. A device, such as a terminal, whose control unit is directly attached to a computer's data channel. No data link or control unit is used. Contrast with remote device.

local system. In CICS intercommunication, the CICS system from whose point of view intercommunication is being discussed.

logging. The recording (by CICS) of recovery information onto journal 01 (the system log), for use during emergency restart

logical unit (LU). In SNA, a port through which a user gains access to the services of a network.

logical unit of work (LUW). A sequence of processing actions (data base changes for example) that must be completed before any of the individual actions can be regarded as committed. When changes are committed (by successful completion of the LUW and recording of the sync point on the system log), they do not need to be backed out after a subsequent failure of the task or system. The end of an LUW is marked in a transaction by a sync point - issued either by the user program or by CICS at the end of task. In the absence of user sync points, the entire task is an LUW.

LU. See **logical unit.**

LUW. See **logical unit of work.**

macro. In CICS, an instruction similar in format to an assembler language instruction. Contrast with command.

main storage. (ISO) Program-addressable storage from which instructions and data can be loaded directly into registers for subsequent execution or processing. See also **real storage, storage, virtual storage.**

map. In CICS, a format established for a page or a portion of a page.

master terminal. In CICS, the terminal at which a designated operator is signed-on.

master terminal operator. Any CICS operator authorized to use the master terminal functions.

MCT. Monitoring control table.

message performance option. The improvement of ISC performance by eliminating sync point co-ordination between the connected systems.

message switching. In a data network, the process of routing messages by receiving, storing, and forwarding complete messages.

*** modularity.** The extent to which a system is composed of modules.

MRO. See **multiregion operation.**

multiprogramming. * Pertaining to the concurrent execution of two or more computer programs by a computer.

multiregion operation (MRO). Communication between CICS systems in the same processor without the use of SNA network facilities. Contrast with ISC.

multitasking. Concurrent execution of application programs within a CICS partition.

multithreading. The use, by several transactions, of a single copy of an application program.

MVT. Multiprogramming with a variable number of tasks.

NACP. Node abnormal condition program.

NCP. Network Control Program. A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

NEP. Node error program.

network. (1) An interconnected group of nodes. (2) The assembly of equipment through which connections are made between data stations.

network configuration. In SNA, the group of links, nodes, machine features, devices, and programs that make up a data processing system, a network, or a communication system.

NLT. Nucleus load table.

nonswitched connection. A connection that does not have to be established by dialing.

online. (1) * Pertaining to a user's ability to interact with a computer. (2) * Pertaining to a user's access to a computer via a terminal. The term "online" is also used to describe a user's access to a computer via a terminal.

operating system. Software that controls the execution of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

* **parameter.** (ISO) A variable that is given a constant value for a specified application and that may denote the application.

partition. A fixed-size subdivision of main storage, allocated to a system task. Contrast with region.

password. A unique string of characters that a program, computer operator, or user must supply to meet security requirements before gaining access to data. The password is confidential, as opposed to user identification.

path length. The amount of processor execution time, for example, per task.

PCT. See **program control table**.

PDS. Partitioned data set.

PEP. Program error program.

* **PL/I.** A programming language designed for use in a wide range of commercial and scientific applications.

PLT. Program list table.

polling. The process whereby stations are invited, one at a time, to transmit. The polling process usually involves the sequential interrogation of several data stations.

PPT. See **processing program table**.

pregenerated system. A CICS system distributed in a form that has already undergone the system generation process.

priority. A rank assigned to a task that determines its precedence in receiving system resources.

processing program table. A table defining all application programs valid for processing under CICS. It also keeps track of whether an application program is in main storage or not.

processor. (ISO) In a computer, a functional unit that interprets and executes instructions.

program check. A condition that occurs when programming errors are detected by an I/O channel.

program control. The CICS element that manages CICS application programs.

program control table. A table defining all transactions that may be processed by the system.

program isolation. Ensuring that only one task at a time can update a particular physical segment of a DL/I data base.

programmable terminal. A user terminal that has computational capability.

pseudo conversational. CICS transactions designed to appear to the operator as a continuous conversation occurring as part of a single transaction.

PTF. Program Temporary Fix. A temporary solution or by-pass of a problem diagnosed by IBM field engineering as the result of a defect in a current unaltered release of the program.

PUT. Program update tape.

quasi-reentrant. Applied to a CICS application program that is serially reusable between entry and exit points because it does not modify itself or store data within itself between calls on CICS facilities.

queue. A line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a message routing system.

RDO. Resource definition online.

real machine. An actual computer and its associated devices. Contrast with virtual machine.

real storage. The main storage in a virtual storage system. Physically, real storage and main storage are identical. Conceptually, however, real storage represents only part of the range of addresses available to the user of a virtual storage system.

recovery routine. A routine that is entered when an error occurs during the performance of an associated operation. It isolates the error, assesses the extent of the error, and attempts to correct the error and resume operation.

reentrant. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

region. In MVT, a variable-size subdivision of the dynamic area that is allocated to a job step or a system task.

remote. In data communication, pertaining to devices that are connected to a data processing system through a data link.

remote device. A device, such as a terminal, connected to a data processing system through a data link. Contrast with local device.

remote system. In CICS intercommunication, a system that the local CICS system accesses via intersystem communication or multiregion operation.

resource. Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs.

* **response time.** (ISO) The elapsed time between the end of an inquiry or demand on a data processing system and the beginning of the response. For example, the length of time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal.

* **rollback.** A programmed return to a prior checkpoint. In CICS, the cancellation by an application program of the changes it has made to all recoverable resources during the current logical unit of work.

RPG II. Report Program Generator, Version 2. A commercially-oriented programming language specifically designed for writing application programs that meet common business data processing requirements.

SAM. Sequential Access Method.

screen page. The amount of data displayed, or capable of being displayed, at any one time on the screen of a terminal.

SCS. SNA character string.

SDF/CICS. Screen Definition Facility. An online application development program product used to define or edit BMS maps interactively.

SDLC. Synchronous Data Link Control. A communications protocol.

security. Prevention of access to or use of data or programs without authorization.

sequential data set. A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape.

service. The carrying out of effective problem determination, diagnosis, and repair on a data processing system or software product.

Service Level Reporter II. A data reduction and analysis program product. Useful for the analysis of CICS operating statistics.

SIT. See **system initialization table.**

SNA. See Systems network architecture.

SNT. Sign-on table.

* **software.** (ISO) Programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system. Contrast with hardware.

SOS. Short on storage.

SQL/DS. Structured Query Language/Data System. A relational data base management facility.

SRT. See **system recovery table.**

starter system. In CICS, a pregenerated system providing at least one version of all system programs. A starter system is ready to use with little or no assembly.

startup. The operation of starting up CICS by the system operator.

startup jobstream. A set of job control statements used to start up CICS.

storage. A functional unit into which data can be placed and from which it can be retrieved. See main storage, storage, virtual storage.

storage control. The CICS element that obtains working storage areas.

storage dump. See **transaction dump.**

storage protection key. An indicator that appears in the current program status word whenever an associated task has control of the system. This indicator must match the storage keys of all main storage blocks that the task is to use.

storage violation dump. A formatted dump taken as a result of a storage error detected by the storage control program, including a dump of the dynamic storage error.

subsystem. A secondary or subordinate system.

supervisory terminal operator. Any CICS operator whose security key(s) allow use of the supervisory terminal functions.

SVC. Supervisor call.

GLOS

switched connection. A connection that is established by dialing.

synchronization point (sync point). A point in the processing of a task (marked by a SYNCPOINT command, end of task, or DL/I TERM call) at which changes to recoverable resources are regarded as **committed**.

system. In CICS, an assembly of hardware and software capable of providing the facilities of CICS for a particular installation.

system activity keypoint. A keypoint written to the system log automatically while CICS is running normally. (See also **activity keypoint**.)

system generation (SYSGEN). In CICS, the process of creating a particular system tailored to the requirements of a data processing installation.

system initialization table. A table containing user-specified data that will control a system initialization process.

system log. The (only) journal (identification='01') that is used by CICS to log changes made to resources for the purpose of backout on emergency restart.

system program. A program providing services in general support of the running of a system.

system recovery table. A table listing the ABEND or abnormal condition codes that CICS will intercept.

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.

task. (1) (ISO) A basic unit of work to be accomplished by a computer. (2) Under CICS, the execution of a transaction for a particular user. Contrast with transaction.

task control. The CICS element that controls all CICS tasks.

task switching. Overlapping of I/O operations and processing between several tasks.

TCA. Task control area.

TCT. Terminal control table.

temporary storage control. The CICS element that provides temporary data storage facilities.

temporary storage table. A table describing temporary storage queues and queue prefixes for which CICS is to provide recovery.

terminal. (1) * A point in a system or communication network at which data can either enter or leave. (2) In CICS, a device, often equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

terminal control. The CICS element that controls all CICS terminal activity.

terminal control table. A table describing a configuration of terminals, logical units, or other CICS systems in a CICS network with which the CICS system may communicate.

terminal operator. The user of a terminal.

terminal paging. A set of commands for retrieving "pages" of an oversize output message in any order.

threading. The process whereby various transactions undergo concurrent execution.

TIOA. Terminal input/output area.

TLT. Terminal list table.

TP access method. Teleprocessing access method. For example, VTAM.

trace control. The CICS element that provides a trace facility.

transaction. A transaction may be regarded as a unit of processing (consisting of one or more application programs) initiated by a single request, often from a terminal. A transaction may require the initiation of one or more tasks for its execution. Contrast with task.

transaction backout. The cancellation, as a result of a transaction failure, of all updates performed by a task.

transaction dump. A dump of the control blocks and storage areas associated with a particular task.

transaction identification code. Synonym for transaction identifier. A group of up to four characters entered by an operator when selecting a CICS transaction.

transaction identifier. Synonymous with transaction identification code.

transaction restart. The restart of a task after a transaction backout.

transient data control. The CICS element that controls sequential data files and intrapartition data.

TST. See **temporary storage table**.

tuning. The process of adjusting system control variables to make the system divide its resources most efficiently for the workload.

turnaround time. (ISO) The elapsed time between submission of a job and the return of the complete output.

TWA. Transaction work area.

TWX. Teletypewriter exchange terminal.

update. To modify a file with current information.

user exit. A point in an IBM-supplied program at which a user exit routine may be given control.

utilities. Informal term for utility programs.

virtual machine (VM). A functional simulation of a computer and its associated devices. Contrast with real machine.

virtual storage (VS). (ISO) The notional storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available and not by the number of main storage locations.

VM/370. IBM Virtual Machine Facility/370.

VSE. Virtual Storage Extended.

VSAM. Virtual Storage Access Method. An access method for direct or sequential processing of fixed- and variable-length records on direct access devices.

VTAM. Virtual Telecommunications Access method.

warm keypoint. A keypoint written to the restart data set during controlled shutdown (after all system activity has ceased). During a subsequent warm restart, information in the warm keypoint is used to reestablish system tables to the status they had at controlled shutdown. (See also **keypoint.**)

working set. (1) The set of a user's pages that must be active in order to avoid excessive paging. (2) The amount of real storage required in order to avoid a thrashing condition.

WTTY. World Trade teletypewriter.

XLT. Transaction list table.

Bibliography

VSE/ADVANCED FUNCTIONS

- VSE/Advanced Functions Installation, SC33-6096
- VSE/Advanced Functions Macro Reference, SC24-5211
- VSE/Advanced Functions Maintain System History Program Reference, SC33-6101
- VSE/Advanced Functions Maintain System History Program User's Guide, SC33-6072
- VSE/Advanced Functions Messages, SC33-6098
- VSE/Advanced Functions Operating Procedures, SC33-6097
- VSE/Advanced Functions Serviceability Aids and Debugging Procedures, SC33-6099
- VSE/Advanced Functions System Control Statements, SC33-6095
- VSE/Advanced Functions System Generation, SC33-6096
- VSE/Advanced Functions System Management Guide, SC33-6094

SYSTEMS NETWORK ARCHITECTURE (SNA)

- Systems Network Architecture - Function Description of Logical Unit Types, GC20-1868
- Systems Network Architecture - Types of Logical Unit to Logical Unit Sessions, GC20-1869

BASIC TELECOMMUNICATIONS ACCESS METHOD

- Basic Telecommunications Access Method Extended Support (BTAM-ES) Programming, SC38-0293

VSAM

- VSE/VSAM Access Method Services User Guide
- VSE/VSAM Programmer's Reference, SC24-5145
- Using VSE/VSAM Commands and Macros, SC24-5144

DL/I DOS/VS

- DL/I DOS/VS Application Programming: High Level Programming Interface, SH24-5009
- DL/I DOS/VS Application Programming Reference Manual, SH12-5411
- DL/I DOS/VS General Information Manual, GH20-1246
- DL/I DOS/VS Guide for New Users, SH24-5001
- DL/I DOS/VS Operator's Reference and Messages and Codes, SH12-5414
- DL/I DOS/VS Resource Definition and Utilities, SH24-5021 (DL/I 1.6 only)
- DL/I DOS/VS System and Application Design Guide, SH12-5413
- DL/I DOS/VS Utilities and Guide for the System Programmer, SH12-5412 (DL/I 1.5 only)

Index

S

W

Customer Information Control System
(CICS/DOS/VS) Version 1 Release 6
Installation and Operations Guide

READER'S COMMENT FORM

Order No. SC33-0070-5

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Number of your latest Technical Newsletter for this publication . . .

If you want an acknowledgment, give your name and address below.

Name . . . .

Job Title . . . . Company . . . .

Address . . . .

. . . . Zip . . . .

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to either address in the Edition Notice on the back of the title page.)

Note: *Staples can cause problems with automated mail sorting equipment*
Please use pressure-sensitive or other gummed tape to seal this form.

Cut Along Dotted Line

SC33-0070-5

Reader's Comment Form

Fold and Tape　　Fold and Tape　　Fold and Tape

No postage necessary if mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS　　PERMIT 40　　ARMONK, NEW YORK

Postage will be paid by addressee:

International Business Machines Corporation
Department 812H
1133 Westchester Avenue
White Plains, New York 10604

Fold and Tape　　Fold and Tape　　Fold and Tape

IBM®

CICS/DOS/VS Version 1 Release 6 Installation and Operations Guide　Printed in U.S.A.　SC33-0070-5

SC33-0070-5

CICS/DOS/VS Version 1 Release 6 Installation and Operations Guide Printed in U.S.A. SC33-0070-5